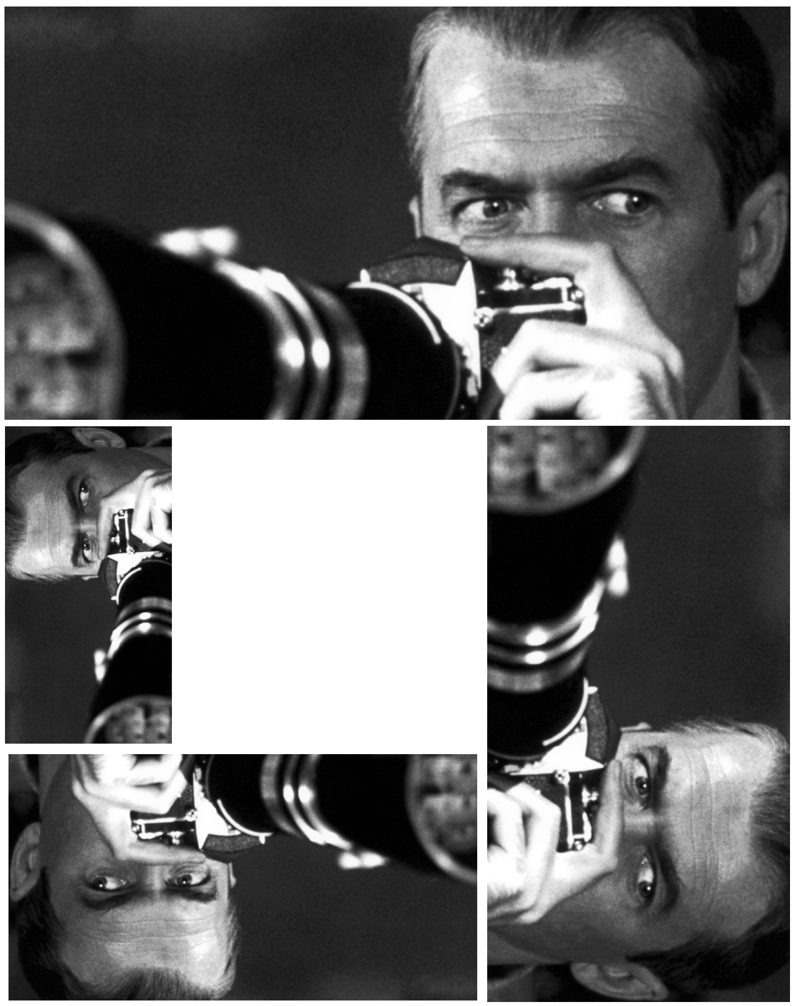
Matemàtica Discreta i Lògica



Francesc J. Ferri

Versió 2.4b (Bilingüe)
Agost 2022

Matemàtica Discreta i Lògica

Francesc J. Ferri
Universitat de València

4 d'agost de 2022

Versió 2.4b

Aquest llibre ha rebut un ajut del Servei de Política Lingüística de la Universitat de València en la convocatòria 2020 d'incentius per a la qualitat lingüística en l'elaboració de materials docents.

Em dic Llop. Solucione problemes.

Índex

Pròleg

L'assignatura *34666 - Matemàtica discreta i lògica* s'imparteix en primer curs d'Enginyeria Informàtica a la Universitat de València. És una assignatura introductòria que hauria de servir de pont entre conceptes matemàtics, lògics i computacionals, importants a l'hora d'assolir certes competències transversals com ara la capacitat per a representar informació i per a resoldre problemes.

En l'assignatura hi ha quatre blocs principals: combinatòria, lògica, recursió, i estructures gràfiques i arborescents. Encara que en queden fora molts continguts típics de cursos i llibres de matemàtica discreta (com ara resolució de recurrències, codificació, teoria de nombres, etc.), també és veritat que de qualsevol dels quatre blocs es podria impartir una assignatura sencera (encara que no en primer curs).

El tret principal del present manual és la col·lecció de problemes. Es tracta majorment de problemes que han sigut utilitzats en exàmens o en exercicis pràctics. Molts admeten solucions obertes, ramificacions i extensions que poden resultar interessants per a la comprensió dels conceptes relacionats.

Per això s'ha reduït la teoria a l'enumeració de conceptes importants, i s'han estès les solucions de problemes seleccionats perquè il·lustren i expliquen la part teòrica corresponent.

El material no s'hauria d'utilitzar com a apunts complets i autocontinguts, sinó com un resum de conceptes que cal ampliar en altres fonts. Tampoc no s'haurien de considerar les solucions com a respostes tipus que cal estudiar. Al contrari, la millor manera d'aprofitar el material consisteix a intentar seriosament la resolució de cada problema abans de mirar la solució.

El llibre s'ha concebut com un projecte obert, de manera que la col·lecció de problemes puga anar creixent. La idea és que estiga disponible en línia de manera oberta i que es puga navegar fàcilment entre les seues parts.

Finalment, val a dir que aquest projecte s'ha beneficiat d'una manera o d'una altra de material (concret) previ, discussions (discretes) i reflexions (lògiques) amb diversos companys del departament amb qui he tingut el plaer de treballar. Principalment, Fernando Barber, Ignacio García, Sergio Casas, Miguel Lozano i Salva Moreno.

Burjassot, 26 de febrer de 2020.

Índex de Problemes

1. Combinatòria

Combinatoria. 1

1.1 Conjunts, seqüències i aplicacions

Conjuntos, secuencias y aplicaciones

1.1.1 Conjunts

Conjuntos

Un **conjunt** és una agrupació o col·lecció de zero o més elements d'algun tipus (per exemple nombres enters, reals, complexos, lletres, o fins i tot altres conjunts) de manera que no importa l'ordre i els elements no es poden repetir.

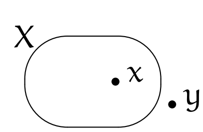
Un **conjunto** *es una agrupación o colección de cero o más elementos de algún tipo (por ejemplo números enteros, reales, complejos, letras, o incluso otros conjuntos) de manera que no importa el orden y los elementos no se pueden repetir.*

Un **multiconjunt** és un conjunt en què els elements es poden repetir un nombre arbitrari de vegades.

Un **multiconjunto** *es un conjunto donde los elementos se pueden repetir un número arbitrario de veces.*

Un element x **pertany** a un conjunt, X, si és un dels que formen la col·lecció. Ho escrivim com a $x \in X$. Si un element y no pertany a X, s'escriu $y \notin X$. Aquesta situació la representem gràficament mitjançant **diagrames de Venn** com a

Un elemento x **pertenece** *a un conjunto, X, si es uno de los que forman la colección. Lo escribimos como $x \in X$. Si un elemento y no pertenece a X, se escribe $y \notin X$. Esta situación la representamos gráficamente mediante* **diagramas de Venn** *como*

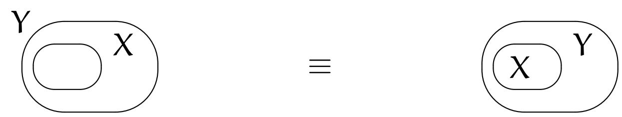


Si tots els elements d'un conjunt X estan en el conjunt Y, diem que Y **conté** X o que X **és contingut** en Y, i ho escrivim com a $X \subseteq Y$. També diem que X és un **subconjunt** de Y.

Si todos los elementos de un conjunto X están en el conjunto Y, decimos que Y **contiene** *a X o que X* **está contenido** *en Y, y lo escribimos como $X \subseteq Y$. También decimos que X es un* **subconjunto** *de Y.*

De la mateixa manera, diem que Y és un **superconjunt** de X. Gràficament

Igualmente, decimos que Y es un **superconjunto** *de X. Gráficamente*

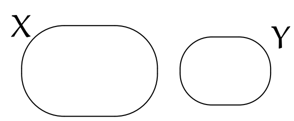


Si $X \subseteq Y$ i $Y \subseteq X$, aleshores necessàriament $X = Y$. Si $X \subseteq Y$ i $X \neq Y$, diem que X és un **subconjunt estricte** de Y. O també que **està estrictament contingut** en Y o que Y **conté estrictament** X. També es pot dir que X és un **subconjunt propi** de, o que **està pròpiament contingut** en Y.

Si $X \subseteq Y$ e $Y \subseteq X$ entonces necesariamente $X = Y$. Si $X \subseteq Y$ y $X \neq Y$ decimos que X es un **subconjunto estricto** *de Y. O también que* **está estrictamente contenido** *en Y o que Y* **contiene estrictamente** *a X. También se dice que X es un* **subconjunto propio** *de, o que* **está propiamente contenido** *en Y.*

Si cap dels elements d'un conjunt X pertany a un altre conjunt Y (i per tant, també al revés) es diu que X i Y, són **conjunts disjunts**.

Si ninguno de los elementos de un conjunto X pertenece a otro conjunto Y (y por lo tanto también al revés) se dice que X e Y son **conjuntos disjuntos***.*



De la mateixa manera, ens referim a qualsevol col·lecció de conjunts com a **disjunts** o **mútuament disjunts** si són tots ells disjunts dos a dos.

De la misma manera, nos referimos a cualquier colección de conjuntos como **disjuntos** *o* **mutuamente disjuntos** *si son todos ellos disjuntos dos a dos.*

El **conjunt buit**, $\emptyset$, és un conjunt que no conté cap element. Com a conseqüència, $\emptyset$ és subconjunt de qualsevol altre conjunt. Fins i tot d'ell mateix.

El **conjunto vacío***, $\emptyset$, es un conjunto que no contiene ningún elemento. Como consecuencia, $\emptyset$ es subconjunto de cualquier otro conjunto. Incluso de sí mismo.*

Anomenem **univers** un conjunt que és superconjunt de tots els conjunts en un determinat context.

Llamamos **universo** *a un conjunto que es superconjunto de todos los conjuntos en un determinado contexto.*

Per a denotar conjunts utilitzem claus com en

Para denotar conjuntos utilizamos llaves como en

$$A = \{a, b, c\}$$

on hem definit el conjunt A per **extensió** (enumerant tots els seus elements).

donde hemos definido el conjunto A por **extensión** *(enumerando todos sus elementos).*

També podem definir un conjunt per **comprensió** si especifiquem alguna propietat que ha de complir un element (de l'univers) si i només si està en el conjunt. Per exemple,

También podemos definir un conjunto por **comprensión** *si especificamos alguna propiedad que tiene que cumplir un elemento (del universo) si y sólo si está en el conjunto. Por ejemplo,*

$$B = \{x \in \mathbb{N} \mid x = 2k\},$$

es llegeix com "el conjunt B és format per tots aquells nombres naturals tals que es poden escriure com a $2k$ (per a algun k, s'entén)".

se lee como "el conjunto B está formado por todos aquellos números naturales de tal manera que se pueden escribir como $2k$ (para a algún k, se entiende)".

En algunes ocasions podem definir conjunts informalment com en $\mathbb{N} = \{1, 2, 3, \ldots\}$. A banda dels **naturals**, $\mathbb{N}$, altres conjunts concrets que utilitzarem sovint són:

En algunas ocasiones podemos definir conjuntos informalmente como en $\mathbb{N} = \{1, 2, 3, \ldots\}$. Aparte de los **naturales***, $\mathbb{N}$, otros conjuntos concretos que utilizaremos frecuentemente son:*

$\mathbb{N}$	naturals *naturales*	$\{1, 2, 3, \ldots\}$
$\mathbb{N}_k$	k-naturals k-*naturales*	$\{1, 2, 3, \ldots, k\}$
$\mathbb{Z}$	enters *enteros*	$\{\ldots, -2, -1, 0, 1, 2, \ldots\}$
$\mathbb{Z}_k$	k-enters k-*enteros*	$\{-k, \ldots, -2, -1, 0, 1, 2, \ldots, k\}$
$\mathbb{Z}^+$	enters no negatius *enteros no negativos*	$\{0, 1, 2, \ldots\}$
$\mathbb{Z}_k^+$	k-enters no negatius k-*enteros no negativos*	$\{0, 1, 2, \ldots, k\}$
$\mathbb{Z}_1^+$	univers binari *universo binario*	$\{0, 1\}$
$\mathbb{Q}$	racionals *racionales*	$\{p/q \mid p \in \mathbb{Z}, q \in \mathbb{N}\}$
$\mathbb{R}$	reals *reales*	{punts de la recta real } *puntos de la recta real*
$[0, 1]$	interval unitari tancat *intervalo unitario cerrado*	$\{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$
$(0, 1)$	interval unitari obert *intervalo unitario abierto*	$\{x \in \mathbb{R} \mid 0 < x < 1\}$

Donat un conjunt X, anomenem **conjunt potència** de X el conjunt format per tots els seus subconjunts,

Dado un subconjunto X, llamamos **conjunto potencia** *de X al conjunto formado por todos sus subconjuntos,*

$$\mathcal{P}(X) = 2^X = \{Y \mid Y \subseteq X\}.$$

Per exemple, el conjunt potència de $\mathbb{Z}_1^+$ és

Por ejemplo, el conjunto potencia de $\mathbb{Z}_1^+$ es

$$\{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$$

La **cardinalitat** d'un conjunt X, que escrivim com a $|X|$ o com a $\mathrm{card}(X)$, és el nombre d'elements que conté. Si $X \subseteq Y$ aleshores $|X| \leq |Y|$. La cardinalitat d'un conjunt és infinita si conté infinits elements.

La **cardinalidad** *de un conjunto X, que escribimos como $|X|$ o como $\mathrm{card}(X)$, es el número de elementos que contiene. Si $X \subseteq Y$ entonces $|X| \leq |Y|$. La cardinalidad de un conjunto es infinita si contiene infinitos elementos.*

Una col·lecció (o conjunt) de conjunts, $\mathcal{B} = \{B_1, \ldots, B_k\}$, s'anomena **partició** d'un conjunt A si tots ells són subconjunts **disjunts** **no buits** de A i tot element de A pertany també a algun dels subconjunts B_i. Els conjunts B_i s'anomenen **blocs** de la partició $\mathcal{B}$. Com a cas especial, es pot acceptar que $\{\emptyset\}$ és una partició de $\emptyset$.

Una colección (o conjunto) de conjuntos, $\mathcal{B} = \{B_1, \ldots, B_k\}$, se denomina **partición** *de un conjunto A si todos ellos son subconjuntos* **disjuntos** **no vacíos** *de A y todo elemento de A pertenece también a alguno de los subconjuntos B_i. Los conjuntos B_i se les llama* **bloques** *de la partición $\mathcal{B}$.*

Donades dues particions, $\mathcal{B}, \mathcal{C}$, d'un mateix conjunt A, es diu que $\mathcal{C}$ és un **refinament** de $\mathcal{B}$ si tot bloc de $\mathcal{C}$ és subconjunt d'algun bloc de $\mathcal{B}$.

Dadas dos particiones, $\mathcal{B}, \mathcal{C}$, de un mismo conjunto A, se dice que $\mathcal{C}$ es un **refinamiento** *de $\mathcal{B}$ si todo bloque de $\mathcal{C}$ es subconjunto de algún bloque de $\mathcal{B}$.*

En el següent exemple, $\mathcal{B}$, és una partició (en 3 blocs) de $\mathbb{Z}$, i $\mathcal{C}$ és un refinament (amb 5 blocs) de $\mathcal{B}$.

En el siguiente ejemplo, $\mathcal{B}$, es una partición (en 3 bloques) de $\mathbb{Z}$, y $\mathcal{C}$ es un refinamento (con 5 bloques) de $\mathcal{B}$.

$$\mathcal{B} = \{\mathbb{N}, \{0\}, \{n \mid -n \in \mathbb{N}\}\},$$
$$\mathcal{C} = \{\{n \mid n = 2k, k \in \mathbb{N}\}, \{n \mid n = 2k-1, k \in \mathbb{N}\}, \{0\}, \{n \mid n = -2k, k \in \mathbb{N}\}, \{n \mid n = 1-2k, k \in \mathbb{N}\}\},$$

1.1.2 Operacions sobre conjunts

Operaciones sobre conjuntos

La **unió** de dos conjunts, A, B, és formada pels elements que estan en A o en B. S'escriu $A \cup B$. Formalment,

La **unión** *de dos conjuntos, A, B, está formada por los elementos que están en A o en B. Se escribe $A \cup B$.Formalmente,*

$$A \cup B = \{x \mid x \in A \vee x \in B\}.$$

La **intersecció** de dos conjunts, A, B, és formada pels elements que estan en A i en B. S'escriu $A \cap B$. Formalment,

La **intersección** *de dos conjunts, A, B, está formada por los elementos que están en A y en B. Se escribe $A \cap B$. Formalmente,*

$$A \cap B = \{x \mid x \in A \wedge x \in B\}.$$

La **diferència conjuntista** entre A i B, és formada pels elements de A que no estan en B. S'escriu $A\backslash B$, o també $A - B$. Formalment,

La **diferencia conjuntista** *entre A y B, está formada por los elementos de A que no están en B. Se escribe $A\backslash B$, o también $A-B$. Formalmente,*

$$A\backslash B = \{x \mid x \in A \wedge x \notin B\}.$$

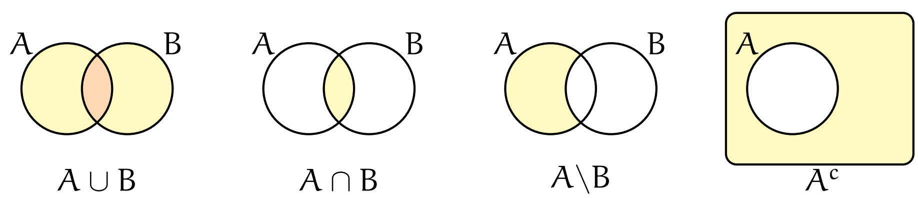
El **complement** del conjunt A (respecte de l'univers U) es defineix com a $\overline{A} = A^c = U\backslash A$. Formalment,

El **complemento** *del conjunt A (respecto al universo U) se define como $\overline{A} = A^c = U\backslash A$. Formalmente,*

$$A^c = \{x \mid x \notin A\}.$$

Aquestes quatre operacions, gràficament es poden representar mitjançant diagrames de Venn de la següent manera.

Estas cuatro operaciones gráficamente se pueden representar mediante diagramas de Venn de la siguiente manera.



1.1.3 Seqüències i tuples

Secuencias y tuplas

Una **seqüència** és una enumeració ordenada de zero o més elements normalment del mateix tipus. Una **tupla** és un element del **producte cartesià** de zero o més conjunts.

Una **secuencia** *es una enumeración ordenada de cero o más elementos normalmente del mismo tipo. Una* **tupla** *es un elemento del* **producto cartesiano** *de cero o más conjuntos.*

Una seqüència (homogènia) és una tupla sobre un únic conjunt d'elements. No distingirem entre seqüències i tuples que escriurem com a $(x_1, x_2, \ldots)$.

Una secuencia (homogénea) es una tupla sobre un único conjunto de elementos. No distinguiremos entre secuencias y tuplas que escribiremos como $(x_1, x_2, \ldots)$.

Si $x_i \in X_i$ per a i=1,2,..., k, aleshores

Si $x_i \in X_i$ para i=1,2,..., k, entonces

$$(x_1, x_2, \ldots, x_k) \in X_1 \times X_2 \times \cdots \times X_k.$$

També podem escriure que

También podemos escribir que

$$X_1 \times X_2 \times \cdots \times X_k = \{(x_1, x_2, \ldots, x_k) \mid x_i \in X_i, i = 1, 2, \ldots, k\}.$$

Una tupla de k elements s'anomena k-tupla. El valor de k és la **dimensió**, **grandària** o **talla** de la tupla. Un **parell** és una 2-tupla. Una **terna**, **trio** o **tripleta** és una 3-tupla. De vegades, anomenarem **vector** qualsevol k-tupla de nombres (reals, en principi).

Una tupla de k elementos se llama k-tupla. El valor de k es la **dimensión**, **tamaño** *o* **talla** *de la tupla. Un* **par** *es una 2-tupla. Una* **terna**, **trio** *o* **tripleta** *es una 3-tupla. A veces, llamaremos* **vector** *a cualquier k-tupla de números (reales, en principio).*

$$(x_1, x_2, \ldots, x_k) \in \mathbb{R}^k.$$

Si estem considerant conjunts sobre un determinat univers els elements del qual es poden numerar, $U = \{e_1, e_2, \ldots, e_n\}$, aleshores podem definir per a cada conjunt, $A \subseteq U$, el seu **vector característic** com a

Si estamos considerando conjuntos sobre un determinado universo cuyos elementos se pueden enumerar, $U = \{e_1, e_2, \ldots, e_n\}$, entonces podemos definir para cada conjunto, $A \subseteq U$, su **vector característico** *como*

$$\chi_A = (x_1, x_2, \ldots, x_n),$$

on

donde

$$x_i = \begin{cases} 1 & \text{si } e_i \in A \\ 0 & \text{si } e_i \notin A \end{cases}$$

Per exemple, els vectors característics dels conjunts $\emptyset$ i $\{2,3\}$ sobre l'univers $\mathbb{N}_6$ són $(0,0,0,0,0,0)$ i $(0,1,1,0,0,0)$, respectivament.

Por ejemplo, los vectores característicos de los conjuntos $\emptyset$ y $\{2,3\}$ sobre el universo $\mathbb{N}_6$ son $(0,0,0,0,0,0)$ y $(0,1,1,0,0,0)$, respectivamente.

Els *vectors* característics sobre universos infinits són de dimensió infinita. Per exemple, $(0,1,0,1,0,\ldots)$ representaria el subconjunt format pels nombres parells de $\mathbb{N}$.

Los vectores *característicos sobre universos infinitos son de dimensión infinita. Por ejemplo, $(0,1,0,1,0,\ldots)$ representaría el subconjunto formado por los números pares de $\mathbb{N}$.*

1.1.4 Relacions i aplicacions

Relaciones y aplicaciones

Una **relació** (binària), R, entre els conjunts A i B és un subconjunt del producte cartesià $A \times B$. És a dir, un conjunt de parells formats per un element de A i un altre de B. O siga,

*Una **relación** (binaria), R, entre los conjuntos A y B es un subconjunto del producto cartesiano $A \times B$. Es decir, un conjunto de pares formados por un elemento de A y otro de B. O sea,*

$$R \subseteq A \times B.$$

En el cas particular que $B = A$ diem que R és una relació en el conjunt A.

En el caso particular en que $B = A$ decimos que R es una relación en el conjunto A.

Diem que un element $a \in A$ està relacionat amb un element $b \in B$ segons R i ho escrivim com a aRb si i només si

Decimos que un elemento $a \in A$ está relacionado con un elemento $b \in B$ según R, y lo escribimos como aRb si y sólo si

$$(a, b) \in R \subseteq A \times B.$$

En cas contrari els elements no estan relacionats i escrivim $a\not{R}b$. Es poden definir relacions entre un nombre k d'elements diferent de 2. Les anomenem **relacions k-àries**. L'enter k és l'**aritat** de la relació.

*En caso contrario los elementos no están relacionados y escribimos $a\not{R}b$. Se pueden definir relaciones entre un número k de elementos diferente de 2. Llamaremos a éstas **relaciones k-arias**. El entero k es la **aridad** de la relación.*

Una relació que siga $R = \emptyset$ (ningú es relaciona amb ningú) o $R = A \times B$ (tots es relacionen amb tots) s'anomena **trivial**.

*Una relación que sea $R = \emptyset$ (nadie se relaciona con nadie) o $R = A \times B$ (todos se relacionan con todos) se la llama **trivial**.*

Una relació entre dos conjunts A i B en què almenys un element de A està relacionat amb almenys un element de B s'anomena **correspondència**.

*Una relación entre dos conjuntos A y B en que al menos un elemento de A está relacionado con al menos uno de B recibe el nombre de **correspondencia**.*

Com que una relació (binària) no és més que un subconjunt del producte cartesià de dos conjunts, sempre és possible definir el vector característic d'una relació a partir de qualsevol enumeració d'aquest producte.

Como una relación (binaria) no es más que un subconjunto del producto cartesiano de dos conjuntos, siempre es posible definir el vector característico de una relación a partir de cualquier enumeración de este producto.

Per exemple, si considerem la igualtat mòdul 2 entre els conjunts $\mathbb{Z}_2^+$ i $\mathbb{Z}_3^+$,

Por ejemplo, si consideramos la igualdad módulo 2 entre los conjuntos $\mathbb{Z}_2^+$ y $\mathbb{Z}_3^+$,

$$nR_2m \quad \textbf{sii} \quad \bmod (n,2) = \bmod (m,2)$$

i l'enumeració

y la enumeración

$$(0,0),(0,1),(0,2),(0,3),(1,0),\ldots,(2,2),(2,3)$$

també anomenada enumeració lèxico-gràfica, el corresponent vector característic de R_2 seria

también llamada enumeración lexicográfica, el correspondiente vector característico de R_2 sería

$$(1,0,1,0,0,1,0,1,1,0,1,0)$$

No obtstant això, en el cas de relacions, resulta més natural i interessant representar aquest vector com una matriu, $\chi_R = [x_{i,j}]$, que anomenarem **matriu característica** de la relació $R \subseteq A \times B$, que es pot definir com

No obstante, en el caso de relaciones, resulta más natural e interesante representar este vector como una matriz, $\chi_R = [x_{i,j}]$, que llamaremos **matriz característica** *de la relación $R \subseteq A \times B$, que se puede definir como*

$$x_{i,j} = \begin{cases} 1 & \text{si } a_i R b_j \\ 0 & \text{si } a_i \not R b_j \end{cases}$$

on

donde

$$A = \{a_1,\ldots,a_n\},\ B = \{b_1,\ldots,b_m\}$$

Per exemple, la matriu característica de la relació R_2 anterior seria

Por ejemplo, la matriz característica de la relación R_2 anterior sería

$$\chi_{R_2} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Una **relació d'equivalència** és una relació binària en un conjunt A que és **reflexiva** (aRa per a qualsevol a), **simètrica** (si aRb aleshores bRa per a qualsevol parell d'elements), i **transitiva** (si aRb i bRc aleshores aRc per a qualsevol terna d'elements).

Una **relación de equivalencia** *es una relación binaria en un conjunto A que es* **reflexiva** *(aRa para cualquier a),* **simétrica** *(si aRb entonces bRa para cualquier par de elementos), y* **transitiva** *(si aRb y bRc entonces aRc para cualquier terna de elementos).*

Exemples de relacions d'equivalència serien la igualtat, $R_=$, o la igualtat mòdul 2, R_2, definides sobre un únic conjunt. Per exemple, sobre el conjunt $\mathbb{Z}_3^+$, les matrius característiques d'aquestes relacions serien

Ejemplos de relaciones de equivalencia serían la igualdad, $R_=$, o la igualdad módulo 2, R_2, definidas sobe un único conjunto. Por ejemplo, sobre el conjunto $\mathbb{Z}_3^+$, las matrices características de estas relaciones serían

$$\chi_{R_=} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \chi_{R_2} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Donada una relació binària d'equivalència R sobre un conjunt A, anomenem **classe d'equivalència** tot subconjunt B de A que compleix

Dada una relación binaria de equivalencia R sobre un conjunto A, llamamos **clase de equivalencia** *a todo subconjunto B de A que cumple*

$$aRb \text{ sii } a \in B, b \in B$$

És a dir, B conté **tots** els elements de A que estan relacionats entre ells.

Es decir, contiene a **todos** *los elementos de A que están relacionados entre ellos.*

La notació $[a]_R$ significa classe d'equivalència de R en A que conté l'element $a \in A$. Per tant, donada una classe d'equivalència B, es té que

La notación $[a]_R$ significa clase de equivalencia de R en A que contiene el elemento $a \in A$. Por lo tanto, dada una clase de equivalencia B, se tiene que

$$[a]_R = B, \forall a \in B$$

El subíndex es pot eliminar si no pot haver-hi confusió.

El subíndice se puede eliminar si no puede haber confusión.

S'anomena **conjunt quocient** del conjunt A i la relació R, o simplement quocient de la relació, i se denota com a A/R, el conjunt format per **totes** les classes d'equivalència induïdes per R en A.

Se denomina **conjunto cociente** *del conjunto A y la relación R, o simplemente cociente de la relación, y se denota como A/R, el conjunto formado por* **todas** *las clases de equivalencia inducidas por R en A.*

$$A/R = \{[a] \mid a \in A\}$$

Els conjunts quocient corresponents a les relacions $R_=$ i R_2 definides anteriorment serien

Los conjuntos cociente correspondientes a las relaciones $R_=$ y R_2 definidas anteriormente serían

$$\mathbb{Z}_3^+/R_= = \{\{0\},\{1\},\{2\},\{3\}\} \qquad \mathbb{Z}_3^+/R_2 = \{\{0,2\},\{1,3\}\}$$

Si considerem la relació R_2 sobre el conjunt infinit $\mathbb{Z}^+$ tindríem també un conjunt quocient de cardinalitat 2,

Si consideramos la relación R_2 sobre el conjunto infinito $\mathbb{Z}^+$ tendríamos también un conjunto cociente de cardinalidad 2,

$$\mathbb{Z}^+/R_2 = \{[0],[1]\}$$

on les classes d'equivalència $[0]$ i $[1]$ contenen els enters parells i senars, respectivament.

donde las clases de equivalencia $[0]$ y $[1]$ contienen los enteros pares e impares, respectivamente.

Una **relació d'ordre** és una relació binària en un conjunt A que és **reflexiva**, **antisimètrica** (si aRb i bRa aleshores $a = b$ per a qualsevol parell), i **transitiva**.

Una **relación de orden** *es una relación binaria en un conjunto A que es* **reflexiva**, **antisimétrica** *(si aRb y bRa entonces $a = b$ para cualquier par), y* **transitiva**.

Serien exemples la relació $\leq$ entre nombres o la relació $\subseteq$ entre conjunts. Per exemple, les relacions $R_\leq$ en $\mathbb{Z}_3^+$ i $R_\subseteq$ en $2^{\mathbb{Z}_1^+} = \{\emptyset,\{0\},\{1\},\{0,1\}\}$ serien relacions d'ordre i tindrien com a matrius característiques,

Serían ejemplos la relación $\leq$ entre números o la relación $\subseteq$ entre conjuntos. Por ejemplo, las relaciones $R_\leq$ en $\mathbb{Z}_3^+$ y $R_\subseteq$ en $2^{\mathbb{Z}_1^+} = \{\emptyset,\{0\},\{1\},\{0,1\}\}$ serían relaciones de orden y tendrían como matrices características,

$$\chi_{R_\leq} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \chi_{R_\subseteq} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

En canvi, les relacions estrictes $<$ i $\subset$ no són relacions d'ordre ja que no es compleix la propietat **reflexiva**.

En cambio, las relaciones estrictas $<$ y $\subset$ no son relaciones de orden ya que no se cumple la propiedad **reflexiva**.

Una **aplicació**, f, del conjunt A al conjunt B és una relació on **cada** element de A està relacionat amb un **únic** element de B. El conjunt A s'anomena **conjunt de partida** o **domini** de f, $\mathrm{Dom}(f)$, i el conjunt B s'anomena **conjunt d'arribada** o **codomini** de f, $\mathrm{Cod}(f)$. Per indicar que f és una aplicació de A a B escrivim

Una **aplicación**, *f, del conjunto A al conjunto B es una relación donde* **cada** *elemento de A está relacionado con un* **único** *elemento de B. El conjunto A se denomina* **conjunto de partida** *o* **dominio** *de f, $\mathrm{Dom}(f)$, y el conjunto B se denomina* **conjunto de llegada** *o* **codominio** *de f, $\mathrm{Cod}(f)$. Para indicar que f es una aplicación de A a B escribimos*

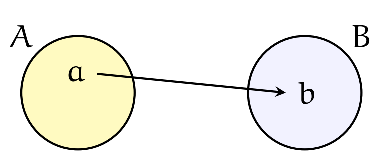
$$f : A \longrightarrow B$$

Si $a \in A$ està relacionat amb $b \in B$ segons f escrivim $f(a) = b$ en lloc de afb. Diem que b és la **imatge** de a o, equivalentment, que a és **antiimatge** de b. El **conjunt imatge** o **rang** de f, $\mathrm{Im}(f)$, és el subconjunt de B format per aquells elements que tenen antiimatge en A.

Si $a \in A$ está relacionado con $b \in B$ según f escribimos $f(a) = b$ en lugar de afb. Decimos que b es la **imagen** *de a o, de manera equivalente, que a es* **antiimagen** *de b. El* **conjunto imagen** *o* **rango** *de f, $\mathrm{Im}(f)$, es el subconjunto de B formado por aquellos elementos que tienen antiimagen en A.*

Gràficament, dibuixem una fletxa des de a fins a b per a indicar que $f(a) = b$.

Gráficamente, dibujamos una flecha desde a hasta b para indicar que $f(a) = b$.



Aquesta mateixa representació gràfica es pot fer servir també per a relacions i per a correspondències (es dibuixa una fletxa de a a b si aRb).

Esta misma representación gráfica se puede utilizar también para relaciones *y para* correspondencias *(se dibuja una flecha de a a b si aRb).*

Una aplicació és **injectiva** si tot element del conjunt imatge té, com a molt, una única antiimatge. Una aplicació és **suprajectiva** o **exhaustiva** si no hi ha elements en el conjunt imatge sense antiimatge. Diem que una aplicació és **bijectiva** si és alhora **injectiva** i **exhaustiva**.

Una aplicación es **inyectiva** *si todo elemento del conjunto imagen tiene, como mucho, una única antiimagen. Una aplicación es* **suprayectiva**, **sobreyectiva** *o* **exhaustiva** *si no hay elementos en el conjunto imagen sin antiimagen. Decimos que una aplicación es* **biyectiva** *si es a la vez* **inyectiva** *y* **exhaustiva**.

Tot element del domini d'una aplicació f ha de tenir imatge i aquesta ha de ser única (per definició d'aplicació). Els elements del codomini poden no tenir antiimatge, tenir una única antiimatge o tenir-ne més d'una. Tots en tindran una o cap si l'aplicació és injectiva. I tots en tindran una o més si f és exhaustiva.

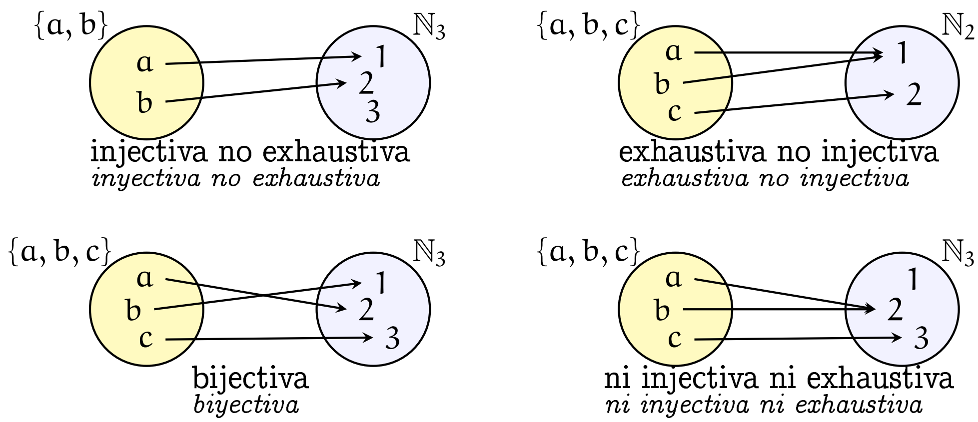
Todo elemento del dominio de una aplicación f ha de tener imagen y ésta tiene que ser única (por definición de aplicación). Los elementos del codominio pueden no tener antiimagen, tener una única antiimagen o tener más de una. Todos tendrán una antiimagen o cap si la aplicación es inyectiva. Y todos tendrán una antiimagen o más si f es exhaustiva.

Els elements de $\mathrm{Im}(f)$ han de tenir exactament una única antiimatge si f és bijectiva o simplement injectiva. I una o més si és exhaustiva.

Los elementos de $\mathrm{Im}(f)$ tienen que tener exactamente una única antiimagen si f es biyectiva o simplemente inyectiva. Y una o más si es exhaustiva.

Alguns exemples d'aplicacions sobre conjunts petits es mostren gràficament a continuació.

Algunos ejemplos de aplicaciones sobre conjuntos pequeños se muestran gráficamente a continuación.



La **composició** de les aplicacions $f : A \to B$ i $g : B \to C$, és una aplicació $f \circ g : A \to C$ de manera que $f \circ g(a) = g(f(a))$ per a tot $a \in A$.

La **composición** *de las aplicaciones* $f : A \to B$ *y* $g : B \to C$, *es una aplicación* $f \circ g : A \to C$ *de manera que* $f \circ g(a) = g(f(a))$ *para todo* $a \in A$.

Donada una aplicació $f : A \to B$, definim la **inversa**, f^{-1}, per a qualsevol element $y \in B$ com a

Dada una aplicación $f : A \to B$, *definimos la* **inversa**, f^{-1}, *para cualquier elemento* $y \in B$ *como*

$$f^{-1}(y) = \{x \in A \mid \ f(x) = y\}.$$

Aquesta inversa es pot veure com una relació entre B i A que no és necessàriament una aplicació. De fet, la **relació inversa** es pot definir per a qualsevol relació exactament de la mateixa manera:

Esta inversa se puede ver como una relación entre B *y* A *que no es necesariamente una aplicación. De hecho, la* **relación inversa** *se puede definir para cualquier relación exactamente de la misma manera:*

Donada una relació, $R \subseteq A \times B$, la **relació inversa**, $R^{-1} \subseteq B \times A$, es defineix de manera que per a tot parell d'elements, $a \in A$, $b \in B$,

Dada una relación, $R \subseteq A \times B$, *la* **relación inversa**, $R^{-1} \subseteq B \times A$, *se define de manera que para todo par de elementos,* $a \in A$, $b \in B$,

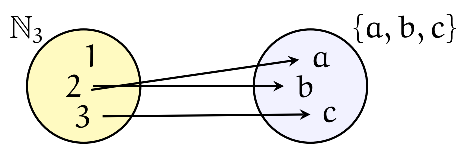
$$bR^{-1}a \ \ \textbf{sii} \ \ aRb,$$

Aquesta relació inversa es pot representar gràficament a partir de la representació gràfica de la relació (directa) només invertint el sentit de les fletxes.

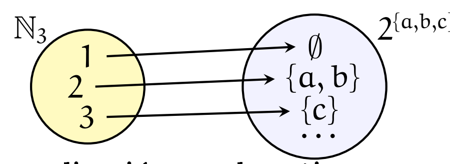
Esta relación inversa se puede representar gráficamente a partir de la representación gráfica de la relación (directa) tan solo invirtiendo el sentido de las flechas.

En el cas particular de les aplicacions, la inversa també es pot veure com una aplicació entre B i el conjunt potència de A, $\mathcal{P}(A)$. Per exemple, la inversa de l'últim dels quatre exemples anteriors es podria veure com a una relació, $f^{-1} \subseteq \mathbb{N}_3 \times \{a, b, c\}$, o també com una aplicació, $f^{-1} : \mathbb{N}_3 \longrightarrow 2^{\{a,b,c\}}$.

En el caso particular de las aplicaciones, la inversa también se puede ver como una aplicación entre B *y el conjunto potencia de* A, $\mathcal{P}(A)$. *Por ejemplo, la inversa del último de los cuatro ejemplos anteriores se podría ver como una relación,* $f^{-1} \subseteq \mathbb{N}_3 \times \{a, b, c\}$, *o también como una aplicación,* $f^{-1} : \mathbb{N}_3 \longrightarrow 2^{\{a,b,c\}}$.



correspondència (relació), no aplicació
correspondencia (relación), no aplicación



aplicació no exhaustiva
aplicación no exhaustiva

Donada qualsevol aplicació $f : A \rightarrow B$ no exhaustiva, es pot definir trivialment una altra aplicació exhaustiva només canviant el codomini pel conjunt imatge de f. És a dir, $f' : A \rightarrow B' \subseteq B$ on $B' = \mathrm{Im}(f) = \{y \in B \mid f(x) = y$ per a algun $x\}$.

Dada cualquier aplicación $f : A \rightarrow B$ *no exhaustiva, se puede definir trivialmente otra aplicación exhaustiva sólo cambiando el codominio por el conjunto imagen de* f. *Es decir,* $f' : A \rightarrow B' \subseteq B$ *donde* $B' = \mathrm{Im}(f) = \{y \in B \mid f(x) = y$ *para algún* $x\}$.

La inversa de l'últim exemple serà exhaustiva si considerem el conjunt $\{\emptyset, \{a, b\}, \{c\}\}$ com a codomini en lloc de la totalitat del conjunt potència.

La inversa del último ejemplo será exhaustiva si consideramos el conjunto $\{\emptyset, \{a, b\}, \{c\}\}$ *como codominio en lugar de la totalidad del conjunto potencia.*

Donada qualsevol aplicació $f : A \rightarrow B$ **injectiva**, sempre podrem definir la seua **aplicació inversa**, $f^{-1} : \mathrm{Im}(f) \rightarrow A$ com a $f^{-1}(y) = x$ si i només si $f(x) = y$.

Dada cualquier aplicación $f : A \rightarrow B$ **inyectiva**, *siempre podremos definir su* **aplicación inversa**, $f^{-1} : \mathrm{Im}(f) \rightarrow A$ *como* $f^{-1}(y) = x$ *si i sólo si* $f(x) = y$.

1.1.5 Aplicacions i cardinalitat

Aplicaciones y cardinalidad

A partir de les definicions es pot arribar a una relació entre les cardinalitats dels dominis i codominis, i el tipus d'aplicació que hi ha entre ells.

A partir de las definiciones se puede llegar a una relación entre las cardinalidades de los dominios y codominios, y el tipo de aplicación que hay entre ellos.

Diem que $|A| \leq |B|$ **sii** $\exists f : A \to B$ injectiva.

Decimos que $|A| \leq |B|$ **sii** $\exists f : A \to B$ *inyectiva.*

Diem que $|A| \geq |B|$ **sii** $\exists f : A \to B$ exhaustiva.

Decimos que $|A| \geq |B|$ **sii** $\exists f : A \to B$ *exhaustiva.*

Diem que $|A| = |B|$ **sii** $\exists f : A \to B$ bijectiva (**regla de la bijecció**)

Decimos que $|A| = |B|$ **sii** $\exists f : A \to B$ *biyectiva (***regla de la biyección***)*

Un conjunt té **cardinalitat** k si es pot trobar una bijecció entre ell i el conjunt $\mathbb{N}_k$.

Un conjunto tiene **cardinalidad** k *si se puede encontrar una biyección entre él y el conjunto* $\mathbb{N}_k$*.*

1.2 Comptatge de conjunts d'elements

Conteo de conjuntos de elementos

Hi ha una sèrie de regles que permeten relacionar les cardinalitats de conjunts diferents de manera que pot resultar relativament senzill deduir la cardinalitat d'un conjunt a partir de la cardinalitat de l'altre.

Hay una serie de reglas que permiten relacionar las cardinalidades de conjuntos diferentes de manera que puede resultar relativamente sencillo deducir la cardinalidad de un conjunto a partir de la cardinalidad del otro.

regla de la bijecció: el cardinal de dos conjunts entre els quals existeix una bijecció és el mateix.

regla de la biyección*: el cardinal de dos conjuntos entre los que existe una biyección es el mismo.*

regla de la divisió: si hi ha una aplicació exhaustiva $f : A \to B$ de manera que tot element de B té exactament k antiimatges (aplicació k a 1), aleshores $|A| = k|B|$.

regla de la división*: si hay una aplicación exhaustiva* $f : A \to B$ *de manera que todo elemento de* B *tiene exactamente* k *antiimágenes (aplicación* k *a 1), entonces* $|A| = k|B|$.

principi de les caixes: Si $|A| > |B|$ aleshores per a qualsevol aplicació, $f : A \rightarrow B$, almenys un element de B ha de tenir més d'una antiimatge. Equivalentment, almenys 2 elements de A han de tenir la mateixa imatge.

principio de las cajas: *Si $|A| > |B|$ entonces para cualquier aplicación, $f : A \rightarrow B$, al menos un elemento de B tiene que tener más de una antiimagen. De manera equivalente, al menos 2 elementos de A tienen que tener la misma imagen.*

principi de les caixes generalitzat: Si $|A| > k|B|$ aleshores per a qualsevol aplicació, $f : A \rightarrow B$, almenys un element de B ha de tenir més de k antiimatges. Equivalentment, almenys $k+1$ elements de A han de tenir la mateixa imatge.

principio de las cajas generalizado: *Si $|A| > k|B|$ entonces para cualquier aplicación, $f : A \rightarrow B$, al menos un elemento de B tiene que tener más de k antiimágenes. De manera equivalente, al menos $k+1$ elementos de A tienen que tener la misma imagen.*

regla del producte: el cardinal del producte cartesià de dos conjunts és el producte dels seus cardinals, $|A \times B| = |A| \cdot |B|$.

regla del producto: *el cardinal del producto cartesiano de dos conjuntos es el producto de sus cardinales, $|A \times B| = |A| \cdot |B|$.*

regla de la suma: el cardinal de la unió de dos conjunts disjunts és la suma de les seues cardinalitats.

regla de la suma: *el cardinal de la unión de dos conjuntos disjuntos es la suma de sus cardinalidades.*

principi d'inclusió-exclusió o **regla de la suma generalitzada**: el cardinal de la unió de dos conjunts és la suma de les seues cardinalitats menys la cardinalitat de la seua intersecció.

principio de inclusión-exclusión *o* **regla de la suma generalizada**: *el cardinal de la unión de dos conjuntos es la suma de sus cardinalidades menos la cardinalidad de su intersección.*

1.3 Variacions, permutacions i combinacions

Variaciones, permutaciones y combinaciones

El conjunt de les **variacions amb repetició de m elements agafats de n en n**, VR_m^n, és el conjunt format per totes les possibles n-tuples de $\mathbb{N}_m$. O siga, $\mathbb{N}_m^n$. El seu cardinal s'obté, per tant, com a

El conjunto de las **variacionnes con repetición de m elementos tomados de n en n**, *VR_m^n, es el conjunto formado por todas las posibles n-tuplas de $\mathbb{N}_m$. O sea, $\mathbb{N}_m^n$. Su cardinal se obtiene por tanto como*

$$|VR_m^n| = |\mathbb{N}_m|^n = m^n$$

en aplicar el **principi del producte**.

al aplicar el **principi del producte**.

A causa del principi de la bijecció, identificarem VR_m^n amb n-tuples sobre qualsevol altre conjunt de cardinalitat m. També de vegades abusarem de la notació i usarem VR_m^n per a referir-nos en realitat al seu cardinal.

Debido al principio de la biyección, identificaremos VR_m^n con n-tuplas sobre cualquier otro conjunto de cardinalidad m. A veces también abusaremos de la notación y usaremos VR_m^n para referir-nos en realidad a su cardinal.

El conjunt de les **variacions sense repetició de m elements agafats de n en n**, V_m^n, és el subconjunt de VR_m^n que conté només les n-tuples sense elements repetits. El seu cardinal s'obté també mitjançant la regla del producte com a

El conjunto de las **variaciones sin repetición de m elementos tomados de n en n**, *V_m^n, es el subconjunto de VR_m^n que contiene sólo las n-tuplas sin elementos repetidos. Su cardinal se obtiene también mediante la* **regla del producto** *como*

$$|V_m^n| = |\mathbb{N}_m| \cdot (|\mathbb{N}_m| - 1) \cdots (|\mathbb{N}_m| - n + 1) = m^{\underline{n}}$$

on $m^{\underline{n}} = m(m-1)\cdots(m-n+1)$ és la **potència decreixent n-èsima de m** (o també **factorial decreixent d'ordre n de m**) sempre que $n \leq m$. En el cas en què $n > m$ el cardinal de V_m^n és òbviament zero.

donde $m^{\underline{n}} = m(m-1)\cdots(m-n+1)$ es la **potencia decreciente n-ésima de m** *(o también* **factorial decreciente de orden n de m***) siempre que $n \leq m$. En el caso en que $n > m$ el cardinal de V_m^n es óbviamente cero.*

El conjunt de les **permutacions de m elements**, P_m, és el conjunt V_m^m. És a dir, totes les m-tuples de $\mathbb{N}_m$ sense elements repetits. Aquest conjunt es correspon amb totes les maneres possibles d'ordenar els m primers nombres naturals (o qualsevol altre conjunt de cardinalitat m). El cardinal de P_m és determinat per

El conjunto de las **permutaciones de m elementos**, P_m, *es el conjunto* V_m^m. *Es decir, todas las m-tuplas de $\mathbb{N}_m$ sin elementos repetidos. Este conjunto se corresponde con todas las formas posibles de ordenar los m primeros números naturales (o cualquier otro conjunto de cardinalidad m). El cardinal de P_m viene dado por*

$$|P_m| = |V_m^m| = m^{\underline{m}} = m!$$

Alternativament, una permutació dels m elements d'un conjunt A es pot veure com una aplicació bijectiva de A en ell mateix. O altrament dit, hi ha una bijecció entre el conjunt P_m i el conjunt de totes les aplicacions bijectives de la forma $f : A \longrightarrow A$ si $|A| = m$.

Alternativamente, una permutacfión de los m elementos de un conjunto A se puede ver como una aplicación biyectiva de A en sí mismo. O dicho de otra manera, existe una biyección entre el conjunto P_m y el conjunto de todas las aplicaciones biyectivas de la forma $f : A \longrightarrow A$ si $|A| = m$.

El conjunt de les **permutacions amb repetició de m elements que es repeteixen $(k_1, k_2, \ldots, k_m)$ vegades**, $PR_{(k_1,k_2,\ldots,k_m)}$, és el conjunt format per les $(k_1 + k_2 + \cdots + k_m)$-tuples de $\mathbb{N}_m$ que contenen k_i repeticions de l'element i, per a $i = 1, \ldots, m$.

El conjunto de las **permutaciones con repetición de m elementos que se repiten $(k_1, k_2, \ldots, k_m)$ veces**, $PR_{(k_1,k_2,\ldots,k_m)}$, *es el conjunto formado por las $(k_1 + k_2 + \cdots + k_m)$-tuplas de $\mathbb{N}_m$ que contienen k_i repeticiones del elemento i, para $i = 1, \ldots, m$.*

El cardinal de $PR_{(k_1,k_2,\ldots,k_m)}$ es pot obtenir a partir del cardinal de les permutacions de $(k_1 + k_2 + \cdots + k_m)$ elements, $P_{(k_1+k_2+\cdots+k_m)}$, en aplicar el **principi de la divisió** com a

El cardinal de $PR_{(k_1,k_2,\ldots,k_m)}$ se puede obtener a partir del cardinal de las permutaciones de $(k_1 + k_2 + \cdots + k_m)$ elementos, $P_{(k_1+k_2+\cdots+k_m)}$, aplicando el **principio de la división** *como*

$$|PR_{(k_1,k_2,\ldots,k_m)}| = \frac{(k_1 + k_2 + \cdots + k_m)!}{k_1!\, k_2! \cdots k_m!} = \frac{(\sum_{i=1}^{m} k_i)!}{\prod_{i=1}^{m} k_i!} = (k_1, \ldots, k_m)!$$

on $(k_1, \ldots, k_m)!$ és el que es coneix com a **coeficients multinomials**

donde $(k_1, \ldots, k_m)!$ es lo que se conoce como **coeficientes multinomiales**.

Per a justificar l'anterior podem raonar de la següent manera. Suposem que les k_i diferents ocurrències de cada element, i, en cada permutació amb repetició de $PR_{(k_1,k_2,\ldots,k_m)}$ es pogueren distingir. En aquest cas, el cardinal que cerquem seria

Para justificar lo anterior podemos razonar de la siguiente manera. Supongamos que las k_i diferentes ocurrencias de cada elemento, i, en cada permutación con repetición de $PR_{(k_1,k_2,\ldots,k_m)}$ se pudiesen distinguir. En ese caso, el cardinal que buscamos sería

$$(\sum_{i=1}^{m} k_i)!$$

Per exemple, les permutacions amb repetició, $PR_{(3,2,1)}$ sobre el conjunt $\{a, b, c\}$ les podríem identificar amb les permutacions, P_6 sobre el conjunt $\{a_1, a_2, a_3, b_1, b_2, c_1\}$. Si considerem una de les permutacions com per exemple

Por ejemplo las permutaciones con repetición de $PR_{(3,2,1)}$ sobre el conjunto $\{a, b, c\}$ las podríamos identificar con las permutaciones, P_6 sobre el conjunto $\{a_1, a_2, a_3, b_1, b_2, c_1\}$. Si consideramos una de las permutaciones como por ejemplo

$$(c_1, b_2, a_3, a_1, b_1, a_2),$$

veiem que es poden permutar les a de $P_3 = 6$ maneres, les b de $P_2 = 2$ maneres i les c de $P_1 = 1$ manera, sense que canvie la permutació amb repetició.

vemos que se pueden permutar las a de $P_3 = 6$ maneras, las b de $P_2 = 2$ maneras y las c de $P_1 = 1$ manera, sin que cambie la permutación con repetición.

En general, la diferència entre distingir o no distingir entre les k_i ocurrències de l'element i és que cada tupla del segon cas es correspon amb $k_i!$ tuples del primer cas, que serien totes les maneres de permutar les k_i ocurrències allà on foren. Com que això es pot fer amb els m elements alhora, hi haurà una correspondència $\prod_{i=1}^{m} k_i!$ a 1 entre els conjunts $P_{(k_1+k_2+\cdots+k_m)}$ i $PR_{(k_1,k_2,\ldots,k_m)}$, per la qual cosa es pot aplicar el **principi de la divisió**.

En general, la diferencia entre distinguir o no entre las k_i ocurrencias del elemento i es que cada tupla del segundo caso se corresponde con $k_i!$ tuplas del primero, que serían todas las formas de permutar las k_i ocurrencias allí donde estuviesen. Como esto mismo se puede hacer on los m elementos a la vez, habrá también una correspondencia $\prod_{i=1}^{m} k_i!$ a 1 entre los conjuntos $P_{(k_1+k_2+\cdots+k_m)}$ y $PR_{(k_1,k_2,\ldots,k_m)}$ por lo que se puede aplicar el **principio de la división**.

El conjunt de les **combinacions de m elements agafats de n en n**, C_m^n, és format per tots els subconjunts de $\mathbb{N}_m$ de cardinalitat n i es correspon amb totes les maneres de triar n elements d'entre m possibles sense que importe l'ordre.

El conjunto de las **combinaciones de m elementos tomados de n en n**, *C_m^n, está formado por todos los subconjuntos de $\mathbb{N}_m$ de cardinalidad n y se corresponde con todas las formas de elegir n elementos de entre m posibles sin que importe el orden.*

Si pensem en els vectors característics dels conjunts que formen C_m^n, veiem que es tracta de m-tuples en $\mathbb{Z}_1^+ = \{0, 1\}$ que contenen exactament n uns i $m - n$ zeros. Per tant, també hi ha una bijecció entre C_m^n i $PR_{(m-n,n)}$, per la qual cosa podem obtenir la cardinalitat de C_m^n com a

Si pensamos en los vectores característicos de los conjuntos que forman C_m^n, vemos que se trata de m-tuplas en $\mathbb{Z}_1^+ = \{0, 1\}$ que contienen exactamente n unos y $m - n$ ceros. Por lo tanto, también hay una biyección entre C_m^n y $PR_{(m-n,n)}$ por lo que podemos obtener la cardinalidad de C_m^n como

$$|C_m^n| = \frac{m!}{n!(m-n)!} = \binom{m}{n}$$

on $\binom{m}{n}$ representa el que es coneix com a **coeficient binomial** o **nombre combinatori** d'ordre (m, n) i es llegeix com "m sobre n".

donde $\binom{m}{n}$ representa lo que se conoce como **coeficiente binomial** *o* **número combinatorio** *de orden (m, n) y se lee como "m sobre n".*

Algunes propietats interessants dels coeficients binomials són

Algunas propiedades interesantes de los coeficientes binomiales son

$$\binom{m}{n} = \binom{m}{m-n} \qquad \sum_{k=1}^{m} \binom{m}{k} = 2^m \qquad \binom{m}{n} = \binom{m-1}{n} + \binom{m-1}{n-1}$$

La primera és trivial. La segona es pot deduir a partir del **principi de la suma** i del fet que hi ha una bijecció entre $\mathcal{P}(\mathbb{N}_m)$ (subconjunts de $\mathbb{N}_m$) i VR_2^m (els seus vectors característics). La tercera és un poc més complicada. La seua deducció així com els casos particulars es deixen com a exercici.

La primera es trivial. La segunda se puede deducir a partir del **principio de la suma** *y del hecho de que existe una biyección entre $\mathcal{P}(\mathbb{N}_m)$ (subconjuntos de $\mathbb{N}_m$) y VR_2^m (sus vectores característicos). La tercera es un poco más compleja. Su deducción así como los casos particulares se dejan como ejercicio.*

El conjunt de les **combinacions amb repetició de m elements agafats de n en n**, CR_m^n, és format per tots els multiconjunts sobre $\mathbb{N}_m$ de cardinalitat n i es correspon amb totes les maneres de triar n elements d'entre m possibles sense que importe l'ordre, però amb l'opció de poder elegir el mateix element més d'una vegada.

El conjunto de las **combinaciones con repetición de m elementos tomados de n en n**, CR_m^n, *está formado por todos los multiconjuntos sobre $\mathbb{N}_m$ de cardinalidad n y se corresponden con todas las formas de elegir n elementos de entre los m posibles sin que importe el orden, pero con la opción de poder elegir el mismo elemento más de una vez.*

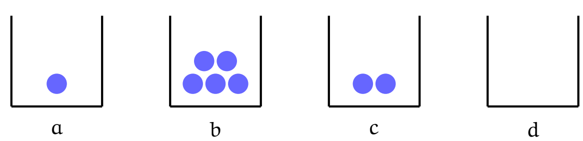
Una bijecció interessant és que CR_m^n es correspon també amb les differents formes de distribuir n objectes indistingibles entre m contenidors il·limitats i distingibles (o numerats). Donat un multiconjunt particular, cada contenidor és un element de $\mathbb{N}_m$ i els objectes que conté són les vegades que aquest element apareix en el multiconjunt.

Una biyección interesante es que CR_m^n se corresponde también con las diferentes formas de distribuir n objetos indistinguibles entre m contenedores ilimitados y distinguibles (o numerados). Dado un multiconjunto particular, cada contenedor es un elemento de $\mathbb{N}_m$ y los objetos que contiene son las veces que ese elemento aparece en el multiconjunto.

Per exemple, la figura següent il·lustra un cas particular de multiconjunts de 4 elements.

Por ejemplo, la siguiente figura ilustra un caso particular de multiconjuntos de 4 elementos.

$\{a, b, b, b, b, b, c, c\} \quad \leftrightarrow$



Una altra bijecció més interessant encara entre CR_m^n i un subconjunt de $\mathbb{Z}_1^+$ és la següent. Representem el nombre d'ocurrències de l'i-èsim element, k_i, en unari. És a dir, com una k_i-tupla d'uns, $(\underbrace{1, 1, \ldots, 1}_{k_i \text{ vegades}})$. Si concatenem les m k_i-tuples, però les separem amb $m-1$ zeros tenim una representació que és única per a tots els possibles multiconjunts de talla $n = k_1 + k_2 + \cdots + k_m$,

Otra biyección aún más interesante entre CR_m^n y un subconjunto de $\mathbb{Z}_1^+$ es la siguiente. Representamos el número de ocurrencias del i-ésimo elemento, k_i, en unario. Es decir, como una k_i-tupla de unos, $(\underbrace{1, 1, \ldots, 1}_{k_i \text{ vegades}})$. Si concatenamos las m k_i-tuplas pero las separamos con $m-1$ ceros tenemos una representación que es única para todos los posibles multiconjuntos de talla $n = k_1 + k_2 + \cdots + k_m$,

$$(\underbrace{1, 1, \ldots, 1}_{k_1}, 0, \underbrace{1, 1, \ldots, 1}_{k_2}, 0, \ldots, 0, \underbrace{1, 1, \ldots, 1}_{k_m}).$$

Això vol dir que podem representar qualsevol multiconjunt de cardinalitat n com una seqüència de $\mathbb{Z}_1^+$ que continga exactament n uns i $m-1$ zeros. I de la mateixa manera, donada qualsevol cadena amb n uns i $m-1$ zeros, es pot interpretar com un multiconjunt de m elements on cada un es pot repetir zero o més vegades però la cardinalitat és n.

Esto quiere decir que podemos representar cualquier multiconjunto de cardinalidad n como una secuencia de $\mathbb{Z}_1^+$ que contenga exactamente n unos y $m-1$ ceros. Y de la misma manera, dada cualquier cadena con n unos y $m-1$ ceros, se puede interpretar como un multiconjunto de m elementos donde cada uno de ellos se puede repetir cero o más veces pero la cardinalidad es n.

En altres paraules, hi ha una bijecció entre les $(n+m-1)$-tuples de $\mathbb{Z}_1^+$ amb n uns i els conjunts de $n+m-1$ elements de talla n, per la qual cosa es pot obtenir el cardinal de CR_m^n com a

En otras palabras, existe una biyección entre las $(n+m-1)$-tuplas de $\mathbb{Z}_1^+$ con n unos y los conjuntos de $n+m-1$ elementos de talla n, por lo que se puede obtener el cardinal de CR_m^n como

$$|CR_m^n| = |C_{n+m-1}^n| = \binom{n+m-1}{n} = \left(\!\!\binom{m}{n}\!\!\right)$$

on $\left(\!\binom{m}{n}\!\right)$ és el que es coneix com a **coeficient multiconjunt**.

donde $\left(\!\binom{m}{n}\!\right)$ es lo que se conoce como **coeficiente multiconjunto**.

1.4 Tècniques visuals de comptatge

Técnicas visuales de conteo

Una forma interessant d'aplicar la regla de la bijecció consisteix a fer-ho entre descripcions visuals o diagramàtiques del que es vol comptar. Exemples interessants són alguns sumatoris que apareixen sovint en computació.

Una forma interesante de aplicar la regla de la biyección consiste en hacerlo entre descripciones visuales o diagramáticas de lo que se quiere contar. Ejemplos interesantes de esto son algunos sumatorios que suelen aparecer en computación.

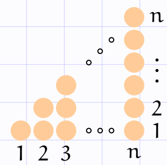
Suma dels n primers enters

Suma de los n primeros enteros

$$S_1(n) = \sum_{i=1}^{n} i = 1 + 2 + \cdots + n$$

Podem representar cada terme del sumatori com una pila de boletes d'altures 1, 2, 3, ... i col·locar-les en fila

Podemos representar cada término del sumatorio como una pila de bolas de alturas 1, 2, 3, ... y colocarlas en fila

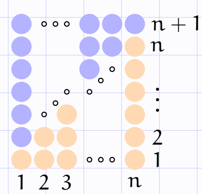


de manera que el valor del sumatori es correspon amb el nombre de boletes. En altres paraules, es pot establir una bijecció entre el valor del sumatori per a cada n i el nombre de boletes en n piles de boletes.

de manera que el valor del sumatorio se corresponde con el número de bolas. En otras palabras, se puede establecer una biyección entre el valor del sumatorio para cada n y el número de bolas en n pilas de bolas.

Però si considerem dues vegades el mateix sumatori i girem la representació gràfica 180º i les ajuntem, tenim que

Pero si consideramos dos veces el mismo sumatorio y giramos la representación gráfica 180º y las juntamos, tenemos que



A partir de la representació gràfica queda clar que

A partir de la representación gráfica queda claro que

$$2S_1(n) = n \cdot (n+1)$$

d'on s'obté

de donde se obtiene

$$\boxed{S_1(n) = \frac{n}{2}(n+1)}$$

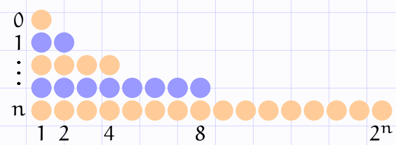
Suma dels $n+1$ dobles successius

Suma de los $n+1$ dobles sucesivos

$$S_d(n) = \sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n$$

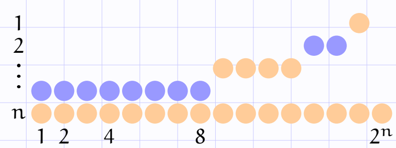
Representarem ara cada un dels $n+1$ termes del sumatori com una fila de boletes de colors alterns i les apilarem

Representaremos ahora cada uno de los $n+1$ términos del sumatorio como una fila de bolas de colores alternos y las apilaremos



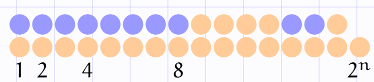
Si ara desplacem les files de la següent manera

Si ahora desplazamos las filas de la siguiente manera



i les compactem totes en només dues files

y las compactamos todas en sólo dos filas



observem clarament que el nombre total de boles és

observamos claramente que el número total de bolas es

$$\boxed{S_d(n) = 2 \cdot 2^n - 1}$$

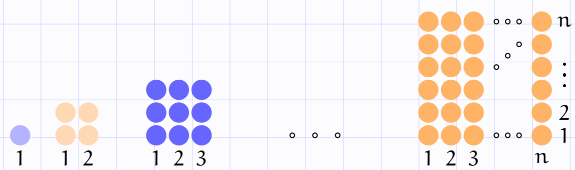
Suma dels n primers quadrats

Suma de los n primeros cuadrados

$$S_2(n) = \sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \cdots + n^2$$

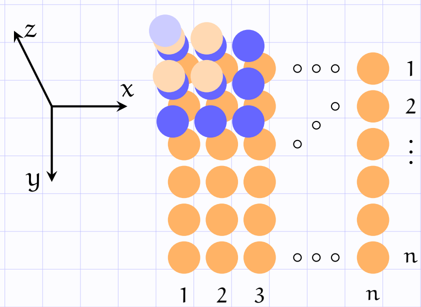
Representem el terme i-èsim del sumatori com una quadrícula de $i \times i$ boletes

Representamos el término i-ésimo del sumatorio como una cuadrícula de $i \times i$ bolas



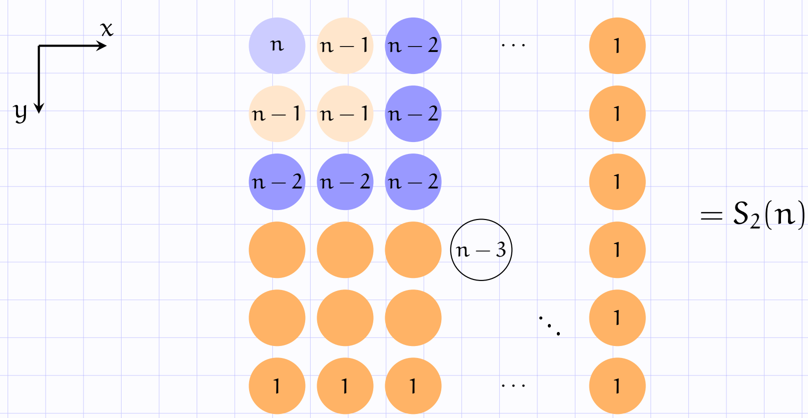
i les apilem totes formant una espècie de piràmide de manera que el seu vèrtex estiga en el cantó superior esquerre.

y las apilamos todas formando una especie de pirámide de manera que su vértice esté en la esquina superior izquierda.



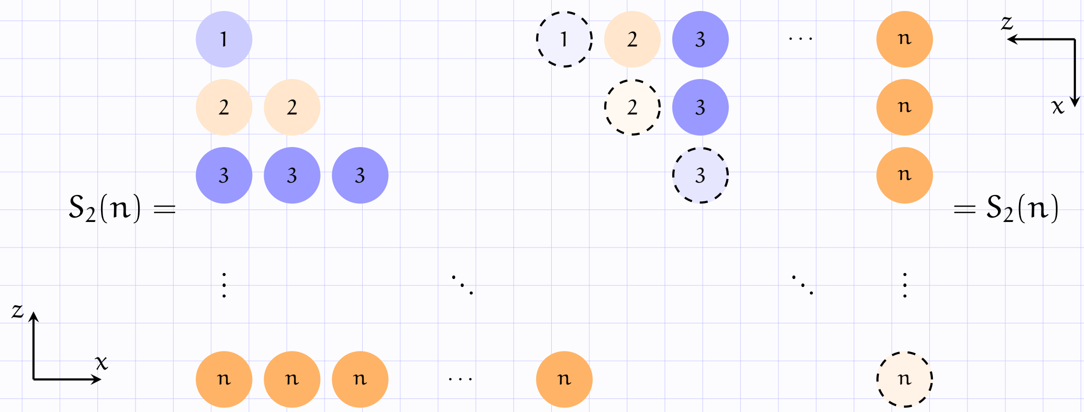
Si mirem aquesta piràmide des de la direcció z i apuntem en cada cercle el nombre de boles en cada posició (x, y), tenim

Si miramos esta pirámide desde la dirección z y apuntamos en cada círculo el número de bolas en cada posición (x, y), tenemos



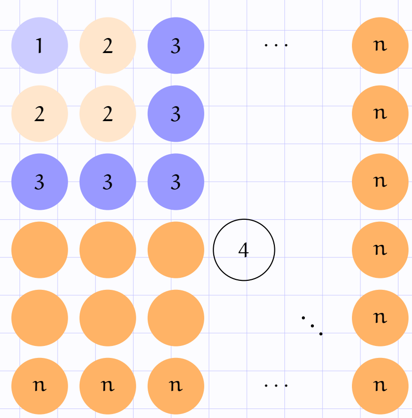
I si fem el mateix però mirant la piràmide des de la direcció y. I, a part, copiem aquest mateix resultat, però canviant files per columnes, obtenim

Y si hacemos lo mismo pero mirando la pirámide desde la dirección y. Y, aparte, copiamos este mismo resultado, pero cambiando filas por columnas, obtenemos



Si superposem ara les dues quadrícules després d'eliminar la diagonal que estava marcada la suma de la qual és exactament $S_1(n)$, arribem a

Si superponemos ahora las dos cuadrículas después de eliminar la diagonal que estaba marcada cuya suma es exactamente $S_1(n)$, llegamos a



$$= 2 \cdot S_2(n) - S_1(n)$$

Si superposem ara a aquesta quadrícula de $n \times n$ boles numerades (o piles de boles) a la que es veu des de direcció z, obtenim una quadrícula de $n \times n$ piles de $n+1$ boles (o boles numerades on totes valen exactament $n+1$). De manera equivalent, podem escriure

Si superponemos ahora a esta cuadrícula de $n \times n$ bolas numeradas (o pilas de bolas) a la que se ve desde la dirección z, obtenemos una cuadrícula de $n \times n$ pilas de $n+1$ bolas (o bolas numeradas donde todas valen exactamente $n+1$). De manera equivalente, podemos escribir

$$n \times n \times (n+1) = S_2(n) + 2 \cdot S_2(n) - S_1(n)$$

d'on s'obté que

de donde se obtiene que

$$3S_2(n) = n^2(n+1) - S_1(n) = (2n+1)S_1(n)$$

i, per tant,

y, por tanto,

$$S_2(n) = \frac{2n+1}{3} S_1(n) = \frac{n}{6}(n+1)(2n+1)$$

Suma dels n primers cubs

Suma de los n primeros cubos

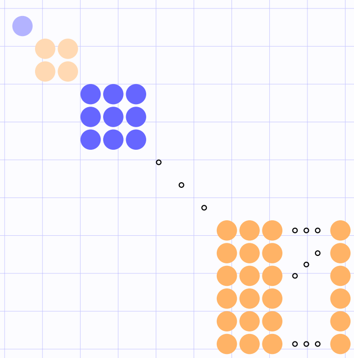
$$S_3(n) = \sum_{i=1}^{n} i^3 = 1^3 + 2^3 + \cdots + n^3$$

Lamentablement no és fàcil generalitzar els procediments anteriors per a aquest sumatori.

Lamentablemente no es fácil generalizar los procedimientos anteriores para este sumatorio.

Considerarem la representació gràfica dels termes del sumatori anterior, $S_2(n)$, en la pàgina 26 i tornarem a dibuixar-los però en diagonal.

Consideraremos la representaciòn gráfica de los términos del sumatorio anterior, $S_2(n)$, en la página 26 y volveremos a dibujarlos pero en diagonal.



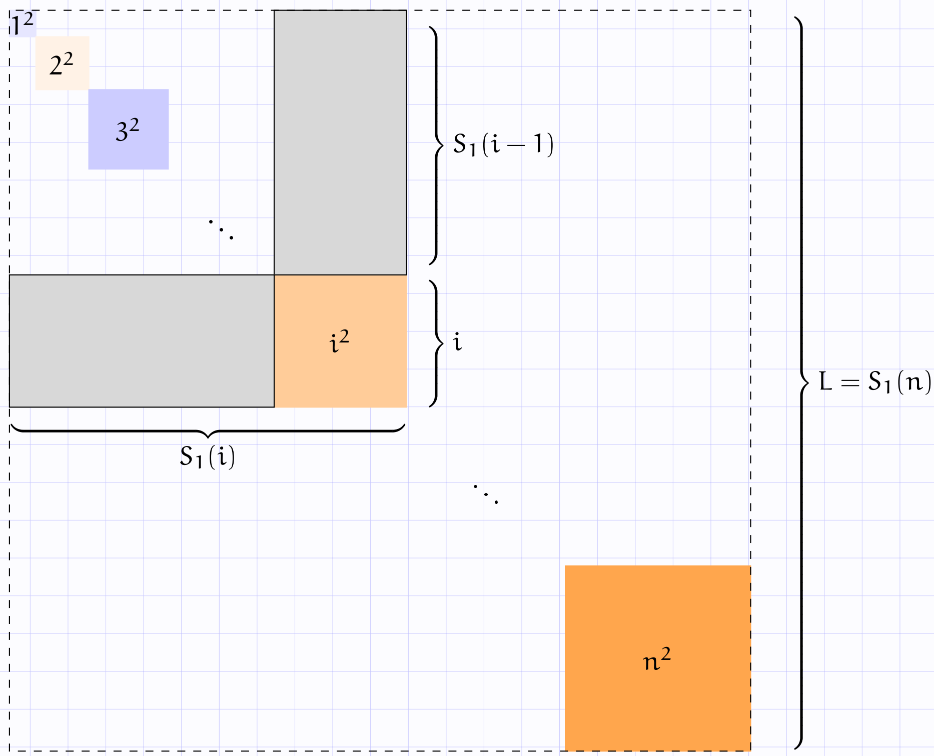
És clar que podem inscriure la representació anterior en una quadrícula de talla $L \times L$ on

Está claro que podemos inscribir la representación anterior en una cuadrícula de talla $L \times L$ donde

$$L = 1 + 2 + 3 + \cdots + n = S_1(n).$$

Representarem els termes com a àrees i associarem al terme i-èsim l'àrea rectangular que està exactament damunt i també la que queda a la seua esquerra (pintades en gris en la figura).

Representaremos los términos como áreas y asociaremos al término i-ésimo el área rectangular que está exactamente encima y también la que queda a su izquierda (pintadas en gris en la figura).



L'àrea associada a l'i-èsim terme aleshores és i^2 més dues vegades l'àrea grisa que és

El área asociada al i-ésimo término entonces es i^2 más dos veces el área gris que es

$$iS_1(i-1)$$

per la qual cosa la i-èsima àrea és

por lo que la i-ésima área es

$$i^2 + 2iS_1(i-1) = i^2 + i^2(i-1) = i^3$$

O siga, que a cada terme li estem associant una àrea que és exactament igual a l'i-èsim cub. I com que la suma d'aquests termes estesos per una banda ha de ser igual a la suma dels cubs, però per l'altra ha de ser igual a l'àrea total de la quadrícula, que és L^2, resulta que

O sea, que a cada término le estamos asociando un área que es exactamente igual al i*-ésimo cubo. Y como la suma de estos términos extendidos por una parte ha de ser igual a la suma de los cubos, pero por otra ha de ser igual al área total de la cuadrícula, que es* L^2*, resulta que*

$$S_3(n) = S_1(n)^2 = \frac{n^2}{4}(n+1)^2$$

1.5 Problemes resolts i comentats
Problemas resueltos y comentados

Problema 1.1: [correspondenciaRelacio]

És el mateix **correspondència** que **relació no trivial**?

*¿Es lo mismo **correspondencia** que **relación no trivial**?*

La pregunta són en realitat dues: 1) Tota relació no trivial és correspondència? i 2) Tota correspondència és relació no trivial?

La pregunta son en realidad dos: 1) ¿Toda relación no trivial es correspondencia? y 2) ¿Toda correspondencia es relación no trivial?

La resposta a 1) és **sí**, ja que almenys algú ha d'estar relacionat amb algú. Si no, seria trivial. La resposta a 2) és **no**, perquè una correspondència en què tots es relacionen amb tots és trivial per definició.

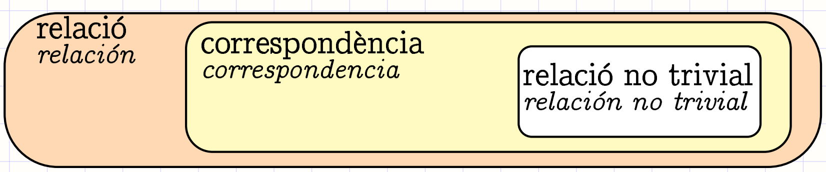
*La respuesta a 1) es **si**, ya que al menos algún elemento ha de estar relacionado con algún otro. Si no, sería trivial. La respuesta a 2) es **no**, porque una correspondencia en la que todos se relacionan con todos es trivial por definición.*

Com a conseqüència, correspondència és més general que relació no trivial. Tota relació no trivial és correspondència, però no al revés.

Como consecuencia, correspondencia es más general que relación no trivial. Toda relación no trivial es correspondencia, pero no al revés.

La situació es mostra gràficament en la figura.

La situación se muestra gráficamente en la figura.



Donats un parell de conjunts, A, B, l'única relació entre ells que no és correspondència és la relació buida, $\emptyset$. I l'única correspondència que és relació trivial és la seua relació complementària, $A \times B$.

Dados un par de conjuntos, A, B, la única relación entre ellos que no es correspondencia es la relación vacía, $\emptyset$. Y la única correspondencia que es relación trivial es su relación complementaria, $A \times B$.

Problema 1.2: [unioDeTres]

Quina és la cardinalitat de $A \cup B$ en funció de les cardinalitats de A i B? I la de $A \cup B \cup C$?

¿Cual es la cardinalidad de $A \cup B$ en función de las cardinalidades de A y B? ¿Y la de $A \cup B \cup C$?

Pel principi d'inclusió-exclusió tenim que

Por el principo de inclusión-exclusión tenemos que

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Si apliquem l'anterior dues vegades, s'arriba a

Si aplicamos lo anterior dos veces se llega a

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C|$$

Problema 1.3: [dabaleArroz]

La frase

La frase

"Dábale arroz a la zorra el abad"

és una frase palindròmica (les lletres són les mateixes si es llegeixen d'esquerra a dreta o de dreta a esquerra) formada per 7 paraules i 25 lletres (sense comptar espais).

es una frase palíndroma (las letras son las mismas si se leen de izquierda a derecha o de derecha a izquierda) formada por 7 palabras y 25 letras (sin contar espacios).

a) Si no considerem espais, quantes frases palindròmiques diferents podem formar amb les mateixes lletres?

a) Si no considamos espacios, ¿cuántas frases palíndromas diferentes podemos formar con las mismas letras?

b) I si considerem espais?

b) ¿Y si consideramos espacios?

c) I quantes frases palindròmiques de 7 paraules podríem formar?

c) ¿Y cuántas frases palíndromas de 7 palabras podríamos formar?

d) I si només considerem paraules de 4 i 5 lletres?

d) ¿Y si sólo conisderamos palabras de 4 y 5 letras?

e) I si considerem paraules de 3, 4 i 5 lletres?

e) ¿Y si consideramos palabras de 3, 4 y 5 letras?

a) La lletra central de la frase és la ℓ i és l'única que apareix un nombre imparell de vegades. Per tant, qualsevol frase palindròmica amb les mateixes lletres haurà de tenir-la en el centre.

a) La letra central de la frase es la ℓ y es la única que aparece un número impar de veces. Por lo tanto, cualquier frase palíndroma con las mismas letras tendrá que tenerla en el centro.

A cada costat de la lletra central tenim 8 lletres diferents, $(d, a, b, e, r, o, z, \ell)$, que es repeteixen $(1, 4, 1, 1, 2, 1, 1, 1)$ vegades, respectivament. 12 en total.

A cada lado de la letra central tenemos 8 letras distintas, $(d, a, b, e, r, o, z, \ell)$, que se repiten $(1, 4, 1, 1, 2, 1, 1, 1)$ veces, respectivamente. 12 en total.

La posició central ha quedat fixada i les lletres de la segona meitat han de ser les mateixes que les de la primera però en sentit invers.

La posición central ha quedado fijada y las letras de la segunda mitad tienen que ser las mismas que las de la primera pero en sentido inverso.

Per tot això, el nombre de frases palindròmiques que es poden formar és determinat per les **permutacions amb repetició de 8 elements que es repeteixen $(4, 2, 1, 1, 1, 1, 1, 1)$ vegades**.

Por todo ello, el número de frases palíndromas que se pueden formar viene determinado por las **permutaciones con repetición de 8 elementos que se repiten $(4, 2, 1, 1, 1, 1, 1, 1)$ veces.**

$$S_a = |PR_{(4,2,1,1,1,1,1,1)}| = \frac{12!}{4! \cdot 2! \cdot (1!)^6} = \frac{11!}{4} = \boxed{9\,979\,200}$$

b) Si considerem espais (entre qualssevol de les 25 lletres), resulta que hi ha 24 possibles posicions on els podem posar (o no). I les formes diferents de posar espais és determinada aleshores per les **variacions amb repetició de 2 elements agafats de 24 en 24**.

b) Si consideramos espacios (entre cualesquiera de las 25 letras), resulta que hay 24 posibles posiciones donde los podemos poner (o no). Y las formas diferentes de poner espacios viene pues determinada por las **variaciones con repetición de 2 elementos tomados de 24 en 24.**

$$E = |VR_2^{24}| = 2^{24} = 16\,777\,216$$

I pel **principi del producte**, el total de frases palindròmiques amb espais serà

Y por el **principio del producto***, el total de frases palíndromas con espacios será*

$$S_b = \boxed{E \times S_a} = 2^{22} \cdot 11! = \boxed{167\,423\,193\,907\,200}$$

c) Si hi ha 7 paraules és perquè hi ha exactament 6 espais que haurem de posar en algunes de les 24 possibles posicions. Per tant, en la fórmula anterior haurem de canviar les variacions amb repetició per **combinacions sense repetició de 24 elements agafats de 6 en 6**.

c) Si hay 7 palabras es por que hay exactamente 6 espacios que habrá que poner en algunas de las 24 posibles posiciones. Por tanto, en la fórmula anterior habrá que cambiar las variaciones con repetición por **combinaciones sin repetición de 24 elementos tomados de 6 en 6**.

$$E_6 = |C_{24}^6| = \binom{24}{6} = \frac{24^{\underline{6}}}{6!} = \frac{\cancel{24} \cdot 23 \cdot 22 \cdot 21 \cdot \cancel{20}^{\,2} \cdot 19}{\cancel{6} \cdot \cancel{5} \cdot \cancel{4} \cdot 3 \cdot \cancel{2} \cdot 1} = 23 \cdot 22 \cdot 7 \cdot 2 \cdot 19 = 134\,596$$

Amb la qual cosa i també pel **principi del producte** obtenim

Con lo cual y también por el **principio del producto** *obtenemos*

$$S_c = \boxed{E_6 \times S_a} = 23 \cdot 22 \cdot 7 \cdot 2 \cdot 19 \cdot \frac{11!}{4} = 23 \cdot 11 \cdot 7 \cdot 1 \cdot 19 \cdot 11! = \boxed{1\ \ 343\,160\,403\,200}$$

d) En total hi ha 25 lletres. Per tant les úniques opcions són i) 5 paraules de 5 lletres, ii) 5 paraules de 4 lletres i 1 de 5.

d) En total hay 25 letras. Por tanto las únicas opciones son i) 5 palabras de 5 letras, ii) 5 palabras de 4 letras y 1 de 5.

En el cas i) només hi ha una opció per a col·locar els espais: cada 5 lletres.

En el caso i) sólo hay una opción para colocar los espacios: cada 5 letras.

En el cas ii) tenim tantes opcions com posicions relatives puga ocupar la paraula de 5 lletres dins la seqüència de les 6 paraules. És a dir, sis opcions.

En el caso ii) tenemos tantas opciones como posiciones relativas pueda ocupar la palabra de 5 letras dentro de la secuencia de las 6 palabras. Es decir, seis opciones.

En general, si tinguérem n paraules d'un tipus i m paraules de l'altre, les possibilitats (per als espais) serien determinades per les **permutacions amb repetició de 2 elements que es repeteixen n i m vegades.** En particular, per a cada un dels casos tenim

En general, si tuviésemos n palabras de un tipo y m palabras del otro, las posibilidades (para los espacios) vendrían determinadas por las **permutaciones con repetición de 2 elementos que se repiten n y m veces.** *En particular, para cada uno de los casos tenemos*

$$S_i = S_a \times PR_{(5,0)} = S_a$$
$$S_{ii} = S_a \times PR_{(5,1)} = S_a \cdot \frac{6!}{5! \cdot 1!} = 6S_a$$

I en aplicar el **principi de la suma** s'obté que

Y aplicando el **principio de la suma** *se obtiene que*

$$S_d = S_i + S_{ii} = \boxed{7S_a} = \boxed{69\,854\,400}$$

e) El raonament és exactament el mateix però amb més casos. En la primera columna de la taula següent s'indiquen els casos. Cada paraula es representa mitjançant un dígit que indica quantes lletres té. Els casos i) i ii) anteriors es marquen com a superíndexs.

e) El razonamiento es exactamente el mismo pero con más casos. En la primera columna de la tabla siguiente se indican los casos. Cada palabra se representa mediante un dígito que indica cuántas letras tiene. Los casos i) y ii) anteriores se marcan como superíndices.

55555 $^{*i)}$		$PR_{(5,0,0)}$	$= \frac{5!}{5!} = 1$
555 4 33		$PR_{(3,1,2)}$	$= \frac{6!}{2!\,3!} = 6 \cdot 5 \cdot 4/2 = 60$
55 ...	55 444 3	$PR_{(2,3,1)}$	$= 60$
	55 33333	$PR_{(2,0,5)}$	$= \frac{7!}{2!\,5!} = 7 \cdot 6/2 = 21$
5 ...	5 44444 $^{*ii)}$	$PR_{(1,5,0)}$	$= \frac{6!}{5!} = 6$
	5 44 3333	$PR_{(1,2,4)}$	$= \frac{7!}{2!\,4!} = 7 \cdot 6 \cdot 5/2 = 105$
4444 333		$PR_{(0,4,3)}$	$= \frac{7!}{4!\,3!} = 7 \cdot 6 \cdot 5/3! = 35$
4 3333 333		$PR_{(0,1,7)}$	$= \frac{8!}{7!} = 8$

Aplicant el **principi de la suma** i el **principi del producte** a cada un dels 8 casos, igual que en l'apartat anterior s'arriba a

*Aplicando el **principio de la suma** y el **principio del producto** a cada uno de los 8 casos, igual que en apartado anterior se llega a*

$$S_e = S_a \times (1 + 60 + 60 + 21 + 6 + 105 + 35 + 8) = \boxed{296\, S_a} = \boxed{2\,953\,843\,200}$$

Problema 1.4:

[futbetDutxes]

10 amics juguen a futbet els divendres.

10 amigos juegan a futbito los viernes.

a) De quantes maneres es podrien distribuir en 2 equips de 5?

a) ¿De cuántas maneras se podrían distribuir en 2 equipos de 5?

b) Si hi ha 7 dels 10 amics que no estan disposats a jugar de porter, de quantes maneres es podran fer els equips?

b) Si hay 7 de los 10 amigos que no están dispuestos a jugar de portero, ¿de cuántas maneras se podrán hacer los equipos?

c) I si dels 3 possibles porters n'hi ha 2 que no volen jugar en el mateix equip, de quantes maneres es podran distribuir?

c) Y si de los 3 posibles porteros hay 2 que no quieren jugar en el mismo equipo, ¿de cuántas maneras se podrán distribuir?

d) I què passaria si els 2 que no volen jugar són dels 7 que no volen ser porters?

d) ¿Y qué pasaría si los 2 que no quieren jugar juntos son de los 7 que no quieren ser porteros?

Dels 10 amics n'hi ha 6 que en acabar es dutxen al mateix temps en alguna de les dues dutxes col·lectives, C1 i C2, que hi ha al pavelló.

e) De quantes maneres es poden dutxar?

f) En el mateix pavelló construiran 3 dutxes més, però individuals (I1, I2, I3). De quantes maneres es podran dutxar els 6 (al mateix temps)?

g) I si, en lloc de 3 dutxes individuals, en construïren una triple (T1), on podrien cabre fins a tres persones?

De los 10 amigos hay 6 que al acabar se duchan al mismo tiempo en alguna de las dos duchas colectivas, C1 y C2, que hay en el pabellón.

e) ¿De cuántas maneras se pueden duchar?

f) En el mismo pabellón van a construir 3 duchas más, pero individuales (I1, I2, I3). ¿De cuántas maneras se podrán duchar los 6 al mismo tiempo?

g) ¿Y si en lugar de 3 duchas individuales construyeran una triple (T1), donde podrían caber hasta tres personas?

a) Són 10 amics. I en total hi ha $|C_{10}^5| = \binom{10}{5}$ subconjunts diferents de grandària 5.

Una vegada format un possible equip, la resta d'amics fins a 10 (el complementari) formen l'altre.

Però cada possible partició en 2 equips es comptarà dues vegades, ja que cada equip i el seu complementari formen part dels $\binom{10}{5}$ subconjunts. Per tant, en aplicar el **principi de la divisió**, el cardinal del conjunt de totes les possibles particions vàlides, S_{10}, és determinat per

a) Son 10 amigos. Y en total hay $|C_{10}^5| = \binom{10}{5}$ subconjuntos de tamaño 5.

Una vez formado un posible equipo, el resto de amigos hasta 10 (el complementario) forman el otro.

Pero cada posible partición en 2 equipos se contará dos veces, ya que cada equipo y su complementario forman parte de los $\binom{10}{5}$ subconjuntos. Por tanto, aplicando el **principio de la división**, *el cardinal del conjunto de todas las posibles particiones válidas, S_{10}, viene dado por*

$$S_{10} = \frac{\binom{10}{5}}{2} = \frac{\cancel{10}^{\,2} \cdot 9 \cdot \cancel{8} \cdot 7 \cdot \cancel{6}}{\cancel{5} \cdot \cancel{4} \cdot \cancel{3} \cdot \cancel{2} \cdot 1} \cdot \frac{1}{\cancel{2}} = 9 \cdot 7 \cdot 2 = \boxed{126}.$$

b) Només hi ha, doncs, 3 possibles porters. I hi ha d'haver un porter en cada equip. Per tant, aquests 3 amics s'han de repartir entre els 2 equips i això es pot fer de $\binom{3}{2} = \binom{3}{1} = 3$ maneres possibles.

b) Sólo hay pues 3 posibles porteros. Y tiene que haber un portero en cada equipo. Por lo tanto, estos 3 amigos se tienen que repartir entre los 2 equipos y eso se puede hacer de $\binom{3}{2} = \binom{3}{1} = 3$ maneras posibles.

A banda, els 7 amics romanents s'han de distribuir entre els dos equips: 4 amb el porter que està sol i 3 amb els 2 porters. Ara no hi ha confusió ni repetició perquè les grandàries són diferents, i aquesta quantitat és $\binom{7}{4} = \binom{7}{3}$.

Aparte, los 7 amigos restantes se tienen que distribuir entre los dos equipos: 4 con el portero que está solo y 3 con los 2 porteros. Ahora no hay confusión ni repetición por que los tamaños son diferentes, y esta cantidad es $\binom{7}{4} = \binom{7}{3}$.

En aplicar el **principi de la multiplicació**, ja que la selecció dels porters és independent de la dels jugadors no porters, el nombre de possibles particions en aquest cas és determinat per

Aplicando el **principio de la multiplicación**, *ya que la selección de los porteros es independiente de la de los jugadores no porteros, el número de posibles particiones en este caso viene dado por*

$$S_{(7,3)} = \binom{3}{1} \times \binom{7}{3} = 3 \cdot \frac{7 \cdot \cancel{6} \cdot 5}{\cancel{3} \cdot \cancel{2} \cdot 1} = 3 \cdot 35 = \boxed{105}.$$

Alternativament, també haguérem pogut comptar les particions impossibles en què els 3 possibles porters estigueren junts en el mateix equip. Aquesta quantitat seria

Alternativamente, también hubiésemos podido contar las particiones imposibles en las que los 3 posibles porteros estuviesen juntos en el mismo equipo. Esta cantidad sería

$$S_3 = \binom{7}{5} = \binom{7}{2} = \frac{7 \cdot 6}{2!} = 21,$$

per la qual cosa les particions possibles s'obtindrien a partir de S_{10} i del **principi de la suma** com a

por lo que las particiones posibles se obtendrían a partir de S_{10} y del **principio de la suma** *como*

$$S_{(7,3)} = S_{10} - S_3 = \boxed{105}.$$

c) En el cas en què 2 dels possibles porters no vulguen jugar junts, les $\binom{3}{1}$ possibilitats de repartir els porters de l'apartat anterior queden reduïdes a 2 (amb qui dels 2 s'ajunta el tercer porter).

c) En el caso que 2 de los posibles porteros no quieran jugar juntos, las $\binom{3}{1}$ posibilidades de repartir los porteros del apartado anterior quedan reducidas a 2 (con cuál de los 2 se queda el tercer portero).

A partir d'ací el càlcul és exactament el mateix.

A partir de aquí el cálculo es exactamente el mismo.

$$S_{(7,2,1)} = 2 \times \binom{7}{3} = 2 \cdot 35 = \boxed{70}.$$

Alternativament, podríem calcular quantes de les particions en $S_{(7,3)}$ tenen junts els dos porters incompatibles (i l'altre porter en l'altre equip). Això correspon a fixar els porters i repartir els restants i seria

Alternativamente, podríamos calcular cuántas de las particiones en $S_{(7,3)}$ tienen juntos los dos porteros incompatibles (y al otro portero en el otro equipo). Esto corresponde a fichar a los porteros y repartir a los restantes y sería

$$S_2 = \binom{7}{4} = \binom{7}{3} = 35.$$

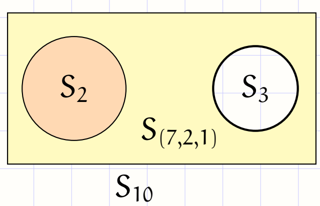
Per tant, a partir de $S_{(7,3)}$ i en aplicar el **principi de la suma** es tindria que

Por tanto, a partir de $S_{(7,3)}$ y aplicando el **principio de la suma** *se tendría que*

$$S_{(7,2,1)} = S_{(7,3)} - S_2 = 105 - 35 = \boxed{70}.$$

El conjunt de totes les particions dels 10 amics en dos equips que estan involucrades en els apartats anteriors es pot representar gràficament en el següent diagrama de Venn.

El conjunto de todas las particiones de los 10 amigos en dos equipos que están involucradas en los apartados anteriores se puede representar gráficamente en el siguiente diagrama de Venn.



La part en color es correspon amb $S_{(7,3)} = S_{(7,2,1)} \cup S_2$, i els tres subconjunts representats dins de S_{10} són subconjunts disjunts. (Estem abusant de la notació i referint-nos indistintament a conjunts i els seus cardinals).

La parte en color se corresponde con $S_{(7,3)} = S_{(7,2,1)} \cup S_2$, y los tres subconjuntos representados dentro de S_{10} son subconjuntos disjuntos. (Estamos abusando de la notación y refiriéndonos indistintamente a conjuntos y sus cardinales).

d) Si els 2 que no volen jugar junts són dels no porters, tornem al raonament de l'apartat b). Les $\binom{3}{1}$ maneres de repartir els porters s'han de combinar ara amb les $|P_2| = 2$ maneres de repartir aquests 2 amics i finalment amb les maneres de distribuir els romanents (3 a un equip i 2 a l'altre).

d) Si los 2 que no quieren jugar juntos son de los no porteros, volvemos al razonamiento del apartado b). Las $\binom{3}{1}$ maneras de repartir los porteros se tienen que combinar ahora con las $|P_2| = 2$ maneras de repartir estos 2 amigos y finalmente con las maneras de distribuir los restantes (3 a un equipo y 2 al otro).

$$S_{(5,2,3)} = \binom{3}{1} \times 2! \times \binom{5}{2} = 3 \cdot 2 \cdot \frac{5 \cdot 4}{2 \cdot 1} = \boxed{60}.$$

La identificació de les configuracions prohibides del càlcula alternatiu resulta ara un poc més complexa. D'una banda hi ha $\binom{3}{1}$ maneres de repartir els porters, però els 2 jugadors poden anar o be amb l'equip de 2 porters (amb la qual cosa faltaria repartir 1 i 4 en cada equip) o be amb l'equip que només en té 1 (amb la qual cosa faltaria repartir-ne 3 i 2). Aleshores i mitjançant el **principi de la suma** es té que

La identificación de las configuraciones prohibidas del cálculo alternativo resulta ahora un poco más compleja. Por un lado hay $\binom{3}{1}$ maneras de repartir los porteros, pero los 2 jugadores pueden ir bien con el equipo de 2 porteros (con lo que faltaría repartir 1 y 4 en cada equipo) o bien con el equipo que sólo tiene 1 (con lo que haría falta repartir 3 y 2). Entonces y mediante el **principio de la suma** *se tiene que*

$$S_2' = \binom{3}{1}\left[\binom{5}{1} + \binom{5}{2}\right] = 45.$$

En aplicar novament el **principi de la suma** com en els casos anteriors s'arriba doncs a

Aplicando nuevamente el **principio de la suma** *como en los casos anteriores se llega pues a*

$$S_{(5,2,3)} = S_{(7,3)} - S_2' = 105 - 45 = \boxed{60}.$$

e) Tal com es fa referència a les dutxes col·lectives, hauria de quedar clar que les dues dutxes **són distingibles** i de **capacitat il·limitada**.

e) Tal y como se hace referencia a las duchas colectivas, tendría que estar claro que las dos duchas son **distinguibles** *y de* **capacidad ilimitada**.

Pert tant, cada un dels 6 amics que es dutxen només ha de decidir en quina de les dues dutxes col·lectives disponibles entra. O siga,

Por lo tanto, cada uno de los 6 amigos que se duchan sólo tienen que decidir en cuál de las dos duchas colectivas disponibles entra. O sea,

$$C_6 = |VR_2^6| = 2^6 = \boxed{64}.$$

f) Si hi haguera 3 dutxes individuals i **distingibles**, cada un dels 6 podria decidir utilitzar-ne alguna o no. Però sempre que no estiguera ja ocupada.

f) Si hubiese 3 duchas individuales y **distinguibles***, cada uno de los 6 podría decidir usar alguna o no. Pero siempre que no estuviese ya ocupada.*

Per tant, caldrà considerar casos quant a l'ocupació de les dutxes individuals: Si ningú es dutxa en les individuals, $O_0 = \binom{3}{0} = 1$. Si només se n'utilitza una d'individual, hi ha 3 possibilitats, $O_1 = \binom{3}{1} = 3$. Si se n'utilitzen dues, $O_2 = \binom{3}{2} = 3$. I en el cas que s'ompliren les tres, $O_3 = \binom{3}{3} = 1$.

Por lo tanto, habrá que considerar casos en cuanto a la ocupación de las duchas individuales: Si nadie se ducha en las individuales, $O_0 = \binom{3}{0} = 1$. Si sólo se utiliza una individual, hay 3 posibilidades, $O_1 = \binom{3}{1} = 3$. Si se utilizan dos, $O_2 = \binom{3}{2} = 3$. Y en el caso en que se llenaran las tres, $O_3 = \binom{3}{3} = 1$.

En resum, les diferents possibilitats d'ocupar les dutxes individuals es poden representar agrupades en quatre casos com s'indica en la següent taula.

En resumen, las diferentes posibilidades de ocupar las duchas individuales se pueden representar agrupadas en cuatro casos como se indica en la siguiente tabla.

i	I1 I2 I3	O_i
0	000	C_3^0
1	001 010 100	C_3^1
2	011 101 110	C_3^2
3	111	C_3^3

I es compleix que

Y se cumple que

$$\sum_{i=0}^{3} O_i = \sum_{i=0}^{3} |C_3^i| = \sum_{i=0}^{3} \binom{3}{i} = 2^3 = |VR_2^3| = 8.$$

Ara caldrà analitzar de manera independent per a cada cas de quantes maneres es poden omplir les dutxes individuals i les col·lectives.

Ahora habrá que analizar de manera independiente para cada caso de cuántas maneras se pueden llenar las duchas individuales y las colectivas.

En el cas O_0 només hi ha una manera d'ocupar les individuals, $I_0 = 1$, els 6 amics van a les col·lectives i es dutxarien de C_6 maneres possibles tal i com s'ha calculat en l'apartat anterior.

En el caso O_0 sólo hay una manera de ocupar las individuales, $I_0 = 1$, los 6 amigos van a las colectivas y se ducharían de C_6 maneras posibles tal y como se calculó en el apartado anterior.

En el cas O_1, un dels 6 amics ocuparia la individual (de $I_1 = 6$ maneres perquè hi ha 6 possibles amics) i els 5 romanents anirien a les col·lectives de $C_5 = |VR_2^5|$ maneres possibles.

En el caso O_1, uno de los 6 amigos ocuparía la individual (de $I_1 = 6$ maneras por que hay 6 posibles amigos) y los 5 restantes irían a las colectivas de $C_5 = |VR_2^5|$ maneras posibles.

El mateix passaria en el cas O_2 però ara hi hauria

Lo mismo pasaría en el caso O_2 pero ahora habría

$$I_2 = |V_6^2| = 6^{\underline{2}} = 30$$

maneres possibles d'ocupar les dues individuals i els 4 romanents anirien a les col·lectives de $C_4 = |VR_2^4|$ maneres possibles.

maneras posibles de ocupar las dos individuales y los 4 restantes irían a las colectivas de $C_4 = |VR_2^4|$ maneras posibles.

Per últim, en el cas O_3, les individuals s'omplirien també de $6^{\underline{3}} = 120$ maneres possibles i les col·lectives de $C_3 = |VR_2^3|$ maneres possibles.

Por último, en el caso O_3, las individuales se llenarían también de $6^{\underline{3}} = 120$ maneres possibles i les col·lectives de $C_3 = |VR_2^3|$ maneres possibles.

Podem completar ara la taula anterior amb tota la informació.

Podemos completar ahora la tabla anterior con toda la información.

casos	individuals *individuales* — dutxes *duchas*		individuals *individuales* — persones *personas*	col·lectives *colectivas*
i	I1 I2 I3	O_i / C_3^i	I_i / V_6^i	C_i / VR_2^{6-i}
0	000	$\binom{3}{0}=1$	$6^{\underline{0}}=1$	2^6
1	001 010 100	$\binom{3}{1}=3$	$6^{\underline{1}}=6$	2^5
2	011 101 110	$\binom{3}{2}=3$	$6^{\underline{2}}=30$	2^4
3	111	$\binom{3}{3}=1$	$6^{\underline{3}}=120$	2^3

Per a obtenir el resultat final caldrà sumar els 4 possibles casos d'ocupació de les individuals (**principi de la suma**) i en cada cas multiplicar les possibilitats d'ocupació per les formes d'omplir les individuals, i també per les formes d'omplir les col·lectives (**principi del producte**).

*Para obtener el resultado final habrá que sumar los 4 posibles casos de ocupación de las individuales (***principio de la suma***) y en cada caso multiplicar las posibilidades de ocupación por las formas de llenar las individuales, y también por las formas de llenar las colectivas (***principio del producto***).*

$$\sum_{i=0}^{3} O_i \cdot I_i \cdot C_{6-i} = \sum_{i=0}^{3} |C_i^3| \cdot |V_6^i| \cdot |VR_2^{6-i}| = \sum_{i=0}^{3} \binom{3}{i} \cdot 6^{\underline{i}} \cdot 2^{6-i} =$$

$$= \underbrace{1 \cdot 1 \cdot 2^6}_{64} + \underbrace{3 \cdot 6 \cdot 2^5}_{576} + \underbrace{3 \cdot 6 \cdot 5 \cdot 2^4}_{1440} + \underbrace{1 \cdot 6 \cdot 5 \cdot 4 \cdot 2^3}_{960} =$$

$$= (1+9)2^6 + (90+60)2^4 = 640 + 880 = \boxed{3040}.$$

g) Si canviem les 3 dutxes individuals (i distingibles) per una única dutxa triple (o equivalentment 3 dutxes individuals indistingibles), l'anàlisi per casos anterior continua sent vàlida però ara només hi ha una forma d'ocupació. És a dir, $O_i = 1$ per a tot i. I la forma d'elegir les persones que van a la triple vindrà donada per **combinacions sense repetició** en lloc de **variacions sense repetició**. És a dir, $I_i = |C_6^i|$ per a tot i.

g) Si cambiamos las 3 duchas individuales (y distinguibles*) por una única ducha triple (o equivalentemente 3 duchas individuales* indistinguibles*), el análisis por casos anterior continua siendo válido pero ahora sólo hay una forma de ocupación. Es decir, $O_i = 1$ para todo i. Y la forma de elegir las personas que van a la triple vendrá dada por* **combinaciones sin repetición** *en lugar de* **variaciones sin repetición**. *Es decir, $I_i = |C_6^i|$ para todo i.*

Tot plegat, el nombre total de possibilitats seria ara

Conjuntamente, el número total de posibilidades sería ahora

$$\sum_{i=0}^{3} I_i \cdot C_{6-i} = \sum_{i=0}^{3} 1 \cdot \binom{6}{i} \cdot 2^{6-i} = \underbrace{1 \cdot 2^6}_{64} + \underbrace{6 \cdot 2^5}_{192} + \underbrace{15 \cdot 2^4}_{240} + \underbrace{20 \cdot 2^3}_{160} = \boxed{656}.$$

Problema 1.5: [gabiesAnimals]

Una botiga d'animals té 5 gàbies numerades i grans per a ocells. Suposant que no sabem distingir 2 ocells de la mateixa espècie,

Una tienda de animales tiene 5 jaulas numeradas y grandes para aves. Suponiendo que no sabemos distinguir 2 aves de la misma especie,

a) de quantes maneres es poden distribuir 1 lloro, 1 cotorra, i un tucà?

a) ¿de cuántas maneras se pueden distribuir 1 loro, 1 cotorra y un tucán?

b) I si foren 2 lloros, 3 cotorres i 1 tucà?

b) ¿Y si fueran 2 loros, 3 cotorras y 1 tucán?

c) I 7 tucans?

c) ¿Y 7 tucanes?

Si ens diuen que les gàbies són grans és perquè hi cabran tots els ocells que vulguem posar-hi. I com que estan numerades, això significa que són **distingibles**.

Si nos dicen que las jaulas son grandes es por que cabrán todos los pájaros que queramos poner. Y como están numeradas, significa que son **distinguibles**.

També és important remarcar que sabem distingir espècies diferents, però no diferents individus de la mateixa espècie.

También es importante remarcar que sabemos distinguir especies diferentes, pero no diferentes individuos de la misma especie.

a) En aquest cas tenim 3 ocells **distingibles** (perquè són d'espècies diferents) cada un dels quals pot estar en qualsevol de les 5 gàbies.

a) En este caso tenemos 3 pájaros **distinguibles** *(por que son de especies diferentes) cada uno de los cuales puede estar en cualquiera de las 5 jaulas.*

El lloro el podem col·locar en qualsevol de les 5 gàbies. Això són 5 possibilitats.

El loro lo podemos colocar en cualquiera de las 5 jaulas. Eso son 5 posibilidades.

De la mateixa manera, tenim altres 5 possibilitats per a col·locar la cotorra i 5 més per al tucà.

De la misma manera, tenemos otras 5 posibilidades para colocar la cotorra y 5 más para el tucán.

I en aplicar el **principi del producte** arribaríem a un total de $5 \cdot 5 \cdot 5 = 125$ possibilitats per a distribuir els tres.

Aplicando el **principio del producto** *llegaríamos a un total de $5 \cdot 5 \cdot 5 = 125$ posibilidades para distribuir a los tres.*

De manera equivalent, podem pensar que a cada un dels 3 ocells li hem d'assignar un d'entre 5 nombres possibles. O siga, **variacions amb repetició de 5 elements agafats de 3 en 3**, VR_5^3.

De manera equivalente, podemos pensar que a cada uno de los 3 pájaros le hemos de asignar uno de entre 5 números posibles. O sea, **variaciones con repetición de 5 elementos tomados de 3 en 3**, VR_5^3.

$$S_a = |VR_5^3| = 5^3 = \boxed{125}.$$

c) Respondrem primer a aquest apartat que és més senzill.

c) Responderemos primero a este apartado que es más sencillo.

La diferència fonamental respecte al primer apartat, a banda que són 7 ocells en lloc de 3, és que ara són ocells **indistingibles**.

La diferencia fundamental respecto al primer apartado, aparte de que son 7 pájaros en lugar de 3, es que ahora son pájaros **indistinguibles**.

Distribuir n ocells (distingibles) en m gàbies numerades ja hem vist en l'apartat anterior que es correspon amb variacions amb repetició de m elements agafats de n en n, VR_m^n.

Distribuir n pájaros (distinguibles en m jaulas numeradas ya hemos visto en el apartado anterior que se corresponde con variaciones con repetición de m elementos tomados de n en n, VR_m^n.

Però si els ocells són indistingibles, el problema equival a distribuir n boles iguals en m caixes numerades. O també, a considerar multiconjunts de m gàbies de cardinalitat n. És a dir, **combinacions amb repetició de m elements agafats de n en n**.

Pero si los pájaros son indistinguibles, el problema equivale a distribuir n bolas iguales en m cajas numeradas. O también, a considerar multiconjuntos de m jaulas de cardinalidad n. Es decir, **combinaciones con repetición de m elementos tomados de n en n**.

En el nostre cas particular, tenim que

En nuestro caso particular, tenemos que

$$S_c = CR_m^n = CR_5^7 = \left(\!\!\binom{5}{7}\!\!\right) = C_{5+7-1}^7 = \binom{11}{7} = \frac{11 \cdot 10 \cdot 9 \cdot \cancel{8}}{\cancel{4} \cdot 3 \cdot \cancel{2} \cdot 1} = \boxed{330}.$$

b) Si ara cal distribuir n_1 ocells d'una espècie i n_2 ocells d'una altra, podem fer-ho **independentment** i aplicar el **principi del producte**.

b) Si ahora hay que distribuir n_1 pájaros de una especie y n_2 pájaros de otra, podemos hacerlo **independientemente** *y aplicar el* **principio del producto**.

Aleshores, les diferents maneres de repartir els 2 lloros, les 3 cotorres i el tucà es poden calcular com a

Entonces, las diferentes maneras de repartir los 2 loros, las 3 cotorras y el tucán se pueden calcular como

$$S_b = CR_5^2 \times CR_5^3 \times CR_5^1 = \left(\!\!\binom{5}{2}\!\!\right) \times \left(\!\!\binom{5}{3}\!\!\right) \times \left(\!\!\binom{5}{1}\!\!\right) = \binom{6}{2} \times \binom{7}{3} \times \binom{5}{1} = 15 \cdot 35 \cdot 5 = \boxed{2625}.$$

Podem veure que d'aquesta manera obtindríem també el resultat del primer apartat on teníem un lloro, una cotorra i un tucà, i per tant

Podemos ver que de esta manera obtendríamos también el resultado del primer apartado donde teníamos un loro, una cotorra y un tucán, y por tanto

$$S_a = CR_5^1 \times CR_5^1 \times CR_5^1 = \boxed{125}.$$

Problema 1.6: [senderisme]

Considera rutes de senderisme en què a cada quilòmetre es pot pujar o baixar 10 m o planar (no canviar d'altitud).

Considera rutas de senderismo de manera que en cada km se puede subir o bajar 10 m o planear (no cambiar de altitud).

a) Quants perfils de ruta diferents de 12 km pot haver-hi amb desnivell positiu acumulat (suma dels metres pujats) de 60 m?

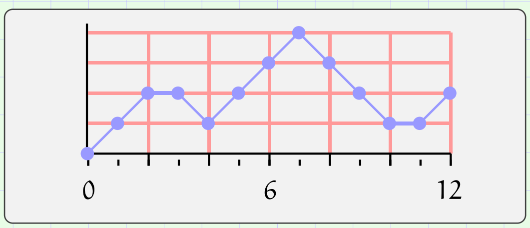
a) ¿Cuántos perfiles de ruta diferentes de 12 km puede haber con desnivel positivo acumulado (suma de los metros subidos) de 60m?

b) I quants de 12 km amb desnivell positiu i negatiu acumulats de 60 i 40m, respectivament?

b) ¿Y cuántos de 12 km con desniveles positivo y negativo acumulados de 60 y 40m, respectivamente?

Com a exemple, en la figura es mostra un perfil de 12 km amb desnivells positiu i negatiu acumulats de 60 i 40 metres, respectivament.

Como ejemplo, en la figura se muestra un perfil de 12 km con desniveles positivo y negativo acumuilados de 60 y 40 metros, respectivamente.



Des del punt de vista del seu perfil, podem representar les diferents rutes de m quilòmetres com a seqüències de longitud m formades per elements del conjunt $\{+1, -1, 0\}$ (pujar, baixar i planar).

Desde el punto de vista de su perfil, podemos representr las diferentes rutas de m kilómetros como secuencias de longitud m formadas por elementos del conjunto $\{+1, -1, 0\}$ (subir, bajar y planear).

Per exemple, la ruta de la figura de l'exemple es podria representar com a

Por ejemplo, la ruta de la figura del ejemplo se podría representar como

$$(+1, +1, 0, -1, +1, +1, +1, -1, -1, -1, 0, +1).$$

El nombre total de rutes de longitud m (en km), S_m, seria aleshores determinat per les **variacions amb repetició de 3 elements agafats de m en m**, VR_3^m.

El número total de rutas de longitud m (en km), S_m, vendría entonces determinado por las **variaciones con repetición de 3 elementos tomados de m en m**, VR_3^m.

Per tant tenim en el cas de l'exemple que

Poer tanto tenemos en el caso del ejemplo que

$$S_{12} = VR_3^{12} = 3^{12}.$$

a) Si dels m quilòmetres de la ruta, n han de ser de pujada, la qual cosa implica un desnivell positiu acumulat de $10n$ metres, això significa que els restants $m - n$ quilòmetres de la ruta han de ser neutres o de baixada.

a) Si de los m kilómetros de la ruta, n tienen que ser de subida, lo que implica un desnivel positivo acumulado de $10n$ metros, esto significa que los restantes $m - n$ kilómetros de la ruta tienen que ser neutros o de bajada.

Aleshores, d'una banda haurem de decidir de quantes maneres es poden col·locar els quilòmetres de pujada dins de la ruta.

Entonces, por una parte tendremos que decidir de cuántas maneras se pueden colocar los kilómetros de subida dentro de la ruta.

I d'altra banda haurem de decidir si cada un dels restants quilòmetres és de baixada o neutre.

Y por otra tendremos que decidir si cada uno de los restantes kilómetros es de bajada o neutro.

La quantitat de casos total serà, doncs, determinada pel **principi del producte**.

La cantidad de casos total será pues determinada por el **principio del producto**.

Els n quilòmetres de pujada dins dels m de la ruta es poden elegir de $\binom{m}{n}$ maneres. O, en altres paraules, es corresponen amb **combinacions de m elements agafats de n en n**.

Los n kilómetros de subida dentro de los m de la ruta se pueden elegir de $\binom{m}{n}$ maneras. O en otras palabras, se corresponden con **combinaciones de m elementos tomados de n en n**.

I les maneres possibles d'etiquetar els $m - n$ romanents com a baixar o planar es correspon amb les **variacions amb repetició de 2 elements agafats de $m - n$ en $m - n$**.

Y las maneras posibles de etiquetar los $m - n$ restantes como bajar o planear se corresponden con las **variaciones con repetición de 2 elementos tomados de $m-n$ en $m-n$**.

El nombre total de rutes de m quilòmetres amb desnivell acumulat de $10n$ metres serà, doncs,

El número total de rutas de m kilómetros con desnivel acumulado de $10n$ metros será pues,

$$S_{(m,n)} = C_m^n \times VR_2^{m-n} = \binom{m}{n} \cdot 2^{m-n}.$$

En particular, el nombre de rutes de 12 km amb 60 m de desnivell positiu acumulat serien

En particular, el número de rutas de 12 km con 60 m de desnivel positivo acumulado serían

$$S_{(12,6)} = \binom{12}{6} \cdot 2^6 = \frac{\cancel{20}^{2} \cdot 19 \cdot \cancel{18}^{2} \cdot 17 \cdot \cancel{16}^{2} \cdot \cancel{15} \cdots \cancel{12}^{2} \cdot 11}{\cancel{10} \cdot \cancel{9} \cdots \cancel{6} \cdot \cancel{5} \cdot 4 \cdot \cancel{3} \cdot 2 \cdot 1} \; 2^6 =$$

$$= 19 \cdot 17 \cdot 13 \cdot 11 \cdot 2^8 = 46189 \cdot 256 = \boxed{11\,824\,384}.$$

b) Ara en les rutes de m quilòmetres, n han de ser de pujada i k han de ser de baixada per a tenir desnivells positiu i negatiu acumulats de $10n$ i $10k$, respectivament.

b) Ahora en las rutas de m kilómetros, n tienen que ser de subida y k tienen que ser de bajada para tener desniveles positivo y negativo acumulados de $10n$ y $10k$ respectivamente.

Per a comptar totes les possibilitats, hem de seleccionar ara k quilòmetres de baixada d'entre els $m-n$ que no són de pujada.

Para contar todas las posibilidades, tenemos que seleccionar ahora k kilómetros de bajada de entre los $m-n$ que no son de subida.

I aplicarem el principi del producte igual que abans, amb la qual cosa el nombre total de rutes de m quilòmetres amb $10n$ metres i $10k$ metres de pujada i baixada acumulada seran

Y aplicaremos el principio del producto igual que antes, con lo que el número total de rutas de m kilómetros con $10n$ metros y $10k$ metros de subida y bajada acumulada serán

$$S_{(m,n,k)} = C_m^n \times C_{m-n}^k = \binom{m}{n} \cdot \binom{m-n}{k}.$$

En particular, el nombre de rutes de 12 km amb 60 m i 40 m de desnivells positiu i negatiu acumulats serien

En particular, el número de rutas de 12 km con 60 m y 40 m de desniveles positivo y negativo acumulados serían

$$S_{(12,6,4)} = \tbinom{12}{6} \cdot \tbinom{6}{4} = \frac{\cancel{12} \cdot 11 \cdot 10 \cdot 9 \cdot \cancel{8}^{2} \cdot 7}{6!} \, \frac{6!}{\cancel{2!} \; \cancel{4} \cdot \cancel{3} \cdot \cancel{2} \cdot 1} = 11 \cdot 10 \cdot 9 \cdot 7 \cdot 2 = \boxed{13\,860}.$$

Problema 1.7: [meitatsSuccessives]

Intenta trobar un argument visual per a calcular el següent sumatori en què cada terme no és un nombre enter sinó una fracció de la unitat.

Intenta encontrar un argumento visual para calcular el siguiente sumatorio en el que cada término no es un número entero sino una fracción de la unidad.

$$\sum_{i=1}^{n} \frac{1}{2^i} = \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^n}$$

Potser la forma més senzilla d'obtenir un resultat per a sumatori de l'enunciat és relacionar-lo amb la suma dels dobles successius, $S_d(n)$, que s'ha vist en la pàgina 25.

Tal vez la forma más sencilla de obtener un resultado para el sumatorio del enunciado es relacionarlo con la suma de los dobles sucesivos, $S_d(n)$, que se ha visto en la página 25.

Primer anomenem $S_m(n)$ el resultat, ho multipliquem tot per 2^n i obtenim

Primero llamamos $S_m(n)$ al resultado, lo multiplicamos todo por 2^n y obtenemos

$$2^n S_m(n) = \sum_{i=1}^{n} 2^{n-i} = \sum_{j=0}^{n-1} 2^j + 2^n - 2^n = S_d(n) - 2^n,$$

on hem fet un canvi d'índex, hem sumat i restat 2^n i hem introduït $S_d(n)$, amb la qual cosa arribem a

donde hemos hecho un cambio de índice, hemos sumado y restado 2^n y hemos introducido $S_d(n)$, con lo que llegamos a

$$2^n S_m(n) = 2 \cdot 2^n - 1 - 2^n,$$

des d'on s'obté que el resultat del sumatori és

desde donde se obtiene que el resultado del sumatorio es

$$\boxed{S_m(n) = 1 - \frac{1}{2^n}.}$$

No obstant això, hi ha un argument visual molt senzill que permet obtenir fàcilment el mateix resultat.

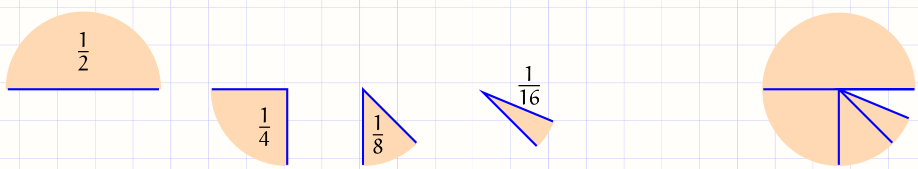
Sin embargo hay un argumento visual muy sencillo que permite obtener fácilmente el mismo resultado.

Primer pensem en la unitat (una unitat de qualsevol cosa) com un cercle. Pensem, per exemple, en una pizza.

Primero pensamos en la unidad (una unidad de cualquier cosa) como un círculo. Pensemos, por ejemplo, en una pizza.

El nostre sumatori consisteix a sumar la meitat de la unitat (de la pizza) amb la meitat de la meitat, i amb la meitat de la meitat de la meitat, i així successivament. De manera gràfica,

Nuestro sumatorio consiste en sumar la mitad de la unidad (de la pizza) con la mitad de la mitad, y con la mitad de la mitad de la mitad, y así sucesivamente. De manera gráfica,



La figura il·lustra el sumatori amb quatre termes. Si observem el dibuix de la dreta on estan "sumades" totes les porcions veiem que el total és igual a la unitat **menys** una porció que és exactament igual que l'últim terme del sumatori. És a dir, en el cas de la figura tindríem

La figura ilustra el sumatorio con cuatro términos. Si observamos el dibujo de la derecha donde están "sumadas" todas las porciones vemos que el total es igual a la unidad **menos** *una porción que es exactamente igual que el último término del sumatorio. Es decir, en el caso de la figura tendríamos*

$$1 - \frac{1}{16}.$$

I aquesta relació és certa per a qualsevol nombre de termes en el sumatori ja que cada nou terme que s'afegeix ocupa la meitat del romanent i el nou romanent acaba sent també de la mateixa grandària.

Y esta relación es cierta para cualquier número de términos en el sumatorio ya que cada nuevo término que se añade ocupa la mitad del resto y el nuevo resto acaba siendo también del mismo tamaño.

Per tant podem deduir l'expressió correcta per al sumatori,

Por lo tanto podemos deducir la expresión correcta para el sumatorio,

$$\sum_{i=1}^{n} \frac{1}{2^i} = 1 - \frac{1}{2^n}.$$

1.6 Problemes proposats

Problemas Propuestos

Problema 1.8: [progressioAritmetica]

El sumatori $S_1(n)$ de la pàgina 24 és un cas particular de suma dels termes d'una progressió aritmètica. Una seqüència de valors, $(a_1, a_2, \ldots, a_n)$, diem que és una **progressió aritmètica** si per a tot $i = 2, \ldots, n$ es compleix que

El sumatorio $S_1(n)$ de la página 24 es un caso particular de suma de los términos de una progresión aritmética. Una secuencia de valores, $(a_1, a_2, \ldots, a_n)$, decimos que es una **progresión aritmética** *si para todo $i = 2, \ldots, n$ se cumple que*

$$a_i = a_{i-1} + d,$$

on d és l'anomenada **diferència** entre dos termes qualssevol de la progressió.

donde d es la llamada **diferencia** *entre cualesquiera dos términos de la progresión.*

Estén l'argument visual utilitzat per obtenir la solució de $S_1(n)$ per a la suma de qualsevol progressió aritmètica.

Extiende el argumento visual utilizado para obtener la solución de $S_1(n)$ para la suma de cualquier progresión aritmética.

La solució és

La solución es

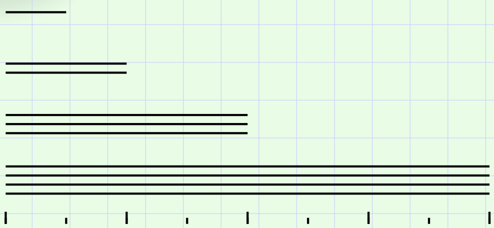
$$\frac{n}{2}(a_1 + a_n) = n\left(a_1 + \frac{n-1}{2}d\right).$$

Problema 1.9: [mesDoblesSuccessius]

Intenta trobar un argument visual per calcular el següent sumatori en què cada terme es correspon amb una quantitat que va duplicant-se multiplicada per un factor (enter) que creix amb l'índex del sumatori. En el dibuix que s'adjunta com a ajut, cada terme del sumatori es representa com a i barres horitzontals de longitud 2^{i-1}.

Intenta encontrar un argumento visual para calcular el siguiente sumatorio en el que cada término se corresponde con una cantidad que va duplicándose multiplicada por un factor (entero) que crece con el índice del sumatorio. En el dibujo que se adjunta como ayuda, cada término del sumatorio se representa como i barras horizontales de longitud 2^{i-1}.

$$\sum_{i=1}^{n} i2^{i-1} = 1 + 2 \cdot 2 + 3 \cdot 4 + 4 \cdot 8 + \cdots + n \cdot 2^{n-1}.$$



La solució és

La solución es

$$n2^n - 2^n + 1.$$

Problema 1.10: [flannagan]

Flannagan és un jugador trampós que sempre té un as de cors en la mànega i l'habilitat d'intercanviar-lo per qualsevol de les 5 cartes de la seua mà sense que ningú se n'adone. Recorda que a la baralla hi ha 13 valors i 4 pals.

Flannagan es un jugador tramposo que siempre tiene un as de corazones en la manga y la habilidad de intercambiarlo por cualquiera de las 5 cartas de su mano sin que nadie se de cuenta. Recuerda que en la baraja hay 13 valores y 4 palos.

a) De quantes maneres diferents pot obtenir un pòquer (4 cartes amb el mateix valor)?

b) I un *full* (3 cartes amb un valor i 2 amb un altre)?

c) Calcula els dos apartats anteriors per a un jugador honrat.

a) ¿de cuantas maneras diferentes puede obtener un poker (4 cartas con el mismo valor)?

b) ¿y un full *(3 cartas con un valor y 2 con otro?*

c) calcula los 2 apartados anteriores para un jugador honrado

Les solucions per a cada un dels apartats són:

Las soluciones para cada uno de los apartados son:

a) $624 + 1128 = 1752$,

b) $3744 + 15840 = 19584$,

c_a) 624,

c_b) 3744.

Problema 1.11: [discretaliaMatricules]

Els Estats Units de Discretàlia (EUD) han decidit utilitzar per als cotxes matrícules de 5 dígits d'entre els 10 possibles: 0123456789. Però cada estat ha afegit condicions a les seues matrícules:

Los Estados Unidos de Discretalia (EUD) han decidido utilizar para los coches matrículas de 5 dígitos de entre los 10 posibles: 0123456789. *Pero cada estado ha añadido condiciones a sus matrículas:*

a) A Variolina del Nord (VN) han decidit usar matrícules que tinguen un màxim de 3 dígits diferents. (consell: prova primer amb 2).

a) En Variolina del Norte (VN) han decidido usar matrículas que tengan un máximo de 3 dígitos diferentes. (consejo: prueba primero con 2).

b) A Nova Vèrtex (NV) només volen matrícules capicues.

b) En Nueva Vértice (NV) solo quieren matrículas capicúa.

c) A Grafifòrnia (GF) exigeixen matrícules sense zeros a l'esquerra.

c) En Grafifornia (GF) exigen matrículas sin ceros a la izquierda.

d) A Arcansas (o Aristansas, AR) acceptaran matrícules que siguen vàlides a Nova Vèrtex o a Grafifòrnia.

d) En Arcansas (o Aristansas, AR) aceptarán matrículas que sean válidas en Nueva Vértice o en Grafifornia.

e) A Ojaio (o Disjuncijaio, OJ) només accepten matrícules que siguen vàlides a Nova Vèrtex **i** a Grafifòrnia (curiosament).

e) En Ojaio (o Disyuncijaio, OJ) solo aceptan matrículas que sean válidas en Nueva Vértice **y** *en Grafifornia (curiosamente).*

f) A Existàlia (EX) no es posa cap condició.

f) En Existalia (EX), no se pone ninguna condición.

g) I a Connexicut (CX), aprofitant que els dígits 0, 6, 8 i 9 es poden girar 180 graus, només accepten matrícules que siguen *girables* sense que canvie el número.

g) Y en Conexicut (CX), aprovechando que los dígitos 0, 6, 8 *y* 9 *se pueden girar 180 grados, solo aceptan matrículas que sean* girables *sin que cambie el número.*

Digues quantes matrícules diferents pot haver-hi en cada estat, raonant amb detall l'obtenció de les respostes.

Di cuántas matrículas diferentes puede haber en cada estado, razonando con detalle la obtención de las respuestas.

A continuació es mostren alguns exemples de matrícules vàlides i no valides en cada estat.

A continuación se muestran algunos ejemplos de matrículas válidas y no válidas en cada estado.

	a) VN	b) NV	c) GF	d) AR	e) OJ	f) EX	g) CX		
si	E-13233	E-13531	E-73210	b∨c	b∧c	-	E-96896	=	96896-∃
no	E-13244	E-11112	E-00700			-	E-00088	≠	88000-∃

Les solucions per a cada un dels apartats són:

Las soluciones para cada uno de los apartados son:

a) VN	b) NV	c) GF	d) AR	e) OJ	f) EX	g) CX
19360	1000	90000	90100	900	100000	32

Problema 1.12:

[futbol]

Suposem un equip de futbol que juga en tres línies. És a dir, amb 4 **defenses** (2-5), 4 **migcampistes** (6-9) i 2 **davanters** (10-11). Anomenem, camí directe a gol una jugada que parteix del porter (1) i passa per un jugador de cada una de les tres línies i acaba en gol.

Supongamos un equipo de fútbol que juega en tres líneas. Es decir, con 4 **defensas** *(2-5), 4* **medios** *(6-9) y 2* **delanteros** *(10-11). Llamamos camino directo a gol una jugada que parte del portero (1) y pasa por un jugador de cada una de las tres líneas y acaba en gol.*

a) Quants camins directes a gol diferents poden haver-hi?

a) ¿Cuántos caminos directos a gol diferentes puede haber?

b) De quantes maneres es poden distribuir els 10 jugadors de camp en les 3 línies si no importa quina posició ocupen dins de la línea?

b) ¿De cuántas maneras se pueden distribuir los 10 jugadores de campo en las 3 líneas si no importa qué posición ocupen dentro de la línea?

c) I si sí que importa?

c) ¿Y si sí que importa?

d) Aquesta forma de jugar s'anomena 1-4-4-2. Quantes altres formes de jugar poden haver amb 3 línies (no buides)?

d) Esta forma de jugar se la llama 1-4-4-2. ¿Cuántas otras formas de jugar puede haber con 3 líneas (no vacías)?

e) I si no restringim el nombre de línies, però no considerem línies amb menys de 2 jugadors?

e) ¿Y si no restringimos el número de líneas, pero no consideramos líneas con menos de 2 jugadores?

Les solucions per a cada un dels apartats són:

Las soluciones para cada uno de los apartados son:

a)	b)	c)	d)	e)
32	3150	3628800	36	34

Problema 1.13: [escales]

Definim una escala musical heptatònica simple com una seqüència de 7+1 notes separades per un interval de **segona major** (2 semitons) o de **segona menor** (1 semitò) i que cobreixen exactament una octava.

Definimos una escala musical heptatónica simple como una secuencia de 7+1 notas separadas por un intervalo de **segunda mayor** *(2 semitonos) o de* **segunda menor** *(1 semitono) y que cubra exactamente una octava.*

Una octava conté 13 notes i 12 semitons que les separen: els 4 dibuixos de tecles de piano mostren les 13 notes consecutives, separades per semitons que cobreixen l'octava que comença en Do i acaba en Do.

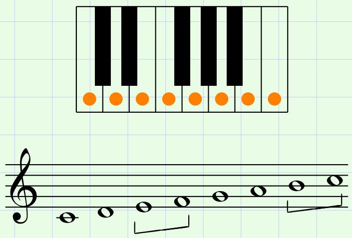
Una octava contiene 13 notas y 12 semitonos que las separan: los 4 dibujos de teclas de piano muestran las 13 notas consecutivas, separadas por semitonos que cubren la octava que comienza en Do y acaba en Do.

En els 3 primers apartats de la figura hi ha 3 exemples d'escales amb notació musical i marcades sobre el teclat amb cercles. L'interval d'una nota a la següent pot ser d'1 o de 2 semitons (en notació musical es marquen per sota les segones menors). Com a curiositat, les escales mostrades són i) l'escala major, ii) l'escala menor, iii) l'escala Lídia.

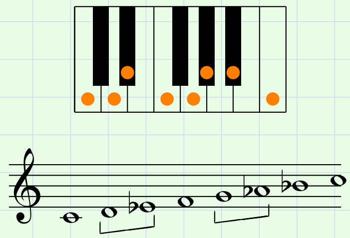
En los 3 primeros apartados de la figura hay 3 ejemplos de escalas en notación musical y marcadas sobre el teclado con círculos. El intervalo de una nota a la siguiente puede ser de 1 o de 2 semitonos (en notación musical se marcan las segundas menores por abajo). Como curiosidad, las escalas mostradas son i) la escala mayor, ii) la escala menor, iii) la escala Lidia.

L'apartat iv) mostra l'escala menor oriental que no seria vàlida (segons la definició anterior) perquè conté 2 intervals de **segona augmentada** (3 semitons), que es marquen per dalt.

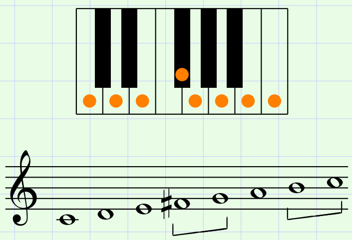
El apartado iv) muestra la escala menor oriental que no sería válida (según la definición anterior) ya que contiene 2 intervalos de **segunda aumentada** *(3 semitonos), que se marcan por arriba.*



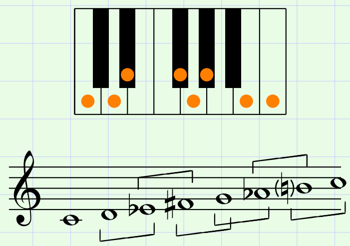
i)



ii)



iii)



iv)

a) Quantes escales heptatòniques simples diferents es poden formar a partir d'una nota?

a) ¿Cuántas escalas heptatónicas simples diferentes se pueden formar a partir de una nota?

b) I si considerem els 3 tipus de segones com en l'exemple iv)?

b) ¿Y si consideramos los 3 tipos de segundas como en el ejemplo iv)?

c) Què passaria si consideràrem **qualssevol** intervals?

c) ¿Qué pasaría si considerásemos **cualesquiera** *intervalos?*

d) I si considerem escales pentatòniques i/o hexatòniques (5 i/o 6 notes, respectivament)?

d) ¿Y si consideramos escalas pentatònicas y/o hexatónicas (5 y/o 6 notas, respectivamente)?

Les solucions per a cada un dels apartats són:

Las soluciones para cada uno de los apartados son:

a)	b)	c)	d)	
21	266	462	330	462

Problema 1.14: [comptaInclusionsSubconj]

Considera un conjunt A de 4 elements qualssevol i el seu conjunt potència 2^A (tots els seus possibles subconjunts). Considera també la relació binària, R, definida per a tot $\alpha, \beta \subseteq A$ com

Considera un conju to A de 4 elementos cualesquiera y su conjunto potencia 2^A (todos sus posibles subconjuntos). Considera también la relación binaria, R, definida para todo $\alpha, \beta \subseteq A$ como

$$\alpha R \beta \quad \textbf{sii} \quad \alpha \subset \beta, \neg \exists \gamma \subseteq A : \alpha \subset \gamma \subset \beta$$

Alguns exemples de la relació R són

Algunos ejemplos de la relación R son

$$\{a_1, a_3\} R \{a_1, a_2, a_3\}, \quad \{a_1, a_3\} R A, \quad \{a_1, a_3\} \not R \{a_1, a_3\}, \quad \{a_1, a_3\} \not R \{a_1, a_2, a_4\}.$$

a) Quin és el cardinal de R? O en altres paraules, quin és el nombre de parells d'elements de A que estan relacionats segons R?

a) ¿Cuál es el cardinal de R? O en otras palabras, ¿cuál es el número de pares de elementos de A que están relacionados según R?

b) Quin és el cardinal de la relació $\subseteq$?

b) ¿Cuál es el cardinal de la relación $\subseteq$?

c) Generalitza els resultats anteriors per a conjunts A de cardinalitat n.

c) Generaliza los resultados anteriores para conjuntos A de cardinalidad n.

Les solucions per a cada un dels apartats són:

Las soluciones para cada uno de los apartados son:

a)	b)	c)	
42	81	$n2^{n-1}$	3^n

■

2. Lògica

Lógica. 2

2.1 Proposicions i equivalències

Proposiciones y equivalencias

Una **proposició** és una afirmació, un fet, o qualsevol altra frase que pot ser **certa** o **falsa**. Per exemple, "els rucs volen".

Una **proposición** *es una afirmación, un hecho, o cualquier otra frase que puede ser* **cierta** *o* **falsa**. *Por ejemplo, "los burros vuelan".*

Els valors **vertader**, V, i **fals**, F, són les (úniques) **constants en lògica proposicional**.

Los valores **verdadero**, V, *y* **falso**, F, *son las (únicas)* **constantes en lógica proposicional**.

Farem servir **variables proposicionals** per a referir-nos a proposicions determinades. Les escriurem normalment $p, q, r, \ldots$

Utilizaremos **variables proposicionales** *para referirnos a proposiciones determinadas. Las escribiremos normalmente como* $p, q, r, \ldots$

És possible construir noves proposicions fent servir **connectives** o **operadors lògics**. En particular, considerarem la **negació**, $\neg$, la **conjunció**, $\wedge$, la **disjunció**, $\vee$, la **implicació**, $\Rightarrow$, i la **coimplicació**, $\Leftrightarrow$.

Es posible construir nuevas proposiciones usando **conectivas** *u* **operadores lógicos**. *En particular, consideraremos la* **negación**, $\neg$, *la* **conjunción**, $\wedge$, *la* **disyunción**, $\vee$, *la* **implicación**, $\Rightarrow$, *y la* **coimplicación**, $\Leftrightarrow$.

Definició recursiva

1. Vertader (V) i Fals (F) són proposicions.
2. Qualssevol variables proposicionals, $p, q, r, \ldots$ són proposicions.
3. Si p és proposició, $\neg p$ també ho és.
4. Si p i q són proposicions, $p \wedge q$, $p \vee q$, $p \Rightarrow q$ i $p \Leftrightarrow q$, també ho són.

Les proposicions definides només segons 1 i 2 s'anomenen **proposicions simples**. La resta de proposicions s'anomenen **proposicions compostes** o també **expressions (proposicionals)**.

Definición recursiva

1. *Verdadero (V) y Falso (F) son proposiciones.*
2. *Cualesquiera variables proposicionales, $p, q, r, \ldots$ son proposiciones.*
3. *Si p es proposición, $\neg p$ también lo es.*
4. *Si p y q son proposiciones, $p \wedge q$, $p \vee q$, $p \Rightarrow q$ y $p \Leftrightarrow q$, también lo son.*

Las proposiciones definidas sólo según 1 y 2 se denominan **proposiciones simples**. *El resto de proposiciones se denominan* **proposiciones compuestas** *o también* **expresiones (proposicionales)**.

En una **expressió** amb diversos operadors, sempre s'interpreten aquests d'esquerra a dreta i segons l'ordre de preferència o **prioritat** donat per la seqüència $(\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow)$.

En una **expresión** *con diversos operadores, siempre se interpretan estos de izquierda a derecha y según el orden de preferència o* **prioridad** *dado por la secuencia* $(\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow)$.

Si es vol canviar l'ordre d'actuació dels operadors en una expressió, es poden utilitzar els parèntesis. Per exemple, les dues expressions següents són equivalents.

Si se quiere cambiar el orden de actuación de los operadores en una expresión, se pueden utilizar los paréntesis. Por ejemplo, las dos expresiones siguientes son equivalentes.

$$p \vee q \vee \neg r \Rightarrow r \wedge p \vee r \Leftrightarrow q \qquad (((p \vee q) \vee (\neg r)) \Rightarrow ((r \wedge p) \vee r)) \Leftrightarrow q$$

En la pràctica escriurem algunes altres connectives lògiques tal i com es recull en la següent taula.

En la práctica y por comodidad escribiremos algunas conectivas lógicas más tal y como se recoge en la siguiente tabla.

connectiva *conectiva*	equivalència *equivalencia*	nom *nombre*
$p \Leftarrow q$	$q \Rightarrow p$	implicació oposada o recíproca *implicación opuesta o recíproca*
$p \not\Leftrightarrow q$	$\neg(p \Leftrightarrow q)$	**disjunció exclusiva** *disyunción exclusiva*
$p \uparrow q$	$\neg(p \wedge q)$	conjunció oposada *conjunción opuesta* (operador **NAND**)
$p \downarrow q$	$\neg(p \vee q)$	disjunció oposada *disyunción opuesta* (operador **NOR**)

A una expressió amb n variables diferents, li corresponen $|V_2^n| = 2^n$ possibles valors de veritat en funció que cada una de les variables siga vertadera o falsa. Aquesta informació en forma de taula rep el nom de **taula de veritat** de l'expressió.

A una expresión con n variables diferentes, le corresponden $|V_2^n| = 2^n$ posibles valores de verdad en función de que cada una de las variables sea verdadera o falsa. Esta información en forma de tabla recibe el nombre de **tabla de verdad** *de la expresión.*

Les taules de veritat corresponents a les connectives bàsiques en lògica proposicional són:

Las tablas de verdad correspondientes a las connectivas bàsicas en lógica proposicional son:

p	q	$\neg q$	$p \wedge q$	$p \vee q$	$p \Rightarrow q$	$p \Leftrightarrow q$
V	V	F	V	V	V	V
V	F	V	F	V	F	F
F	V		F	V	V	F
F	F		F	F	V	V

Diem que dues expressions, e_1 i e_2, són **equivalents** si tenen la mateixa taula de veritat. S'escriu

Decimos que dos expresiones, e_1 y e_2, son **equivalentes** *si tienen la misma tabla de verdad. Se escribe*

$$e_1 \equiv e_2.$$

Una expressió que sempre és vertadera s'anomena **tautologia**. Una expressió que sempre és falsa s'anomena **contradicció**.

Una expresión que siempre es verdadera se denomina **tautología**. *Una expresión que siempre es falsa se denomina* **contradicción**.

2.1.1 Equivalències amb disjuncions i conjuncions

Equivalencias con disyunciones y conjunciones

Llei del tercer exclòs | ***Ley del tercio excluso***

$$p \vee \neg p \equiv V$$

Llei de contradicció | ***Ley de contradicción***

$$p \wedge \neg p \equiv F$$

Lleis d'identitat (elements neutres) | ***Leyes de identidad (elementos neutros)***

$$p \vee F \equiv p \qquad p \wedge V \equiv p$$

Lleis de dominació | ***Leyes de dominación***

$$p \vee V \equiv V \qquad p \wedge F \equiv F$$

Lleis d'idempotència | ***Leyes de idempotencia***

$$p \vee p \equiv p \qquad p \wedge p \equiv p$$

Llei de doble negació | ***Ley de doble negación***

$$\neg\neg p \equiv \neg(\neg p) \equiv p$$

Lleis commutatives | ***Leyes conmutativas***

$$p \vee q \equiv q \vee p \qquad p \wedge q \equiv q \wedge p$$

Lleis associatives | ***Leyes asociativas***

$$(p \vee q) \vee r \equiv p \vee (q \vee r) \qquad (p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$$

Lleis distributives (factor comú)	***Leyes distributivas (factor común)***
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$

Lleis de De Morgan	***Leyes de De Morgan***
$\neg(p \vee q) \equiv \neg p \wedge \neg q$	$\neg(p \wedge q) \equiv \neg p \vee \neg q$

Lleis d'absorció de la conjunció/disjunció	***Leyes de absorción de la conjunción/disyunción***
$(p \vee q) \wedge p \equiv p$	$(p \wedge q) \vee p \equiv p$

Una variable proposicional o la seua negació s'anomena **literal**. Una **clàusula disjuntiva** o simplement **clàusula** és qualsevol disjunció de literals.

Una variable proposicional o su negación se llama **literal***. Una* **cláusula disyuntiva** *o simplemente* **cláusula** *es cualquier disyunción de literales.*

$$(\ell_1 \vee \ell_2 \vee \cdots \vee \ell_k)$$

Anàlogament, una **clàusula conjuntiva** és qualsevol conjunció de literals.

Análogamente, una **cláusula conjuntiva** *es cualquier conjunción de literales.*

$$(\ell_1 \wedge \ell_2 \wedge \cdots \wedge \ell_k)$$

Un tipus de clàusules que seran importants més endavant són les anomenades **clàusules de Horn** que són aquelles disjuncions de literals on no pot haver-hi més d'un literal no negat.

Un tipo de cláusulas que serán importantes más adelante son las llamadas **cláusulas de Horn** *que són aquellas disyunciones de literales donde no puede haber más de un literal no negado.*

$$\neg p \vee q \vee \neg r \qquad\qquad \neg p \vee \neg q$$

2.1.2 Equivalències bàsiques amb implicacions

Equivalencias básicas con implicaciones

Definició de la implicació | ***Definición de la implicación***

$$p \Rightarrow q \equiv \neg p \vee q$$

Definició de la coimplicació | ***Definición de la coimplicación***

$$p \Leftrightarrow q \equiv (p \Rightarrow q) \wedge (q \Rightarrow p)$$

Contraposició lògica | ***Contraposición lógica***

$$p \Rightarrow q \equiv \neg q \Rightarrow \neg p$$

2.1.3 Algunes propietats interessants de les implicacions

Algunas propiedades interesantes de las implicaciones

Distributivitat per l'esquerra respecte de la disjunció | **Distributividad por la izquierda *respecto a la disyunción***

$$\boxed{p \Rightarrow (q \vee r) \equiv (p \Rightarrow q) \vee (p \Rightarrow r)}$$

$$p \Rightarrow (q \vee r) \overset{\text{(def.)}}{\equiv} \neg p \vee (q \vee r) \overset{\text{(idemp.)}}{\equiv} \neg p \vee q \vee \neg p \vee r \overset{\text{(def.)}}{\equiv} (p \Rightarrow q) \vee (p \Rightarrow r)$$

Distributivitat per l'esquerra respecte de la conjunció | **Distributividad por la izquierda *respecto a la conjunción***

$$\boxed{p \Rightarrow (q \wedge r) \equiv (p \Rightarrow q) \wedge (p \Rightarrow r)}$$

$$p \Rightarrow (q \wedge r) \overset{\text{(def.)}}{\equiv} \neg p \vee (q \wedge r) \overset{\text{(distr.)}}{\equiv} (\neg p \vee q) \wedge (\neg p \vee r) \overset{\text{(def.)}}{\equiv} (p \Rightarrow q) \wedge (p \Rightarrow r)$$

Pseudodistributivitat per la dreta respecte de la disjunció/conjunció

Pseudodistributividad por la derecha respecto a la disyunción/conjunción

Es pot traure factor comú per la dreta però canviant conjuncions per disjuncions.

Se puede sacar factor común por la derecha pero cambiando conjunciones por disyunciones.

$$\boxed{(p \vee q) \Rightarrow r \equiv (p \Rightarrow r) \wedge (q \Rightarrow r)}$$

$$(p \vee q) \Rightarrow r \overset{\text{(def.)}}{\equiv} \neg(p \vee q) \vee r \overset{\text{(Morgan)}}{\equiv} (\neg p \wedge \neg q) \vee r \overset{\text{(distr.)}}{\equiv}$$
$$\equiv (\neg p \vee r) \wedge (\neg q \vee r) \overset{\text{(def.)}}{\equiv} (p \Rightarrow r) \wedge (q \Rightarrow r)$$

Pseudodistributivitat per la dreta respecte de la conjunció/disjunció

Pseudodistributividad por la derecha respecto a la conjunción/disyunción

Es pot traure factor comú per la dreta però canviant disjuncions per conjuncions.

Se puede sacar factor común por la derecha pero cambiando disyunciones por conjunciones.

$$\boxed{(p \wedge q) \Rightarrow r \equiv (p \Rightarrow r) \vee (q \Rightarrow r)}$$

$$(p \wedge q) \Rightarrow r \overset{\text{(def.)}}{\equiv} \neg(p \wedge q) \vee r \overset{\text{(Morgan)}}{\equiv} (\neg p \vee \neg q) \vee r \overset{\text{(idemp.)}}{\equiv}$$
$$\equiv \neg p \vee r \vee \neg q \vee r \overset{\text{(def.)}}{\equiv} (p \Rightarrow r) \vee (q \Rightarrow r)$$

Les **clàusules de Horn** amb exactament un literal positiu (no negat) s'anomenen **clàusules definides** i es poden escriure com una implicació l'antecedent de la qual és una conjunció de literals positius. En altres paraules,

Las **cláusulas de Horn** *con exactamente un literal positivo (no negado) se llaman* **cláusulas definidas** *y se pueden escribir como una implicación cuyo antecedente es una conjunción de literales positivos. En otras palabras,*

$$\neg p_1 \vee \cdots \vee \neg p_k \vee q \equiv (\neg p_1 \vee q) \vee \cdots \vee (\neg p_k \vee q) \equiv$$
$$\equiv (p_1 \Rightarrow q) \vee \cdots \vee (p_k \Rightarrow q) \overset{(\text{pdistr})}{\equiv} (p_1 \wedge \cdots \wedge p_k) \Rightarrow q.$$

2.2 Implicacions, deduccions i inferència

Implicaciones, deducciones e inferencia

En una implicació, $p \Rightarrow q$, la proposició a l'esquerra, p, rep el nom d'**antecedent** i la proposició a la dreta és el **conseqüent**.

En una implicación, $p \Rightarrow q$, la proposición de la izquierda, p, recibe el nombre de **antecedente**, *y la proposición de la derecha es el* **consecuente**.

És important adonar-se que la implicació sempre és certa si l'antecedent és fals. Això vol dir que per a comprovar la veracitat d'una implicació, només cal plantejar-se el cas en què l'antecedent és vertader. Si en aquest cas el conseqüent també és vertader, la implicació serà certa.

Es importante darse cuenta de que la implicación siempre es cierta si el antecedente es falso. Esto significa que para comprobar la veracidad de una implicación, sólo necesitamos plantearnos el caso en que el antecedente es verdadero. Si en ese caso el consecuente también es verdadero, la implicación será cierta.

Alternativament, podem veure el conseqüent com una proposició que és certa quan l'antecedent ho és. En altres paraules, podem deduir la veritat de q a partir de la veritat de p (sempre que siga cert que $p \Rightarrow q$).

Alternativamente, podemos ver el consecuente como una proposición que es cierta cuando el antecedente lo es. En otras palabras, podemos deducir la verdad de q a partir de la verdad de p (siempre que sea cierto que $p \Rightarrow q$).

En general, diem que una expressió, Q, es pot **deduir** a partir d'una altra expressió, P, si Q és certa quan P ho és. En aquest context, P rep el nom de **premissa** i Q s'anomena **conclusió semàntica** o simplement **conclusió**. Si P és una conjunció de dos o més subexpressions, $P = P_1, P_2, \ldots$, totes les P_i s'anomenen **premisses**.

En general, decimos que una expresión, Q, se puede **deducir** *a partir de otra expresión, P, si Q es cierta cuando P lo es. En este contexto, P recibe el nombre de* **premisa** *y Q se denomina* **conclusión semántica** *o simplemente* **conclusión**. *Si P es una conjunción de dos o más subexpresiones, $P = P_1, P_2, \ldots$, todas las P_i se denominan* **premisas**.

Quan una conclusió, Q, es dedueix d'un conjunt de premisses, $P_1, P_2, \ldots, P_k$, escrivim

Cuando una conclusión, Q, se deduce de un conjunto de premisas, $P_1, P_2, \ldots, P_k$, escribimos

$$P_1, P_2, \ldots, P_k \vdash Q$$

Aquesta expressió s'anomena també **teorema**. El conjunt de premisses es denomina també **hipòtesi** i la conclusió s'anomena **tesi**. Del fet de comprovar que la conclusió es pot deduir a partir de les premisses se'n diu també **demostració**.

Esta expresión se denomina también **teorema**. *Las premisas reciben también el nombre de* **hipótesis** *y la conclusión se denomina* **tesis**. *Al hecho de comprobar que la conclusión se puede deducir a partir de las premisas se le denomina también* **demostración**.

Si $P \Rightarrow Q$ aleshores $P \vdash Q$, i al revés

Si $P \Rightarrow Q$ entonces $P \vdash Q$, y al revés

O altrament expressat,

O de manera alternativa,

$$\boxed{P \vdash Q \text{ sii } P \Rightarrow Q}$$

A partir de la taula de veritat de $P \Rightarrow Q$ és obvi que Q és cert si P ho és. I, per tant,

A partir de la tabla de verdad de $P \Rightarrow Q$ es obvio que Q es cierto si P lo es. Y, por tanto,

$$P \vdash Q.$$

De la mateixa manera, quan $P \vdash Q$, l'únic cas prohibit és que P siga vertader i Q fals. I això coincideix exactament amb la taula de veritat de $P \Rightarrow Q$.

De la misma forma, cuando $P \vdash Q$, el único caso prohibido es que P sea verdadero y Q falso. Y esto coincide exactamente con la tabla de verdad de $P \Rightarrow Q$.

2.2.1 Algunes propietats de $\vdash$

Algunas propiedades de $\vdash$

Propietat reflexiva

Propiedad reflexiva

$$\boxed{p \vdash p}$$

Evident.

Evidente.

Propietat antisimètrica | ***Propiedad antisimétrica***

$$p \equiv q \text{ si } p \vdash q \wedge q \vdash p$$

Si q ha de ser cert quan p ho és i al revés, necessàriament hauran de ser tots dos certs o tots dos falsos. És a dir,

Si q tiene que ser cierto cuando p lo es y al revés, necesariamente tendrán que ser los dos ciertos o los dos falsos. Es decir,

$$p \equiv q.$$

Una conseqüència immediata de l'anterior i de la propietat que relaciona deducció i implicació és que

Una consecuencia inmediata de lo anterior y de la propiedad que relaciona deducción e implicacion es que

$$P \equiv Q \text{ sii } P \Leftrightarrow Q$$

Propietat transitiva | ***Propiedad transitiva***

$$p \vdash r \text{ si } p \vdash q \wedge q \vdash r$$

Quan p és cert, q ho ha de ser. I r també, perquè $q \vdash r$. Aleshores serà també cert que

Cuando p es cierto, q tiene que serlo. Y r también, por que $q \vdash r$. Entonces será cierto también que

$$p \vdash r.$$

A conseqüència de les tres propietats anteriors, la relació $\vdash$ és una relació binària d'ordre dins de les expressions en lògica proposicional.

Como consecuencia de las tres propiedades anteriores, la relación $\vdash$ es una relación binaria de orden dentro de las expresiones en lógica proposicional.

Fites inferior i superior | ***Cota inferior y superior***

$$F \vdash p \vdash V$$

La demostració és òbvia a partir de la definició de $\vdash$.

La demostración es obvia a partir de la definición de $\vdash$.

Quan escrivim $p \vdash V$ estem dient que qualsevol premissa p és bona per a deduir una cosa que és certa independentment de p.

Cuando escribimos $p \vdash V$ estamos diciendo que cualquier premisa p es buena para deducir algo que es cierto independientemente de p.

Quan escrivim $F \vdash p$ estem expressant que sempre es pot deduir qualsevol cosa quan la premissa no es compleix mai. Aquest cas concret s'anomena **inferència inconsistent** perquè en realitat no estem deduint res.

Cuando escribimos $F \vdash p$ estamos expresando que siempre se puede deducir cualquier cosa cuando la premisa no se cumple nunca. Este caso concreto se denomina **inferencia inconsistente** *por que en realidad no estamos deduciendo nada.*

L'expressió $F \vdash p$ no s'hauria de confondre amb l'expressió $\vdash p$ que és sovint utilitzada per a indicar que p és una tautologia. O, equivalentment, que la veritat de p es pot deduir a partir d'un conjunt buit de premisses.

La expresión $F \vdash p$ no debería confundirse con la expresión $\vdash p$ que se usa a menudo para indicar que p es una tautología. O, de manera equivalente, que la verdad de p se puede deducir a partir de un conjunto vacío de premisas.

De fet, la conjunció d'un conjunt buit de premisses és V (l'element neutre de $\wedge$) per la qual cosa les expressions $\vdash p$ i $V \vdash p$ són equivalents i signifiquen que $p \equiv V$.

De hecho, la conjunción de un conjunto vacío de premisas es V (el elemento neutro de $\wedge$) por lo que las expresiones $\vdash p$ y $V \vdash p$ son equivalentes y significan que $p \equiv V$.

$P \Rightarrow Q$ en llenguatge natural | ***$P \Rightarrow Q$ en lenguaje natural***

Hi ha moltes maneres de referir-se a la implicació en la parla informal. Heus-ne ací algunes:

Existen muchas maneras de referirse a la implicación en el habla informal. Algunas de ellas son:

• si P aleshores/llavors Q. • P implica Q. • P, llavors/aleshores Q. • quan P, Q. • Q si P. • no P o Q (només en alguns casos com per exemple: "No cantes o plourà")	• *si* P *entonces* Q. • P *implica* Q. • P, *luego/entonces* Q. • *cuando* P, Q. • Q *si* P.

P $\Leftarrow$ Q **en llenguatge natural**	P $\Leftarrow$ Q ***en lenguaje natural***
També hi ha maneres informals d'expressar la implicació contrària:	*También hay maneras informales de expresar la implicación contraria:*
• només si P aleshores Q. • Q només si P. • només quan P, Q.	• *sólo/solamente si* P *entonces* Q. • Q *sólo si* P. • *sólo cuando* P, Q.

P $\Leftrightarrow$ Q **en llenguatge natural**	P $\Leftrightarrow$ Q ***en lenguaje natural***
De la mateixa manera, la coimplicació (en aquest cas sempre podem intercanviar P i Q):	*De la misma manera, la coimplicación (en este caso siempre podemos intercambiar* P *y* Q*):*
• si i només si P aleshores Q. • P si i només si Q. • P sii Q.	• *si y sólo si* P *entonces* Q. • P *si y sólo si* Q. • P sii Q.

2.2.2 Regles d'inferència estàndard

Reglas de inferencia estándar

Hi ha alguns esquemes que permeten deduir expressions a partir de premisses amb una certa forma. Això és el que es coneix com a **regles d'inferència**. Algunes són més que òbvies, mentre que altres requereixen un poc més d'atenció.

Hay algunos esquemas que permiten deducir expresiones a partir de premisas con una cierta forma. Esto es lo que se conoce como **reglas de inferencia**. *Algunas de ellas son más que obvias, mientras otras requieren un poco más de atención.*

Totes tenen el seu origen en la lògica clàssica i el seu objectiu és que es puguen combinar per a dur a terme deduccions més complexes.

Todas tienen su origen en la lógica clásica y su objetivo es que se puedan combinar para dar lugar a deducciones más complejas.

Una manera alternativa i més gràfica d'expressar una regla $P_1, P_2, \ldots \vdash Q$ és

Una forma alternativa y más gráfica de expresaer una regla $P_1, P_2, \ldots \vdash Q$ *es*

$$\begin{array}{c} P_1 \\ P_2 \\ \vdots \\ \hline Q \end{array}$$

És a dir, una línia per a cada una de les premisses i per a la conclusió. D'aquesta manera és fàcil aplicar noves regles usant premisses o conclusions anteriors. Només cal identificar clarament cada una de les línies perquè quede clar quina regla s'aplica i sobre quines premisses per a l'obtenció de cada nova línia.

Es decir, una línea por cada una de las premisas y para la conclusión. De esta manera es fácil aplicar nuevas reglas usando premisas o conclusiones anteriores. Sólo es necesario identificar claramente cada una de las líneas para que quede claro qué regla se aplica y sobre qué premisas para la obtención de cada nueva línea.

EC. Eliminació de la conjunció | ***Eliminación de la Conjunción***

$$\frac{p \wedge q}{p} \qquad\qquad \frac{p \wedge q}{q}$$

Aquesta regla s'obté directament a partir de la taula de veritat de la conjunció.

Esta regla se obtiene directamente a partir de la tabla de verdad de la conjunción.

IC. Introducció de la conjunció | *Introducción de la Conjunción*

$$\frac{\begin{array}{c}p\\ q\end{array}}{p \wedge q}$$

Més que d'una regla, es tracta d'expressar el mateix de dues maneres diferents.

Más que de una regla, se trata de expresar lo mismo de dos maneras distintas.

$$p, q \equiv p \wedge q$$

La deducció en el sentit contrari es pot obtenir molt fàcilment a partir de l'aplicació consecutiva de les dues regles **EC** sobre $p \wedge q$.

La deducción en el sentido contrario se puede obtener fácilmente a partir de la aplicación consecutiva de las dos reglas **EC** *sobre* $p \wedge q$.

ID. Introducció de la disjunció | *Introducción de la Disyunción*

$$\frac{p}{p \vee q} \qquad\qquad \frac{q}{p \vee q}$$

Molt similar a la regla **EC**. També s'obté directament a partir de la taula de veritat de la disjunció.

Muy similar a la regla **EC**. *También se obtiene directamente a partir de la tabla de verdad de la disyunción.*

EN. Eliminació de la negació | *Eliminación de la Negación*

$$\frac{\neg\neg p}{p}$$

És en realitat una equivalència (llei de doble negació).

Es en realidad una equivalencia (ley de doble negación).

IN. Introducció de la negació, o també reducció a l'absurd

Introducción de la Negación, o también reducción al absurdo

$$\frac{\begin{array}{c}[\mathrm{p}]\\ \vdots \\ \mathrm{F}\end{array}}{\neg \mathrm{p}}$$

És una manera molt usual de raonar. Suposem certa una expressió, p, i ho indiquem mitjançant claudàtors. Si a partir d'ací aconseguim deduir F, és a dir, arribem a una contradicció, és perquè p ha de ser fals i aleshores $\neg\mathrm{p}$ és cert.

Es una forma muy usual de razonar. Suponemos cierta una expresión, p*, y lo indicamos mediante corchetes. Si a partir de ahí conseguimos deducir* F*, es decir, llegamos a una contradicción, es por que* p *ha de ser falso y entonces* $\neg\mathrm{p}$ *es cierto.*

ED. Eliminació de la disjunció, o també demostració per casos

Eliminación de la Disyunción, o también demostración por casos

$$\frac{\begin{array}{c}\mathrm{p} \vee \mathrm{q}\\ [\mathrm{p}]\\ \vdots \\ \mathrm{r}\\ [\mathrm{q}]\\ \vdots \\ \mathrm{r}\end{array}}{\mathrm{r}}$$

És també molt usual i es pot estendre a disjuncions de tres o més proposicions. Si en suposar per separat cada una de les opcions en una disjunció, s'arriba a la mateixa conclusió, aleshores aquesta conclusió es pot deduir en general.

Es también muy usual y se puede extender a disyunciones de tres o más proposiciones. Si al suponer por separado cada una de las opciones en una disyunción, se llega a la misma conclusión, entonces esta conclusión se puede deducir en general.

EI. Eliminació de la implicació, o també modus ponens

Eliminación de la Implicación, o también modus ponens

$$\frac{\begin{array}{c} p \Rightarrow q \\ p \end{array}}{q}$$

És una de les formes més clàssiques de raonar. El nom complet és *modus ponendo ponens*: mode que en afirmar afirma.

Es una de las formas más clásicas de razonar. El nombre completo es modus ponendo ponens*: modo que al afirmar, afirma.*

Si una implicació és certa i també ho és el seu antecedent, aleshores ho serà també el conseqüent.

Si una implicación es cierta y también lo es su antecedente, entonces lo será también el consecuente.

II. Introducció de la implicació, o també demostració de la implicació

Introducción de la Implicación, o también demostración de la implicación

$$\frac{\begin{array}{c} [p] \\ \vdots \\ q \end{array}}{p \Rightarrow q}$$

Si suposem cert p, i a partir d'això arribem a q aleshores serà cert que

Si suponemos cierto p, y a partir de esto llegamos a q entonces será cierto que

$$p \Rightarrow q.$$

Les vuit regles d'inferència estàndard (introducció/eliminació de la negació/conjunció/disjunció/implicació) són les que farem servir per a dur a terme deduccions i demostracions més complexes.

Las ocho reglas de inferencia estándar (introducción/eliminación de la negación/conjunción/disyunción/implicación) son las que usaremos para llevar a cabo deducciones y demostraciones más complejas.

No obstant això, hi ha altres regles d'inferència clàssiques tan famoses o més que les que nosaltres anomenem estàndard. A continuació n'enunciem algunes i les deduirem a partir de les estàndard.

No obstante, hay otras reglas de inferencia clásicas tan famosas o más que las que nosotros llamamos estándar. A continuación enunciamos algunas de éstas y las deduciremos a partir de las estándar.

modus tollens

$$\frac{\begin{array}{c} p \Rightarrow q \\ \neg q \end{array}}{\neg p}$$

És com el modus ponens, però ara neguem el conseqüent per a demostrar la falsedat de l'antecedent. El nom complet és *modus tollendo tollens*: mode que en negar nega.

Es como el modus ponens pero ahora negamos el consecuente para demostrar la falsedad del antecedente. El nom complet és modus tollendo tollens*: modo que al negar, niega.*

Demostrarem la deducció,

Vamos a demostrar la deducción,

$$p \Rightarrow q, \neg q \vdash \neg p,$$

primer mitjançant regles estàndard:

primero mediante reglas estándar:

1)	$[p]$	comença **IN**, suposem el contrari del que volem deduir ... *comienza* **IN**, *suponemos lo contrario de lo que queremos deducir*
2)	$p \Rightarrow q$	... premissa 1
3)	q	... **EI** (modus ponens) en 1), 2)
4)	$\neg q$	... premissa 2
5)	$q \wedge \neg q \equiv F$	... **IC** en 3), 4)
6)	$\neg p$	acaba **IN** en 1)

Alternativament, podem aplicar la contraposició lògica (l'última de les equivalències notables en la pàgina 66) per a modificar la implicació i aleshores demostrar-ho només aplicant **EI** (modus ponens).

Alternativamente, podemos aplicar la contraposición lógica (la última de las equivalencias notables en la página 66) para modificar la implicación y entonces demostrarlo sólo aplicando **EI** *(modus ponens).*

modus tollendo ponens

o també sil·logisme disjuntiu | ***o también* silogismo disyuntivo**

$$\frac{\begin{array}{c} p \vee q \\ \neg p \end{array}}{q} \qquad \frac{\begin{array}{c} p \vee q \\ \neg q \end{array}}{p}$$

modus tollendo ponens: mode que en negar afirma. Aquesta regla d'inferència s'il·lustra en l'exercici 2.6

modus tollendo ponens*: modo que al negar, afirma. Esta regla de inferencia se ilustra en el ejercicio 2.6*

modus ponendo tollens

$$\frac{\begin{array}{c} \neg(p \wedge q) \\ p \end{array}}{\neg q} \qquad \frac{\begin{array}{c} \neg(p \wedge q) \\ q \end{array}}{\neg p}$$

modus ponendo tollens: mode que en negar afirma. Aquesta regla es pot obtenir a partir de l'anterior només canviant cada proposició per la seua negació.

modus ponendo tollens*: modo que al negar, afirma. Esta regla se puede obtener a partir de la anterior sólo cambiando cada proposición por su negación.*

sil·logisme hipotètic | **silogismo hipotético**

$$\frac{\begin{array}{c} p \Rightarrow q \\ q \Rightarrow r \end{array}}{p \Rightarrow r}$$

Aquesta regla d'inferència s'obté a partir de la transitivitat de la implicació. Utilitzant regles d'inferència seria:

Esta regla de inferencia se obtiene a partir de la transitividad de la implicació. Utilizando reglas de inferencia sería:

1)	[p]	començа **II**, suposem l'antecedent *comienza* **II**, *suponemos el antecedente*
2)	$p \Rightarrow q$	... premissa 1
3)	q	... **EI** en 1), 2)
4)	$q \Rightarrow r$	... premissa 2
5)	r	... **EI** en 3), 4)
6)	$p \Rightarrow r$	acaba **II** en 1)

Llei d'absorció | Ley de absorción

$$\frac{p \Rightarrow q}{p \Rightarrow p \wedge q}$$

Aquesta regla d'inferència és en realitat una equivalència.

Esta regla de inferencia es en realidad una equivalencia.

$$p \Rightarrow p \wedge q \overset{(1)}{\equiv} \neg p \vee (p \wedge q) \overset{(2)}{\equiv} (\neg p \vee p) \wedge (\neg p \vee q) \overset{(3)}{\equiv} V \wedge (p \Rightarrow q) \overset{(4)}{\equiv} p \Rightarrow q$$

En cada pas s'ha aplicat la definició de la implicació (1), la propietat (llei) distributiva (2), la llei del tercer exclòs i la definició de la implicació (3), i la llei d'identitat de la conjunció (4). Totes elles equivalències notables enumerades en la secció 2.1.1, pàgina 64.

En cada paso se ha aplicado la definición de la implicación (1), la propiedad (ley) distributiva (2), la ley del tercio excluso y la definición de la implicación (3), y la ley de identidad de la conjunción (4). Todas ellas equivalencias notables enumeradas en la Sección 2.1.1, página 64.

Alternativament, podem aplicar la distributivitat per l'esquerra de la implicació (a), la definició de la implicació (b), la llei del tercer exclòs (c), i la llei d'identitat de la conjunció (d).

Alternativamente, podemos aplicar la distributividad por la izquierda de la implicación (a), la definición de la implicación (b), la ley del tercio excluso (c), y la ley de identidad de la conjunción (d).

$$p \Rightarrow p \wedge q \overset{(a)}{\equiv} (p \Rightarrow p) \wedge (p \Rightarrow q) \overset{(b)}{\equiv} (\neg p \vee p) \wedge p \Rightarrow q \overset{(c)}{\equiv} V \wedge p \Rightarrow q \overset{(d)}{\equiv} p \Rightarrow q$$

Llei d'exportació | Ley de exportación

$$p \Rightarrow (q \Rightarrow r) \equiv (p \wedge q) \Rightarrow r$$

És una equivalència que es pot aplicar com a regla d'inferència en els dos sentits. La demostració es deixa com a exercici.

Es una equivalencia que se puede aplicar como regla de inferencia en los dos sentidos. Su demostración se deja como ejercicio.

dilema constructiu | **dilema constructivo**

$$\begin{array}{c} p \Rightarrow r \\ q \Rightarrow s \\ p \vee q \\ \hline r \vee s \end{array}$$

Aquesta regla d'inferència és una versió disjuntiva del modus ponens. La demostració es deixa com a exercici.

Esta regla de inferencia es una versión disyuntiva del modus ponens. Su demostración se deja como ejercicio.

dilema destructiu | **dilema destructivo**

$$\begin{array}{c} p \Rightarrow r \\ q \Rightarrow s \\ \neg r \vee \neg s \\ \hline \neg p \vee \neg q \end{array}$$

Aquesta regla d'inferència és una versió disjuntiva del modus tollens. La demostració es deixa com a exercici.

Esta regla de inferencia es una versión disyuntiva del modus tollens. Su demostración se deja como ejercicio.

resolució | **resolución**

$$\begin{array}{c} p \vee q \\ \neg p \vee r \\ \hline q \vee r \end{array}$$

Aquesta regla d'inferència relaciona i simplifica clàusules. La demostració es deixa com a exercici.

Esta regla de inferencia relaciona y simplifica cláusulas. Su demostración se deja como ejercicio.

2.3 Lògica de predicats

Lógica de predicados

Un **predicat** k-ari o d'ordre k és una aplicació del producte cartesià de k dominis o universos (contextos) al conjunt $\{V, F\}$,

*Un **predicado** k-ario o de orden k es una aplicación del producto cartesiano de k dominios o universos (contextos) al conjunto $\{V, F\}$,*

$$P : D_1 \times D_2 \times \cdots \times D_k \longrightarrow \{V, F\},$$

de manera que $P(a_1, a_2, \ldots, a_k)$, una vegada fixats els $a_i \in D_i$, ha de ser o vertader o fals.

de manera que $P(a_1, a_2, \ldots, a_k)$, una vez fijados los $a_i \in D_i$, tiene que ser o verdadero o falso.

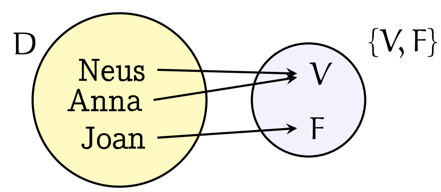
Alternativament, un predicat es pot veure també com una relació k-ària en els conjunts $D_1, D_2, \ldots, D_k$ de manera que podem escriure (abusant de la notació) que

Alternativamente, un predicado se puede ver también como una relación k-aria en los conjuntos $D_1, D_2, \ldots, D_k$ de manera que podemos escribir (abusando de la notación) que

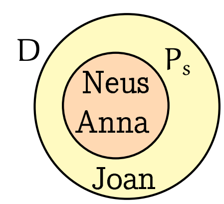
$$P = \{(a_1, a_2, \ldots, a_k) \mid P(a_1, a_2, \ldots, a_k) \equiv V\} \subseteq D_1 \times D_2 \times \cdots \times D_k.$$

Exemples de dominis poden ser: persones, animals, nombres, símbols, etc. Exemples de predicats amb $k = 1$ i $k = 2$ sobre un mateix domini format per tres persones es mostren a continuació com a aplicació i com a relació (i com a correspondència en el cas $k = 2$).

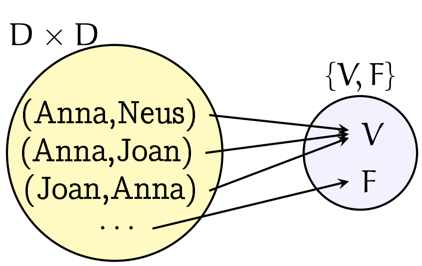
Ejemplos de dominios pueden ser: personas, animales, números, símbolos, etc. Ejemplos de predicados con $k = 1$ y $k = 2$ sobre un mismo dominio formado por tres personas se muestran a continuación como aplicación y como relación (y como correspondencia en el caso $k = 2$).



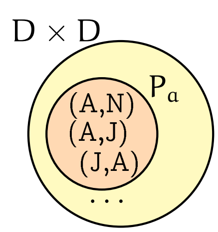
predicat P_s com a aplicació
predicado P_s como aplicación



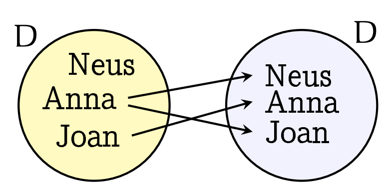
predicat P_s com a relació
predicado P_s como relación



P_a com a aplicació
P_a como aplicación



P_a com a relació
P_a como relación



P_a com a correspondència
P_a como correspondencia

Cada element concret de cada un dels dominis implicats són **constants**. A més a més podem utilitzar **variables** per a referir-nos a elements **genèrics** o **indeterminats** de cada un dels dominis. Normalment escriurem les variables com a x_i o y_i.

Cada elemento concreto de cada uno de los dominios implicados son **constantes**. *Además podemos utilizar* **variables** *para referirnos a elementos* **genéricos** *o* **indeterminados** *de cada uno de los dominios. Normalmente escribiremos las variables como x_i o y_i.*

Les constants i/o les variables es poden combinar per a donar lloc a nous elements (de nous dominis) mitjançant les anomenades **funcions** o **functors**. Alguns exemples de funcions que poden servir com a arguments de predicats serien

Las constantes y/o las variables se pueden combinar para dar lugar a nuevos elementos (de nuevos dominios) mediante las denominadas **funciones** *o* **functores**. *Algunos ejemplos de funciones que pueden servir como argumentos de predicados serían*

$$f(a_1, x_2) \qquad g(a_1, f(x_1, x_2)) \qquad h(a_1, h(x_1, x_2))$$

Tant les constants com les variables i les funcions poden ser arguments dels predicats i s'anomenen en general **termes**.

Tanto las constantes como las variables y las funciones pueden ser argumentos de los predicados y se denominan en general **términos**.

Una **fórmula atòmica** és un predicat k-ari aplicat sobre k termes.

Una **fórmula atómica** *es un predicado k-ario aplicado sobre k términos.*

$$P(t_1, t_2, \ldots, t_k)$$

Fórmula ben formada (fbf)

Fórmula bien formada (fbf)

1. Si A és una fórmula atòmica, aleshores és fbf.
2. Si A i B són fbfs, aleshores també ho seran

1. Si A es una fórmula atómica, entonces es fbf.

2. Si A y B son fbfs, entonces también lo serán

$$\neg A, A \wedge B.A \vee B, A \Rightarrow B, A \Leftrightarrow B$$

3. Si $A(.., v, ..)$ és una fbf i $v \in D$ és una de les variables que conté, aleshores també ho seran

3. Si $A(.., v, ..)$ es una fbf y $v \in D$ es una de las variables que contiene, entonces también lo serán

$$\forall v : v \in D', A(.., v, ..) \qquad \exists v, v \in D' : A(.., v, ..)$$

La fórmula

La fórmula

$$\forall v : v \in D', A(.., v, ..)$$

es llegeix "per a tot element v tal que v pertany al conjunt D', $A(.., v, ..)$". I significa que el predicat A és cert per a **tots** els valors de v dins del subdomini D' que pot ser qualsevol subconjunt del domini corresponent a l'argument del predicat on apareix v. Aquest nou operador s'anomena **quantificador universal** o simplement **generalitzador**.

se lee "para todo elemento v tal que v pertenece al conjunto D', $A(.., v, ..)$". Y significa que el predicado A es cierto para **todos** *los valores de v dentro del subdominio D' que puede ser cualquier subconjunto del dominio correspondiente al argumento del predicado donde aparece v. Este nuevo operador se denomina* **cuantificador universal** *o simplemente* **generalizador**.

La fórmula

La fórmula

$$\exists v, v \in D' : A(.., v, ..)$$

es llegeix "existeix un element v, que pertany al conjunt D', tal que $A(.., v, ..)$". I significa que el predicat A és cert per a **algun** dels valors de v en D'. Aquest nou operador s'anomena **quantificador existencial** o simplement **particularitzador**.

se lee "existe un elemento v, que pertenece al conjunto D', tal que $A(.., v, ..)$".' Y significa que el predicado A es cierto para **alguno** *de los valores de v en D'. Este nuevo operador se denomina* **cuantificador existencial** *o simplemente* **particularizador**.

Quan el conjunt D' associat als quantificadors siga el mateix domini D associat a l'argument corresponent del predicat i/o aquest siga prou clar pel context, escriurem simplement

Cuando el conjunto D' asociado a los cuantificadores sea el mismo dominio D asociado al argumento correspondiente del predicado y/o éste esté suficientemente claro por el contexto, escribiremos simplemente

$$\forall v, A(.., v, ..) \qquad \exists v : A(.., v, ..)$$

Els quantificadors universal i existencial es poden escriure com a conjuncions i disjuncions, respectivament. Si $D = \{a_1, \ldots, a_n\}$ tenim que

Los cuantificadores universal y existencial se pueden escribir como conjunciones y disyunciones, respectivamente. Si $D' = \{a_1, \ldots, a_n\}$ tenemos que

$$\forall v : v \in D', A(.., v, ..) \equiv \bigwedge_{v \in D'} A(.., v, ..) \equiv A(.., a_1, ..) \wedge \cdots \wedge A(.., a_n, ..)$$
$$\exists v, v \in D' : A(.., v, ..) \equiv \bigvee_{v \in D'} A(.., v, ..) \equiv A(.., a_1, ..) \vee \cdots \vee A(.., a_n, ..)$$

amb l'única particularitat que aquestes conjuncions o disjuncions poden ser infinites.

con la única particularidad de que estas conjunciones o disyunciones pueden ser infinitas.

Si una fórmula ben formada **no** conté variables o si les que conté estan associades a quantificadors, s'anomena **fórmula tancada**. En cas contrari s'anomena **fórmula oberta**. Les variables no associades a quantificadors en una fórmula s'anomenen **variables lliures** mentre que les associades són **variables lligades**.

Si una fórmula bien formada **no** *contiene variables o si las que contiene están asociadas a cuantificadores, se la llama* **fórmula cerrada**. *En caso contrario se denomina* **fórmula abierta**. *Las variables no asociadas a cuantificadores en una fórmula se las llama* **variables libres** *mientras que les asociadas son* **variables ligadas**.

2.4 Inferència amb predicats

Inferencia con predicados

A una fórmula tancada li correspon un valor de veritat: cert o fals. En canvi, el valor de veritat d'una fórmula oberta dependrà dels valors que puguen prendre les seues variables que no estiguen lligades a quantificadors.

A una fórmula cerrada le corresponden un valor de verdad: cierto o falso. En cambio, el valor de verdad de una fórmula abierta dependerá de los valores que puedan tomar sus variables que no estén ligadas a cuantificadores.

Amb fórmules tancades o obertes però sense quantificadors es pot raonar de la mateixa manera que amb proposicions. En particular, es poden aplicar totes les regles d'inferència de la lògica proposicional. L'única particularitat de les fórmules obertes és que el raonament depèn dels possibles valors de les variables lliures. Però per a raonar amb quantificadors, necessitarem introduir dos nous parells de regles d'inferència.

Con fórmulas cerradas o abiertas pero sin cuantificadores se puede razonar de la misma manera que con proposiciones. En particular, se pueden aplicar todas las reglas de inferencia de la lógica proposicional. La única particularidad de las fórmulas abiertas es que el razonamiento depende de los posibles valores de las variables libres. Pero para razonar con cuantificadores, necesitaremos introducir don nuevos pares de reglas de inferencia.

EG. Eliminació del generalitzador | ***Eliminación del Generalizador***

$$\frac{\forall x, P(x)}{P(a)}$$

En la fórmula $P(a)$, a és un element qualsevol del domini que ens puga interessar. Pot ser una variable o una constant que haja aparegut anteriorment. O pot ser una nova variable sense cap mena de restricció. Normalment usarem les lletres $a, b, c, \ldots$

En la fórmula $P(a)$, a es un elemento cualquiera del dominio que nos pueda interesar. Puede ser una variable o una constante que haya aparecido anteriormente. O puede ser una nueva variable sin ningún tipo de restricción. Normalmente usaremos las letras $a, b, c, \ldots$

IG. Introducció del generalitzador | ***Introducción del Generalizador***

$$\frac{P(a)}{\forall x, P(x)}$$

Només en el cas que a siga una variable sobre la qual no hi haja cap mena de restricció, podem substituir la fórmula $P(a)$ pel corresponent generalitzador.

Sólo en el caso que a sea una variable sobre la que no haya ningún tipo de restricción, podemos sustituir la fórmula $P(a)$ por el correspondiente generalizador.

IP. Introducció del particularitzador | ***Introducción del Particularizador***

$$\frac{P(a)}{\exists x : P(x)}$$

La regla és formalment idèntica a l'anterior. Però hi ha una diferència fonamental: ara es pot aplicar independentment del que siga a.

La regla es formalmente idéntica a la anterior. Pero hay una diferencia fundamental: ahora se puede aplicar independientemente de lo que sea a.

EP. Eliminació del particularitzador | *Eliminación del Particularizador*

$$\begin{array}{c} \exists x : P(x) \\ [P(a_i)] \\ \vdots \\ Q \\ \hline Q \end{array}$$

Podem eliminar l'i-èsim particularitzador si introduïm una nova variable, a_i. A partir d'aquest moment continuem raonant fins que arribem a una fórmula, Q, que no depèn de a_i.

Podemos eliminar el i-ésimo particularizador si introducimos una nueva variable, a_i. A partir de ese momento continuamos razonando hasta que llegamos a una fórmula, Q, que no depende de a_i.

2.4.1 Algunes propietats interessants amb quantificadors

Lleis de De Morgan amb quantificadors | *Leyes de De Morgan con cuantificadores*

$$\neg(\forall x \in D, P(x)) \equiv \exists x \in D : \neg P(x)$$
$$\neg(\exists x \in D, P(x)) \equiv \forall x \in D : \neg P(x)$$

Reordenació de quantificadors | *Reordenación de cuantificadores*

$$\forall x, \forall y, P(x,y) \equiv \forall y, \forall x, P(x,y)$$
$$\exists x : \exists y : P(x,y) \equiv \exists y : \exists x : P(x,y)$$

Normalment i per simplificar, escriurem

Normalmente y para simplificar escribiremos

$$\forall x, y, z, \ldots, P(x,y,z,\ldots) \qquad \exists x, y, z, \ldots : P(x,y,z,\ldots)$$

en lloc de

en lugar de

$$\forall x, \forall y, \forall z, \ldots P(x, y, z, \ldots) \qquad \exists x : \exists y : \exists z : \cdots : \; P(x, y, z, \ldots)$$

Un detall important és que quantificadors diferents no es poden intercanviar.

Un detalle importante es que cuantificadores diferentes no se pueden intercambiar.

$$\forall x, \exists y : P(x, y) \not\equiv \exists y : \forall x, P(x, y)$$

Exemple: existeix per a tot?

Ejemplo: ¿existe para todo?

És cert que

¿Es cierto que

$$\exists y : \forall x, \; P(x, y) \vdash \forall x, \exists y \; : \; P(x, y)?$$

Mitjançant regles estàndard a partir de la premissa es té que

Mediante reglas estándar a partir de la premisa se tiene que

1)	$[\forall x \; : \; P(x, a_1)]$	**EP**, introduïm variable particular ... **EP**, *introducimos variable particular ...*
2)	$P(a, a_1)$	... **EG**, introduïm variable genèrica **EG**, *introducimos variable genérica*
3)	$\exists y : P(a, y)$	**IP** en 2. Acaba **EP** de 1.
4)	$\forall x, \exists y \; : \; P(x, y)$	**IG** en 3, s'obté la conclusió *se obtiene la conclusión*

Exemple: per a tot existeix?

Ejemplo: ¿para todo existe?

És cert que

¿Es cierto que

$$\forall x, \exists y \; : \; P(x, y) \vdash \exists y : \forall x, \; P(x, y)?$$

Si intentem aplicar regles estàndard ara,

Si intentamos aplicar reglas estándar ahora,

1)	$\exists y \; : \; P(a, y)$	**EG**, introduïm variable genèrica **EG**, *introducimos variable genérica*
2)	$[P(a, a_1)]$	**EP**, introduïm variable particular ... **EG**, *introducimos variable particular ...*
3)		

El següent pas que ens interessaria aplicar és la introducció del generalitzador sobre l'expressió de la línia 2, que és una suposició.

El siguiente paso que nos interesaría aplicar es la introducción del generalizador sobre la expresión de la línea 2, que es una suposición.

Però, el fet important és que aquesta suposició introdueix una **restricció** entre les variables a i a_1. En particular, per a que l'expressió siga certa, la variable a_1 ha de dependre de la variable a.

Sin embargo, el hecho importante es que esta suposición introduce una **restricción** *entre las variables a y a_1. En particular, para que la expresión sea cierta, la variable a_1 tiene que depender de la variable a.*

En altres paraules, $P(a, a_1)$ no pot ser certa per a **qualsevol** valor de a y **el mateix** valor de a_1.

Dicho de otra manera, $P(a, a_1)$ no puede ser cierta para **cualquier** *valor de a y* **el mismo** *valor de a_1.*

Com que hi ha una restricció sobre la variable genèrica, a, no és possible aplicar la regla **IG** i no es pot completar la regla **EP** anterior.

Como hay una restricción sobre la variables genérica, a, no es posible aplicar la regla **IG** *y no se puede completar la regla* **EP** *anterior.*

Podem aleshores afirmar que

Podemos entonces afirmar que

$$\forall x, \exists y \;:\; P(x,y) \nvdash \exists y : \forall x, \;P(x,y)?$$

Lamentablement no. Per a poder negar la pregunta inicial, el més fàcil és trobar un contraexemple que en el present cas és molt senzill.

Lamentablemente no. Para poder negar la pregunta inicial, lo más fácil es encontrar un contraejemplo que en el caso presente es muy sencillo.

Considerem un univers amb 2 elements, $\{a_1, a_2\}$, de manera que

Consideramos un universo con 2 elementos, $\{a_1, a_2\}$, de manera que

$$P(a_1, a_2) = P(a_2, a_1) = V, \quad P(a_1, a_1) = P(a_2, a_2) = F.$$

La deducció inicial en aquest cas particular esdevé

La deducción inicial en este caso particular se convierte en

$$(P(a_1,a_1)\vee P(a_1,a_2))\wedge(P(a_1,a_1)\vee P(a_1,a_2)) \vdash (P(a_1,a_1)\wedge P(a_1,a_2))\vee(P(a_1,a_1)\wedge P(a_1,a_2))$$

d'on directament s'obté

de donde directamente se obtiene

$$\mathsf{V} \vdash \mathsf{F}$$

la qual cosa és impossible. | *que es imposible.*

2.5 Problemes resolts i comentats
Problemas resueltos y comentados

Problema 2.1: [equivalencies]

Considera les expressions següents, calcula les corresponents taules de veritat i digues quines són equivalents.

Considera las expresiones siguientes, calcula las correspondientes tablas de verdad y di cuáles de ellas son equivalentes.

$(p \Rightarrow q) \Rightarrow r$	$p \Rightarrow (q \Rightarrow r)$	$(p \Rightarrow q) \wedge (q \Rightarrow r)$	$(p \Rightarrow r) \vee (q \Rightarrow r)$

Deixarem les taules de veritat per al final i analitzarem primer les diferents expressions.

Dejaremos las tablas de verdad para el final y analizaremos primero las diferentes expresiones.

Observem primer que en l'última expressió podem aplicar la **pseudodistributivitat per la dreta** de la implicació (podem traure factor comú per la dreta però canviant disjunció per conjunció),

Observamos primero que en la última expresión podemos aplicar la **pseudodistributividad por la derecha** *de la implicación (podemos sacar factor común por la derecha pero cambiando disyunción por conjunción),*

$$(p \Rightarrow r) \vee (q \Rightarrow r) \overset{\text{(pdistr.)}}{\equiv} (p \wedge q) \Rightarrow r,$$

amb la qual cosa arribem a una implicació que tindria el mateix conseqüent que la primera expressió. No obstant això, els antecedents són molt diferents i no sembla que puguen ser equivalents la primera i la quarta expressions.

con lo que llegamos a una implicación que tendría el mismo consecuente que la primera expresión. Sin embargo, los antecedentes son muy diferentes y no parece que puedan ser equivalentes la primera y la cuarta expresiones.

Una altra possibilitat és aplicar la **distributivitat per l'esquerra** a la segona expressió després d'haver aplicat la definició de la implicació al conseqüent (en aquest cas ja tenim l'expressió factoritzada i el que fem és introduir-hi el factor),

Otra posibilidad es aplicar la **distributividad por la izquierda** *a la segunda expresión después de haber aplicado la definición de la implicación al consecuente (en este caso ya tenemos la expresión factorizada y lo que hacemos es introducir el factor),*

$$p \Rightarrow (q \Rightarrow r) \overset{\text{(def.)}}{\equiv} p \Rightarrow (\neg q \vee r) \overset{\text{(distr.)}}{\equiv} (p \Rightarrow \neg q) \vee (p \Rightarrow r).$$

D'aquesta manera arribem a una disjunció d'implicacions com en el cas de la quarta expressió.

De esta manera llegamos a una disyunción de implicaciones como en el caso de la cuarta expresión.

Encara que les implicacions són diferents, és fàcil veure que si apliquem la definició de la implicació i la llei d'idempotència tant en el cas de la segona com en el de la quarta expressió s'arriba a l'expressió equivalent (que és per cert, una **clàusula de Horn**),

Aunque las implicaciones son diferentes, es fácil ver que si aplicamos la definición de la implicación y la ley de idempotencia tanto en el caso de la segundo como en el de la cuarta expresión se llega a la expresión equivalente (que es por cierto una **cláusula de Horn**),

$$\neg p \vee \neg q \vee r,$$

la qual cosa significa que ambdues expressions són equivalents.

lo cual significa que ambas expresiones son equivalentes.

Com a curiositat, i com a conseqüència de la "simetria" respecte de les variables p i q, també s'ha de complir que

Como curiosidad, y como consecuencia de la "simetría" respecto de las variables p y q, también se tiene que cumplir que

$$p \Rightarrow (q \Rightarrow r) \equiv q \Rightarrow (p \Rightarrow r).$$

Les taules de veritat de les quatre expressions que ens demanen (marcades en la primera fila) són

Las tablas de verdad de las cuatro expresiones que se nos piden (marcadas en la primera fila) son

p	q	r	$p \Rightarrow q$	$(p \Rightarrow q) \Rightarrow r$	$q \Rightarrow r$	$p \Rightarrow (q \Rightarrow r)$	$(p \Rightarrow q) \wedge (q \Rightarrow r)$	$q \Rightarrow r$	$(p \Rightarrow r) \vee (q \Rightarrow r)$
V	V	V	V	V	V	V	V	V	V
V	V	F	V	**F**	F	**F**	**F**	F	**F**
V	F	V	F	V	V	V	**F**	V	V
V	F	F	F	V	V	V	**F**	F	V
F	V	V	V	V	V	V	V	V	V
F	V	F	V	**F**	F	V	**F**	V	V
F	F	V	F	V	V	V	V	V	V
F	F	F	F	**F**	V	V	V	V	V

I a partir d'aquestes es veu com efectivament, les úniques expressions que són equivalents entre elles són la segona i la quarta. (S'han marcat en les quatre columnes els valors F per a facilitar la comparació.)

Y a partir de éstas se ve como efectivamente, las únicas expresiones que son equivalentes entre ellas son la segunda y la cuarta. (Se han marcado en las cuatro columnas los valores F para facilitar la comparación).

Problema 2.2: [implicaForallDistr]

A partir de l'expressió e_1, es pot deduir l'expressió e_2? I e_1 a partir de e_2? Justifica la resposta.

A partir de la expresión e_1, ¿se puede deducir la expresión e_2? ¿Y e_1 a partir de e_2? Justifica la respuesta.

$$e_1 : \forall x,\ (P(x) \Rightarrow Q(x))$$
$$e_2 : \forall x,\ P(x) \Rightarrow \forall x,\ Q(x)$$

Primer farem una deducció informal mentre intentem entendre què signifiquen les expressions.

Primero haremos una deducción informal mientras intentamos entender qué significan las expresiones.

La primera fórmula ens diu que existeix una implicació entre els predicats P i Q **per a cada element** del domini. O expressat d'una altra manera,

La primera fórmula nos dice que existe una implicación entre los predicados P y Q **para cada elemento** *del dominio. O expresado de otra manera,*

$$P(x_1) \Rightarrow Q(x_1),\ P(x_2) \Rightarrow Q(x_2),\ \ldots$$

En canvi, la segona és una **única** implicació l'antecedent de la qual afirma P per a tots els elements del domini. Aquest únic antecedent el podem expressar de manera equivalent com un conjunt de premisses,

En cambio, la segunda es una **única** *implicación cuyo antecedente afirma P para todos los elementos del dominio. Este único antecedente lo podemos expresar de manera equivalente como un conjunto de premisas,*

$$P(x_1),\ P(x_2),\ \ldots$$

Per això, si són certes totes les implicacions de què consta e_1, i aplicant el **modus ponens** tantes vegades com elements continga el domini, a partir de les premisses es dedueix el conseqüent que afirma Q per a tot el domini.

*Por eso, si son ciertas **todas** las implicaciones de que consta e_1, y aplicando el **modus ponens** tantas veces como elementos contenga el dominio, a partir de las premisas se deduce el consecuente que afirma Q para todo el dominio.*

Més formalment, com que es tracta de deduir e_2 que és una implicació, podem usar la regla **II** que consisteix a suposar l'antecedent, $\forall x,\ P(x)$, i intentar arribar al conseqüent, $\forall x,\ Q(x)$. Esquemàticament escrivim

*Más formalmente, como se trata de deducir e_2 que es una implicación, podemos usar la regla **II** que consiste en suponer el antecedente, $\forall x,\ P(x)$, e intentar llegar al consecuente, $\forall x,\ Q(x)$. Esquemáticamente escribimos*

1)	e_1	introduïm la premissa *introducimos la premisa*
2)	$[\forall x,\ P(x)]$	suposem cert l'antecedent (**II** sobre la conclusió) ... *suponemos cierto el antecedente (**II** sobre la conclusión)*
3)	$P(a) \Rightarrow Q(a)$	eliminem el generalitzador (**EG**) en 1 (elegim $x = a$) *eliminamos el generalizador (**EG**) en 1 (elegimos $x = a$)*
4)	$P(a)$	eliminem el generalitzador (**EG**) en 2 (usem $x = a$) *eliminamos el generalizador (**EG**) en 2 (usamos $x = a$)*
5)	$Q(a)$	modus ponens (regla **EI**) en 3,4
6)	$\forall x,\ Q(x)$	introduïm el generalitzador (regla **IG**) en 5 *introducimos el genealizador (regla **IG**) en 5*
7)	$\forall x,\ P(x) \Rightarrow \forall x,\ Q(x)$	... acaba la regla **II** aplicada en 2

Per tant, es compleix que

Por tanto se cumple que

$$\boxed{e_1 \vdash e_2.}$$

Si ens plantegem la implicació e_2 com a premissa, resulta que aquesta significa que ha de ser cert P per a tot el domini perquè siga cert Q (també per a tot el domini). I això no implica necessàriament cap regla separada per als elements del domini. Açò ens fa sospitar que tal vegada $e_2 \nvdash e_1$. Però, com podem demostrar-ho?

*Si nos planteamos la implicación e_2 como premisa, resulta que esta significa que tiene que ser cierto P **para todo** el dominio para que sea cierto Q (también para todo el dominio). Y esto no implica necesariamente ninguna regla serparada para los elementos del dominio. Esto nos hace sospechar que tal vez $e_2 \nvdash e_1$. ¿Pero cómo demostrarlo?*

La forma més senzilla de demostrar que una expressió **no** es pot deduir a partir d'una altra consisteix a trobar un **contraexemple**. És a dir, un cas concret per al domini de manera que existisca una assignació de veritat que faça certa e_2 i falsa e_1 al mateix temps.

La forma más sencilla de demostrar que una expresión **no** *se puede deducri a partir de otra consiste en encontrar un* **contraejemplo**. *Es decir, un caso concreto para el dominio de manera que exista una asignación de verdad que haga cierta e_2 y falsa e_1 al mismo tiempo.*

Aquest domini ha de tenir com a mínim 2 elements, ja que per a dominis d'un o de cap element, les dues expressions coincideixen. En canvi, per a dominis de 2 elements tenim que $e_2 \vdash e_1$ esdevé

Este dominio ha de tener como mínimo 2 elementos, ya que par dominios de un o ningún elemento, las dos expresiones coinciden. En cambio, para dominios de 2 elementos tenemos que $e_2 \vdash e_1$ se convierte en

$$(P(x_1) \wedge P(x_2)) \Rightarrow (Q(x_1) \wedge Q(x_2)) \ \vdash (P(x_1) \Rightarrow Q(x_1)) \wedge (P(x_2) \Rightarrow Q(x_2)),$$

que en el cas concret en què

que en el caso concreto en que

$$P(x_1) = Q(x_2) = F, \quad P(x_2) = Q(x_1) = V,$$

ens dona

nos da

$$\underbrace{F \wedge V}_{F} \Rightarrow \underbrace{V \wedge F}_{F} \ \vdash \underbrace{(F \Rightarrow V)}_{V} \wedge \underbrace{(V \Rightarrow F)}_{F},$$

d'on s'obté que $V \vdash F$ la qual cosa és òbviament falsa.

de donde se obtiene que $V \vdash F$ lo que es obviamente falso.

Com que hem trobat un cas concret on no es compleix, aleshores **no** és cert que $e_2 \vdash e_1$ i per tant

Como hemos encontrado un caso concreto donde no se cumple, entonces **no** *es cierto que $e_2 \vdash e_1$ y por tanto*

$$\boxed{e_2 \nvdash e_1.}$$

Problema 2.3: [iiAlternativa]

Demostra que les dues regles següents són equivalents:

Demuestra que las dos reglas siguientes son equivalentes:

$\boxed{A \vdash B \Rightarrow C}$ $\boxed{A, B \vdash C}$

Per a demostrar que les dues regles d'inferència són equivalents, podem deduir cada una d'elles assumint l'altra com a vàlida. Suposem primer que la primera és una regla d'inferència vàlida i l'anomenem **regla AA**. Aleshores,

Para demostrar que las dos reglas de inferencia son equivalentes, podemos deducir cada una de ellas asumiendo la otra como válida. Suponemos primero que la primera es una regla de inferencia válida y la llamamos **regla AA***. Entonces,*

0)		$A \vdash B \Rightarrow C$ (regla AA)
1)	A	premissa 1
2)	B	premissa 2
3)	$B \Rightarrow C$	regla AA en 1
4)	C	eliminació de la implicació (**EI**) en 2,3 *eliminación de la implicación (**EI**) en 2,3*

És a dir, és possible deduir la conclusió C a partir de les premisses A i B si la regla AA és certa.

Es decir, es posible deducir la conclusión C a partir de las premisas A y B si la regla AA es cierta.

Si suposem ara que la segona és vàlida i l'anomenem **regla BB** tenim també que

Si suponemos ahora que la segunda es válida y la llamamos **regla BB** *tenemos que*

0)		$A, B \vdash C$ (regla BB)
1)	A	premissa
2)	$[B]$	introducció de la implicació (**II**). Suposem l'antecedent ... *introducción de la implicación (**II**). Suponemos el antecedente ...*
3)	C	regla BB en 1,2
4)	$B \Rightarrow C$	... acaba la regla **II** de 2

O siga, que hem pogut deduir $B \Rightarrow C$ a partir de A tot suposant certa la regla BB.

O sea, que hemos podido deducir $B \Rightarrow C$ a partir de A suponiendo cierta la regla BB.

En lloc d'utilitzar les regles d'inferència, podem raonar directament a partir de la definició.

En lugar de utilizar las reglas de inferencia, podemos razonar directamente a partir de la definición.

Si suposem que la regla AA és certa, això vol dir que si $A \equiv V$, aleshores $B \Rightarrow C$ també ha de ser vertader segons la definició. Si volem demostrar que la regla BB és certa, hem de mostrar que quan $A \equiv B \equiv V$ s'obté que $C \equiv V$.

Si suponemos que la regla AA es cierta, esto quiere decir que si $A \equiv V$, entonces $B \Rightarrow C$ también tiene que ser verdadero según la definición. Si queremos demostrar que la regla BB es cierta, tenemos que mostrar que cuando $A \equiv B \equiv V$ se obtiene que $C \equiv V$.

En particular, si $B \equiv V$ i $(B \Rightarrow C) \equiv V$ (perquè suposem certa la regla AA), podem deduir la veritat de C (usant **modus ponens** o a partir de la taula de veritat de les dues expressions), per la qual cosa queda demostrat que es pot deduir C a partir de A i B (regla BB).

En particular, si $B \equiv V$ y $(B \Rightarrow C) \equiv V$ (porque suponemos cierta la regla AA), podemos deducir la verdad de C (usando **modus ponens** *o a partir de la tabla de verdad de las dos expresiones), por lo que queda demostrado que se puede deducir C a partir de A y B (regla BB).*

En sentit contrari, suposem certa la regla BB (a partir de $A \equiv B \equiv V$ es dedueix C), partim del fet que $A \equiv V$, i ens preguntem si és cert o no que $B \Rightarrow C$.

En sentido contrario, suponemos cierta la regla BB (a partir de $A \equiv B \equiv V$ se deduce C), partimos del hecho que $A \equiv V$, y nos preguntamos si es cierto o no que $B \Rightarrow C$.

Sabem que $A \equiv V$ però pot ser que B siga vertader o fals. En el primer cas, podem deduir la veritat de C (per la regla BB), per la qual cosa la implicació és certa. En el segon cas ($B \equiv F$), resulta que també la implicació és certa independentment del valor de veritat de B. Per tant, com que podem deduir la implicació en qualsevol cas, la regla AA és certa.

Sabemos que $A \equiv V$ pero puede ser que B sea verdadero o falso. En el primer caso, podemos deducir la verdad de C (por la regla BB), por lo que la implicación es cierta. En el segundo caso ($B \equiv F$), resulta que también la implicación es cierta independientemente del valor de verdad de B. Por tanto, como podemos deducir la implicación en cualquier caso, la regla AA es cierta.

Problema 2.4: [lesDuesAfirmacions]

Intenta establir la veritat o falsedat de cada una de les dues afirmacions següents:

a) El que vaig a dir a continuació és **fals**.

b) Tot el que dic és **fals**.

Intenta establecer la verdad o falsedad de cada una de las dos afirmaciones siguientes:

a) Lo que voy a decir a continuación es **falso**.

b) Todo lo que digo es **falso**.

Considerarem les variables proposicionals a i b per representar cada una de les dues afirmacions (proposicions) anteriors.

Consideraremos las variables proposicionales a y b para representar cada una de las dos afirmaciones (proposiciones) anteriores.

La primera, a, diu que la següent, b, és falsa. Mentre que la segona, b, diu que tot (tant a com b) és fals. És a dir,

La primera, a, dice que la siguiente, b, es falsa. Mientras que la segunda, b, dice que todo (tanto a como b) es falso. Es decir,

$$a \overset{\text{def}}{\equiv} \neg b$$
$$b \overset{\text{def}}{\equiv} \neg a \wedge \neg b$$

En substituir la definició de a en la de b s'obté

Al sustituir la definición de a en la de b se obtiene

$$b \overset{\text{def}}{\equiv} \neg(\neg b) \wedge \neg b \equiv b \wedge \neg b \equiv F$$

per la qual cosa arribem a que b és **fals**. I en substituir açò en la primera definició s'obté de la mateixa manera que a és **certa**. Per tant,

por lo que llegamos a que b es **falsa**. *Y al sustituir esto en la primera definición se obtiene de la misma manera que a es* **cierta**. *Por lo tanto,*

$$\boxed{a \equiv V} \qquad \boxed{b \equiv F}$$

Alternativament, podem escriure les definicions de les dues afirmacions com un conjunt de premisses:

Alternativamente, podemos escribir las definiciones de las dos afirmaciones como un conjunto de premisas:

$$a \Leftrightarrow \neg b$$
$$b \Leftrightarrow \neg(a \vee b)$$

on hem aplicat la llei de De Morgan. Si despleguem ara les coimplicacions i les manipulem,

donde hemos aplicado la ley de De Morgan. Si desplegamos ahora las coimplicaciones y las manipulamos,

$$\begin{aligned}
P_1 &\equiv a \Rightarrow \neg b & &\equiv b \Rightarrow \neg a \\
P_2 &\equiv \neg b \Rightarrow a & &\equiv \neg a \Rightarrow b \\
P_3' &\equiv b \Rightarrow \neg a \wedge \neg b & &\equiv \underbrace{(b \Rightarrow \neg a)}_{P_1} \wedge (b \Rightarrow \neg b) \equiv P_1 \wedge \underbrace{\neg b}_{\equiv P_3} \\
P_4' &\equiv \neg(a \vee b) \Rightarrow a & &\equiv a \vee b \equiv P_2
\end{aligned}$$

vegem que podem resumir les anteriors afirmacions només amb 3 premisses simples una de les quals és ja la falsedat de b.

vemos que podemos resumir las anteriores afirmaciones sólo con 3 premisas simples, una de las cuales es ya la falsedad de b.

$$P_1 \equiv a \Rightarrow \neg b \equiv b \Rightarrow \neg a$$
$$P_2 \equiv \neg b \Rightarrow a \equiv \neg a \Rightarrow b$$
$$P_3 \equiv \neg b$$

De la premissa P_3 i de la P_2 es dedueix la veracitat de a en aplicar la regla **EI**. Si escrivim la deducció completa pas a pas tenim

De la premisa P_3 y de la P_2 se deduce la veracidad de a al aplicar la regla **EI**. *Si escribimos la deducción completa paso a paso tenemos*

P_2)	$\neg b \Rightarrow a$	premissa 2
P_3)	$\neg b$	premissa 3
4)	a	regla **EI** en P_2, P_3
5)	$a \wedge \neg b$	regla **IC** en P_3, 4

En aquesta deducció es veu que no s'ha necessitat la primera premissa per a arribar al resultat.

En esta deducción se ve que no se ha necesitado la primera premisa para llegar al resultado.

Si analitzem un poc més la deducció vegem que es tracta en realitat d'una equivalència entre les premisses P_2 i P_3 i el resultat.

Si analizamos un poco más la deducción vemos que se trata en realidad de una equivalencia entre las premisas P_2 y P_3 y el resultado.

$$P_2 \wedge P_3 \equiv (\neg b \Rightarrow a) \wedge \neg b \equiv (b \vee a) \wedge \neg b \equiv \cancelto{F}{(b \wedge \neg b)} \vee (a \wedge \neg b) \equiv a \wedge \neg b$$

Problema 2.5: [lesDuesAfirmacionsBis]

Intenta establir la veritat o falsedat de cada una de les dues afirmacions següents:

Intenta establecer la verdad o falsedad de cada una de las dos afirmaciones siguientes:

a) El que vaig a dir a continuació és **fals**.

a) Lo que voy a decir a continuación es **falso**.

b) El que he dit abans és **fals**.

b) Lo que he dicho antes es **falso**.

Considerarem les variables proposicionals a i b per representar cada una de les dues afirmacions (proposicions) anteriors.

Consideraremos las variables proposicionales a y b para representar cada una de las dos afirmaciones (proposiciones) anteriores.

La primera, a, diu que la següent, b, és falsa. Mentre que la segona, b, diu que la primera, a, és falsa. És a dir,

La primera, a, dice que la siguiente, b, es falsa. Mientras que la segunda, b, dice que la primera, a, es falsa. Es decir,

$$a \stackrel{\text{def}}{\equiv} \neg b$$
$$b \stackrel{\text{def}}{\equiv} \neg a$$

Contràriament al que passava en l'exercici 2.4, la substitució de la definició de a en la de de b, o al revés, dóna lloc a una expressió de la forma,

Contrariamente a lo que pasaba en el ejercicio 2.4, la sustitución de la definición de a en la de b, o al revés, da lugar a una expresión de la forma,

$$p \stackrel{\text{def}}{\equiv} \neg p$$

que es correspon amb una paradoxa del tipus "aquesta afirmació és falsa" a partir de la qual no se'n pot establir ni la veritat ni la fasedat, per la qual cosa haurem de considerar un raonament alternatiu.

que se corresponde con una paradoja del tipo "esta afirmación es falsa" a partir de la que no se puede establecer ni su verdad ni su falsedad, por lo que habrá que considerar un razonamiento alternativo.

Suposem que a és certa, la qual cosa vol dir que b ha de ser falsa. b afirma la falsedat de a, però com que és falsa, a és efectivament vertadera.

Suponemos que a es cierta, lo que quiere decir que b tiene que ser falsa. b afirma la falsedad de a, pero como es falsa, a es efectivamente verdadera.

Si suposem en canvi que a és falsa, aleshores b hauria de ser vertadera. I b efectivament afirma la falsedat de a.

Si suponemos en cambio que a es falsa, entonces b tendría que ser verdadera. Y b efectivamente afirma la falsedad de a.

En resum, a partir de les dues afirmacions només podem arribar a què a és vertadera i b falsa, o a que a és falsa i b vertadera. I això ho podem expressar simplement com $a \not\equiv b$ la qual cosa és equivalent a

En resumen, a partir de las dos afirmaciones sólo podemos llegar a que a es verdadera y b falsa, o a que a es falsa y b verdadera. Y esto lo podemos expresar simplemente como $a \not\equiv b$ lo que es equivalente a

$$(a \equiv V \wedge b \equiv F) \vee (a \equiv F \wedge b \equiv V)$$

O be fent servir expressions purament proposicionals com a

O bien usando expresiones puramente proposicionales como

$$(a \wedge \neg b) \vee (\neg a \wedge b) \equiv \neg(a \Leftrightarrow b) \equiv a \not\Leftrightarrow b$$

si acceptem el símbol $\not\Leftrightarrow$ per representar la **disjunció exclusiva**.

si aceptamos es símbolo $\not\Leftrightarrow$ para representar la **disyunción exclusiva**.

Si considerem les dues afirmacions com a premisses les podem escriure com a

Si consideramos las dos afirmaciones como premisas las podemos escribir como

$$a \Leftrightarrow \neg b$$
$$b \Leftrightarrow \neg a$$

I una vegada desplegades les coimplicacions donen lloc només a

Y una vez desplegadas las coimplicaciones dan lugar sólo a

$$P_1 \equiv \neg a \Rightarrow b \equiv a \vee b$$
$$P_2 \equiv b \Rightarrow \neg a \equiv \neg b \vee \neg a \equiv \neg(a \wedge b)$$

La conjunció de les premisses és exactament la disjunció exclusiva obtinguda abans. I les úniques coses que es poden deduir a partir d'aquestes premisses serien les que s'il·lustren en la següent taula de veritat.

La conjunción de las premisas es exactamente igual a la disyunción exclusiva obtenida antes. Y lo único que se puede deducir a partir de estas premisas sería lo que se ilustra en la siguiente tabla de verdad.

a	b	$P_1 \wedge P_2 \equiv a \not\Leftrightarrow b$	$a \vee b$	$\neg(a \wedge b)$	V
V	V	F	V	F	V
V	F	**V**	**V**	**V**	**V**
F	V	**V**	**V**	**V**	**V**
F	F	F	F	V	V

Com a conclusió i resultat final, no és possible de cap manera establir la veritat o la falsedat de cap de les dues afirmacions.

Como conclusión y resultado final, no es posible de ninguna manera establecer la verdad o la falsedad de ninguna de las dos afirmaciones.

Problema 2.6: [tollendoPonens]

Demostra que és cert que

Demuestra que es cierto que

$$P \vee Q, \neg P \vdash Q$$

Aquesta regla que s'utilitza en lògica clàssica, s'anomena **modus tollendo ponens** o també **sil·logisme disjuntiu** (pàgina 78). Un exemple d'aquesta regla el trobem en la coneguda frase atribuïda al personatge més famós de Conan Doyle: *"Una vegada que es descarta l'impossible, el que queda és la veritat per improbable que semble"*.

Esta regla que se utiliza en lógica clásica, se denomina **modus tollendo ponens** *o también* **silogismo disyuntivo** *(página 78). Un ejemplo de esta regla lo encontramos en la conocida frase atribuida al personaje más famoso de Conan Doyle:* "Una vez descartado lo imposible, lo que queda es la verdad por improbable que parezca".

Per a demostrar-la podem usar les regles d'inferència estàndard.

Para demostrarla podemos usar las reglas de inferencia estándar.

1)	$P \vee Q$	premissa 1
2)	$\neg P$	premissa 2
3)	$[\neg Q]$	suposem el contrari del que volem (**IN**) ... *suponemos lo contrario de lo que queremos (**IN**) ...*
4)	$[P]$	... Comencem demostració per casos (**ED**) en 1. Primer cas ... *Comenzamos demostración por casos (**ED**) en 1. Primer caso ...*
5)	$P \wedge \neg P \equiv F$	... ... introducció de la conjunció (**IC**) en 2, 4 *introducción de la conjunción (**IC**) en 2, 4*
6)	$[Q]$	... ... Segon cas ... *Segundo caso ...*
7)	$Q \wedge \neg Q \equiv F$	... ... introducció de la conjunció (**IC**) en 3, 6 *introducción de la conjunción (**IC**) en 3, 6*
8)	F	... acaba la regla **ED** en 1. Contradicció en qualsevol cas *acaba la regla **ED** en 1. Contradicción en cualquier caso*
9)	Q	acaba la regla **IN**

L'anterior és una demostració per reducció a l'absurd: comencem negant el que volem demostrar i comprovem que s'arriba necessàriament a una contradicció.

Lo anterior es una demostración por reducción al absurdo: comenzamos negando lo que queremos demostrar y comprobamos que se llega necesariamente a una contradicción.

Una alternativa a l'anterior és convertir la disjunció de la primera premissa en una implicació per a poder aplicar el **modus ponens** o regla **EI**.

Una alternativa a lo anterior es convertir la disyunción de la primera premisa en una implicación para poder aplicar el **modus ponens** *o regla* **EI**.

1)	$P \vee Q$	premissa 1
2)	$\neg P$	premissa 2
3)	$\neg P \Rightarrow Q$	**definició de la implicació** en 1 **definición de la implicación** *en 1*
4)	Q	eliminem la implicació (**EI**) en 2,3. *eliminamos la implicación (**EI**) en 2,3.*

Alternativament, podem veure que la conjunció de les dues premisses és

Alternativamente, podemos ver que la conjunción de las dos premisas es

$$(P \vee Q) \wedge \neg P \equiv (P \wedge \neg P) \vee (Q \wedge \neg P) \equiv F \vee (Q \wedge \neg P) \equiv Q \wedge \neg P$$

I d'aquesta conjunció es dedueix Q mitjançant la regla d'eliminació de la conjunció (**EC**). El mateix en forma de taula seria

*Y de esta conjunción se deduce Q mediante la regla de eliminación de la conjunción (**EC**). Lo mismo en forma de tabla sería*

1)	$P \vee Q$	premissa 1
2)	$\neg P$	premissa 2
3)	$(P \vee Q) \wedge \neg P$	introducció de la conjunció (**IC**) en 1,2 *introducción de la conjunción (**IC**) en 1,2*
4)	$Q \wedge \neg P$	equivalent a 3 *equivalente a 3*
5)	Q	eliminació de la conjunció (**EC**) en 4 *eliminación de la conjunción (**EC**) en 4*

Problema 2.7: [tollendoPonensDisj]

Explica què significa la següent deducció i demostra-la a partir de les taules de veritat.

Explica qué significa la siguiente deducción y demuéstrala a partir de las tablas de verdad.

$$P \vee Q, \neg P \vdash Q \vee R$$

Tenim dues premisses. Una disjunció de dues proposicions i la negació d'una d'elles. La conclusió és una altra disjunció de la proposició no negada en les premisses i una altra proposició que no apareix en cap premissa.

Tenemos dos premisas. Una disyunción de dos proposiciones y la negación de una de ellas. La conclusión es otra disyunción de la proposición no negada en las premisas y otra proposición que no aparece en ninguna premisa.

Formalment, és com el **modus tollendo ponens** o **sil·logisme disjuntiu** (pàgina 78), però amb la introducció d'una disjunció en la conclusió. Per això mateix, és relativament fàcil demostrar-la aplicant *modus tollendo ponens* primer (com en l'exercici 2.6) i la regla d'**introducció de la disjunció**, **ID**, després.

Formalmente, es como el **modus tollendo ponens** *o* **silogismo disyuntivo** *(página 78), pero con la introducción de una disyunción en la conclusión. Por eso mismo, es relativamente fácil demostrarla aplicando* modus tollendo ponens *primero (como en el ejercicio 2.6) y la regla de* **introducción de la disyunción**, **ID**, *después.*

Deixem aquesta opció de demostració com a exercici.

Dejamos esta opción de demostración como ejercicio.

Per a demostrar-ho mitjançant taules de veritat necessitarem primer calcular-les per a la conjunció de les premisses d'una banda, i per a la conclusió de l'altra.

Para demostrarlo mediante tablas de verdad necesitaremos primero calcularlas para la conjunción de las premisas por un lado, y para la conclusión por otro.

De les files de la taula de veritat, marcarem aquelles que es corresponen amb la veritat de les premisses que són les crítiques a l'hora de decidir si es pot deduir o no la conclusió.

De las filas de la tabla de verdad, marcaremos aquellas que se corresponden con la verdad de las premisas que son las críticas a la hora de decidir si se puede deducir o no la conclusión.

					premisses *premisas*	conclusió *conclusión*
P	Q	R	$P \vee Q$	$\neg P$	$(P \vee Q) \wedge \neg P$	$Q \vee R$
V	V	V	V	F	F	V
V	V	F	V	F	F	V
V	F	V	V	F	F	V
V	F	F	V	F	F	F
F	V	V	V	V	**V**	**V**
F	V	F	V	V	**V**	**V**
F	F	V	F	V	F	V
F	F	F	F	V	F	F

Podem observar que en els dos casos en què les premisses són certes, la conclusió també ho és. I per tant, la deducció queda demostrada. Observem, finalment, que la resta de la taula de veritat de la conclusió (en gris) no calia calcular-la.

Podemos observar que en los dos casos en que las premisas son ciertas, la conclusión también lo es. Y por tanto, la deducción queda demostrada. Observamos, finalmente, que el resto de la tabla de verdad de la conclusión (en gris) no era necesario calcularla.

Problema 2.8: [quantificadorsInvertits]

Analitza la deducció següent i demostra-la utilitzant les regles d'inferència estàndard.

Analiza la siguiente deducción y demuéstrala utilizando las reglas de inferència estándar.

$$\forall X, \exists Y : (P(X,Y) \wedge Q(X)) \vdash \forall Y, \exists X : (P(X,Y) \vee Q(X))$$

Es tracta de dues fórmules, una premissa i una conclusió, amb quantificadors niats amb l'ordre invertit en cada cas.

Se trata de dos fórmulas, una premisa y una conclusión, con cuantificadores anidados con el orden invertido en cada caso.

Tenim un predicat binari i un altre unari, que apareixen connectats mitjançant conjunció i disjunció en la premissa i la conclusió, respectivament.

Tenemos un predicado binario y otro unario, que aparecen conectados mediante conjunción y disyunción en la premisa y la conclusión, respectivamente.

Però la part més important la formen els quantificadors invertits. La premissa diu que per a tot element del domini, X, n'hi ha d'haver un altre amb el qual es relacione, $P(X,\cdot)$, **i a més a més** s'ha de complir $Q(X)$.

Pero la parte más importante la forman los cuantificadores invertidos. La premisa dice que para todo elemento del dominio, X, tiene que haber otro con el cual se relacione, $P(X,\cdot)$, **y además** *se tiene que cumplir $Q(X)$.*

Com que la fórmula oberta sense els quantificadors és una conjunció, es pot deduir separadament

Como la fórmula abierta sin los cuantificadores es una conjunción, se puede deducir separadamente

$$\forall X, \exists Y : P(X,Y) \qquad \forall X, Q(X)$$

com a subconclusions a partir de la premissa. (De fet, la conjunció d'aquestes és equivalent a la premissa.)

como subconclusiones a partir de la premissa. (De hecho, la conjunción de éstas es equivalente a la premisa.)

En canvi, la conclusió diu que per a tot element, Y, ha d'haver-n'hi un altre que, o bé es relacione, $P(\cdot, Y)$, o bé complisca Q.

En cambio, la conclusión dice que para todo elemento, Y, tiene que haber otro que, o bien se relacione, $P(\cdot, Y)$, o bien cumpla Q.

Afortunadament, com que es tracta d'una disjunció, podrem deduir la conclusió si arribem a una de les dues fórmules següents

Afortunadamente, como se trata de una disyunción, podremos deducir la conclusión si llegamos a una de las dos fórmulas siguientes

$$\forall Y, \exists X : P(X, Y) \qquad \exists X : Q(X)$$

A la primera no hi ha manera d'arribar llevat que P siga commutatiu. Però la segona sí que és fàcilment deduïble a partir de la segona subconclusió abans esmentada.

A la primera no hay manera de llegar excepto si P es conmutativo. Pero la segunda sí que es fácilmente deducible a partir de la segunda subconclusión antes comentada.

Per tant, l'estratègia de deducció de la conclusió a partir de la premissa, haurà de passar per la segona subconclusió per a arribar a l'última de les fórmules, amb la qual cosa deduiríem la conclusió.

Por tanto, la estrategia de deducción de la conclusión a partir de la premisa, tendrá que pasar por la segunda subconclusión para llegar a la última de las fórmulas, con lo que deduciríamos la conclusión.

La deducció formal usant regles d'inferència reprodueix aquesta estratègia d'una manera una mica més directa.

La deducción formal usando reglas de inferencia reproduce esta estrategia de una manera un poco más directa.

1)	$\forall X, \exists Y : (P(X, Y) \wedge Q(X))$	premissa
2)	$\exists Y : (P(a, Y) \wedge Q(a))$	**EG** en 1 (elegim) *(elegimos)* $X = a$
3)	$[P(a, b_1) \wedge Q(a)]$	**EP** en 2 (introduïm) *(introducimos)* b_1 ...
4)	$Q(a)$	... **EC** en 3 (b_1 ja ha desaparegut) *(b_1 ya ha desaparecido)* ... acaba **EP**
5)	$P(a, c) \vee Q(a)$	**ID** en 4 (introduïm c, vegeu text) *(introducimos c, ver texto)*
6)	$\exists X : P(X, c) \vee Q(X)$	**IP** en 5
7)	$\forall Y, \exists X : P(X, Y) \vee Q(X)$	**IG** en 6

En la línia 5, hem introduït una variable, c, que pot ser qualsevol element del domini, ja que no és en absolut important que la fórmula $P(a, c)$ siga certa perquè ho siga la disjunció. Aquesta matisació és important perquè després (línia 7) voldrem introduir un generalitzador associat a aquesta variable.

En la línea 5, hemos introducido una variable, c, que puede ser cualquier elemento del dominio, ya que no es en absoluto importante que la fórmula $P(a, c)$ sea cierta para que lo sea la disyunción. Esta matización es importante porque después (línea 7) querremos introducir un generalizador asociado a esta variable.

Problema 2.9: [quadratParell]

Demostra l'afirmació: **Si un enter parell és el quadrat d'un altre, aleshores aquest altre és també parell.** Per a això, pots seguir el següent esquema informal: imagina que el quadrat és parell però el nombre no, i intenta arribar a una contradicció.

Demuestra la afirmación, **Si un entero par es el cuadrado de otro, entonces este otro es también par.** *Para esto, puedes seguir el siguiente esquema informal: imagina que el cuadrado es par pero el número no, y intenta llegar a una contradicción.*

Fes primer una demostració informal utilitzant llenguatge natural i després expressa tant l'enunciat com la demostració fent servir la lògica de primer ordre.

Haz primero una demostración informal usando lenguaje natural y después expresa tanto el enunciado como la demostración usando la lógica de primer orden.

L'enunciat ens suggereix una demostració per reducció a l'absurd. I l'estratègia consisteix a suposar que un enter senar al quadrat dóna com a resultat un nombre parell.

El enunciado nos sugiere una demostración por reducción al absurdo. Y la estrategia consiste en suponer que un entero impar al cuadrado da como resultado un número par.

Però si ho pensem un poc, un enter no parell multiplicat per ell mateix, mai no podrà donar com a resultat un nombre parell.

Pero si lo pensamos un poco, un entero no par multiplicado por él mismo, nunca podrá dar como resultado un número par.

Podem per exemple imaginar la descomposició en factors primers del quadrat i la del nombre. Si en el quadrat no pot aparèixer el factor 2, és impossible que en el nombre hi haguera cap factor 2 (Perquè aleshores hauria d'aparèixer també en el quadrat).

Podemos por ejemplo imaginar la descomposición en factores primos del cuadrado y la del número. Si en el cuadrado no puede aparecer el factor 2, es imposible que en el número hubiese ningún factor 2 (Porque entonces tendría que aparecer también en el cuadrado).

Ara expressarem formalment l'enunciat del que volem demostrar i després la seua demostració.

Ahora expresaremos formalmente el enunciado de lo que queremos demostrar y después su demostración.

$$\boxed{\forall n \in \mathbb{Z}^+,\ \left(\exists k \geq 0\ :\ n^2 = 2k\right) \Rightarrow (\exists \ell \geq 0\ :\ n = 2\ell)}$$

Des del punt de vista de la lògica de predicats, podem eliminar el generalitzador i escriure

Desde el punto de vista de la lógica de predicados, podemos eliminar el generalizador y escribir

$$\underbrace{\exists k \geq 0\ :\ n^2 = 2k}_{P_1} \vdash \underbrace{\exists \ell \geq 0\ :\ n = 2\ell}_{Q},$$

de manera que si som capaços de demostrar aquesta deducció per al valor genèric, n, quedarà demostrat el resultat general.

de manera que si somo capaces de demostrar esta deducción para el valor genérico, n, quedará demostrado el resultado general.

1)	$\exists k \geq 0 \ : \ n^2 = 2k$	premissa 1
2)	$[n^2 = 2k_1]$	**EP** en 1 (introduïm k_1) ... *introducimos*
3)	$[\neg(\exists \ell \geq 0 \ : \ n = 2\ell)]$	... **IN** ...
3')	$\forall \ell \geq 0, \ n \neq 2\ell$	... ... De Morgan en 3
4)	$\exists \ell \geq 0 \ : \ (n = 2\ell \vee n = 2\ell + 1)$	... ... introduïm una tautologia *introducimos una tautología*
5)	$[n = 2k_2 \vee n = 2k_2 + 1]$	... ... **EP** en 4 (introduïm k_2) ... *introducimos*
6)	$n \neq 2k_2$	... ... **EG** en 3'
7)	$n = 2k_2 + 1$	... ... modus tollendo ponens en 5, 6
8)	$n^2 = (2k_2+1)^2 = 2\underbrace{(k_2+1)2k_2}_{k_3}+1$	... ... elevem al quadrat *elevamos al cuadrado*
9)	$n^2 = 2k_1 \wedge n^2 = 2k_3 + 1$	... ... **IC** en 2, 8
10)	F	... ... equivalent a 9 si $n, k_1, k_3 \in \mathbb{Z}^+$ *equivalente a 9 si*
11)	$\exists \ell \geq 0 \ : \ n = 2\ell$	... acaba **EP** 4 (k_2), acaba **IN**
12)	$\exists \ell \geq 0 \ : \ n = 2\ell$	acaba **EP**1 (k_1)

Problema 2.10: [inferenciaDosF]

En la següent expressió, identifica premisses i conclusió i fes la demostració pas a pas especificant totes les regles d'inferència usades.

En la siguiente expresión, identifica premisas y conclusión y haz la demostración paso a paso especificando todas las reglas de inferencia usadas.

$$\exists Y : (\forall X, p(X,Y) \Rightarrow \neg q(X) \ ) \ \ \vdash \ \ \exists Y : \ q(Y) \Rightarrow \neg p(Y,Y).$$

Explica què canviaria en la demostració si en la premissa intercanviàrem l'ordre dels quantificadors, És a dir, si la premissa fóra

Explica qué cambiaría en la demostración si en la premisa intercambiáramos el orden de los cuantificadores. Es decir, si la premisa fuera

$$\forall X, \ \exists Y : \ (\ p(X,Y) \Rightarrow \neg q(X) \ ).$$

Es tracta d'una única premissa amb dos quantificadors niats,

Se trata de una única premisa con dos cuantificadores anidados,

$$P_1 \equiv \exists Y : \ (\forall X, p(X,Y) \Rightarrow \neg q(X) \ ),$$

i una conclusió amb un únic particularitzador,

y una conclusión con un único particularizador,

$$\exists Y : q(Y) \Rightarrow \neg p(Y, Y).$$

Quantificadors a banda, la premissa és una implicació i la conclusió també. Per tant, en algun moment haurem d'aplicar la regla **II** i haurem de suposar el predicat q per a algun valor del domini corresponent.

Cuantificadores aparte, la premisa es una implicación y la conclusión también. Por tanto, en algún momento tendremos que aplicar la regla **II** *y tendremos que suponer el predicado* q *para algún valor del dominio correspondiente.*

Aquest predicat q s'haurà de combinar amb la implicació de la premissa per a arribar al conseqüent de la conclusió. Més detalladament i en forma de taula,

Este predicado q *se tendrá que combinar con la implicación de la premisa para llegar al consecuente de la conclusión. Más detalladamente y en forma de tabla,*

1)	$\exists Y : (\forall X,\ p(X, Y) \Rightarrow \neg q(X)\ )$	premissa
2)	$\forall X,\ p(X, a_1) \Rightarrow \neg q(X)$	**EP** en 1 (introduïm *introducimos* a_1)
3)	$p(a_1, a_1) \Rightarrow \neg q(a_1)$	... **EG** en 2 (elegim *elegimos* $X = a_1$)
4)	$[q(a_1)]$	... **II** ...
5)	$\neg p(a_1, a_1)$	... ... *modus tollens* en 3, 4
6)	$q(a_1) \Rightarrow \neg p(a_1, a_1)$	... acaba **II** de 4
7)	$\exists Y : q(Y) \Rightarrow \neg p(Y, Y)$	**IP** en 6, acaba **EP** en 1 (a_1)

La deducció anterior seria un poc més llarga si en lloc d'usar el modus tollens en la línia 5 s'hagueren d'usar regles estàndar.

La deducción anterior sería un poco más larga si en lugar de usar el modus tollens *en la línea 5 hubiese que usar reglas estándar.*

En el cas que en la premissa s'intercanviaren els quantificadors com s'especifica en l'enunciat tindríem que

En el caso que en la premisa se intercambiaran los cuantificadores como se especifica en el enunciado tendríamos que

1)	$\forall X : (\exists Y,\ p(X,Y) \Rightarrow \neg q(X)\ )$	premissa
2)	$\exists Y :\ p(a,Y) \Rightarrow \neg q(a)$	**EG** en 1 (elegim $\underset{elegimos}{}$ $X = a$)
3)	$p(a,a_1) \Rightarrow \neg q(a)$	**EP** en 2 (introduïm $\underset{introducimos}{}$ a_1) ...
4)	$[q(a)]$	... **II** ...
5)	$\neg p(a,a_1)$	... ... modus tollens en 3, 4
6)	$q(a) \Rightarrow \neg p(a,a_1)$	... acaba **II** de 4
7)	$\exists Y,Z : q(Y) \Rightarrow \neg p(Y,Z)$	**IP** en 6 (x2), acaba **EP** en 2

Com que ara estem obligats a eliminar primer el generalitzador (i introduir un valor genèric, a), el valor que s'introdueix en eliminar el particularitzador no es pot elegir de manera que els dos coincidisquen, com abans. De fet, el valor particular a_1 depén del valor de a.

Como ahora estamos obligados a eliminar primero el generalizador (e introducir un valor genérico, a), el valor que se introduce al eliminar el particularizador no se puede elegir de manera que los dos coincidan, como antes. De hecho, el valor particular de a_1 depende del valor de a.

En altres paraules, no es pot arribar a la conclusió amb la mateixa extratègia que abans.

En otras palabras, no se puede llegar a la conclusión con la misma estrategia que antes.

No obstant això, si volem demostrar que, efectivament la conclusió no és deduïble a partir de la premissa, hem de mostrar almenys un contraexemple. És a dir, un cas particular en què la premissa siga certa i la conclusió falsa.

Sin embargo, si queremos demostrar que, efectivamente la conclusión no es deducible a partir de la premisa, tenemos que mostrar al menos un contraejemplo. Es decir, un caso particular en que la premisa sea cierta y la conclusión falsa.

Aquest contraexemple el podem construir sobre un domini amb dos elements, $\{1,2\}$ de manera que els valors de veritat siguen

Este contraejemplo lo podemos construir sobre un dominio con dos elementos, $\{1,2\}$ de manera que los valores de verdad sean

$p(1,2)$	$p(2,1)$	$q(1)$	$q(2)$	$p(1,1)$	$p(2,2)$
F	F	V	V	V	V

Segons l'assignació de veritat elegida tenim que la premissa és certa,

Según la asignación de verdad elegida tenemos que la premisa es cierta,

$$\underbrace{(\overbrace{(\underbrace{p(1,1)}_{V} \Rightarrow \underbrace{\neg q(1)}_{F})}^{F} \vee \overbrace{(\underbrace{p(1,2)}_{F} \Rightarrow \neg q(1))}^{V})}_{V} \wedge \underbrace{(\overbrace{(\underbrace{p(2,1)}_{F} \Rightarrow \neg q(2))}^{V} \vee \overbrace{(\underbrace{p(2,2)}_{V} \Rightarrow \underbrace{\neg q(2)}_{F})}^{F})}_{V} \equiv V,$$

però la conclusió és falsa,

pero la conclusión es falsa,

$$\overbrace{(\underbrace{q(1)}_{V} \Rightarrow \underbrace{\neg p(1,1)}_{F})}^{F} \vee \overbrace{(\underbrace{q(2)}_{V} \Rightarrow \underbrace{\neg p(2,2)}_{F})}^{F} \equiv F.$$

La deducció a partir de la segona premissa és falsa per que, segons l'enunciat, es tracta de dos quantificadors niats aplicats sobre una implicació. Però si haguérem escrit la premissa com

La deducción a partir de la segunda premisa es falsa porque, según el enunciado, se trata de dos cuantificadores anidados aplicados sobre una implicación. Pero si hubiésemos escrito la premisa como

$$\forall X, \; (\exists Y: \; p(X,Y) \;) \Rightarrow \neg q(X)).$$

de manera que el particularitzador només actua sobre el predicat, p, i no sobre la implicació, aleshores la deducció si que seria certa. La demostració d'aquesta nova deducció sobre la segona premissa modificada es deixa com a exercici.

de manera que el particularizador sólo actua sobre el predicado, p, y no sobre la implicación, entonces la deducción sí que sería cierta. La demostración de esta nueva deducción sobre la segunda premisa modificada se deja como ejercicio.

Es pot comentar que aquesta doble interpretació quant a l'abast dels quantificadors no té sentit en el cas de la primera premissa ja que la variable del predicat unari, q, està associada al quantificador extern.

Se puede comentar que esta doble interpretación en cuanto al alcance de los cuantificadores no tiene sentido en el caso de la primera premisa ya que la variable del predicado unario, q, está asociada al cuantificador externo.

Problema 2.11: [tresFormules]

Considera les tres fórmules que es donen a continuació i raona si alguna es pot o no deduir a partir de les altres dues. Si és el cas, explicita la deducció corresponent fent servir les regles d'inferència estàndard.

Considera las tres fórmulas que se dan a continuación y razona si alguna de ellas se puede o no deducir a partir de las otras dos. Si es el caso, explicita la deducción correspondiente utilizando las reglas de inferència estándar.

$$f_1 \equiv \boxed{\exists x : R(x)} \quad f_2 \equiv \boxed{\exists y : \forall x, \ ((P(x) \vee Q(y)) \Rightarrow R(y))} \quad f_3 \equiv \boxed{P(p)}$$

Dues de les fórmules són fets o afirmacions de veracitat dels predicats R (per a alguns valors del domini) i P (per a un element concret, p). Mentre que l'altra fórmula combina implicacions per a determinats parells d'elements del domini.

Dos de las fórmulas son hechos o afirmaciones de veracidad de los predicados R (para algunos valores del dominio) y P (para un elemento concreto, p). Mientras la otra fórmula combina implicaciones para determinados pares de elementos del dominio.

A partir de fets no sembla fàcil deduir implicacions per la qual cosa descartarem en principi f_2 com a conclusió.

A partir de hechos no parece fácil deducir implicaciones por lo que descartaremos en principio f_2 como conclusión.

Això vol dir que f_2 és una premissa i que l'altra ha de ser un fet o conjunt de fets que es puguen combinar amb les implicacions de f_2.

Esto quiere decir que f_2 es una premisa y que la otra tiene que ser un hecho o conjunto de hechos que se puedan combinar con las implicaciones de f_2.

Però com que totes les implicacions de f_2 tenen R com a conseqüent, això vol dir que l'única fórmula que té sentit plantejar-se com a conclusió és f_1.

Pero como todas las implicaciones de f_2 tienen R como consecuente, esto quiere decir que la única fórmula que tiene sentido plantearse como conclusión es f_1.

Demostrarem, per tant, que

Demostraremos, por tanto, que

$$\underbrace{P(p)}_{\text{premissa 1}} \ , \ \underbrace{\exists y : \forall x, \ ((P(x) \vee Q(y)) \Rightarrow R(y))}_{\text{premissa 2}} \vdash \underbrace{\exists x : R(x)}_{\substack{\text{conclusió} \\ \textit{conclusión}}}$$

1)	$\exists y: \forall x, ((P(x) \vee Q(y)) \Rightarrow R(y))$	premissa 2
2)	$[\forall x, ((P(x) \vee Q(a_1)) \Rightarrow R(a_1))]$	**EP** ... (introduïm a_1) *introducimos*
3)	$(P(p) \vee Q(a_1)) \Rightarrow R(a_1)$	... **EG** (elegim $x = p$) *elegimos*
4)	$P(p)$	... premissa 1
5)	$P(p) \vee Q(a_1)$	... **ID** en 4
6)	$R(a_1)$	... **EI** en 3, 5
7)	$\exists x: R(x)$	**IP** en 6, acaba **EP** (a_1 ha desaparegut) *ha desaparecido*

Si repensem ara f_2 com a conclusió, i ens fixem en què el generalitzador només abarca la disjunció, la podem aleshores reescriure com a

Si repensamos ahora f_2 como conclusión, y nos fijamos en que el generalizador sólo abarca la disyunción, la podremos entonces reescribir como

$$\exists y, x: (P(x) \vee Q(y) \Rightarrow R(y))$$

I per a que siga certa, només cal trobar una combinació de valors que facen cert algun $R(y)$. I sabem que això pot passar si f_1 és certa.

Y para que sea cierta, sólo hace falta encontrar una combinación de valores que hagan cierto algún $R(y)$. Y sabemos que esto puede pasar si f_1 es cierta.

Anem doncs a intentar la deducció

Vamos pues a intentar la deducción

$$f_1, f_3 \vdash f_2$$

1)	$\exists x: R(x)$	premissa 1
2)	$P(p)$	premissa 2
3)	$R(a_1)$	**EP** ... (introduïm a_1) *introducimos*
4)	$[\forall x, ((P(x) \vee Q(a_1))]$	... **II** (suposem antecedent a_1)... *suponemos antecedente*
5)	$(\forall x, ((P(x) \vee Q(a_1))) \Rightarrow R(a_1)$	... acaba **II**
6)	$\exists y: \forall x, ((P(x) \vee Q(y)) \Rightarrow R(y))$	**IP** en 5, acaba **EP** (a_1 ha desaparegut) *ha desaparecido*

Si considerem finalment f_3 com a conclusió no es veu cap manera de deduir-la a partir de les altres fórmules. Però per a estar segurs de que no és deduïble, cal trobar un contraexemple.

Si consideramos finalmente f_3 como conclusión no se ve ninguna manera de deducirla a partir de las otras fórmulas. Pero para estar seguros de que no es deducible, hay que encontrar un contraejemplo.

Considerem el cas més senzill en què el domini de P només conté l'element p i el domini de Q i R (que han de ser el mateix) només tenen un element. En eixe cas, totes les fórmules esdevenen proposicionals i la nostra deducció es pot escriure com a

Consideremos el caso más sencillo en que el dominio de P sólo contiene el elemento p y el dominio de Q y R (que tienen que ser el mismo) sólo tienen un elemento. En ese caso, todas las fórmulas son proposicionales y nuestra deducción se puede escribir como

$$R, P \vee Q \Rightarrow R \vdash P$$

Si suposem que la primera premissa, R, és certa, tenim que

Si suponemos que la primera premisa, R, es cierta, tenemos que

$$(P \vee Q \Rightarrow V \equiv V) \vdash P$$

la qual cosa és falsa ja que P pot no ser cert.

lo cual es falso ya que P puede no ser cierto.

2.6 Problemes proposats

Problemas Propuestos

Problema 2.12: [absorcions]

Considera les lleis d'absorció de la conjunció i de la disjunció (pàgina 65),

Considera la leyes de absorción de la conjunción y de la disyunción (página 65),

$$(p \vee q) \wedge p \equiv p \qquad (p \wedge q) \vee p \equiv p$$

i demostra-les,

y demuéstralas,

a) usant taules de veritat.

a) usando tablas de verdad.

b) usant altres lleis d'equivalència.

b) usando otras leyes de equivalencia.

c) usant exclusivament regles d'inferència.

c) usando exclusivamente reglas de inferència.

d) En resum, quina és la forma més fàcil de demostrar les dues lleis d'absorció?

d) En resumen, ¿cuál es la forma más fácil de demostrar las dos leyes de absorción?

Problema 2.13: [lesTresAfirmacions]

Intenta establir la veritat o falsedat de cada una de les tres afirmacions següents:

Intenta establecer la verdad o falsedad de cada una de las tres afirmaciones siguientes:

a) La següent afirmació és **falsa**.

a) La siguiente afirmación es **falsa**.

b) El que vaig a dir a continuació és **cert**.

b) Lo que voy a decir a continuación es **cierto**.

c) No sé si certes o falses, però les dues afirmacions anteriors són **el mateix**.

c) No sé si falsas o ciertas, pero las dos afirmaciones anteriores son **lo mismo**.

■

Problema 2.14: [lesSisAfirmacions]

Sabries dir quines de les següents afirmacions són certes?

a) Totes les posteriors (a aquesta).

b) Cap de les que segueixen.

c) Només una de les anteriors.

d) Totes les anteriors.

e) Cap de les anteriors.

f) Cap de les anteriors.

Resol primer el problema amb la suposició addicional que **només una** de les afirmacions és certa.

¿Sabrías decir cuales de las siguientes afirmaciones son ciertas?

a) Todas las posteriores (a ésta).

b) Ninguna de las que siguen.

c) Sólo una de las anteriores.

d) Todas las anteriores.

e) Ninguna de las anteriores.

f) Ninguna de las anteriores.

Resuelve primero el problema con la suposición adicional de que **sólo una** *de las afirmaciones es cierta.*

■

Problema 2.15: [tresPredicats]

En la següent expressió, identifica premisses i conclusió i fes la demostració pas a pas especificant totes les regles d'inferència usades.

En la siguiente expresión, identifica premisas y conclusión y haz la demostración paso a paso especificando todas las reglas de inferencia usadas.

$$\forall X : ((\neg P(X) \vee Q(X)) \Rightarrow R(X)) \wedge \exists X : Q(X) \vdash \exists X : (P(X) \vee R(X))$$

Problema 2.16: [germans]

Considera les relacions de parentiu familiars normals de primer i segon grau (pares, fills, avis, etc.), i tradueix les afirmacions a la lògica de primer ordre. Digues si constitueixen o no una contradicció i si es pot deduir alguna cosa sobre la família concreta.

Considera las relaciones de parentesco familiares normales de primer y segundo grado (padres, hijos, abuelos, etc.) y traduce las afirmaciones siguientes a la lógica de primer orden. Di si constituyen o no una contradicción y si se puede deducir algo sobre la família concreta.

Bohigues tenia un germà. El germà de Bohigues va morir. No obstant això, l'home que va morir no va tenir mai cap germà.

Bohigues tenía un hermano. El hermano de Bohigues murió. Sin embargo, el hombre que murió nunca tuvo ningún hermano.

Problema 2.17: [irracional]

Una demostració senzilla de la irracionalitat de $\sqrt{2}$ consisteix a suposar el contrari, és a dir que podem escriure

Una demostración sencilla de la irracionalidad de $\sqrt{2}$ consiste en suponer lo contrario, es decir que podemos escribir

$$\sqrt{2} = \frac{a}{b}$$

i que, a més a més, aquesta fracció no es pot simplificar més (és irreductible).

y que, además, esta fracción no se puede simplificar más (es irreducible).

Si això fos cert, en elevar al quadrat i aïllar, tindríem que

Si eso fuera cierto, elevando al cuadrado y despejando tendríamos que

$$a^2 = 2b^2$$

la qual cosa implica que a^2 és parell.

lo que implica que a^2 es par.

Si a^2 és parell, també ho ha de ser a (problema 2.9),

Si a^2 es par, también lo tiene que ser a (problema 2.9), la qual cosa implica que b no pot ser parell, perquè en aquest cas $\frac{a}{b}$ es podria simplificar.

lo que implica que b no puede ser par, por que en ese caso $\frac{a}{b}$ se podría simplificar.

Però si a és parell s'ha de poder escriure $a = 2k$ per a algun k, amb la qual cosa

Pero si a es par se tiene que poder escribir $a = 2k$ para algún k, con lo cual

$$a^2 = 2k\cancel{2}k = \cancel{2}b^2$$

I, per tant, b^2 és parell i b també.

Y, por tanto, b^2 es par y b también.

O siga, que b ha de ser parell i senar alhora, la qual cosa és impossible. Per tant, la suposició inicial que $\sqrt{2}$ és racional ha de ser falsa.

O sea, que b tiene que ser par e impar al mismo tiempo, lo que es imposible. Por tanto, la suposición inicial de que $\sqrt{2}$ es racional tiene que ser falsa.

Descriu aquesta demostració, incloent-hi l'enunciat, usant la lògica de predicats i les corresponents regles d'inferència. El resultat del problema 2.9 el pots fer servir com una regla d'inferència més.

Describe esta demostración, incluyendo el enunciado, usando la lógica de predicados y las correspondientes reglas de inferencia. El resultado del problema 2.9 lo puedes usar como una regla de inferencia más.

Problema 2.18: [enTirant]

En la següent expressió, i en el text a continuació, identifica premisses i conclusió i fes la corresponent deducció fent servir les regles d'inferència estàndard.

En la siguiente expresión, y en el texto a continuación, identifica premisas y conclusión y haz la correspondiente deducción usando las reglas de inferencia estándar.

$$\forall X : \exists Y : (P(X) \Rightarrow (Q(X,Y) \vee R(Y))),\ P(k),\ \forall X : \neg R(X) \vdash \exists X, Y : Q(X,Y)$$

Tot cavaller errant té la seua amada, de manera que o bé aquest l'estima bojament o és que ella és monja (o totes dues coses). En Tirant és un cavaller errant. I no hi ha monges en el regne. Com que l'estimada d'en Tirant ha d'existir (com la de qualsevol cavaller errant), i com que resulta que no hi ha monges en aquest regne, aleshores hi ha almenys un cavaller i una dama de manera que el primer estima bojament la segona.

Todo caballero andante tiene su amada, de manera que o bien éste la ama alocadamente o es que ella es monja (o las 2 cosas). Don Tirante es un caballero andante. Y no hay en el reino monja alguna. Puesto que la amada de Don Tirante tiene que existir (como la de cualquier caballero andante), y resulta que no hay monjas en este reino, entonces hay al menos un caballero y una dama de manera que el primero ama alocadamente a la segunda.

3. Inducció i recursió

Inducción y recursión. 3

3.1 Definicions i predicats recursius

Definiciones y predicados recursivos

Una definició es pot entendre com una manera d'especificar un concepte nou a partir d'altres conceptes més bàsics o ja coneguts. En el context de la lògica, la definició d'un predicat és una doble implicació amb una fórmula, de manera que podem establir unívocament el valor de veritat o falsedat del predicat en funció dels seus arguments. Per exemple, els predicats $P(n)$ i $I(n)$ que són certs si n és parell/senar i falsos en cas contrari es podrien definir com a

Una definición se puede entender como una manera de especificar un concepto nuevo a partir de otros conceptos más básicos o ya conocidos. En el contexto de la lógica, la definición de un predicado es una doble implicación con una fórmula, de manera que podemos establecer unívocamente el valor de verdad o falsedad del predicado en función de sus argumentos. Por ejemplo, los predicados $P(n)$ y $I(n)$ que son ciertos si n es par/impar y falsos en caso contrario se podrían definir como

$$\boxed{P(n)} \Leftrightarrow \boxed{\exists k \in \mathbb{Z}^+ \ : \ n = 2 \cdot k}$$
$$\boxed{I(n)} \Leftrightarrow \boxed{\neg P(n)}$$

Un altre exemple en matemàtiques és la definició de límit d'una successió

Otro ejemplo en matemáticas es la definición de límite de una sucesión

$$\boxed{\lim_{n\to\infty} s_n = a} \Leftrightarrow \boxed{\forall \varepsilon \in \mathbb{R}, \ \varepsilon > 0, \ \exists n_0 \in \mathbb{N} \ : \ \forall n > n_0, \ |s_n - a| \leq \varepsilon}$$

Parlem de **definició recursiva** quan el concepte l'especifiquem en funció d'ell mateix. Per exemple i de manera informal, "un nombre enter és parell si (i solament si) el seu immediat anterior no ho és".

Hablamos de **definición recursiva** *cuando el concepto lo especificamos en función de él mismo. Por ejemplo y de manera informal, "un número entero es par si (y solamente si) su inmediato anterior no lo es".*

Tota definició recursiva ha de tenir un o més **casos base** de manera que quan s'aplica la definició recursiva a qualsevol element del domini sempre s'arribe a algun cas base.

Toda definición recursiva tiene que tener uno o más **casos base** *de manera que cuando se aplica la definición recursiva a cualquier elemento del dominio siempre se llegue a algún caso base.*

Per exemple, el cas base de l'exemple anterior podria ser que el nombre zero és parell. De manera que la definició recursiva expressada formalment seria

Por ejemplo, el caso base del ejemplo anterior podría ser que el número cero es par. De manera que la definición recursiva expresada formalmente sería

$$P(n) = \begin{cases} V & \text{si } n = 0 \quad (\text{CB}, \underset{\textit{caso base}}{\text{cas base}}) \\ \neg P(n-1) & \text{si } n > 0 \quad (\text{CR}, \underset{\textit{caso recursivo}}{\text{cas recursiu}}) \end{cases}$$

i si l'aplicàrem a l'enter 3 tindríem una cosa com ara

y si lo aplicáramos al entero 3 tendríamos algo como

$$P(3) = \neg P(2) = \neg\neg P(1) = P(1) = \neg P(0) = \neg V = F.$$

La definició anterior que és una igualtat amb condicions és el que anomenem **relació de recurrència**.

La definición anterior que es una igualdad con condiciones es lo que denominamos **relación de recurrencia**.

Una forma alternativa en lògica per a la mateixa definició seria una conjunció (implícita) entre el(s) cas(os) base i el cas recursiu on tindríem una coimplicació, ja que la relació entre un enter i el següent o l'anterior és la mateixa.

Una forma alternativa en lógica para la misma definición sería una conjunción (implícita) entre el(los) caso(s) base y el caso recursivo donde tendríamos una coimplicación, ya que la relación entre un entero y el siguiente o el anterior es la misma.

$$\begin{cases} P(0) & (\text{CB}) \\ \forall n > 0,\ P(n) \Leftrightarrow \neg P(n-1) & (\text{CR}) \end{cases}$$

i calcularíem $P(3)$ de la manera següent. Primer obtindríem un total de 4 premisses en aplicar la regla CR per a $n = 1, 2, 3$ i la regla CB.

y calcularíamos $P(3)$ de la manera siguiente. Primero obtendríamos un total de 4 premisas al aplicar la regla CR para $n = 1, 2, 3$ y la regla CB.

$$\underbrace{(P(3) \Leftrightarrow \neg P(2))}_{p_1} \wedge \underbrace{(P(2) \Leftrightarrow \neg P(1))}_{p_2} \wedge \underbrace{(P(1) \Leftrightarrow \neg P(0))}_{p_3} \wedge \underbrace{P(0)}_{p_4}$$

I a partir d'aquestes podríem fer la següent deducció

Y a partir de estas podríamos hacer la siguiente deducción

1)	$P(0)$	p_4
2)	$P(0) \Rightarrow \neg P(1)$	Def. coimp + **EC**+ Contrap. lòg. en p_3
3)	$\neg P(1)$	**EI** en 1) 2)
4)	$\neg P(1) \Rightarrow P(2)$	Def. coimp + **EC** en p_2
5)	$P(2)$	**EI** en 3) 4)
6)	$P(2) \Rightarrow \neg P(3)$	Def. coimp + **EC**+ Contrap. lòg. en p_1
7)	$\neg P(3)$	**EI** en 5) 6)

És important adonar-se que per a poder afirmar alguna cosa a partir d'una definició recursiva s'està fent un raonament que connecta el que es vol demostrar amb un cas base per a poder encadenar una seqüència d'implicacions (línies 2, 4 i 6) des del cas base fins a la conclusió corresponent (línia 7).

Es importante darse cuenta de que para poder afirmar algo a partir de una definición recursiva se está haciendo un razonamiento que conecta lo que se quiere demostrar con un caso base para poder encadenar una secuencia de implicaciones (líneas 2, 4 y 6) desde el caso base hasta la conclusión correspondiente (línea 7).

En lògica són particularment interessants les definicions que fan servir implicacions en el mateix sentit en què després seran aplicades en les demostracions de veritat corresponents. És a dir, definicions de predicats, Q, en principi sobre $\mathbb{Z}^+$ com ara

En lógica son particularmente interesantes las definiciones que usan implicaciones en el mismo sentido en que después serán aplicadas en las demostraciones de verdad correspondientes. Es decir, definiciones de predicados, Q, en principio sobre $\mathbb{Z}^+$ como

$$\left\{ \begin{array}{rll} & \forall x \le b, \ \boxed{Q(x)} & (\text{CB, un o més casos base} \ \ _{\textit{uno o más casos base}}) \\ \forall x \ : & \boxed{x > b}, \ \boxed{\mathcal{C}_i(Q(\alpha_i(x))) \Rightarrow Q(x)} & (\text{CR}_i\text{, cas recursiu i-èsim} \ \ _{\textit{caso recursivo i-ésimo}}) \end{array} \right.$$

on α_i és un functor que a partir d'un element del domini, x, ens dona un element "anterior", $\alpha_i(x) < x$, i $\mathcal{C}_i$ és una fbf que conté Q i que normalment és una conjunció de predicats.

donde α_i es un functor que a partir de un elemento del dominio, x, nos da un elemento "anterior", $\alpha_i(x) < x$, y $\mathcal{C}_i$ es una fbf que contiene Q y que normalmente es una conjunción de predicados.

Per a obtenir una definició d'aquest tipus en el cas de l'exemple anterior podem transformar-la de manera que en la definició apareguen les implicacions en el sentit en què seran usades en les deduccions.

Para obtener una definición de este tipo en el caso del ejemplo anterior podemos transformarla de manera que en la definición aparezcan las implicaciones en el sentido en que serán usadas en las deducciones.

$$\begin{cases} \qquad\qquad P(0) & (CB) \\ \forall n > 0, \quad P(n-1) \Rightarrow \neg P(n) & (CR_1) \\ \forall n > 1, \quad \neg P(n-1) \Rightarrow P(n) & (CR_2) \end{cases}$$

Amb aquesta definició els raonaments acabarien en CR_1 o en CR_2 segons es vulga demostrar la falsedat o la veritat de P, respectivament.

Con esta definición los razonamientos acabarían en CR_1 o en CR_2 según se quiera demostrar la falsedad o la verdad de P, respectivamente.

També podem combinar els dos casos recursius anteriors (i fer algunes manipulacions més) per a arribar a la següent definició equivalent.

También podemos combinar los dos casos recursivos anteriores (y hacer algunas manipulaciones más) para llegar a la siguiente definición equivalente.

$$\begin{cases} \qquad\qquad P(0) & (CB_1) \\ \qquad\qquad \neg P(1) & (CB_2) \\ \forall n > 1, \quad P(n-2) \Rightarrow P(n) & (CR_1) \\ \forall n > 2, \quad \neg P(n-2) \Rightarrow \neg P(n) & (CR_2) \end{cases}$$

on ara relacionem la veritat (falsedat) de dos nombres parells (senars) consecutius.

donde ahora relacionamos la verdad (falsedad) de dos números pares (impares) consecutivos.

Sobretot si ens plantegem la possibilitat d'automatitzar raonaments, és encara més important restringir les definicions recursives a implicacions on el conseqüent no aparega negat.

Sobre todo si nos planteamos la posibilidad de automatizar razonamientos, es aún más importante restringir las definiciones recursivas a implicaciones donde el consecuente no aparezca negado.

Això planteja un problema a l'hora de demostrar la falsedat. Per això, en alguns sistemes de raonament automàtics i en el llenguatge de programació Prolog en particular, se segueix la convenció anomenada **negació per fallada**. Això vol dir, amb poques paraules, que si no podem deduir la **veritat** d'un predicat a partir de la seua "definició", aleshores és fals.

Esto plantea un problema a la hora de demostrar la falsedad. Por eso, en algunos sistemas de razonamiento automáticos y en el lenguaje de programación Prolog en particular, se sigue la convención llamada **negación por fallo***. Esto quiere decir, con pocas palabras, que si no podemos deducir la* **verdad** *de un predicado a partir de su "definición", entonces es falso.*

Per exemple, el predicat P anterior es podria "definir" també com a

Por ejemplo, el predicado P anterior se podría "definir" también como

$$\begin{cases} \quad P(0) & \text{(CB)} \\ \forall n > 1, \quad P(n-2) \Rightarrow P(n) & \text{(CR)} \end{cases}$$

la qual cosa es traduiria en Prolog com a

lo que se traduciría en Prolog como

```
parell(0).          % CB
parell(N) :-
   N>1,              % CR
   M is N-2,         % avaluació del functor corresponent
                       evaluación del functor correspondiente
   parell(M).
```

Ara considerarem un altre exemple de definició recursiva usant lògica de predicats:

Ahora conseideraremos otro ejemplo de definición recursiva usando lógica de predicados:

$$\begin{cases} \quad S(1,1) & \text{(CB)} \\ \forall n > 1, \forall y \in Z^+, \; S(y, n-1) \Rightarrow S(y+n, n) & \text{(CR)} \end{cases}$$

El predicat $S(y,n)$ és cert si y és la suma dels n primers enters.

El predicado $S(y,n)$ es cierto si y es la suma de los n primeros enteros.

Aquesta última "definició" podem modificar-la lleugerament

Esta última "definición" podemos modificarla ligeramente

$$\begin{cases} \quad S(1,1) \\ \forall n > 1, \forall x \in Z^+, \; x \geq n, \; S(x-n, n-1) \Rightarrow S(x,n) \end{cases}$$

I fins i tot introduir noves variables per a l'antecedent de la implicació

E incluso introducir nuevas variables para el antecedente de la implicación

$$\begin{cases} S(1,1) \\ \forall n > 1, \forall x \in Z^+,\ x \geq n,\ \exists y, m\ :\ y = x - n \wedge m = n - 1 \wedge S(y,m) \Rightarrow S(x,n) \end{cases}$$

la qual cosa admet una traducció quasi literal al llenguatge de programació Prolog.

lo que admite una traducción casi literal al lenguaje de programación Prolog.

```
suma(1,1).
suma(X,N) :-
   N>1,
   M is N-1,
   suma(Y,M),
   X is Y+N.
```

No obstant això, el predicat anterior (convenientment estés per al valor zero) es correspon en realitat amb una funció,

No obstante, el predicado anterior (convenientemente extendido para el valor cero) se corresponde en realidad con una función.

$$s : \mathbb{Z}^+ \longrightarrow \mathbb{Z}^+,$$

que podríem representar alternativament mitjançant la recurrència

que podríamos representar alternativamente mediante la recurrencia

$$s(n) = \begin{cases} 0 & \textbf{si } n = 0 \\ s(n-1) + n & \textbf{si } n > 0 \end{cases}$$

Una relació de recurrència és en realitat un sistema d'equacions, que també es pot escriure com a

Una relación de recurrencia es en realidad un sistema de ecuaciones, que también se puede escribir como

$$\begin{cases} s(0) &= 0 \\ s(n) &= s(n-1) + n, \quad \forall\, n > 0. \end{cases}$$

No obstant això, les relacions de recurrència se solen interpretar com una operació o un càlcul:

No obstante, las relaciones de recurrencia se suelen interpretar como una operación o un cálculo:

$$s(n) \leftarrow \begin{cases} 0 & \text{si } n = 0 \\ s(n-1) + n & \text{si } n > 0 \end{cases}$$

on el símbol $\leftarrow$ significa "es calcula com a" o "es reescriu com a" segons que estiguem en un context de programació imperativa o funcional.

donde el símbolo $\leftarrow$ significa "se calcula como" o "se reescribe como" según estemos en un contexto de programación imperativa o funcional.

Per exemple, el predicat $s(n)$ de l'últim exemple es podria implementar en Haskell i en Python de la següent manera:

Por ejemplo, el predicado $s(n)$ del último ejemplo se podría implementar en Haskell y en Python de la siguiente manera:

```haskell
suma ::  Integer -> Integer
suma 0 = 0
suma n | n > 0 = n + suma (n-1)
```

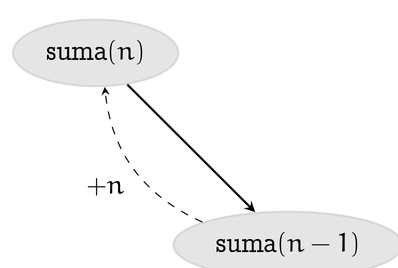
```python
def suma (n):
   if n==0:
      return 0
   else:
      return n + suma(n-1)
```

És molt important observar que totes les definicions recursives dels predicats anteriors juntament amb l'última recurrència així com totes les implementacions es corresponen amb la mateixa **idea recursiva** que en aquest cas consisteix a relacionar la suma dels n primers enters amb la suma dels $n-1$ primers enters.

Es muy importante observar que todas las definiciones recursivas de los predicados anteriores junto con la última recurrencia así como todas las implementaciones se corresponden con la misma **idea recursiva** *que en este caso consiste en relacionar la suma de los n primeros enteros con la suma de los $n-1$ primeros enteros.*

Aquesta idea també es pot representar de manera gràfica:

Esta idea también se puede representar de manera gráfica:

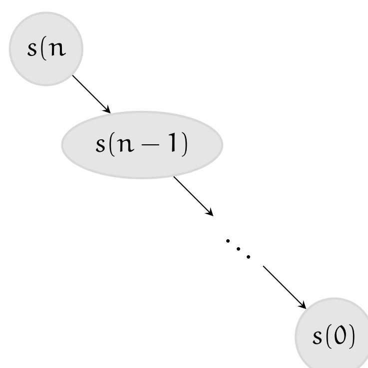


En aquest diagrama la fletxa descendent indica la relació recursiva entre el problema, $s(n)$, i el subproblema, $s(n-1)$, que es representen com a nodes. La fletxa puntejada ascendent indica que la subsolució o resultat del subproblema es fa servir per a obtenir la solució del problema per a n.

En este diagrama la flecha descendiente indica la relación recursiva entre el problema, $s(n)$, y el subproblema, $s(n-1)$, que se representan como nodos. La flecha punteada ascendiente indica que la subsolución o resultado del subproblema se usa para obtener la solución del problema para n.

Com que el subproblema (o subproblemes) es poden tornar a relacionar amb altres subproblemes recursivament, en expandir el diagrama anterior s'obté el que anomenarem **arbre de recursió**, que en el cas concret de l'exemple donaria un arbre amb una sola branca que representaríem simplificadament com a

Como el subproblema (o subproblemas) se puede(n) volver a relacionar con otros subproblemas recursivamente, al expandir el diagrama anterior se obtiene lo que llamaremos **árbol de recursión***, que en el caso concreto del ejemplo daría un árbol con una sola rama que representaríamos simplificadamente como*



En general serà convenient separar la idea recursiva d'una particular implementació o tipus de fórmula per a expressar-la.

En general será conveniente separar la idea recursiva de una particular implementación o tipo de fórmula para expresarla.

Més concretament, per a qualsevol funció, f,

Más concretamente, para cualquier función, f,

$$f : D \longrightarrow C$$

sempre existirà el corresponent predicat, P_f,

siempre existirá el correspondiente predicado, P_f,

$$P_f : D \times C \longrightarrow \{V, F\}$$

de manera que

de manera que

$$\forall c \in C\ \forall d \in D,\ \boxed{c = f(d)} \Leftrightarrow \boxed{P_f(c, d)}$$

A partir d'una mateixa idea recursiva sempre podrem donar definicions recursives que utilitzen f o P_f de manera equivalent.

A partir de una misma idea recursiva siempre podremos dar definiciones recursivas que utilicen f o P_f de manera equivalente.

Normalment sol ser més senzill escriure una relació recursiva per a f que no una definició lògica per a P_f. I més encara si la fórmula lògica ha de ser "executable" en, o traduïble a sistemes com ara Prolog.

Normalmente suele ser más sencillo escribir una relación recursiva para f que una definición lógica para P_f. Y más aún si la fórmula lógica tiene que ser "ejecutable" en, o traducible a sistemas como Prolog.

3.1.1 Estructures recursives

Estructuras recursivas

De vegades interessa representar informació usant estructures definides recursivament com per exemple el que ací anomenarem llista la definició de la qual donem a continuació.

A veces interesa representar información usando estructuras definidas recursivamente como por ejemplo lo que aquí llamaremos lista *cuya definición damos a continuación.*

• La llista buida, $\emptyset$, és una llista.

• *La lista vacía, $\emptyset$, es una lista.*

• Si L és una llista, la mateixa amb l'addició a l'esquerra d'un element e qualsevol, que anomenarem $\mathrm{constr}(e, L)$, també és una llista.

• *Si L es un alista, la misma con la adición a la izquierda de un elemento e cualquiera, que llamaremos $\mathrm{constr}(e, L)$, también es una lista.*

Si $\mathbb{L}$ és el domini format per totes les possibles llistes, i $\mathcal{U}$ és qualsevol domini, la funció constr la descriurem com

Si $\mathbb{L}$ es el dominio formado por todas las posibles listas, y $\mathcal{U}$ es cualquier dominio, la función constr la describiremos como

$$\mathrm{constr}\ :\ \mathcal{U} \times \mathbb{L} \longrightarrow \mathbb{L}.$$

Aquesta funció s'anomena en altres contextos constructor de la corresponent estructura per que permet obtindre qualsevol llista del domini a partir del cas base, $\emptyset$. Per comoditat, abreviarem l'expressió $\mathrm{constr}(e, L)$ com a $e|L$.

Esta función se denomina en otros contextos constructor *de la correspondiente estructura por que permite obtener cualquier lista del dominio a partir del caso base, $\emptyset$. Por comodidad, abreviaremos la expresión $\mathrm{constr}(e, L)$ como $e|L$.*

A partir de la definició, una llista no és més que una seqüència d'elements de qualsevol tipus. No obstant això, les escriurem usant claudàtors per que quede clar que estem parlant de llistes definides recursivament. Exemples de llistes serien

A partir de la definición, una lista no es más que una secuencia de elementos de cualquier tipo. Sin embargo, las escribiremos usando corchetes para que quede claro que estamos hablando de listas definidas recursivamente. Ejemplos de listas serían

$$[\,] = \emptyset \qquad [1, 2, a, pep] \qquad [a, b, [1, c, 2], 0] \qquad [[\,][[\,]]]$$

La segona d'aquestes llistes es pot escriure també com

La segunda de estas listas se puede escribir también como

$$1|[2, a, pep] = 1|2|[a, pep] = 1|2|a|[pep] = 1|2|a|pep|\emptyset$$

Associades a les llistes definirem també les funcions bàsiques:

Asociadas a las listas definiremos también las funciones básicas:

$$cap : \mathbb{L} \longrightarrow U,$$

que torna l'element que es troba en el primer lloc de la llista, i

que devuelve el elemento que se encuentra en el primer lugar de la lista, y

$$cua : \mathbb{L} \longrightarrow \mathbb{L},$$

que torna la mateixa llista sense l'element que es trobava en el primer lloc de la llista.

que devuelve la misma lista sin el elemento que se encontraba en el primer lugar de la lista.

Per a qualsevol llista no buida s'ha de complir necessàriament que

Para cualquier lista no vacía se debe cumplir necesariamente que

$$L = constr(cap(L), cua(L)) = cap(L)|cua(L)$$

Una vegada definida una estructura recursiva, es poden plantejar funcions recursives que les utilitzen. Per exemple, la funció recursiva que a partir d'una llista d'enters calcula la seua suma es podria especificar com

Una vez definida una estructura recursiva se pueden plantear funciones recursivas que las utilicen. Por ejemplo, la función recursiva que a partir de una lista de enteros calcula su suma se podría especificar como

$$s(L) = \begin{cases} 0 & \text{si } L = \emptyset \\ s(cua(L)) + cap(L) & \text{si } L \neq \emptyset \end{cases}$$

Aquest tipus de funcions sobre llistes ademeten una traducció quasi directa a llenguatges com Prolog:

Este tipo de funciones sobre listas admiten una traducción casi directa a lenguajes como Prolog:

```
suma([],0).
suma([E|L],S) :-
   suma(L,SS),
   S is SS+E.
```

3.1.2 Tipus de recursió i exemples

Tipos de recursión y ejemplos

Diem que una recursió és **lineal** quan la relació és només amb **un sol** cas anterior. S'anomena lineal perquè podem dibuixar el procés de càlcul per a un valor donat com una única cadena que acaba en un cas base.

Decimos que una recursión es **lineal** *cuando la relación és únicamente con* **un solo** *caso anterior. Se la llama lineal por que podemos dibujar el proceso de cálculo para un valor dado como una única cadena que acaba en un caso base.*

Factorial

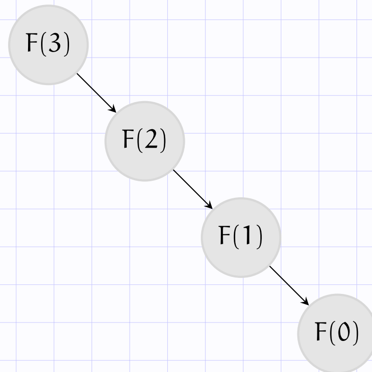
$$F(n) = \begin{cases} 1 & \text{si } n = 0 \\ n \cdot F(n-1) & \text{si } n > 0 \end{cases}$$

Aquest exemple és exactament igual que el de la suma dels n primers enters només canviant la suma per la multiplicació i tenint en compte que el cas base es correspon amb l'element neutre de cada operació.

Este ejemplo es exactamente igual que el de la suma de los n primeros enteros sólo cambiando la suma por la multiplicación y teniendo en cuenta que el caso base se corresponde con el elemento neutro de cada operación.

L'arbre de recursió per a $n = 3$ seria

El árbol de recursión para $n = 3$ sería



Aquesta representació arborescent es correspon amb la següent seqüència de substitucions, reescriptures o càlculs.

Esta representación arborescente se corresponde con la siguiente secuencia de sustituciones, reescrituras o cálculos.

$$F(3) = 3 \cdot F(2) = 3 \cdot 2 \cdot F(1) = 3 \cdot 2 \cdot 1 \cdot F(0) = 3 \cdot 2 \cdot 1 \cdot 1 = 3! = 6.$$

Una recursió és **no lineal** o també **múltiple** si la relació és al mateix temps amb dos o més casos anteriors. En aquest cas el procés de càlcul o **arbre de recursió** té més d'una branca.

*Una recursión es **no lineal** o también **múltiple** si la relación es al mismo tiempo con dos o más casos anteriores. En este caso el proceso de cálculo o **árbol de recursión** tiene más de una rama.*

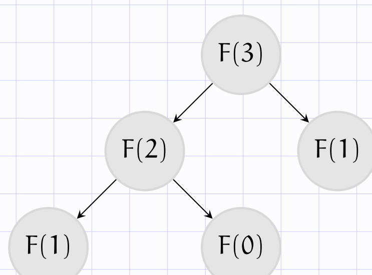
Successió de Fibonacci

Sucesión de Fibonacci

$$F(n) = \begin{cases} n & \text{si } n \leq 1 \\ F(n-1) + F(n-2) & \text{si } n > 1 \end{cases}$$

L'arbre de recursió per a $n = 3$ seria

El árbol de recursión para $n = 3$ sería



El càlcul de $F(3)$ representat per l'arbre anterior seria

El cálculo de $F(3)$ representado por el árbol anterior sería

$$F(3) = F(2) + F(1) = F(1) + F(0) + F(1) = 1 + 0 + 1 = 2.$$

La recurrència anterior la podem escriure com un predicat, $P_F(n, x)$, que siga cert quan $x = F(n)$.

La recurrencia anterior la podemos escribir como un predicado, $P_F(n, x)$, que sea cierto cuando $x = F(n)$.

$$\begin{cases} P_F(0,0) \wedge P_F(1,1) \\ \forall n > 1, \forall x_1, x_2 \in Z^+,\ P_F(n-1, x_1) \wedge P_F(n-2, x_2) \Rightarrow P_F(n, x_1 + x_2) \end{cases}$$

I aquest predicat es pot escriure també com un conjunt de regles en Prolog. Escrivim també el programa en Haskell que calcularia la funció F.

Y este predicado se puede escribir también como un conjunto de reglas en Prolog. Escribimos también el programa en Haskell que calcularía la función F.

```
fib(0,0).
fib(1,1).
fib(N,R) :-
   N>1,
   M1 is N-1,
   M2 is N-2,
   fib(M1,X1),
   fib(M2,X2),
   R is X1+X2.
```

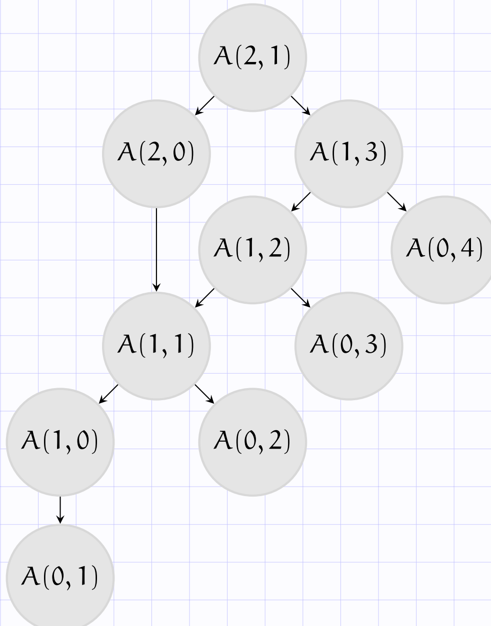
```
fib ::  Integer -> Integer
fib 0 = 0
fib 1 = 1
fib n |n>1 = fib (n-1) + fib (n-2)
```

Anomenem **recursió niada** quan es fa referència a algun cas anterior com a **argument** d'un altre cas anterior.

*Hablamos de **recursión anidada** cuando se hace referencia a algún caso anterior como **argumento** de otro caso anterior.*

Funció d'Ackermann | ***Función de Ackermann***

$$A(m,n) = \begin{cases} n+1 & \text{si } m = 0 \\ A(m-1, 1) & \text{si } m > 0 \wedge n = 0 \\ A(m-1, A(m, n-1)) & \text{si } m > 0 \wedge n > 0 \end{cases}$$



El càlcul de $A(2,1)$ seria

El cálculo de $A(2,1)$ sería

$$A(2,1) = A(1, A(2,0)) = A(1, \underline{A(1,1)}) = A(1, A(0, A(1,0))) = A(1, A(0, \underbrace{A(0,1)}_{2})) =$$

$$= A(1, \underbrace{A(0,2)}_{3}) = A(1,3) = A(0, A(1,2)) = A(0, A(0, \underline{A(1,1)})) =$$

Observem que en aquest punt arribem al càlcul de $A(1,1)$ per segona vegada. I com que ja sabem que $A(1,1) = 3$, continuem directament.

Observamos que en este punto llegamos al cálculo de $A(1,1)$ por segunda vez. Y como ya sabemos que $A(1,1) = 3$, contiinuamos directamente.

$$= A(0, \underbrace{A(0,3)}_{4}) = A(0,4) = 5.$$

L'especificació d'aquesta recursió en lògica de predicats i en Prolog es deixa com a exercici.

La especificación de esta recursión en lógica de predicados y en Prolog se deja como ejercicio.

El programa recursiu en Haskell que calcula la funció A és pràcticament una traducció literal de la relació de recurrència.

El programa recursivo en Haskell que calcula la función A es prácticamente una traducción literal de la relación de recurrencia.

```
ack ::  Integer -> Integer -> Integer
ack 0 n | n>=0 = n+1
ack m 0 | m>0 = ack (m-1) 1
ack m n | m>0,n>0 = ack (m-1) (ack m (n-1))
```

3.2 Principis d'inducció

Principios de inducción

El principi d'inducció és un esquema de raonament per a deduir fórmules que han de ser certes per a un domini infinit numerable.

El principio de inducción es un esquema de razonamiento para deducir fórmulas que tienen que ser ciertas para un dominio infinito numerable.

Normalment el que es vol demostrar és de la forma $\forall n,\ P(n)$. I aquesta alternativa a utilitzar les regles d'inferència en lògica de predicats es pot enunciar informalment de la següent manera.

Normalmente lo que se quiere demostrar es de la forma $\forall n,\ P(n)$. *Y esta alternativa a utilizar las reglas de inferencia en lógica de predicados se puede enunciar informalmente de la siguiente manera.*

Principi d'inducció (informalment)

1. Un predicat, P, és cert per a un primer valor.
2. Suposem aquest predicat, P, cert per a un valor genèric.
3. Si a partir de l'anterior podem demostrar que P és cert per al valor següent,
4. aleshores es pot assegurar que P és cert per a tots els valors a partir del primer.

Principio de inducción (informalmente)

1. *Un predicado,* P, *es cierto para un primer valor.*
2. *Suponemos este predicado,* P, *cierto para un valor genérico.*
3. *Si a partir de lo anterior podemos demostrar que* P *es cierto para el valor siguiente,*
4. *entonces se puede asegurar que* P *es cierto para todos los valores a partir del primero.*

L'ítem 1 és el que anomenem cas base o **base d'inducció** (**BI**), i normalment s'identifica amb el valor 0 o el valor 1 o, en general, amb el valor enter més baix per al qual es puga complir P.

El ítem 1 es lo que llamamos caso base o **base de inducción** *(***BI***), y normalmente se identifica con el valor 0 o el valor 1 o, en general, con el valor entero más bajo para el que se pueda cumplir* P.

L'ítem 2 és l'anomenada **hipòtesi d'inducció** (**HI**), que consisteix a suposar cert el que es vol demostrar.

El ítem 2 es la llamada **hipótesis de inducción** *(***HI***), que consiste en suponer cierto lo que se quiere demostrar.*

El **pas d'inducció** (**PI**) de l'ítem 3 és la demostració de la veracitat de P per al valor següent, la qual cosa (per la regla **II**) demostra la veracitat d'una implicació entre qualssevol valors consecutius.

El **paso de inducción** *(***PI***) del ítem 3 es la demostración de la veracidad de P para el valor siguiente, lo cual (por la regla* **II***) demuestra la veracidad de una implicación entre cualesquiera valores consecutivos.*

I són precisament aquestes implicacions encadenades les que ens permeten arribar a la conclusió (ítem 4) gràcies a l'aplicació múltiple de la regla **EI** començant pel cas base.

Y son precisamente estas implicaciones encadenadas las que nos permiten llegar a la conclusión (ítem 4) gracias a la aplicación múltiple de la regla **EI** *empezando por el caso base.*

Més formalment,

Más formalmente,

Principi d'inducció (bàsica)

Principio de inducción (básica)

$$\begin{array}{ll} P(n_0) & \textbf{BI} \\ & (*)\ \textbf{HI} \\ \forall n \geq n_0,\ \overbrace{P(n)}^{(*)} \Rightarrow P(n+1) & \textbf{PI} \\ \hline \forall n \geq n_0,\ P(n) & \end{array}$$

Exemple: És cert que

Ejemplo: ¿Es cierto que

$$S_1(n) = \sum_{i=1}^{n} i = \frac{n}{2}(n+1)\ \forall n?$$

Primer comprovarem el valor de la fórmula per a alguns valors petits de n.

Primero comprobaemos el valor de la fórmula para algunos valores pequeños de n.

n	$S_1(n)$	$\frac{n}{2}(n+1)$
0	0	0
1	1	$\frac{1}{2}(2) = 1$
2	$1+2=3$	$\frac{2}{2}(3) = 3$
3	$1+2+3=6$	$\frac{3}{2}(4) = 6$

Com veiem, es pot agafar el cas $n = 0$ com a **base d'inducció**.

Como vemos, se puede tomar el caso $n = 0$ como **base de inducción**.

Suposem com a **hipòtesi d'inducció** que per a un **valor genèric** que anomenem n es compleix que

*Suponemos como **hipótesis de inducción** que para un **valor genérico** que llamamos n se cumple que*

$$\left[S_1(n) = \frac{n}{2}(n+1)\right].$$

Escrivim la **HI** entre claudàtors per a indicar que es tracta d'una suposició.

*Escribimos la **HI** entre corchetes para indicar que se trata de una suposición.*

El **pas d'inducció** consisteix a arribar a la fórmula per a l'enter següent, és a dir a

*El **paso de inducción** consiste en llegar a la fórmula para el entero siguiente, es decir a*

$$S_1(n+1) = \frac{n+1}{2}(n+2),$$

a partir de la hipòtesi, per a la qual cosa necessitarem relacionar dos casos consecutius.

a partir de la hipótesis, para lo que necesitaremos relacionar dos casos consecutivos.

En el cas de sumatoris quasi sempre resulta molt fàcil relacionar dos casos consecutius mitjançant la separació de l'últim terme,

En el caso de sumatorios casi siempre resulta muy fácil relacionar dos casos consecutivos mediante la separación del último término,

$$S_1(n+1) = \sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = S_1(n) + (n+1).$$

És interessant adonar-se que la relació entre dos termes consecutius del sumatori juntament amb el fet que $S_1(0) = 0$ es correspon amb la **definició recursiva** del mateix sumatori,

*Es interesante darse cuenta de que la relación entre los dos términos consecutivos del sumatorio junto con el hecho de que $S_1(0) = 0$ se corresponde con la **definición recursiva** del mismo sumatorio,*

$$S_1(n) = \begin{cases} 0 & \text{si } n = 0 \\ S_1(n-1) + n & \text{si } n > 0. \end{cases}$$

Podem ara encadenar la relació recursiva amb la hipòtesi d'inducció i aleshores

Podemos ahora encadenar la relación recursiva con la hipótesis de inducción y entonces

$$S_1(n+1) \overset{(\text{def})}{=} S_1(n)+(n+1) \overset{(\text{HI})}{=} \frac{n}{2}(n+1)+(n+1) = (\frac{n}{2}+1)(n+1) = \frac{n+2}{2}(n+1),$$

que és, precisament, on es volia arribar.

que es, precisamente, donde se quería llegar.

Podem resumir la demostració esquemàticament de la següent manera:

Podemos resumir la demostración esquemáticamente de la siguiente manera:

BI: $S_1(0) = 0,$
HI: $\left[S_1(n) = \frac{n}{2}(n+1)\right],$
PI: $S_1(n+1) \overset{(\text{def})}{=} S_1(n) + (n+1) \overset{(\text{HI})}{=} \frac{n}{2}(n+1) + (n+1) = \frac{n+2}{2}(n+1).$

3.2.1 Principi d'inducció forta

Principio de inducción fuerte

En alguns casos, la relació entre un cas i el següent és difícil d'especificar però en canvi és fàcil la relació amb casos anteriors que no siguen l'immediatament anterior.

En algunos casos, la relación entre un caso y el siguiente es difícil de especificar pero en cambio es fácil la relación con casos anteriores que no sean el inmediatamente anterior.

En aquests casos, es pot utilitzar una versió més forta (amb una suposició més àmplia) del principi d'inducció.

En estos casos, se puede utilizar una versión más fuerte (con una suposición más amplia) del principio de inducción.

Principi d'inducció forta | ***Principio de inducción fuerte***

$$\begin{array}{ll} P(n_0) & \textbf{BI} \\ & (*)\ \textbf{HI} \\ \forall n > n_0\ (\overbrace{\forall k < n\ P(k)}^{(*)}) \Rightarrow P(n) & \textbf{PI} \\ \hline \forall n \geq n_0\ P(n) & \end{array}$$

Exemple: l'n-èsim terme de la successió de Fibonacci, f_n, compleix que

Ejemplo: el n-ésimo término de la sucesión de Fibonacci, f_n, cumple que

$$f_n < 2^n.$$

Recordem primer la definició de la successió de Fibonacci,

Recordemos primero la definición de la sucesión de Fibonacci,

$$f_0 = 0,\ f_1 = 1,\ \ \forall n > 1,\ f_n = f_{n-1} + f_{n-2},$$

i comprovem els primers valors de n,

y comprobemos los primeros valores de n,

n	f_n	2^n
0	0	1
1	1	2
2	1	4
3	2	8

La demostració usant el principi d'inducció forta es podria resumir com a

La demostración usando el principio de inducción fuerte se podría resumir como

BI: $f_0 = 0 < 2^0 = 1$
$f_1 = 1 < 2^1 = 2$

HI: $[\forall k < n,\ f_k < 2^k]$

PI: $f_n \stackrel{(\mathrm{def})}{=} f_{n-1} + f_{n-2} \stackrel{(\mathrm{HI})}{<} 2^{n-1} + 2^{n-2} = (2+1)2^{n-2} < 4 \cdot 2^{n-2} = 2^n$

És important remarcar que en aquest cas són necessaris dos casos base perquè cada cas (recursiu) està sempre relacionat amb els dos casos immediatament anteriors.

Es importante remarcar que en este caso son necesarios dos casos base por que cada caso (recursivo) está siempre relacionado con los dos casos inmediatamente anteriores.

3.2.2 Principi d'inducció general o Noetheriana

Principio de inducción general o Noetheriana

El principi d'inducció forta es pot generalitzar per a dominis més enllà dels enters en els quals es puga definir una relació d'ordre o fins i tot una relació encara més simple.

El principio de inducción fuerte se puede generalizar para dominios más allá de los enteros en los cuales se pueda definir una relación de orden o incluso una relación todavía más simple.

Preordre ben fundat | ***Preorden bien fundado***

Una relació binària, $\preceq$, és un **preordre ben fundat** si compleix les propietats

Una relación binaria, $\preceq$, es un **preorden bien fundado** *si cumple las propiedades*

a) **reflexiva** ($a \preceq a, \ \forall a$),

b) **transitiva** ($a \preceq b \wedge b \preceq c \Rightarrow a \preceq c, \ \forall a, b, c$),

c) i en el domini no hi ha successions infinites estrictament decreixents de la forma

c) y en el dominio no hay sucesiones infinitas estrictamente decrecientes de la forma

$$a_1 \succ a_2 \succ \cdots$$

on la relació $\succ$ es defineix com a

donde la relación $\succ$ se define como

$$a_i \succ a_{i+1} \Leftrightarrow (a_{i+1} \preceq a_i \wedge a_{i+1} \neq a_i)$$

Una conseqüència immediata d'un preordre ben fundat és l'existència d'un conjunt no buit, $\mathcal{M}$, d'elements **minimals** que constituiran els casos base d'inducció.

Una consecuencia inmediata de un preorden bien fundado es la existencia de un conjunto no vacío, $\mathcal{M}$, de elementos **minimales** *que constituirán los casos base de inducción.*

La inducció Noetheriana és exactament igual que la inducció forta només canviant la relació d'ordre usual sobre els enters pel preordre ben fundat definit sobre el domini corresponent.

La inducción Noetheriarna es exactamente igual que la inducción fuerte sólo cambiando la relación de orden usual sobre los enteros por el preorden bien fundado definido sobre el dominio correspondiente.

Principi d'inducció general o Noetheriana | *Principio de inducción general o Noetheriana*

$$\cfrac{\begin{array}{ll} \forall z_0 \in \mathcal{M},\ P(z_0) & \textbf{BI} \\ & (*)\ \textbf{HI} \\ \forall z \notin \mathcal{M}\ \Big(\overbrace{\forall t \prec z\ P(t)}^{(*)}\Big) \Rightarrow P(z) & \textbf{PI} \end{array}}{\forall z\ P(z)}$$

Exemple: El valor de la funció d'Ackermann, $A(m,n)$, compleix que | ***Ejemplo: ¿El valor de la función de Ackermann, $A(m,n)$, cumple que***

$$A(m,n) > 2^{m+n-1}?$$

¿A partir de quins valors de m i n? | ***¿A partir de qué valores de m y n?***

El primer que cal fer és definir un preordre ben fundat sobre parells d'enters de manera que aquesta relació siga compatible amb la relació de recurrència de la funció d'Ackermann en la pàgina 135.

Lo primero que hay que hacer es definir un preorden bien fundado sobre parejas de enteros de manera que esta relación sea compatible con la relación de recurrencia de la función de Ackermann en la página 135.

Com que la referència niada a un cas anterior apareix com a segon argument d'una altra, basarem el preordre en el primer element del parell d'enters.

Como la referencia anidada a un caso anterior aparece como segundo argumento de otra, basaremos el preorden en el primer elemento del par de enteros.

En particular, definim la relació, $\preceq$, entre parells d'enters com a

En particular, definimos la relación, $\preceq$, entre pares de enteros como

$$(k,\ell) \preceq (m,n) \Leftrightarrow (k < m) \vee (k = m \wedge \ell \geq n).$$

D'aquesta manera, els casos anteriors a què fa referència la definició recursiva són també anteriors segons $\preceq$.

De esta manera, los casos anteriores a que hace referencia la definición recursiva son también anteriores según $\preceq$.

Posposarem intencionadament la base d'inducció i enunciarem directament la hipòtesi.

Pospondremos intencionadamente la base de inducción y enunciaremos directamente la hipótesis.

HI:

$$\left[\forall (k,\ell) \prec (m,n),\ A(k,\ell) > 2^{k+\ell-1}\right].$$

I a continuació el pas d'inducció.

Y a continuación el paso de inducción.

PI:

Considerem primer el cas en què n és igual a zero.

Consideremos primero el caso en que n es igual a cero.

$$A(m,n) \overset{(n=0)}{=} A(m-1,1) \overset{(\text{HI})}{>} 2^{m-1} = 2^{m+0-1} = 2^{m+n-1}.$$

Vegem que el **PI** es compleix per a valors arbitraris de $m > 0$.

*Vemos que el **PI** se cumple para valores arbitrarios de $m > 0$.*

En el cas alternatiu, $n > 0$, es té que

En el caso alternativo, $n > 0$, se tiene que

$$A(m,n) \overset{(n>0)}{=} A(m-1, A(m,n-1)) \overset{(\text{HI})}{>} A(m-1, 2^{m+n-2}) \overset{(\text{HI})}{>} 2^{m-2+2^{m+n-2}} \geq 2^{m+n-1}.$$

En l'última desigualtat hem considerat que

En la última desigualdad hemos considerado que

$$2^{m+n-2} = 2^{m-2} \cdot 2^n \overset{(m-2\geq 0)}{\geq} 2^0 \cdot 2^n = 2^n \overset{(n\geq 0)}{\geq} n+1,$$

per la qual cosa el pas d'inducció en aquest segon cas només serà cert si

por lo que el paso de inducción en este segundo caso sólo será cierto si

$$\boxed{m \geq 2.}$$

Com veiem, la demostració és correcta llevat de la base d'inducció. Analitzem ara els valors més petits de la funció per a esbrinar quins són els valors associats a la base d'inducció.

Como vemos, la demostración es correcta excepto la base de inducción. Analicemos ahora los valores más pequeños de la función para averiguar cuales son los valores asociados a la base de inducción.

En el cas $m = 0$ tenim

En el caso $m = 0$ tenemos

$$A_0(n) = A(0,n) = n+1,$$

i és clar que **no** es compleix que

*y está claro que **no** se cumple que*

$$\boxed{A_0(n) > 2^{n-1}, \quad \forall n \geq 0.}$$

En el cas $m = 1$,

En el caso $m = 1$,

$$A_1(n) = A(1,n) = \begin{cases} A(0,1) = 2 & \text{si } n = 0 \\ A(0, A(1, n-1)) = A_1(n-1) + 1 & \text{si } n > 0, \end{cases}$$

o siga que $A_1(n) = n+2$. I tampoc no es compleix que

o sea que $A_1(n) = n+2$. Y tampoco se cumple que

$$\boxed{A_1(n) > 2^{n}, \quad \forall n \geq 0.}$$

Si $m = 2$ tenim que

Si $m = 2$ tenemos que

$$A_2(n) = A(2,n) = \begin{cases} A(1,1) = 3 & \text{si } n = 0 \\ A(1, A(2, n-1)) = A_2(n-1) + 2 & \text{si } n > 0, \end{cases}$$

amb la qual cosa $A_2(n) = 2n+3$. I tampoc no es compleix que

con lo que $A_2(n) = 2n+3$. Y tampoco se cumple que

$$\boxed{A_2(n) > 2^{n+1}, \quad \forall n \geq 0.}$$

Si considerem ara $m = 3$ obtenim la recurrència

Si consideramos ahora $m = 3$ obtenemos la recurrencia

$$A_3(n) = A(3,n) = \begin{cases} A(2,1) = 5 & \text{si } n = 0 \\ A(2, A(3, n-1)) = 2A_3(n-1) + 3 & \text{si } n > 0, \end{cases}$$

que, a diferència de les anteriors, no és gens fàcil de calcular. Però en realitat només necessitem saber que la funció $f(n) = 2^{n+2}$ ve determinada per la recurrència

que, a diferencia de las anteriores, no es nada fácil de calcular. Pero en realidad sólo necesitamos saber que la función $f(n) = 2^{n+2}$ viene determinada por la recurrencia

$$f(n) = \begin{cases} 4 & \textbf{si } n = 0 \\ 2f(n-1) & \textbf{si } n > 0, \end{cases}$$

per a concloure fàcilment que

para concluir fácilmente que

$$\boxed{A_3(n) > 2^{n+2} \quad \forall n \geq 0.}$$

Per tant i finalment, podem dir que es compleix que $A(m,n) \geq 2^{m+n-1}$ per a parells de valors (m,n) majors o iguals (segons $\preceq$) que $(3,0)$.

Por tanto y finalmente podemos decir que se cumple que $A(m,n) \geq 2^{m+n-1}$ para pares de valores (m,n) mayores o iguales (según $\preceq$) que $(3,0)$.

En altres paraules, podem enunciar la **base d'inducció** com a

En otras palabras, podemos enunciar la **base de inducción** *como*

BI: $$A(3,0) = 5 > 2^{3-1} = 4,$$

i el que finalment s'ha demostrat com a

y lo que finalmente se ha demostrado como

$$\boxed{A(m,n) > 2^{m+n-1} \quad \forall m \geq 3, \ \forall n \geq 0.}$$

3.3 Problemes resolts i comentats

Problemas resueltos y comentados

Problema 3.1: [billarREC]

Les 15 boles de billar americà a l'inici formen un triangle equilàter de costat 5. I les 3 boles més internes en formen un de costat 2. De la mateixa manera, podríem afegir 21 boles més per a formar un triangle de costat 8. Si anomenem $B(n)$ el nombre de boles en tot el triangle de costat n,

Las 15 bolas de billar americano al inicio forman un triángulo equilátero de lado 5. Y las 3 bolas más internas forman uno de lado 2. De la misma forma, podríamos añadir 21 bolas más para formar un triángulo de lado 8. Si llamamos $B(n)$ al número de bolas en todo el triángulo de lado n,

a) Escriu la relació anterior entre els casos $B(2) = 3$, $B(5) = 15$ i $B(8) = 36$ com una definició recursiva vàlida en general.

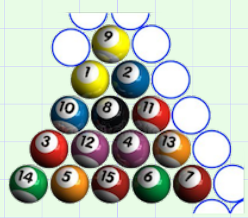
b) Raona quin seria el valor de $B(n)$ per a tot n.

c) Defineix domini, codomini i imatge per a l'aplicació B i digues si és injectiva, exhaustiva i/o bijectiva.

a) escribe la relación anterior entre los casos $B(2) = 3$, $B(5) = 15$ y $B(8) = 36$ como una definición recursiva válida en general.

b) Razona cuál sería el valor de $B(n)$ pera todo n.

c) Define dominio, codominio e imagen para la aplicación B y di si es inyectiva, exhaustiva y/o biyectiva.



a) La relació entre casos consecutius que es comenta és

a) La relación entre casos consecutivos que se comenta es

$$B(5) = B(2) + 12, \qquad B(8) = B(5) + 21.$$

En general, si tenim un triangle de $B(n)$ boles de costat n, el següent triangle que podem formar en afegir una fila de boles que envolte totalment el triangle tindrà costat $n + 3$.

En general, si tenemos un triángulo de $B(n)$ bolas de lado n, el siguiente triángulo que podemos formar añadiendo una fila de bolas que envuelva totalmente el triángulo tendrá lado $n + 3$.

I el nombre de boles afegides serà tres vegades el costat menys tres (ja que les tres boles en els vèrtexs s'hauran comptat dues vegades. En resum,

Y el número de bolas añadidas será tres veces el lado menos tres (ya que las tres bolas en los vértices se habrán contado dos veces. En resumen,

$$B(n+3) = B(n) + 3(n+3) - 3.$$

O alternativament

O alternativamente

$$B(n) = B(n-3) + 3n - 3.$$

Perquè aquesta relació siga una definició recursiva vàlida cal afegir-hi els casos base. En aquest cas, caldran 3 casos base perquè la definició siga vàlida per a tot enter n (major o igual a 0).

Para que esta relación sea una definición recursiva válida hay que añadir los casos base. En este caso, harán falta 3 casos base para que la definición sea válida para todo entero n (mayor o igual a 0).

$$\begin{cases} B(0) = 0,\ B(1) = 1,\ B(2) = 3, & \\ B(n) = B(n-3) + 3n - 3, & \text{si } n \geq 3. \end{cases}$$

b) Podem intentar trobar el valor de $B(n)$ per inspecció. Si apliquem la definició recursiva diverses vegades, arribem a

b) Podemos intentar encontrar el valor de $B(n)$ por inspección. Si aplicamos la definición recursiva varias veces, llegamos a

$$\begin{aligned} B(n) = B(n-3) + 3(n-1) = B(n-6) + 3(n-4) + 3(n-1) = \\ = B(n-9) + 3(n-7) + 3(n-4) + 3(n-1) = \cdots \end{aligned}$$

O en general a

O en general a

$$B(n) = B(n-3k) + 3\sum_{\ell=1}^{k}(n - 3\ell + 2) = B(n-3k) + 3k(n+2) - 3^2\frac{k}{2}(k+1).$$

Ara podríem estudiar en quins casos $B(n-3k)$ arriba a ser un cas base, substituir-ho en l'expressió anterior i intentar arribar a una fórmula general per a $B(n)$.

Ahora podríamos estudiar en qué casos $B(n-3k)$ llega a ser un caso base, sustituirlo en la expresión anterior e intentar llegar a una fórmula general para $B(n)$.

En lloc d'això, deduirem relacions de recurrència equivalents però més senzilles per a $B(n)$.

En lugar de eso, deduciremos relaciones de recurrencia equivalentes pero más sencillas para $B(n)$.

Si tenim un triangle de boles de costat n i li llevem les boles de **dos costats**, i no tres com en la recurrència anterior, podem arribar a la recurrència equivalent,

Si tenemos un triángulo de bolas de lado n y le quitamos las bolas de **dos lados**, *y no tres como en la recurrencia anterior, podemos llegar a la recurrencia equivalente,*

$$\begin{cases} B(0) = 0, \ B(1) = 1, \\ B(n) = B(n-2) + 2n - 1, \quad \text{si } n \geq 2. \end{cases}$$

Alternativament, podem considerar un triangle de boles de costat n i llevar-li les boles de només **un costat**, amb la qual cosa tenim que

Alternativamente, podemos considerar un triángulo de bolas de lado n y quitarle las bolas de sólo **un lado**, *con lo cual tenemos que*

$$\begin{cases} B(0) = 0, \\ B(n) = B(n-1) + n, \quad \text{si } n \geq 1. \end{cases}$$

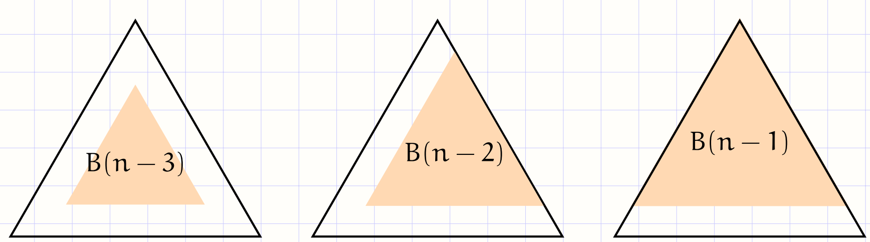
I aquesta última recurrència sí que es correspon trivialment amb el sumatori $\sum_{i=1}^{n} i$, per la qual cosa

Y esta última recurrencia sí que se corresponde trivialmente con el sumatorio $\sum_{i=1}^{n} i$, por lo que

$$B(n) = \frac{n}{2}(n+1).$$

Podem resumir les tres recurrències obtingudes per a $B(n)$ gràficament de la següent manera.

Podemos resumir las tres recurrencias obtenidas para $B(n)$ gráficamente de la siguiente manera.



c) Ara que coneixem els valors que pren B, podem descriure-la com una aplicació de la forma

c) Ahora que conocemos los valores que toma B, podemos describirla como una aplicación de la forma

$$B : \mathbb{Z}^+ \longrightarrow \mathbb{Z}^+,$$

que així definida és injectiva i no exhaustiva. Injectiva perquè per a tot parell de valors del domini, (n, m), tals que $n \neq m$ necessàriament $B(n) \neq B(m)$. I no exhaustiva perquè valors del codomini com 4 o 5 no tenen antiimatge segons B.

que así definida es inyectiva y no exhaustiva. Inyectiva porque para todo par de valores del dominio, (n, m), tales que $n \neq m$ necesariamente $B(n) \neq B(m)$. Y no exhaustiva porque valores del codominio como 4 o 5 no tienen antiimagen según B.

Conseqüentment, B tampoc no és bijectiva.

Por consiguiente, B tampoco es biyectiva.

Finalment, el conjunt imatge seria,

Finalmente, el conjunto imagen sería,

$$\operatorname{Im}(B) = \{m \in \mathbb{Z}^+ \mid \exists n \in \mathbb{Z}^+ \ : \ m = \frac{n}{2}(n+1)\} = \{0, 1, 3, 6, 10, 15, 21, 28, 36, \dots\}$$

Problema 3.2: [duesRecurrencies]

Considera les relacions de recurrència que defineixen les funcions B i C, i demostra per inducció que $B(n) = C(n)$, per a tot $n \geq 0$.

Considera las relaciones de recurrencia que definen las funciones B y C, y demuestra por inducción que $B(n) = C(n)$, para todo $n \geq 0$.

$$\begin{cases} B(0) = 0, \\ B(n) = B(n-1) + n, \quad \text{si } n \geq 1 \end{cases} \qquad \begin{cases} C(0) = 0, \ C(1) = 1, \ C(2) = 3, \\ C(n) = C(n-3) + 3n - 3, \quad \text{si } n \geq 3 \end{cases}$$

Aquestes recurrències són dues de les tres que hem deduït informalment en l'exercici sobre les boles de billar americà (pàgina 149). Però ara es tracta de demostrar formalment per inducció que les dues són exactament equivalents.

Estas recurrencias son dos de las tres que hemos deducido informalmente en el ejercicio sobre las bolas de billar americano (página 149). Pero ahora se trata de demostrar formalmente por inducción que las dos son exactamente equivalentes.

Utilitzarem inducció forta sobre els enters (no negatius) guiats per la recurrència de B. Per tant, tenim com a base d'inducció que

Utilizaremos inducción fuerte sobre los enteros (no negativos) guiados por la recurrencia de B. Por tanto, tenemos como base de inducción que

BI: $$B(0) = 0 = C(0).$$

Ara establim la hipòtesi.

Ahora establecemos la hipotesis.

HI: $$[B(k) = C(k),\ \forall k < n]\,.$$

I a continuació ja podem demostrar el pas d'inducció.

Y a continuación ya podemos demostrar el paso de inducción.

PI:

$$B(n) \overset{(n\geqslant 1)}{=} B(n-1) + n \overset{(n\geqslant 2)}{=} B(n-2) + (n-1) + n \overset{(n\geqslant 3)}{=}$$
$$= B(n-3) + (n-2) + (n-1) + n = B(n-3) + 3n - 3 \overset{(n\geqslant 3)}{=} C(n).$$

Veiem que el pas d'inducció queda demostrat, però només si $n \geq 3$.

Vemos que el paso de inducción queda demostrado, pero sólo si $n \geq 3$.

Aleshores, per a completar la demostració necessitem o bé ampliar la base d'inducció amb els casos $n = 1$ i $n = 2$, o equivalentment comprovar aquests dos casos separadament dins del pas d'inducció.

Entonces, para completar la demostración necesitamos o bien ampliar la base de inducción con los casos $n = 1$ y $n = 2$, o equivalentemente comprobar estos dos casos separadamente dentro del paso de inducción.

En qualsevol cas, tenim que

En cualquier caso, tenemos que

$$B(1) \overset{(def)}{=} B(0) + 1 = 1 = C(1),$$

$$B(2) \overset{(def)}{=} B(1) + 2 = 3 = C(2),$$

amb la qual cosa cloem la demostració.

con lo que finalizamos la demostración.

Problema 3.3: [imparellParell]

Considera la definició recursiva de la funció f i demostra per inducció que $f(n) > \frac{n-1}{2}$. Justifica el tipus d'inducció i raona clarament els diferents passos.

Considera la definición recursiva de la función f y demuestra por inducción que $f(n) > \frac{n-1}{2}$. Justifica el tipo de inducción i razona claramente los diferentes pasos.

$$f(n) = \begin{cases} 1 & \text{si } n = 1, \\ f(n-1)+1 & \text{si } n = 2\ell \ \text{per a algun} \ \underset{\textit{para algún}}{} \ \ell \geq 1, \\ f(n-2)+1 & \text{si no.} \end{cases}$$

Calculem la funció per als primers valors de n i comprovem la desigualtat.

Calculamos la función para los primeros valores de n y comprobamos la desigualdad.

n	$f(n)$		$\frac{n-1}{2}$
1	1	$>$	0
2	2	$>$	$\frac{1}{2}$
3	2	$>$	1
4	3	$>$	$\frac{3}{2}$
5	3	$>$	2

A la vista de la taula podem fixar com a cas base $n = 1$.

A la vista de la tabla podemos fijar como caso base $n = 1$.

BI: $$f(1) = 1 > \frac{1-1}{2} = 0.$$

Com que la recurrència no relaciona sols valors de n consecutius, farem servir inducció forta.

Como la recurrencia no relaciona sólo valores de n consecutivos, usaremos inducción fuerte.

HI: $$\left[f(k) > \frac{k-1}{2},\ \forall k < n\right].$$

I a continuació estudiarem què passa amb el valor n. Però haurem de distingir entre el cas que n siga parell o senar.

Y a continuación estudiaremos qué pasa con el valor de n. Pero tendremos que distinguir entre el caso en que n sea par o impar.

PI:

$$[n = 2\ell] \qquad f(n) = f(n-1) + 1 \overset{\text{(HI)}}{>} \frac{n-2}{2} + 1 = \frac{n}{2} > \frac{n-1}{2}$$

$$[n \neq 2\ell] \qquad f(n) = f(n-2) + 1 \overset{\text{(HI)}}{>} \frac{n-3}{2} + 1 = \frac{n-1}{2}$$

Com que en els dos casos arribem al mateix resultat, podem concloure que es compleix el pas d'inducció en general.

Como en los dos casos llegamos al mismo resultado, podemos concluir que se cumple el paso de inducción en general.

Com a alternativa a la demostració anterior, podem transformar la recurrència original en una altra d'equivalent.

Como alternativa a la demostración anterior, podemos transformar la recurrencia original en otra equivalente.

Primer reescrivim la recurrència,

Primero reescribimos la recurrencia,

$$\begin{cases} f(1) &= 1 \\ f(n) &= f(n-1)+1 \quad \text{si } \exists \ell \geq 1 \ : \ n = 2\ell, \\ f(n) &= f(n-2)+1 \quad \text{si } \exists \ell \geq 1 \ : \ n = 2\ell+1. \end{cases}$$

I a continuació fem la substitució $f(n-1) = f(n-2)+1$ en l'última equació, ja que $n = 2\ell+1$ implica que $n-1 = 2\ell$ per a $\ell \geq 1$.

Y a continuación hacemos la sustitución $f(n-1) = f(n-2)+1$ en la última ecuación, ya que $n = 2\ell+1$ implica que $n-1 = 2\ell$ para $\ell \geq 1$.

$$\begin{cases} f(1) &= 1 \\ f(n) &= f(n-1)+1 \quad \text{si } \exists \ell \geq 1 \ : \ n = 2\ell, \\ f(n) &= f(n-1) \qquad\quad \text{si } \exists \ell \geq 1 \ : \ n = 2\ell+1. \end{cases}$$

Aquesta última recurrència la podem escriure usant la funció $\text{mod}(m,n)$ que dona el residu de la divisió entera entre m i n.

Esta última recurrencia la podemos escribir usando la función $\text{mod}(m,n)$ que da el resto de la división entera entre m y n.

$$\begin{cases} f(1) &= 1 \\ f(n) &= f(n-1) + \text{mod}(n-1, 2) \quad \text{si } n \geq 2, \end{cases}$$

A partir d'aquesta recurrència equivalent es pot fer una demostració per inducció (bàsica) perquè la relació es dona només entre enters consecutius. Això es deixa com a exercici.

A partir de esta recurrencia equivalente se puede hacer una demostración por inducción (básica) porque la relación se da sólo entre enteros consecutivos. Esto se deja como ejercicio.

Alternativament, podem resoldre la recurrència, ja que es correspon trivialment amb el sumatori,

Alternativamente, podemos resolver la recurrencia, ya que se corresponde trivialmente con el sumatorio,

$$f(n) = f(1) + \sum_{i=1}^{n-1} \mathrm{mod}(i, 2) = 1 + \underbrace{1 + 0 + 1 + 0 + \cdots}_{n-1}$$

la solució del qual és

cuya solución es

$$1 + \underbrace{1 + 0 + 1 + 0 + \cdots + 0}_{n-2} + 1 = 2 + \frac{n-2}{2} = 1 + \frac{n}{2},$$

si n és parell (tant el primer com l'últim terme del sumatori són 1), i

si n es par (tanto el primero como el último término del sumatorio son 1), y

$$1 + \underbrace{1 + 0 + 1 + 0 + \cdots + 0}_{n-1} = 1 + \frac{n-1}{2},$$

en cas que n siga senar i per tant exactament la meitat dels termes del sumatori són 1.

en el caso que n sea impar y por tanto exactamente la mitad de los términos del sumatorio son 1.

A partir de l'anterior podem escriure la solució de la recurrència com a

A partir de lo anterior podemos escribir la solución de la recurrencia como

$$f(n) = 1 + \lfloor \frac{n}{2} \rfloor = \lceil \frac{n+1}{2} \rceil,$$

d'on es dedueix trivialment que

de donde se deduce trivialmente que

$$f(n) > \frac{n-1}{2}.$$

Problema 3.4: [teSentit]

Per a cada una de les funcions, f i g, definides més avall, contesta:

a) Si la definició té sentit, calcula els valors de la funció fins $n = 4$ i, si pots, una expressió $\forall n$. Si alguna de les definicions no té sentit, explica clarament per què.

b) Quin tipus d'inducció hauries d'usar per a demostrar coses relacionades amb la funció?

c) Identifica domini, codomini i imatge, i digues si la funció és injectiva, exhaustiva i/o bijectiva.

Para cada una de las funciones, f y g, definidas abajo, contesta:

a) Si la definición tiene sentido, calcula los valores de la función hasta $n = 4$ y, si puedes, una expresión $\forall n$. Si alguna de las definiciones no tiene sentido, explica claramente por qué.

b) ¿Qué tipo de inducción deberías usar para demostrar cosas relacionadas con la función?

c) Identifica dominio e imagen y di si la función es inyectiva, exhaustiva y/o biyectiva.

$$\begin{cases} f(n) = 2f(n+1) - f(n) - 1 & \text{si } n \geq 0 \\ f(0) = 0 \end{cases} \qquad \begin{cases} g(n) = g(n-1) - g(n-2) + 1 & \text{si } n > 0 \\ g(n) = 0 & \text{si no} \end{cases}$$

a) Si entenem el cas recursiu de la primera recurrència com una fórmula per a calcular o "reescriure" el valor de $f(n)$, aleshores aquesta definició recursiva no té sentit perquè estaríem definint el cas n en funció d'ell mateix (definició circular) i d'un cas posterior, $n+1$, la qual cosa donaria lloc a una seqüència infinita de càlculs que no acabarien en cap cas base.

No obstant això, el que ens dóna la definició no és més que una equació. I se'ns diu que és vàlida per a tot $n \geq 0$.

Si ens hi fixem bé, el cas $n = 0$ està cobert tant per a la part recursiva com per al cas base. Si escrivim les dues equacions per a $n = 0$, tenim que

a) Si entendemos el caso recursivo de la primera recurrencia como una fórmula para calcular o "reescribir" el valor de $f(n)$, entonces esta definición recursiva no tiene sentido porque estaríamos definiendo el caso n en función de él mismo (definición circular) y de un caso posterior, $n+1$, lo que daría lugar a una secuencia infinita de cálculos que no acabarían en ningún caso base.

Sin embargo, lo que nos da la definición no es más que una ecuación. Y se nos dice que es válida para todo $n \geq 0$.

Si nos fijamos bien, el caso $n = 0$ está cubierto tanto por la parte recursiva como por el caso base. Si escribimos las dos ecuaciones para $n = 0$, tenemos que

$$\begin{cases} f(0) = 2f(1) - f(0) - 1 & , \\ f(0) = 0 & , \end{cases}$$

la qual cosa significa que

lo que significa que

$$f(1) = \frac{1}{2}.$$

De la mateixa manera podem ara aplicar la mateixa equació als casos $n = 1, 2, \ldots$ i calcular els primers valors de la funció:

De la misma manera podemos ahora aplicar la misma ecuación a los casos $n = 1, 2, \ldots$ y calcular los primeros valores de la función:

n	$f(n)$
0	0
1	$\frac{1}{2}$
2	1
3	$\frac{3}{2}$
4	2

En resum, estem interpretant el cas recursiu com una fórmula per a calcular $f(n+1)$ a partir de $f(n)$. De fet, ho podem reescriure com a

En resumen, estamos interpretando el caso recursivo como una fórmula para calcular $f(n+1)$ a partir de $f(n)$. De hecho, lo podemos reescribir como

$$2f(n+1) = 2f(n) + 1, \text{ si } n \geq 0.$$

Si ara dividim les dues parts de l'equació per 2 i fem un canvi de variable, arribem a la següent relació de recurrència equivalent:

Si ahora dividimos las dos partes de la ecuación por 2 y hacemos un cambio de variable, llegamos a la siguiente relación de recurrencia equivalente:

$$\begin{cases} f(n) = f(n-1) + \frac{1}{2}, & \text{si } n \geq 1, \\ f(0) = 0 & . \end{cases}$$

A partir d'aquesta recurrència es veu que el valor de $f(n)$ és determinat per un sumatori,

A partir de esta recurrencia se ve que el valor de $f(n)$ viene dado por un sumatorio,

$$\boxed{f(n)} = \cancelto{0}{f(0)} + \sum_{i=1}^{n} \frac{1}{2} = \boxed{\frac{n}{2}}.$$

La recurrència que defineix la funció g és una fórmula que ens permet calcular-la per a tot valor de n a partir de dos valors anteriors.

La recurrencia que define la función g es una fórmula que nos permite calcularla para todo valor de n a partir de dos valores anteriores.

L'única cosa que té d'especial és que el cas base inclou el valor zero i **tots els negatius**. De fet, necessitem almenys el valor -1 per a poder calcular la funció a partir de $n = 1$.

Lo único especial es que el caso base se incluye el valor cero y **todos los negativos**. *De hecho, necesitamos al menos el valor -1 para poder calcular la función a partir de $n = 1$.*

Els valors de la funció fins a $n = 5$ es mostren en la taula següent.

Los valores de la función hasta $n = 5$ se muestran en la tabla siguiente.

n	$g(n)$	$\mathrm{mod}(n, 6)$
$\vdots$	0	
-1	0	
0	0	0
1	1	1
2	2	2
3	2	3
4	1	4
5	0	5

Si continuem calculant, veiem que $g(6) = 0$, amb la qual cosa podem deduir que es compleix que

Si continuamos calculando, vemos que $g(6) = 0$, con lo que podemos deducir que se cumple que

$$g(n) = g(n+6), \quad \forall n \geq -1.$$

O en altres paraules, es tracta d'una funció periòdica el període de la qual és 6.

O en otras palabras, se trata de una función periódica cuyo período es 6.

Per a poder escriure una expressió vàlida per a tot $n \geq 0$ podem fer servir la funció $\mathrm{mod}(n, 6)$ que també és periòdica i amb el mateix període. De fet, es compleix que

Para poder escribir una expresión válida para todo $n \geq 0$ podemos usar la función mod$(n, 6)$ que también es periódica y con el mismo período. De hecho, se cumple que

$$g(n) = \operatorname{mod}(n,6) \quad \underset{\textit{siempre que}}{\text{sempre que}} \quad \operatorname{mod}(n,6) \leq 2$$

En canvi quan $3 \leq \operatorname{mod}(n,6) \leq 5$ els valors decreixen i es compleix que

En cambio cuando $3 \leq \operatorname{mod}(n,6) \leq 5$ los valores decrecen y se cumple que

$$g(n) = 5 - \operatorname{mod}(n,6) \quad \underset{\textit{cuando}}{\text{quan}} \quad 3 \leq \operatorname{mod}(n,6) \leq 5$$

Per tant, podem expressar el valor de la funció g per a tot $n \geq 0$ com a

Por tanto, podemos expresar el valor de la función g para todo $n \geq 0$ como

$$g(n) = \begin{cases} 0, & \text{si } n < 0, \\ \operatorname{mod}(n,6), & \text{si } n \geq 0 \wedge 0 \leq \operatorname{mod}(n,6) \leq 2, \\ 5 - \operatorname{mod}(n,6), & \text{si } n \geq 0 \wedge 3 \leq \operatorname{mod}(n,6) \leq 5. \end{cases}$$

O també com a

O también como

$$\boxed{g(n) = \begin{cases} 0, & \text{si } n < 0, \\ \min(\operatorname{mod}(n,6), 5 - \operatorname{mod}(n,6)), & \text{si no.} \end{cases}}$$

b) En el cas de la primera recurrència, com que el que s'estableix és una relació entre el valor de la funció per a dos enters consecutius, es podrà utilitzar el principi d'inducció bàsica.

b) En el caso de la primera recurrencia, como lo que se establece es una relación entre el valor de la función para dos enteros consecutivos, se podrá utilizar el principio de inducción básica.

A títol d'exemple, podem demostrar que la funció que defineix la recurrència és $f(n) = \frac{n}{2}$, de la següent manera.

A título de ejemplo, podemos demostrar que la función que define la recurrencia es $f(n) = \frac{n}{2}$, de la siguiente manera.

BI: $f(0) = 0 = \frac{0}{2}$.

HI: $\left[f(n) = \frac{n}{2}\right]$.

PI:

Escrivim primer el cas recursiu

Escribimos primero el caso recursivo

$$f(n) = 2f(n+1) - f(n) - 1.$$

I ara apliquem la hipòtesi d'inducció,

Y ahora aplicamos la hipótesis de inducción,

$$\frac{n}{2} = 2f(n+1) - \frac{n}{2} - 1,$$

d'on s'obté que

de donde se obtiene que

$$2f(n+1) = n+1,$$

i d'ací el resultat $f(n+1) = \frac{n+1}{2}$ que demostra el pas d'inducció.

y de aquí el resultado $f(n+1) = \frac{n+1}{2}$ que demuestra el paso de inducción.

En el cas de la segona recurrència la relació s'estableix entre un enter i els dos anteriors, per la qual cosa necessitarem utilitzar el principi d'inducció forta.

En el caso de la segunda recurrencia la relación se establece entre un entero y los dos anteriores, por lo que necesitaremos utilizar el principio de inducción fuerte.

Com a exemple, demostrarem per inducció que $g(n) \leq 2$ per a tot $n \geq 0$.

Como ejemplo, demostraremos por inducción que $g(n) \leq 2$ pera todo $n \geq 0$.

Enunciem primer la hipòtesi i posposem la base d'inducció.

Enunciamos primero la hipótesis y posponemos la base de inducción.

HI: $[g(k) \leq 2, \ \forall k < n].$

En el pas d'inducció intentarem desfer-nos dels termes negatius que ens donaran problemes a l'hora d'aplicar la hipòtesi, ja que en aquest cas es tracta d'una fita superior.

En el paso de inducción intentaremos deshacernos de los términos negativos que darán problemas a la hora de aplicar la hipótesis, ya que en este caso se trata de una cota superior.

PI:

$$g(n) \overset{(n\geqslant 1)}{=} g(n-1) - g(n-2) + 1 \overset{(n\geqslant 2)}{=} \cancel{g(n-2)} - g(n-3) - \cancel{g(n-2)} + 2.$$

Hem arribat a una relació directa entre $g(n)$ i $g(n-3)$,

Hemos llegado a una relación directa entre $g(n)$ y $g(n-3)$,

$$g(n) = 2 - g(n-3) \quad \text{si} \quad n \geq 2$$

que podem tornar a aplicar perquè aparega només $g(n-6)$ en positiu i poder aplicar la hipòtesi. Aleshores,

que podemos volver a aplicar para que aparezca sólo $g(n-6)$ en positivo y poder aplicar la hipótesis. Entonces,

$$g(n) \overset{(n\geq 5)}{=} g(n-6) - \cancel{2} + \cancel{2} \overset{(\mathrm{HI})}{\leq} 2,$$

amb la qual cosa conclou el pas d'inducció.

con lo que concluye el paso de inducción.

Com que el pas d'inducció només és cert per a $n \geq 5$, els valors menors o iguals a 4 s'hauran d'incloure en la base d'inducció.

Como el paso de inducción sólo es cierto para $n \geq 5$, los valores menores o iguales a 4 se tendrán que incluir en la base de inducción.

BI:

$$\begin{aligned}
g(n) &= 0 \leq 2 \quad \text{si} \quad n \leq 0 \\
g(1) &= g(0) - g(-1) + 1 = 1 \leq 2 \\
g(2) &= g(1) - g(0) + 1 = 2 \leq 2 \\
g(3) &= g(2) - g(1) + 1 = 2 \leq 2 \\
g(4) &= g(3) - g(2) + 1 = 1 \leq 2
\end{aligned}$$

c) La funció f la podem descriure com a

c) La función f la podemos describir como

$$f : \mathbb{Z}^+ \longrightarrow \mathbb{R},$$

i, segons hem vist, és injectiva perquè

y, según hemos visto, es inyectiva porque

$$f(n) = \frac{n}{2} \neq \frac{m}{2} = f(m) \text{ si } n \neq m.$$

A més a més, és no exhaustiva i, per tant, no bijectiva. El conjunt imatge de f és

Además, es no exhaustiva y, por tanto, no biyectiva. el conjunto imagen de f es

$$\mathrm{Im}(f) = \{x \in \mathbb{R} \mid \exists n \in \mathbb{Z}^+ \; : \; x = \frac{n}{2}\} = \{x \in \mathbb{R} \mid 2x \in \mathbb{Z}^+\}.$$

Quant a la funció g la descrivim com a

En cuanto a la función g la describimos como

$$g : \mathbb{Z} \longrightarrow \{0, 1, 2\},$$

i no és injectiva (ja que, per exemple, $g(0) = g(6)$) i per tant no bijectiva. Però sí que és exhaustiva tal com l'hem definida, ja que els conjunts codomini i imatge de g coincideixen.

y no es inyectiva (ya que, por ejemplo, $g(0) = g(6)$) y por tanto no biyectiva. Pero sí que es exhaustiva tal como la hemos definido, ya que los conjuntos codominio e imagen de g coinciden.

Problema 3.5: [fibonacci]

Considera la definició recursiva de la successió de Fibonacci, $F(n)$, i demostra per inducció que $F(n) \leq n!$.

Considera la definición recursiva de la sucesión de Fibonacci, $F(n)$, y demuestra por inducción que $F(n) \leq n!$.

Com que en la recurrència de Fibonacci, s'estableix una relació entre tres termes consecutius, necessitarem utilitzar inducció forta.

Como en la recurrencia de Fibonacci, se establece una relación entre tres términos consecutivos, necesitaremos utilizar inducción fuerte.

A més a més, pel tipus de relació, necessitarem dos casos base consecutius que complisquen el que es vol demostrar.

Además, por el tipo de relación, necesitaremos dos casos base consecutivos que cumplan lo que se quiere demostrar.

BI:

$$F(0) = 0 \leq 1 = 0!$$
$$F(1) = 1 \leq 1 = 1!$$

La hipòtesi d'inducció serà

La hipótesis de inducción será

HI: $$[F(k) \leq k!, \ \forall k < n]$$

I en el pas d'inducció estudiem què passa per al valor n.

Y en el paso de inducción estudiamos qué pasa para el valor n.

PI:

$$F(n) \overset{(n>1)}{=} F(n-1) + F(n-2) \overset{(HI)}{\leq} (n-1)! + (n-2)! = (n-1+1)(n-2)! < n!$$

Com que hem arribat al resultat correcte per a n fent servir la hipòtesi d'inducció, podem assegurar que la desigualtat és certa per a tot n.

Como hemos llegado al resultado correcto para n usando la hipótesis de inducción, podemos asegurar que la desigualdad es cierta para todo n.

Problema 3.6: [pseudoFibonacci]

Considera la definició recursiva de la successió de Fibonacci, $F(n)$, i la definició recursiva de la funció $G(n)$. És possible expressar $G(n)$ en funció de $F(n)$ de manera no recursiva? Com? Raona la resposta.

Considera la definición recursiva de la sucesión de Fibonacci, $F(n)$, y la definición recursiva de la función $G(n)$. ¿Es posible expresar $G(n)$ en función de $F(n)$ de manera no recursiva? ¿Cómo? Razona la respuesta.

$$G(n) = \begin{cases} n & \text{si } 0 \leq n \leq 1 \\ G(n-1) + G(n-2) + 1 & \text{si } n \geq 2 \end{cases}$$

Calcularem els primers valors de les dues funcions per a il·lustrar-ne el comportament.

Calcularemos los primeros valores de las dos funciones para ilustrar el comportamiento.

n	$F(n)$	$G(n)$
0	0	0
1	1	1
2	1	2
3	2	4
4	3	7
5	5	12

Ara podríem cercar alguna relació entre els valors de les dues funcions, per a proposar una fórmula i demostrar-la per inducció.

Ahora podríamos buscar alguna relación entre los valores de las dos funciones, para proponer una fórmula y demostrarla por inducción.

En lloc d'això raonarem a partir de les dues definicions recursives. Escriurem primer la definició recursiva de G per als valors n i $n-1$,

En lugar de esto razonaremos a partir de las dos definiciones recursivas. Escribiremos primero la definición recursiva de G para los valores n y $n-1$,

$$G(n) = G(n-1) + G(n-2) + 1,$$
$$G(n-1) = G(n-2) + G(n-3) + 1.$$

Si ara restem les dues equacions anteriors, podem arribar a una nova equació que haurà de ser vàlida per a tot $n \geq 3$.

Si ahora restamos las dos ecuaciones anteriores, podemos llegar a una nueva ecuación que tendrá que ser válida para todo $n \geq 3$.

$$G(n) - G(n-1) = \underbrace{G(n-1) - G(n-2)} + \underbrace{G(n-2) - G(n-3)}.$$

Podem veure que l'equació anterior encara es pot simplificar més. Però no ho hem fet amb tota la intenció, perquè ara definirem una nova funció, H, com a

Podemos ver que la ecuación anterior aún se puede simplificar más. Pero no lo hemos hecho con toda la intención, porque ahora definiremos una nueva función, H, como

$$H(n) = G(n) - G(n-1), \ \forall n \geq 1$$

Independentment del(s) cas(os) base de la funció H, s'haurà de complir que

Independientemente del o de los casos base de la función H, se tendrá que cumplir que

$$H(n) = H(n-1) + H(n-2), \ \forall n \geq 3.$$

Aquesta relació és exactament la mateixa que la que defineix F llevat dels casos base. Però es compleix que

Esta relación es exactamente la misma que la que define F excepto los casos base. Pero se cumple que

$$H(2) = G(2) - G(1) = 1 = F(2)$$
$$H(1) = G(1) - G(0) = 1 = F(1)$$

i podem fixar $H(0) = 0 = F(0)$ en la definició de H.

y podemos fijar $H(0) = 0 = F(0)$ en la definición de H.

Com a conseqüència, podem assegurar que es compleix que

Como consecuencia, podemos asegurar que se cumple que

$$F(n) = G(n) - G(n-1), \ \forall n \geq 1.$$

Com veiem, hem expressat F en funció de G (Tot i que caldria afegir que $F(0) = G(0)$). Però l'enunciat de l'exercici ens demana exactament el contrari.

Como vemos, hemos expresado F en función de G (Aunque habría que añadir que $F(0) = G(0)$). Pero el enunciado del ejercicio nos pide exactamente lo contrario.

Podem aleshores escriure la relació anterior, junt amb el fet que $G(0) = 0$, alternativament com a

Podemos entonces escribir la relación anterior, junto con el hecho de que $G(0) = 0$, alternativamente como

$$G(n) = \begin{cases} 0 & \text{si } n = 0, \\ G(n-1) + F(n) & \text{si } n \geq 1, \end{cases}$$

la qual cosa no és més que una definició recursiva per a G en funció de F la solució de la qual és

lo cual no es más que una definición recursiva para G en función de F cuya solución es

$$\boxed{G(n) = \sum_{i=1}^{n} F(i), \ \forall n \geq 0,}$$

que és la definició de G en funció de F que se'ns demanava.

que es la definición de G en función de F que se nos pedía.

En el cas que haguérem arribat a l'expressió anterior per inspecció a partir de la primera taula de valors, caldria aleshores demostrar el resultat per inducció, tal i com es resumeix a continuació.

En el caso de que hubiésemos llegado a la expresión anterior por inspección a partir de la primera tabla de valores, habría entonces que demostrar el resultado por inducción, tal como se resume a continuación.

BI: $G(0) = 0 = \sum_{i=1}^{0} F(i)$, $G(1) = 1 = \sum_{i=1}^{1} F(i) = F(1)$.

HI: $$\left[G(k) = \sum_{i=1}^{k} F(i),\ \forall k < n\right].$$

PI:

$$G(n) \overset{(n\leqslant 2)}{=} G(n-1) + G(n-2) + 1 \overset{(HI)}{=} \sum_{i=1}^{n-1} F(i) + \sum_{i=1}^{n-2} F(i) + 1 =$$

$$= \sum_{j=2}^{n} F(j-1) + \sum_{j=3}^{n} F(j-2) + 1 = \left(F(1) + \sum_{j=3}^{n} F(j)\right) + \underbrace{1}_{F(2)} = \sum_{j=1}^{n} F(j).$$

Problema 3.7: [sumaGeometrica]

Demostra per inducció:

Demuestra por inducción:

$$S(n) = \sum_{i=0}^{n} \frac{1}{2^i} = 1 + \frac{1}{2} + \frac{1}{4} + \cdots < 2.$$

Normalment quan es tracta de demostrar coses per inducció relacionades amb un sumatori se sol descompondre aquest deixant de banda l'últim terme del sumatori.

Normalmente cuando se trata de demostrar cosas por inducción relacionadas con un sumatorio se suele descomponer éste dejando aparte el último término del sumatorio.

En aquest cas concret tindríem

En este caso concreto tendríamos

$$S(n) = S(n-1) + \frac{1}{2^n}.$$

Però ocorre que si intentem aplicar la corresponent hipòtesi d'inducció en aquest cas concret arribaríem a

Pero ocurre que si intentamos aplicar la correspondiente hipótesis de inducción en este casos concreto llegaríamos a

$$S(n) = S(n-1) + \frac{1}{2^n} \overset{(HI)}{<} 2 + \frac{1}{2^n}.$$

I resulta que la quantitat per la qual podem fitar el valor de $S(n)$ fent servir la **HI** és estrictament superior a 2 per a tot n.

*Y resulta que la cantidad por la que podemos acotar el valor de $S(n)$ usando la **HI** es estrictamente superior a 2 para todo n.*

La conclusió és que la relació recursiva amb què hem caracteritzat el nostre sumatori no ens permet demostrar aquest resultat directament mitjançant inducció.

La conclusión es que la relación recursiva con la que hemos caracterizado nuestro sumatorio no nos permite demostrar este resultado directamente mediante inducción.

Una alternativa podria consistir a demostrar algun resultat més general a partir del qual es puga deduir el nostre, com per exemple

Un alternativa podría consistir en demostrar algún resultado más general a partir del que se pueda deducir el nuestro, como por ejemplo

$$S(n) = 2 - \frac{1}{2^n} < 2.$$

Una altra alternativa més interessant consisteix a obtenir una caracterització recursiva diferent del sumatori. En el nostre cas tenim que, si $n \geq 1$,

Otra alternativa más interesante consiste en obtener una caracterización recursiva diferente del sumatorio. En nuestro caso tenemos que, si $n \geq 1$,

$$S(n) = 1 + \frac{1}{2} + \frac{1}{4} + \cdots \frac{1}{2^n} = 1 + \frac{1}{2}\left(1 + \frac{1}{2} + \cdots \frac{1}{2^{n-1}}\right) = 1 + \frac{1}{2}S(n-1).$$

Amb aquesta caracterització alternativa la inducció bàsica ens serveix perfectament per a demostrar el resultat com es mostra a continuació.

Con esta caracterización alternativa la inducción básica nos sirve perfectamente para demostrar el resultado como se muestra a continuación

BI: $S(0) = 1 < 2$,

HI: $[S(n) < 2]$,

PI: $S(n+1) = 1 + \frac{1}{2}S(n) \overset{(HI)}{<} 1 + \frac{2}{2} = 2.$

3.4 Problemes proposats

Problemas Propuestos

Problema 3.8: [exponenciacio]

Considera la funció $G(n)$ definida més avall. Demostra per inducció que $G(n) \geq n!$

Considera la función $G(n)$ definida más abajo. Demuestra por inducción que $G(n) \geq n!$

$$G(n) = (((2^{\overbrace{2)^2)^{\cdots})^2}^{n-1}}, \qquad G(3) = (2^2)^2 = 16, \quad G(4) = ((2^2)^2)^2 = 256.$$

Problema 3.9: [binomials]

Considera la següent propietat dels coeficients binomials

Considera la siguiente propiedad de los coeficientes binomiales

$$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1}.$$

a) Digues quin(s) cas(os) base cal afegir perquè tinga sentit com a definició recursiva.

a) Di qué caso(s) base(s) hay que añadir para que tenga sentido como definición recursiva.

b) Anomena la funció com a F i defineix-la formalment explicitant els conjunts domini, codomini i imatge.

b) Nombra la función como F y defínela formalmente explicitando los conjuntos dominio, codominio e imagen.

c) Digues si és injectiva, exhaustiva i/o bijectiva.

c) Di si es inyectiva, exhaustiva y/o biyectiva.

d) Demostra per inducció que el valor de la funció (a partir de la definició recursiva) és igual a

d) Demuestra por inducción que el valor de la función (a partir de la definición recursiva) es igual a

$$\frac{n!}{m!(n-m)!}.$$

Problema 3.10: [induccioMultiple]

A partir de la definició recursiva de la funció f_n,

a) Calcula els seus valors fins a $n = 4$.

b) Demostra per inducció que $f_n > n!$

c) Demostra per inducció que $f_n < (n+1)!$

A partir de la definición recursiva de la función f_n,

a) Calcula sus valores hasta $n = 4$.

b) Demuestra por inducción que $f_n > n!$

c) Demuestra por inducción que $f_n < (n+1)!$

$$f_n = \begin{cases} \sum_{i=1}^{n} i f_{i-1} & \text{si} \quad n > 0 \\ 1 & \text{si no} \end{cases}$$

Problema 3.11: [multiinduccio]

Donada la definició recursiva de N_n,

Dada la definición recursiva de N_n,

$$N_n = \begin{cases} 1 & \text{si } n = 0 \\ \sum_{k=0}^{n-1} N_{n-k-1} N_k & \text{si } n > 0 \end{cases}$$

a) demostra per inducció que

a) demuestra por inducción que

$$N_n \geq 2^{n-1}.$$

b) ¿Es podria demostrar que

b) ¿Se podría demostrar que

$$N_n \geq 2^n?$$

Explica què i com hauries de canviar en la primera demostració o explica per què no es pot.

Explica cómo y qué deberías cambiar en la primera demostració o explica por qué no se puede.

4. Grafs i arbres

Grafos y árboles. 4

4.1 Grafs: definicions i propietats

Grafos: definiciones y propiedades

Un **graf** és un parell de conjunts, (V, A), de manera que

a) el conjunt de **nodes** o **vèrtexs**, V, és un conjunt d'elements arbitrari, i

b) el conjunt d'**arcs** o **arestes**, A, és una **relació binària** entre elements de V.

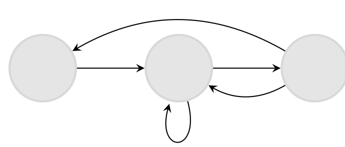
*Un **grafo** es un par de conjuntos, (V, A), de manera que*

*a) el conjunto de **nodos** o **vértices**, V, es un conjunto de elementos arbitrario, y*

*b) el conjunto de **arcos** o **aristas**, A, es una **relación binaria** entre elementos de V.*

Sovint representem els nodes com a punts o cercles i els arcs com a fletxes que uneixen parells de nodes.

A menudo representamos los nodos como puntos o círculos y los arcos como flechas que unen pares de nodos.



Un **graf no dirigit** (moltes vegades **graf**, simplement) és un graf el conjunt de nodes del qual és **no buit** i on el conjunt d'arcs és format per **parells no ordenats** de nodes. És a dir,

*Un **grafo no dirigido** (muchas veces simplemente **grafo**) es un grafo cuyo conjunto de nodos es no vacío y donde el conjunto de arcos está formado por **pares no ordenados** de nodos. Es decir,*

$$A \subseteq \{\{u, v\} \ : \ u, v \in V\}.$$

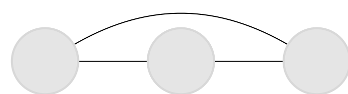
Alternativament es pot dir que en un graf no dirigit, A és una **relació binària simètrica** i **antireflexiva** ($a \not R a$, $\forall a$) en V. És a dir,

Alternativamente se puede decir que en un grafo no dirigido, A es una **relación binaria simétrica** *y* **antireflexiva** *($a \not R a$, $\forall a$) en V. Es decir,*

$$((u,v) \in A \Leftrightarrow (v,u) \in A) \ \wedge \ ((u,v) \in A \Rightarrow u \neq v)\,.$$

Quan un graf és no dirigit, se'l sol representar mitjançant cercles units per línies.

Cuando un grafo es no dirigido, se le suele representar mediante círculos unidos por líneas.



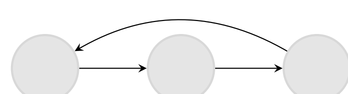
Un **graf dirigit** o també **digraf** és un graf el conjunt de nodes del qual és **no buit** i el conjunt d'arcs del qual és format per **parells ordenats** de nodes **diferents**. És a dir,

Un **grafo dirigido** *o también* **digrafo** *es un grafo cuyo conjunto de nodos es no vacío y cuyo conjunto de arcos está formado por* **pares ordenados** *de nodos* **diferentes**. *Es decir,*

$$A \subseteq \{(u,v) \in V \times V \ : \ u \neq v\}.$$

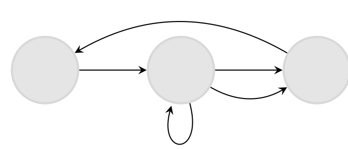
També es pot dir que en un graf dirigit, A és una **relació binària antireflexiva**. Els grafs dirigits es representen mitjançant cercles units per fletxes i, per definició, no poden contenir ni **arcs paral·lels** (arcs entre un mateix parell ordenat de nodes) ni **bucles** (arcs entre un node i ell mateix).

También se puede decir que en un grafo dirigido, A es una **relación binaria antireflexiva**. *Los grafos dirigidos se representan mediante círculos unidos por flechas y, por definición, o pueden contener ni* **arcos paralelos** *(arcos entre un mismo par ordenado de nodos) ni* **bucles** *(arcos entre un nodo y él mismo).*



Un **multigraf** o **graf amb arcs paral·lels** és un graf, (V, A), (del tipus que siga) on A és un **multiconjunt** en lloc d'un conjunt d'arcs.

Un **multigrafo** *o* **grafo con arcos paralelos** *es un grafo, (V, A), (del tipo que sea) donde A es un* **multiconjunto** *en lugar de un conjunto de arcos.*



Un **graf amb bucles** és un graf, (V, A), (del tipus que siga) on admetem relacions entre un node i ell mateix com a elements de A.

Un **grafo con bucles** *es un grafo, (V, A), (del tipo que sea) donde admitimos relaciones entre un nodo y él mismo como elementos de A.*

En alguns casos podrem definir el que anomenarem **graf nul**, o **graf buit**, $\emptyset = (\emptyset, \emptyset)$, que és un graf especial que no conté cap node i, per tant, cap arc.

En algunos casos podremos definir lo que llamaremos **grafo nulo**, *$\emptyset = (\emptyset, \emptyset)$, que es un grafo especial que no contiene ningún nodo y, por tanto, ningún arco.*

Un **graf buit de n nodes** o **d'ordre n**, $\emptyset_n = \emptyset_V = (V, \emptyset)$, és un graf el conjunt de nodes del qual, V, té n nodes i no conté cap arc. Es té que

Un **grafo vacío de n nodos** *o* **de orden n**, *$\emptyset_n = \emptyset_V = (V, \emptyset)$, es un grafo cuyo conjunto de nodos, V, tiene n nodos y no contiene ningún arco. Se tiene que*

$$\emptyset = \emptyset_0 = \emptyset_\emptyset.$$

Dos nodes són **adjacents** si existeix un arc entre ells. Dos arcs són **adjacents** si tenen un node en comú.

Dos nodos son **adyacentes** *si existe un arco entre ellos. Dos arcos son* **adyacentes** *si tienen un nodo en común.*

Diem que un arc **incideix** o és **incident** en cada un dels nodes que el formen.

Decimos que un arco **incide** *o es* **incidente** *en cada uno de los nodos que lo forman.*

El **grau** d'un node u, $\text{gr}(u)$, és el nombre d'arcs que incideixen en u. En els grafs dirigits es pot distingir entre **grau d'entrada** o **igrau**, $\text{gi}(u)$ (nombre d'arcs que arriben a u), i **grau d'eixida** o **ograu**, $\text{go}(u)$ (nombre d'arcs que ixen de u).

El **grado** *de un nodo u, $\text{gr}(u)$, es el número de arcos que inciden en u. En los grafos dirigidos se puede distinguir entre* **grado de entrada** *o* **igrado**, *$\text{gi}(u)$ (número de arcos que llegan a u), y* **grado de salida** *o* **ogrado**, *$\text{go}(u)$ (número de arcos que salen de u).*

Proposició

$$\boxed{\forall G = (V, A),\ \exists k \in \mathbb{Z}^+\ :\ \sum_{u \in V} \text{gr}(u) = 2k}$$

En altres paraules, la suma dels graus dels nodes de qualsevol graf és parell.

En otras palabras, la suma de los grados de los nodos de cualquier grafo es par.

La demostració és evident, ja que cada arc contribueix en 1 al grau de cada un dels dos nodes als quals incideix. Per tant, quan sumem per a tots els nodes tenim que

La demostración es evidente ya que cada arco contribuye en 1 al grado de cada uno de los dos nodes a los que incide. Por tanto, cuando sumamos para todos los nodos tenemos que

$$\sum_{u \in V} \mathrm{gr}(u) = 2 \cdot |A|.$$

Corol·lari

En tot graf hi ha d'haver necessàriament un nombre **parell** de **nodes amb grau imparell**.

*En todo grafo tiene que haber necesariamente un número **par** de **nodos con grado impar**.*

4.1.1 Connexió i descomposició

Conexión y descomposición

Diem que un graf, $G' = (V', A')$, és **subgraf** d'un altre, $G = (V, A)$, i ho escrivim com a $G' \subseteq G$, si

*Decimos que un grafo, $G' = (V', A')$, es **subgrafo** de otro, $G = (V, A)$, y lo escribimos como $G' \subseteq G$, si*

$$V' \subseteq V \wedge A' \subseteq A.$$

Donat un graf, $G = (V, A)$, i un subconjunt, $V' \subseteq V$, anomenem **subgraf induït per V'** el subgraf (V', A') on

*Dado un grafo, $G = (V, A)$, y un subconjunto, $V' \subseteq V$, llamamos **subgrafo inducido por V'** al subgrafo (V', A') donde*

$$A' = \{(u, v) \in A \ : \ u, v \in V'\}.$$

És a dir, és aquell subgraf de G que conté **tots** els arcs de G que enllacen nodes de V'. De vegades, si G' té $m \leq |V|$ nodes, direm que és un **subgraf d'ordre m** de G.

*Es decir, es aquel subgrafo de G que contiene **todos** los arcos de G que enlazan nodos de V'. A veces, si G' tiene $m \leq |V|$ nodos diremos que es un **subgrafo de orden m** de G.*

Donats dos grafs, $G_1 = (V_1, A_1)$ i $G_2 = (V_2, A_2)$, definim la seua **unió** com a

*Dados dos grafos, $G_1 = (V_1, A_1)$ y $G_2 = (V_2, A_2)$, definimos su **unión** como*

$$G_1 \cup G_2 = (V_1 \cup V_2, A_1 \cup A_2).$$

Diem que un graf, $G = (V, A)$, és un **graf complet** o que és un **clique**, si hi ha un arc en A per a tot parell de nodes en V.

Decimos que un grafo, $G = (V, A)$, es un **grafo completo** *o que es un* **clique**, *si hay un arco en A para todo par de nodos en V.*

Un graf, $G = (V, A)$, és **graf bipartit** respecte d'una bipartició (U, W) de V si tots els seus arcs relacionen nodes de U amb nodes de W o al revés. Equivalentment, el seu conjunt d'arcs ha de complir

Un grafo, $G = (V, A)$, es **bipartido** *respecto a una bipartición (U, W) de V si todos sus arcos relacionan nodos de U con nodo nodos de W o al revés. Equivalentemente, su conjunto de arcos tiene que cumplir*

$$A = \{(u, v) \ : \ u \in U \Leftrightarrow v \in W\}.$$

Donat un graf, $G = (V, A)$, anomenem **recorregut** tota successió de nodes i/o arcs adjacents que podem representar com a

Dado un grafo, $G = (V, A)$, llamamos **recorrido** *a toda sucesión de nodos y/o arcos adyacentes que podemos representar como*

$$(x_1, (x_1, x_2), x_2, (x_2, x_3), x_3, \ldots, x_{n-1}, (x_{n-1}, x_n), x_n),$$
$$((x_1, x_2), (x_2, x_3), \ldots, (x_{n-1}, x_n)),$$
$$(x_1, x_2, x_3, \ldots, x_{n-1}),$$

sempre que

siempre que

$$\forall i, \ x_i \in V, \ \forall i < n, \ (x_i, x_{i+1}) \in A.$$

Diem que el recorregut comença en x_1 i acaba en x_n. Un exemple de recorregut que comença en v_2 i acaba en v_4 per al graf de la figura de la pàgina 176 seria

Decimos que el recorrido comienza en x_1 y acaba en x_n. Un ejemplo de recorrido que comienza en v_2 y acaba en v_4 para el grafo de la figura de la página 176 sería

$$(v_2, v_3, v_1, v_2, v_4), \quad \text{o també} \quad (a_2, a_4, a_1, a_3).$$

o también

Un **camí** és un recorregut on tots els nodes que conté són **diferents**. Un exemple de camí per al graf de la mateixa figura seria

Un **camino** *es un recorrido donde todos los nodos que contiene son diferentes. Un ejemplo de camino para el grafo de la misma figura sería*

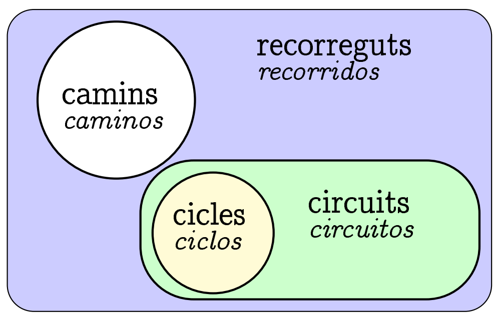
$$(v_3, v_1, v_2, v_4) \underset{\textit{o también}}{\text{o també}} (a_4, a_1, a_3)$$

Un **recorregut circular** o **circuit** és un recorregut que comença i acaba en un mateix node. Un **cicle** és un circuit on l'únic node que es repeteix és l'inicial.

Un **recorrido circular** *o* **circuito** *es un recorrido que comienza y acaba en un mismo nodo. Un* **ciclo** *es un circuito donde el único nodo que se repite es el inicial.*

La relació entre els diferents tipus de recorreguts es mostra gràficament en la figura.

La relación entre los distintos tipos de recorridos se muestra gráficamente en la figura.



Exemples de circuit i cicle per al graf de la figura que es mostra després, són, respectivament,

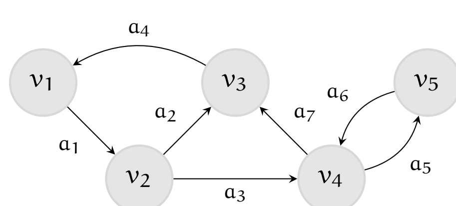
Ejemplos de circuito y ciclo para el grafo de la figura que se muestra después, vendrían dados respectivamente por

$$(v_2, v_3, v_1, v_2, v_4, v_3), \underset{\textit{o también}}{\text{o també}} (a_2, a_4, a_1, a_3, a_7),$$

i

y por

$$(v_3, v_1, v_2, v_4, v_3), \underset{\textit{o también}}{\text{o també}} (a_4, a_1, a_3, a_7).$$



Un recorregut/circuit en un graf $G = (V, A)$, s'anomena **eulerià** si recorre tots els **arcs** en A però només una vegada.

Un recorrido/circuito en un grafo $G = (V, A)$, *se denomina* **Euleriano** *si recorre todos los* **arcos** *en* A *pero sólo una vez.*

Un recorregut (circuit) en un graf $G = (V, A)$, s'anomena **camí (cicle) hamiltonià** si recorre tots els **nodes** en V però només una vegada (llevat del primer/últim en el cas de circuits).

Un recorrido (circuito) en un grafo $G = (V, A)$, se denomina **camino (ciclo) Hamiltoniano** *si recorre todos los* **nodos** *en V pero sólo una vez (excepto el primero/último en el caso de circuitos).*

4.1.2 Representació

Representación

Donat un graf, $G = (V, A)$, i $V = \{u_1, \ldots, u_n\}$, es defineix la seua **matriu d'adjacència** com una matriu, W, cada un dels coeficients de la qual val

Dado un grafo, $G = (V, A)$, y $V = \{u_1, \ldots, u_n\}$, se define su **matriz de adyacencia** *como una matriz, W, cuyos coeficientes valen*

$$w_{ij} = \begin{cases} 1 & \text{si} \quad (u_i, u_j) \in A, \\ 0 & \text{si no.} \end{cases}$$

La matriu d'adjacència d'un graf no dirigit és simètrica. I els valors de la diagonal són zero si parlem de grafs sense bucles.

La matriz de adyacencia de un grafo no dirigido es simétrica. Y los valores de la diagonal son cero si hablamos de grafos sin bucles.

Per exemple, la matriu d'adjacència del graf de l'última figura seria:

Por ejemplo, la matriz de adyacencia del grafo de la última figura sería:

$$W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

4.1.3 Connexió

Conexión

Un graf s'anomena **graf acíclic** o **sense cicles** si no conté cap cicle d'un o més arcs.

Un grafo se llama **grafo acíclico** *o* **sin ciclos** *si no contiene ningún ciclo de uno o más arcos.*

Un graf $G = (V, A)$ és **connex** si entre qualsevol parell dels seus nodes hi ha un **camí** que els uneix. És a dir, si

Un grafo $G = (V, A)$ es **conexo** *si entre cualquier par de sus nodos hay un* **camino** *que los une. Es decir, si*

$$\forall u, v \in V,\ \exists v_1, \ldots, v_k \in V,\ k \geq 0\ :\ C_{u,v} = (u, v_1, \ldots, v_k, v),$$

i $C_{u,v}$ és un camí en G.

y $C_{u,v}$ es un camino en G.

Un graf connex i acíclic s'anomena **arbre**.

Un grafo conexo y acíclico se denomina **árbol**.

Teorema (d'Euler)

Un graf (no dirigit) connex conté un **circuit** eulerià sii tots els seus nodes tenen grau parell.

Un grafo (no dirigido) conexo contiene un **circuito** *Euleriano* sii *todos sus nodos tienen grado par.*

Corol·lari

Un graf (no dirigit) connex conté un **recorregut no circular** eulerià sii només dos dels seus nodes tenen grau imparell.

Un grafo (no dirigido) conexo contiene un **recorrido no circular** *Euleriano* sii *sólo dos de sus nodos tienen grado impar.*

En el graf de l'últim exemple, el recorregut

En el grafo del último ejemplo, el recorrido

$$(v_2, v_3, v_1, v_3, v_4, v_5, v_4, v_3),$$

és eulerià. Sabem que no hi pot haver un circuit eulerià perquè els nodes v_2 i v_3 tenen grau 3.

es Euleriano. Sabemos que no puede haber un circuito Euleriano porque los nodos v_2 y v_3 tienen grado 3.

En el mateix graf, el recorregut

En el mismo grafo, el recorrido

$$(v_5, v_4, v_3, v_1, v_2),$$

és un camí hamiltonià i no existeix cap cicle hamiltonià. En canvi, sí que es pot trobar un cicle hamiltonià en el subgraf induït per $\{v_1, v_2, v_3, v_4\}$, per exemple:

es un camino Hamiltoniano y no existe ningún ciclo Hamiltoniano. En cambio si que se puede encontrar un ciclo Hamiltoniano en el subgrafo inducido por $\{v_1, v_2, v_3, v_4\}$, por ejemplo

$$(v_4, v_3, v_1, v_2, v_4).$$

4.1.4 Potències i clausures

Potencias y clausuras

La **potència n-èsima d'un graf** $G = (V, A)$, és un graf, $G^n = (V, A^n)$, on

La **potencia n-ésima de un grafo** $G = (V, A)$, *es un grafo,* $G^n = (V, A^n)$, *donde*

$$(u, v) \in A^n \Leftrightarrow \exists v_1, \dots, v_k \in V,\ k \geq n - 1\ :\ C_{u,v} = (u, v_1, \dots, v_k, v),$$

sent $C_{u,v}$ un camí en G. En altres paraules, els arcs de G^n són camins en G de longitud n.

siendo $C_{u,v}$ *un camino en* G. *En otras palabras, los arcos de* G^n *son* caminos *en* G *de longitud* n.

Per a qualsevol relació binària, R, R^n és la **potència n-èsima de la relació** i es defineix com a

Para cualquier relación binaria, R, R^n *es la* **potencia n-ésima de la relación** *y se define como*

$$aR^nb \Leftrightarrow \exists x_1, \dots, x_{n-1}\ :\ a = x_0, b = x_n, \forall 0 \leq i < n,\ x_i R x_{i+1}.$$

Es compleix que

Se cumple que

$$G^0 = \emptyset_V, \qquad G^1 = G.$$

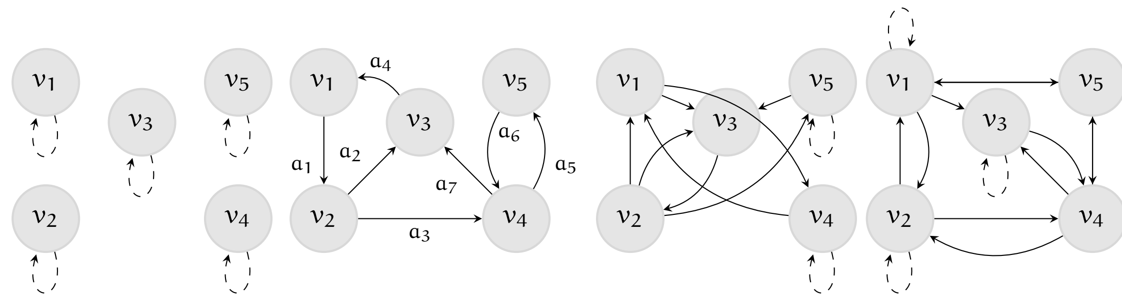
La **clausura transitiva** o de vegades simplement **clausura d'un graf** és el graf $G^+ = (V, A^+)$, on

La **clausura transitiva** *o a veces simplemente* **clausura de un grafo** *es el grafo* $G^+ = (V, A^+)$, *donde*

$$A^+ = \bigcup_{n \geq 1} A^n.$$

Per a qualsevol relació binària, R, R^+ és la **clausura transitiva de la relació** que es defineix també com la menor relació transitiva que conté R.

Para cualquier relación binaria, R, R^+ *es la* **clausura transitiva de la relación** *que se define también como la menor relación transitiva que contiene* R.



G^0 G^1 G^2 G^3

En la figura es mostren les primeres potències del graf de la figura anterior (que coincideix amb G_1). En alguns d'aquests grafs es mostren bucles, ja que en aquests casos poden tenir interès. Per exemple, el bucle sobre v_3 en G_3 significa que hi ha un **cicle** que comença i acaba en v_3 de longitud 3, que és

En la figura se muestran las primeras potencias del grafo de la figura anterior (que coincide con G_1). En algunos de estos grafos se muestran bucles ya que en estos casos pueden tener interés. Por ejemplo, el bucle sobre v_3 en G_3 significa que hay un **ciclo** *que comienza y acaba en v_3 de longitud 3, que es*

$$(v_3, v_1, v_2, v_3).$$

La clausura transitiva del graf anterior es correspon amb un graf complet, ja que es tracta d'un graf connex.

La clausura transitiva del grafo anterior se corresponde con un grafo completo ya que se trata de un grafo conexo.

Proposició

La clausura transitiva de tot graf **connex** és un graf **complet**.

La clausura transitiva de todo grafo **conexo** *es un grafo* **completo**.

A partir de la definició de connex, és evident que hi ha d'haver un camí de longitud k, $1 \leq k \leq |V|$ entre tot parell de nodes. I aquest camí apareixerà com un arc en la corresponent potència del graf.

A partir de la definición de conexo, es evidente que tiene que haber un camino de longitud k, $1 \leq k \leq |V|$ entre todo par de notos. Y este camino aparecerá como un arco en la correspondiente potencia del grafo.

En el cas que considerem grafs amb bucles, o relacions binàries en general, té sentit definir la **clausura reflexiva**. En el cas d'una relació, R, es defineix com la relació reflexiva més petita que la conté. O equivalentment, la mateixa relació però on a més a més tot element del domini està relacionat amb ell mateix. O siga,

En el caso de que consideremos grafos con bucles, o relaciones binarias en general, tiene sentido definir la **clausura reflexiva**. *En el caso de una relación, R, se define como la relación reflexiva más pequeña que la contiene. O equivalentemente, la misma relación pero donde además todo elemento del dominio está relacionado consigo mismo. O sea,*

$$R^0 \cup R.$$

En el cas de grafs, això es correspon amb afegir bucles en tots els nodes (que no els tingueren ja).

En el caso de grafos, esto se corresponde con añadir bucles en todos los nodos (que no los tuviesen ya).

La **clausura transitiva i reflexiva** d'una relació, R, és la relació transitiva i reflexiva més petita que la conté i és equivalent a

La **clausura transitiva y reflexiva** *de una relación, R, es la relación transitiva y reflexiva más pequeña que la contiene y es equivalente a*

$$R^* = \bigcup_{i \geq 0} R^i.$$

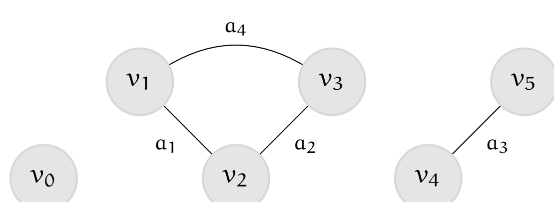
Donat un graf $G = (V, A)$, anomenem **component connexa** de G qualsevol subgraf **connex** de G que siga **maximal** respecte de la relació d'ordre induïda per la inclusió entre grafs.

Dado un grafo $G = (V, A)$, llamamos **componente conexa** *de G a cualquier subgrafo* **conexo** *de G que sea* **maximal** *respecto a la relación de orden inducida por la inclusión entre grafos.*

En la figura es mostra un graf d'ordre 6 que té 3 components connexes que són determinades pels subgrafs induïts per

En la figura se muestra un grafo de orden 6 que tiene 3 componentes conexas que vienen dadas por los subgrafos inducidos por

$$\{v_0\}, \quad \{v_1, v_2, v_3\}, \quad \{v_4, v_5\}.$$



Proposició	***Proposición***
Tot graf no dirigit, $G = (V, A)$, té almenys $\lvert V\rvert - \lvert A\rvert$ components connexes.	*Todo grafo no dirigido, $G = (V, A)$, tiene al menos $\lvert V\rvert-\lvert A\rvert$ componentes conexas.*

La demostració és per inducció sobre el nombre d'arcs.	*La demostración es por inducción sobre el número de arcos.*
Siga $A = \{a_1, \dots, a_{\lvert A\rvert}\}$ i	*Sea $A = \{a_1, \dots, a_{\lvert A\rvert}\}$ y*

$$G_n = (V, \{a_1, \dots, a_n\}), \ \forall n, \ 0 \leq n \leq |A|.$$

Anomenem c_n el nombre de components connexes del subgraf G_n.	*Llamamos c_n al número de componentes conexas del subgrafo G_n.*

BI:

La base d'inducció es correspon amb $n = 0$ arcs. En aquest cas G_0 té tantes components connexes com nodes i per tant	*La base de inducción se corresponde con $n = 0$ arcos. En ese caso G_0 tiene tantas componentes conexas como nodos y por tanto*

$$c_0 = |V| \geq |V| - 0 = |V| - |A|.$$

HI:

$$[c_n \geq |V| - n].$$

PI:

Tenim el subgraf G_n que té c_n components connexes i volem caracteritzar el nombre de components connexes del subgraf G_{n+1}, c_{n+1}.	*Tenemos el subgrafo G_n que tiene c_n componentes conexas y queremos caracterizar el número de componentes conexas del subgrafo G_{n+1}, c_{n+1}.*
Però l'única diferència entre G_n i G_{n+1} és l'arc a_{n+1}. I hi ha dos casos possibles:	*Pero la única diferencia entre G_n y G_{n+1} es el arco a_{n+1}. Y hay dos casos posibles:*
a) a_{n+1} uneix dos nodes dins de la mateixa component connexa. Aleshores	*a) a_{n+1} une dos nodos dentro de la misma componente conexa. Entonces*

$$c_{n+1} = c_n \overset{(HI)}{\geq} |V| - n > |V| - (n+1).$$

b) a_{n+1} uneix dos nodes en **diferents** components connexes. Aleshores

*b) a_{n+1} une dos nodos en **diferentes** componentes conexas. Entonces*

$$c_{n+1} = c_n - 1 \overset{(HI)}{\geq} |V| - n - 1 = |V| - (n+1).$$

Com que en els dos casos arribem al mateix, podem concloure finalment que

Como en los dos casos llegamos a lo mismo, podemos concluir finalmente que

$$\forall G = (V, A) \ : \ |A| = n, \ c_n \geq |V| - n = |V| - |A|.$$

Corol·lari

Tot graf no dirigit connex té almenys $|V| - 1$ arcs. És a dir,

Todo grafo no dirigido tiene al menos $|V| - 1$ arcos. Es decir,

$$|A| \geq |V| - 1.$$

4.1.5 Demostració del teorema d'Euler

Demostración del teorema de Euler

La demostració del teorema d'Euler és senzilla en un sentit però significativament més complexa en l'altre. La formulació més general amb multigrafs amb bucles no només no afegeix complexitat si no que fa una part de la demostració més natural.

La demostración del teorema de Euler es sencilla en un sentido pero significativamente más compleja en el otro. La formulación más general con multigrafos con bucles no sólo no añade complejidad si no que hace una parte de la demostración más natural.

Per conveniència demostrarem primer un lema que estableix l'existència de cicles en un multigraf (sense bucles).

Por conveniencia demostraremos primero un lema que establece la existencia de ciclos en un multigrafo (sin bucles).

Lema

Si G és un multigraf no dirigit i connex on el grau de tots els seus nodes és major o igual que 2, aleshores ha d'existir algun cicle en G.

Si G es un multigrafo no dirigido y conexo donde el grado de todos sus nodos es mayor o igual que 2, entonces tiene que existir algún ciclo en G.

La demostració fa servir el **principi de les caixes**. Comencem per un node arbitrari, u_1, dels $n \geq 2$ nodes de G i construim una seqüència de nodes diferents, cada un adjacent a l'anterior, fins que arribem a un node u_k que siga adjacent a algun node u_ℓ anterior en la seqüència ($\ell < k$).

La demostración utiliza el **principio de las cajas**. *Empezamos con un nodo arbitrario, u_1, de los $n \geq 2$ nodos de G y construimos una secuencia de nodos diferentes, cada uno adyacente al antarior, hasta que lleguemos a un nodo u_k que sea adyacente a algún nodo u_ℓ anterior en la secuencia ($\ell < k$).*

$$(u_1, u_2, \ldots, u_k)$$

Aquesta seqüència sempre es podrà construir ja que tots els nodes tenen com a mínim altres 2 nodes adjacents. A més a més, el valor de k complirà necessàriament,

Esta secuencia siempre se podrá construir ya que todos los nodos tienen como mínimo otros 2 nodos adyacentes. Además, el valor de k cumplirá necesariamente,

$$2 \leq k \leq n$$

i en tots els casos el node u_k tindrà (almenys) un node adjacent, $u_\ell, \ell < k$, a banda de u_{k-1} per tindre grau major o igual a 2.

y en todos los casos el nodo u_k tendrá (al menos) un nodo adyacente, $u_\ell, \ell < k$, aparte de u_{k-1} por tener grado mayor o igual a 2.

Com a conseqüència podem afirmar que

Como consecuencia podemos afirmar que

$$(u_\ell, u_{\ell+1}, \ldots, u_{k-1}, u_k, u_\ell)$$

és un cicle en G.

es un ciclo en G.

Teorema d'Euler

Teorema de Euler

Siga $G = \langle V, A \rangle$ un multigraf amb bucles no dirigit i connex. G conté un **circuit** Eulerià sii tots els seus nodes tenen grau parell major o igual que 2.

Sea $G = \langle V, A \rangle$ un multigrafo con bucles no dirigido y conexo. G contiene un **circuito** *Euleriano* sii *todos sus nodos tienen grado par mayor o igual a 2.*

Organitzarem la demostració en dues parts, una pera cada sentit de la coimplicació. També, per comoditat, expressarem com a

Organizaremos la demostración en dos partes, una para cada sentido de la coimplicación. También, por comodidad, expresaremos como

$$C_{CE}(G)$$

el fet que G conté un cicle Eulerià.

el hecho de que G contiene un ciclo Euleriano.

$\boxed{C_{CE}(G) \Rightarrow \forall u \in V, \exists k \geq 1 : gr(u) = 2k}$

La demostració és per reducció a l'absurd. I el que suposem és que hi ha en G un node, v, el grau del qual és senar.

La demostración es por reducción al absurdo. Y lo que suponemos es que hay en G un nodo, v, cuyo grado es impar.

Però resulta que el circuit Eulerià que ha d'existir ha de passar per tots els arcs (una vegada) i per tots els nodes (per ser connex) una o més vegades.

Pero resulta que el circuito Euleriano que tiene que existir tiene que pasar por todos los arcos (una vez) y por todos los nodos (por ser conexo) una o más veces.

Això vol dir que haurà d'entrar i eixir, una o més vegades de cada node. Però en el cas de v, per tindre grau senar, arribaria un moment en què no es podria eixir. La qual cosa és impossible.

Esto significa que tendrá que entrar y salir, una o más veces de cada nodo. Pero en el caso de v, por tener grado impar, llegaría un momento en que no se podría salir, lo que es imposible.

$\boxed{\forall u \in V, \exists k \geq 1 : gr(u) = 2k \Rightarrow C_{CE}(G)}$

La demostració és per inducció sobre el nombre d'arcs, n, en el multiconjunt d'arcs de G. Primer que res, establim una numeració arbitrària de tots els arcs.

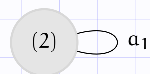
La demostración es por inducción sobre el número de arcos, n, en el multiconjunto de arcos de G. Primero, establecemos una numeración arbitraria de todos los arcos.

$$A = \{a_1, \ldots, a_n\}$$

BI:

Per a un sol arc, l'única possibilitat és un únic node amb un bucle (per tant amb grau 2, mostrat entre parèntesis). En eixe cas, el bucle constitueix un circuit Eulerià.

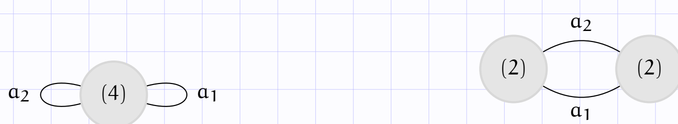
Para un solo arco, la única posibilidad es un único nodo con un bucle (por tanto con grado 2, mostrado entre paréntesis). En ese caso, el bucle constituye un circuito Euleriano.



circuit: (a_1)
circuito:

Amb 2 arcs hi ha dos casos. Un node amb dos bucles (grau 4) , o dos nodes amb dos arcs paral·lels (grau 2 cada un). En els 2 casos, la seqüència formada pels 2 arcs és un circuit Eulerià.

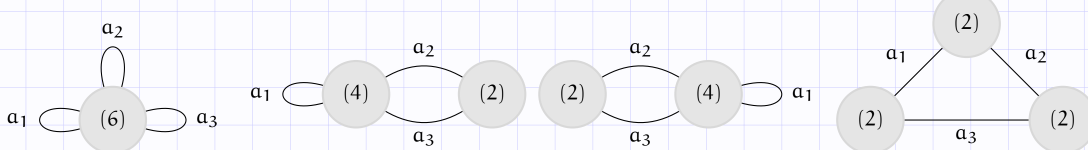
Con 2 arcos hay dos casos. Un nodo con dos bucles (grado 4), o dos nodos con dos arcos paralelos (grado 2 cada uno). En los 2 casos, la secuencia formada por los 2 arcos es un circuito Euleriano.



circuit: (a_1, a_2)
circuito:

Entre els casos que hi ha amb 3 arcs, apareix el graf més simple (no multigraf i sense bucles) que estaria format per tres nodes totalment connectats entre ells i que conté també un circuit Eulerià.

Entre los casos que hay con 3 arcos, aparece el grafo más simple (no multigrafo y sin bucles) que estaría formado por tres nodos totalmente conectados entre ellos y que contiene también un circuito Euleriano.



circuit: (a_1, a_2, a_3)
circuito:

Estrictament només necessitem el cas $n = 1$ com a base d'inducció.

Estrictamente sólo necesitamos el caso $n = 1$ como base de inducción.

HI:

Com a hipòtesi d'inducció suposarem que tot multigraf amb bucles connex amb nodes de grau parell major que zero, i amb estrictament menys de n arcs, conté un circuit Eulerià.

Como hipótesis de inducción supondremos que todo multigrafo con bucles conexo con nodos de grado par mayor que cero, y estrictamente menos de n arcos, contiene un circuito Euleriano.

Altrament dit, necessitarem **inducció forta**. I podem expressar formalment la hipòtesi com

Dicho de otra manera, necesitaremos ***inducción fuerte****. Y podemos expresar formalmente la hipótesis como*

$$\left[\forall G' = \langle V', A'\rangle : |A'| = \ell < n, \left(\forall u \in V',\ gr(u) \geq 2\right) \Rightarrow C_{CE}(G')\right]$$

PI:

Considerarem ara un multigraf ,G, amb exactament n arcs i m nodes, tots ells amb grau parell major o igual que 2. És a dir,

Consideraremos ahora un multigrafo, G, con exactamente n arcos y m nodos, todos ellos con grado par mayor o igual que 2. Es decir,

$$G = \langle V, A\rangle, \quad V = \{v_1, \ldots, v_m\}, \quad A = \{a_1, \ldots, a_n\}$$

El multigraf G ha de contindre un cicle. Be per l'existència de bucles, be com a conseqüència del lema anterior. Ja que fins i tot sense comptar els bucles tots els nodes tenen graus majors o iguals que 2 per ser G connex.

El multigrafo G tiene que contener un ciclo. Bien por la existencia de bucles, bien como consecuencia del lema anterior. Ya que incluso sin contar los bucles todos los nodos tienen grados mayores o iguales que 2 por ser G conexo.

Si expressem aquest cicle com a seqüència de nodes i arcs,

Si expresamos este ciclo como secuencia de nodos y arcos,

$$\phi = (u_1, b_1, u_2, b_2, \ldots, u_k, b_k, u_1), \quad \begin{array}{ll} 1 \leq k \leq n & \\ \forall i, & u_i \in V \wedge b_i \in A, \\ \forall i,j : i \neq j, & u_i \neq u_j \wedge b_i \neq b_j \end{array}$$

Si eliminem ara el conjunt d'arcs

Si eliminamos ahora el conjunto de arcos

$$B = \{b_1, \ldots, b_k\},$$

del multigraf G, el resultat serà un multigraf G' amb els mateixos nodes i amb $n' = n - k$ arcs on

del multigrafo G, el resultado será un multigrafo G' con los mismos nodos y con $n' = n - k$ arcos donde

$$0 \leq n' \leq n - 1$$

El multigraf G' no serà necessàriament connex i tindrà entre 1 i m components connexes,

El multigrafo G' no será necesariamente conexo y tendrá entre 1 y m componentes conexas,

$$C_1, \ldots, C_p, \quad 1 \leq p \leq m,$$

que contindran tots els nodes de G.

que contendrán todos los nodos de G.

El grau dels nodes de ϕ haurà decrescut en 2 i per tant tots els nodes de G' tindran grau parell només que alguns (dels de ϕ) podran tindre grau zero.

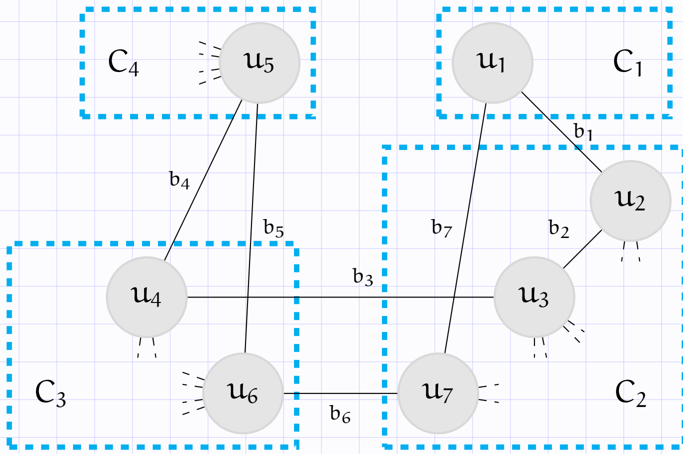
El grado de los nodos de ϕ habrá decrecido en 2 y por lo tanto todos los nodos de G' tendrán grado par solo que algunos (de los de ϕ) podrán tener grado cero.

Quant a les components connexes, n'hi haurà de dos tipus: a) les formades per un únic node de ϕ de grau zero, i b) les formades per un o més nodes de ϕ i possiblement altres, tots ells amb grau major o igual a 2.

En cuanto a las componentes conexas, las habrá de dos tipos: a) las formadas por un único nodo de ϕ de grado cero, y b) las formadas por uno o más nodos de ϕ y posiblemente otros, todos ellos con grado mayor o igual a 2.

No pot haver cap component sense nodes de ϕ si el multigraf original era connex. En la figura es mostra un exemple amb 4 components connexes i un cicle ϕ amb 7 arcs.

No puede haber ninguna componente conexa sin nodos de ϕ si el multigrafo original era conexo. En la figura se muestra un ejemplo con 4 componentes conexas y un ciclo ϕ con 7 arcos.



En el cas que totes les components connexes són nodes de grau zero, el circuit ϕ ja és un circuit Eulerià per al multigraf G.

En el caso en que todas las componentes conexas son nodos de grado cero, el circuito ϕ ya es un circuito Euleriano para el multigrafo G.

En cas contrari es pot construir un circuit Eulerià de la següent manera. Comencem en u_1 i anem recorreguent el cicle ϕ i afegint els seus arcs al circuit en construcció començant per l'arc b_1.

En caso contrario se puede construir un circuito Euleriano de la siguiente manera. Empezamos con u_1 y vamos recorriendo el ciclo ϕ y añadiendo sus arcos al circuito en construcción empezando por el arco b_1.

Quan arribem per primera vegada a una component connexa, si el corresponent node, u_i, té grau zero continuem (la component només té el node u_i) i si no, la component connexa és un multigraf connex amb estrictament menys de n arcs.

Cuando llegamos por primera vez a una componente conexa, si el nodo correspondiente, u_i, tiene grado cero continuamos (la componente sólo tiene el nodo u_i) y si no, la componente conexa es un multigrafo conexo con estrictamente menos de n arcos.

Per **hipòtesi d'inducció**, la component contindrà un circuit Eulerià que comença i acaba en u_i i que podrem insertar en el circuit. A continuació afegim l'arc b_i i continuem.

Por **hipótesis de inducción**, *la componente contendrá un circuito Euleriano que empieza y acaba en u_i y que podremos insertar en el circuito. A continuación añadimos el arco b_i y continuamos.*

Si en algun moment tornem a una component connexa ja visitada, ens limitarem a entrar i eixir del corresponent node u_j usant els arcs b_{j-1} i b_j ja que tots els altres arcs d'aquesta component ja s'hauran insertat en el circuit la primera vegada que s'ha visitat.

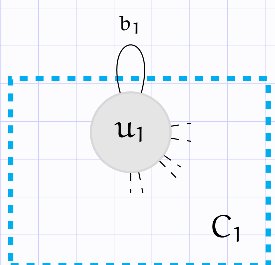
Si en algún momento volvemos a una componente conexa ya visitada, nos limitamos a entrar y salir del correspondiente nodo u_j usando los arcos b_{j-1} y b_j ya que todos los otros arcos de esta componente ya se habrán insertado en el circuito la primera vez que se ha visitado.

Després d'afegir l'últim arc de ϕ, b_k, arribarem de nou al node u_1 i haurem completat un circuit Eulerià.

Después de añadir el último arco de ϕ, b_k, llegaremos de nuevo al nodo u_1 y habremos completado un circuito Euleriano.

El cas més senzill és quan el circuit és un bucle, $\phi = (u_1, b_1, u_1)$. Aleshores només hi hauria una única component connexa que seria tot el multigraf sense el bucle ϕ i per tant amb exactament $n-1$ arcs. Aquesta component contindria un circuit Eulerià per **hipòtesi d'inducció** i només caldria afegir el bucle per tal d'obtindre un circuit Eulerià de tot G.

El caso más sencillo es cuando el circuito es un bucle, $\phi = (u_1, b_1, u_1)$. Entonces sólo habría una única componente conexa que seria todo el multigrafo sin el bucle ϕ y por lo tanto con exactamente $n-1$ arcos. Esta componente conexa contendría un circuito Euleriano por **hipótesis de inducción** *y sólo haría falta añadir el bucle para obtener un circuito Euleriano de todo G.*



4.2 Arbres: definicions i propietats
Árboles: definiciones y propiedades

Ja hem definit un arbre com un graf connex i acíclic. Però aquesta definició no respon a la idea de jerarquia que normalment associem als arbres. Per això, aquests arbres s'anomenen també **arbres sense arrel**.

Ya hemos definido un árbol como un grafo conexo y acíclico. Pero esta definición no responde a la idea de jerarquía que normalmente asociamos a los árboles. Por este motivo, estos árboles se les llama también **árboles sin raíz**.

En un arbre sense arrel no hi ha cap node que siga origen de cap jerarquia. Tampoc no podem parlar de cap relació jeràrquica entre nodes (pare, fill, ancestre, etc.).

En un árbol sin raíz no hay ningún nodo que sea origen de ninguna jerarquía. Tampoco se puede hablar de ninguna relación jerárquica entre nodos (padre, hijo, ancestro, etc.).

Entre qualssevol dos nodes d'un arbre sense arrel hi ha un únic camí.

Entre cualesquiera dos nodos de un árbol sin raíz hay un único camino.

A qualsevol subgraf d'un arbre que siga arbre l'anomenem **subarbre**.

A cualquier subgrafo de un árbol que sea árbol se le llama **subárbol**.

Si suprimim un arc en un arbre sense arrel, el resultat és un graf no connex amb dues components connexes cada una de les quals és un subarbre.

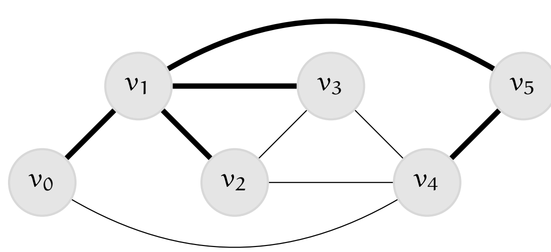
Si suprimimos un arco en un árbol sin raíz, el resultado es un grafo no conexo con dos componentes conexas cada una de las cuales es un subárbol.

Donat un graf connex $G = (V, A)$, diem que $G' = (V, A')$ és un **arbre generador** o **arbre d'extensió** de G sii $G' \subseteq G$ i G' és un arbre.

Dado un grafo conexo $G = (V, A)$, decimos que $G' = (V, A')$ es un **árbol generador** *o* **árbol de extensión** *de G sii $G' \subseteq G$ y G' es un árbol.*

En la figura es mostra un graf i un dels seus arbres generadors (marcat amb línies més grosses).

En la figura se muestra un grafo y uno de sus subárboles generadores (marcado con líneas más gruesas).



Proposició	***Proposición***
Tot graf connex conté almenys un arbre generador.	*Todo grafo conexo contiene al menos un árbol generador.*
Demostració (informal):	**Demostración (informal):**
Suposem que el graf no és arbre (altrament ja estaria demostrat). En aquest cas contindrà almenys un cicle.	*Supongamos que el grafo* no *es árbol (en otro caso ya estaría demostrado). En ese caso contendrá al menos un ciclo.*
Si eliminem qualsevol arc del cicle tenim un nou graf connex amb un arc menys i podem tornar al principi del raonament.	*Si eliminamos cualquier arco del ciclo tenemos un nuevo grafo conexo con un arco menos y podemos volver al principio del razonamiento.*
En repetir aquest raonament arribarem necessàriament a un graf connex i sense cicles.	*Al repetir este razonamiento llegaremos necesariamente a un grafo conexo y sin ciclos.*

4.2.1 Arbres amb arrel

Árboles con raíz

Un **arbre amb arrel** és un arbre (graf connex i acíclic) en què distingim un node especial que anomenem **arrel**.

Un **árbol con raíz** *es un árbol (grafo conexo y acíclico) en el que distinguimos un nodo especial que llamamos* **raíz**.

Com a conseqüència de la definició anterior tenim que:

Como consecuencia de la definición anterior tenemos que:

a) Els nodes que no són arrel s'organitzen en $m \geq 0$ conjunts disjunts que indueixen arbres.

a) Los nodos que no son raíz se organizan en $m \geq 0$ conjuntos disjuntos que inducen árboles.

b) Els $m \geq 0$ nodes adjacents a l'arrel, r, se'ls anomena **fills** de r i se'ls distingeix com a arrels dels corresponents $m \geq 0$ subarbres. També es diu que r és **pare** dels seus nodes adjacents.

b) Los $m \geq 0$ nodos adyacentes a la raíz, r, se les llama **hijos** *de r y se los distingue como raíces de los correspondientes $m \geq 0$ subárboles. También se dice que r es* **padre** *de sus nodos adyacentes.*

c) Els nodes d'un arbre amb arrel s'organitzen també en nivells. El **nivell** o **profunditat d'un node** és la longitud (en nodes) del camí des de l'arrel fins al node. L'arrel està a profunditat 1, els fills a profunditat 2, etc.

c) Los nodos de un árbol con raíz se organizan también en niveles. El ***nivel*** *o* ***profundidad de un nodo*** *es la longitud (en nodos) del camino desde la raíz hasta él. La raíz está a profundidad 1, sus hijos a profundidad 2, etc.*

d) La **profunditat d'un arbre** es defineix com el nivell màxim dels seus nodes. I la **talla** o **grandària** de l'arbre és el nombre de nodes que conté.

d) La ***profundidad de un árbol*** *se define como el nivel máximo de sus nodos. Y la* ***talla*** *o* ***tamaño*** *del árbol es el número de nodos que contiene.*

e) Els nodes que no tenen fills s'anomenen **fulles**.

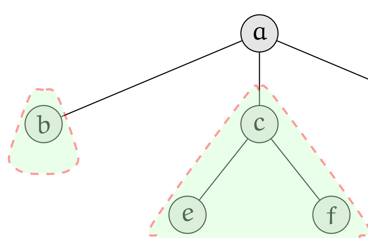
e) Los nodos que no tienen hijos se llaman ***hojas***.

Ens referirem a l'arbre que té per arrel r com a $T(r)$ o moltes vegades simplement com a r. La profunditat d'aquest arbre l'escriurem com a $h(T(r))$ o $h(r)$ i la seua grandària com a $|T(r)|$ o $|r|$.

Nos referiremos al árbol cuya raíz es r como $T(r)$ o muchas veces simplemente como r. La profundidad de este árbol la escribiremos como $h(T(r))$ o $h(r)$ y su tamaño como $|T(r)|$ o $|r|$.

En la figura es mostra un arbre amb arrel de grandària 7 i profunditat 3 que té per arrel el node a. S'indiquen a més a més els 3 subarbres de a que tenen per arrels b, c i d. L'arbre té 4 fulles en total: b, e, f i g que són a profunditat (o nivell) 2, 3, 3 i 3 respecte de l'arbre a.

En la figura se muestra un árbol con raíz de tamaño 7 y profundidad 3 cuya raíz es el nodo a. Se indican además los 3 subárboles de a cuyas raíces son b, c y d. El árbol tiene 4 hojas en total: b, e, f y g que se encuentran a profundidad (o nivel) 2, 3, 3 y 3 respecto al árbol a.



Els arbres amb arrel es poden veure com a estructures de dades que representen diferents classes de jerarquies, per la qual cosa admeten també i de manera natural definicions recursives com la que esbossem a continuació.

Los árboles con raíz se pueden ver como estructuras de datos que representan diferentes tipos de jerarquías por lo que admiten también y de forma natural definiciones recursivas como la que esbozamos a continuación.

a) Un graf d'un node és un arbre amb arrel (l'arrel és l'únic node).

a) Un grafo de un nodo es un arbol con raíz (la raíz es el único nodo).

b) Un arbre amb arrel, $T(r)$, de grandària major que 1 és format per $k > 1$ subarbres amb arrels

b) Un árbol con raíz, $T(r)$, de tamaño mayor que 1 está formado por $k > 1$ subárboles con raíces

$$T(r_1), \ldots, T(r_k),$$

que estan units amb r mitjançant arcs.

que están unidos con r mediante arcos.

4.2.2 Arbres ordenats o m-aris

Árboles ordenados o m-arios

Un **arbre ordenat** d'ordre m o **arbre m-ari** és un arbre amb arrel on **tot** node té exactament m subarbres **distingibles** que poden ser buits.

*Un **árbol ordenado** de orden m o **árbol m-ario** es un árbol con raíz donde **todo** nodo tiene exactamente m subárboles **distinguibles** que pueden ser vacíos.*

L'**arbre buit** (cap node) es considera un cas especial d'arbre ordenat.

*El **árbol vacío** (ningún nodo) se considera un caso especial de árbol ordenado.*

Es pot dir que en els arbres ordenats m-aris cada node té una **seqüència** de fills de longitud m, mentre que en els arbres amb arrel cada node té un **conjunt** de fills.

*Se puede decir que en los árboles ordenados m-arios cada nodo tiene una **secuencia** de hijos de longitud m, mientras que en los árboles con raíz cada nodo tiene un **conjunto** de hijos.*

En un arbre m-ari hi ha un màxim de m^{h-1} nodes a profunditat h.

En un árbol m-ario hay un máximo de m^{h-1} nodos a profundidad h.

Un arbre m-ari està o és un **arbre ple fins a profunditat h** sii té exactament

*Un árbol m-ario está o es un **árbol lleno hasta profundidad h** sii tiene exactamente*

$$\sum_{i=1}^{h} m^{i-1} = \frac{m^h - 1}{m - 1}$$

nodes a profunditat menor o igual que h.

nodos a profundidad menor o igual que h.

Un arbre m-ari està o és un **arbre ple** si està ple fins a la seua profunditat.

Un árbol m-ario está o es un **árbol lleno** *si está lleno hasta su profundidad.*

En un arbre m-ari **ple** de n nodes es compleix que

En un árbol m-ario **lleno** *de n nodos se cumple que*

$$n = \frac{m^h - 1}{m - 1},$$

o alternativament

o alternativamente

$$h = \log_m (1 + n(m-1)) \overset{(n \geq 1)}{\leq} 1 + \log_m n.$$

Una característica fonamental dels arbres ordenats com a estructura de dades és la seua profunditat. I en molts casos és clau que aquesta siga logarítmica respecte al nombre de nodes.

Una característica fundamental de los árboles ordenados como estructura de datos es su profundidad. Y en muchos casos es clave que ésta sea logarítmica respecto al número de nodos.

L'exigència de profunditats logarítmiques motiva diverses definicions que afegeixen condicions a l'estructura particular dels arbres ordenats.

La exigencia de profundidades logarítmicas motiva diversas definiciones que añaden condiciones a la estructura particular de los árboles ordenados.

Un arbre m-ari de profunditat h està o és un **arbre complet** sii

a) està **ple fins a profunditat $h-1$**,

b) per als m fills de qualsevol node, $s_1, \ldots, s_m$ es compleix que

Un árbol m-ario de profundidad h está o es un **árbol completo** sii

a) está **lleno hasta profundidad $h-1$**,

b) para los m hijos de cualquier nodo, $s_1, \ldots, s_m$ se cumple que

$$|s_1| \geq \ldots \geq |s_m|.$$

Un arbre m-ari està o és un **arbre equilibrat** sii

Un árbol m-ario está o es un **árbol equilibrado** sii

a) la màxima diferència entre qualsevol parell dels m subarbres és de 1,

b) els m subarbres són equilibrats.

a) la máxima diferencia entre cualquier par de los m subárboles es de 1,

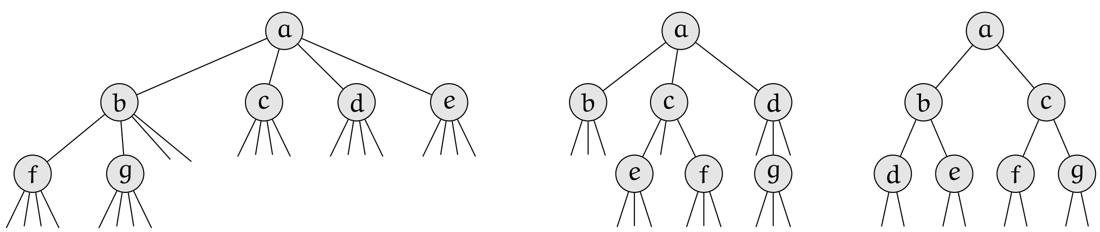
b) los m subárboles son **equilibrados**.

Els **arbres binaris** són el cas particular dels m-aris quan $m = 2$. Però són especialment importants tant per motius teòrics com, especialment, pràctics i d'implementació.

Los **árboles binarios** *son el caso particular de los m-arios cuando $m = 2$. Pero son especialmente importantes tanto por motivos teórios como, especialmente prácticos y de implementación.*

Quan $m = 2$ els dos fills de cada node se solen anomenar **esquerre** i **dret**, respectivament.

Cuando $m = 2$ los dos hijos de cada nodo se suelen llamar **izquierdo** *y* **derecho**, *respectivamente.*



En la figura es mostren 3 arbres ordenats d'ordres 4, 3 i 2 respectivament. Els arbres contenen 22, 15 i 8 subarbres buits, 5, 4 i 4 fulles, encara que tots tres tenen 7 nodes i són de profunditat 3.

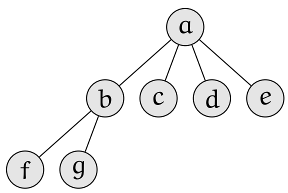
En la figura se muestran 3 árboles ordenados de órdenes 4, 3 y 2 respectivamente. Los árboles contienen 22, 15 y 8 subárboles vacíos, 5, 4 y 4 hojas, aunque los tres tienen 7 nodos y son de profundidad 3.

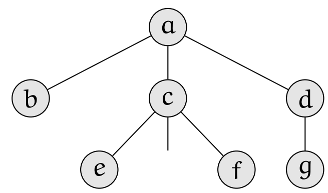
L'arbre de l'esquerra, que és 4-ari, és un arbre **ple fins al nivell 2** i **complet**, mentre que l'arbre de la dreta és **ple** (la qual cosa implica **ple fins al nivell 3** i **complet**).

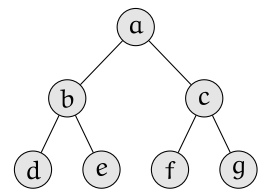
El árbol de la izquierda, que es 4-ario, es un árbol **lleno hasta el nivel 2** *y* **completo**, *mientras que el árbol de la derecha es* **lleno** *(lo que implica* **lleno hasta el nivel 3** *y* **completo**).

Llevat que es tinga un interès especial, no serà normal representar tots els subarbres buits com s'ha fet en la figura anterior. No obstant això, si un node té menys fills que l'ordre de l'arbre ha de quedar clar quin dels m fills és cada un. En particular, els arbres anteriors els dibuixarem també com en la figura següent.

A no ser que se tenga un interés especial, no será normal representar todos los subárboles vacíos como se ha hecho en la figura anterior. Sin embargo, si un nodo tiene menos hijos que el orden del árbol, tiene que quedar claro cual de los m hijos es cada uno. En particular, los árboles anteriores los dibujaremos también como en la siguiente figura.







4.3 Grafs i arbres amb contingut

Grafos y árboles con contenido

Fins ara els grafs i arbres són estructures que representen diferents tipus de relacions dins d'un conjunt de nodes el nom dels quals no és important. De fet, sovint els nodes es representen com a cercles que no tenen cap nom.

Hasta ahora los grafos y árboles son estructuras que representan diferentes tipos de relaciones dentro de un conjunto de nodos cuyos nombres no son importantes. De hecho, a menudo los nodos se representan como círculos que no tienen ningún nombre.

Però de vegades és interessant emmagatzemar o assignar alguna informació als nodes o als arcs. Aquesta informació pot ser de qualsevol mena, però els casos més importants i interessants són quan es tracta d'un **pes** (valor numèric normalment positiu que indica una quantitat d'alguna cosa: diners, distància, cost, etc.), una **clau numèrica** (valor numèric enter que pot ser únic o no: normalment un número d'ordre, una prioritat, etc.) o una **clau alfanumèrica** (cadena de caràcters que representa un identificador normalment únic: un DNI, un nom d'algú, un codi, etc.).

Pero a veces es interesante almacenar o asignar algún tipo de información a los nodos o a los arcos. Esta información puede ser de cualquier tipo pero los casos más importantes e interesantes son cuando se trata de un **peso** *(valor numérico normalmente positivo que indica una cantidad de algo: dinero, distancia, coste, etc.), una* **clave numérica** *(valor numérico que puede ser único o no: normalmente un número de orden, una prioridad, etc.) o una* **clave alfanumérica** *(cadena de caracteres que representa un identificador normalmente único: un DNI, un nombre de alguien, un código, etc.).*

Un **graf amb nodes ponderats** és un graf $G = (V, A)$ i una aplicació,

Un **grafo con nodos ponderados** *es un grafo* $G = (V, A)$ *y una aplicación,*

$$p : V \longrightarrow \mathbb{R}^+.$$

Un **graf amb arcs ponderats** o simplement **graf ponderat** és un graf $G = (V, A)$ i una aplicació,

Un **grafo con arcos ponderados** *o simplemente* **grafo ponderado** *es un grafo* $G = (V, A)$ *y una aplicación,*

$$p : A \longrightarrow \mathbb{R}^+.$$

A vegades representarem ambdós tipus de grafs ponderats com a

A veces representaremos ambos tipos de grafos ponderados como

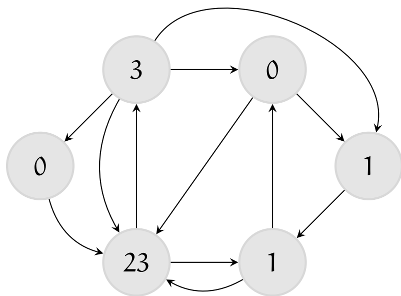
$$G = (V, A, p).$$

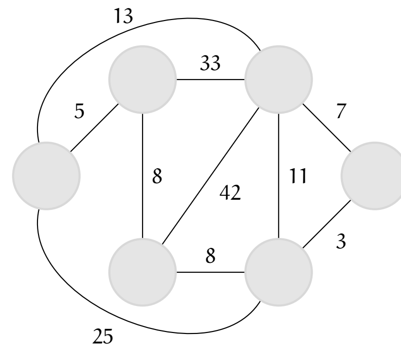
En els grafs ponderats es pot associar un pes a cada camí i també a tot el graf. Alguns exemples de problemes amb grafs ponderats són l'obtenció de camins de pes mínim, de l'arbre d'extensió minimal o de la bipartició de cost mínim (o màxim).

En los grafos ponderados se puede asociar un peso a cada camino y también a todo el grafo. Algunos ejemplos de problemas con grafos ponderados son la obtención de caminos de peso mínimo, del árbol de extensión minimal o de la bipartición de coste mínimo (o máximo).

En la figura següent a l'esquerra es mostra un graf (dirigit) amb nodes ponderats que podria correspondre a relacions d'amistat en una mini-xarxa social on el pes de cada node significa el nivell d'influència o popularitat de cada membre.

En la figura siguiente a la izquierda se muestra un grafo (dirigido) con nodos ponderados que podría corresponder a relaciones de amistad en una minired social donde el peso de cada nodo significa el nivel de influencia o popularidad de cada miembro.





En la mateixa figura a la dreta es mostra un graf ponderat que podria correspondre a un conjunt de ciutats i les connexions per carretera entre elles. En aquest cas el pes de cada arc podria correspondre a la distància en quilòmetres de cada tram de carretera.

En la misma figura a la derecha se muestra un grafo ponderado que podría corresponder a un conjunto de ciudades y las conexiones por carretera entre ellas. En este caso el peso de cada arco podría corresponder a la distancia en quilómetros de cada tramo de carretera.

Un **arbre amb claus** és un **arbre** (del tipus que siga) format per un conjunt de nodes, V, un conjunt d'arcs, A, i una aplicació $c : V \longrightarrow \mathcal{C}$, de manera que $c(u)$ és la clau associada al node u.

*Un **árbol con claves** es un **árbol** (del tipo que sea) formado por un conjunto de nodos, V, un conjunto de arcos, A, y una aplicación $c : V \longrightarrow \mathcal{C}$, de manera que $c(u)$ es la clave asociada al nodo u.*

Quan no puga haver-hi confusió, escriurem u per a referir-nos a $c(u)$. És més, moltes vegades quan els nodes només continguen claus i aquestes siguen úniques definirem directament

Cuando no pueda haber confusión, escribiremos u para referirnos a $c(u)$. Es más, muchas veces cuando los nodos sólo contengan claves y éstas sean únicas definiremos directamente

$$\forall u \in V,\ c(u) = u.$$

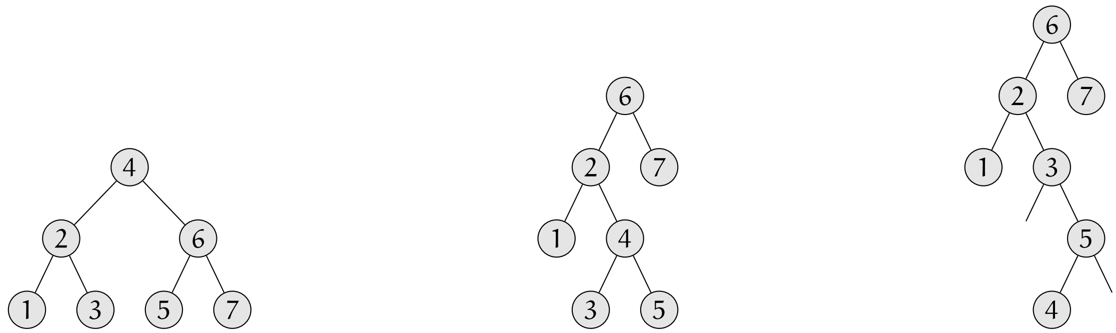
Un **arbre binari de cerca** és un arbre binari amb claus de manera que en $\mathcal{C}$ hi ha una relació d'ordre tal que per a tot node, a, es compleix que

*Un **árbol binario de búsqueda** es un árbol binario con claves de manera que en $\mathcal{C}$ hay una relación de orden tal que para todo nodo, a, se cumple que*

$$\forall x, y \in V\ :\ x \in \mathrm{esq}(a) \wedge y \in \mathrm{dre}(a),\ x < a < y.$$

En la figura següent es mostren tres arbres binaris de cerca de 7 nodes i de profunditats 3, 4 i 5 respectivament.

En la figura siguiente se muestran tres árboles binarios de búsqueda de 7 nodos y de profundidades 3, 4 y 5 respectivamente.



Un **monticle (de mínims)** és un arbre amb claus numèriques de manera que en $\mathcal{C}$ hi ha una relació d'ordre tal que per a tot node, a, es compleix que

Un **montículo (de mínimos)** *es un árbol con claves numéricas de manera que en $\mathcal{C}$ hay una relación de orden tal que para todo nodo, a, se cumple que*

$$\forall x \in V \ : \ a = \mathrm{pare}(x), \ a \leq x.$$

A vegades es poden definir monticles que són **conjunts** d'arbres. És a dir, **boscos**.

A veces, se pueden definir montículos que son **conjuntos** *de árboles. Es decir,* **bosques**.

Els monticles són en general estructures arborescents que són ideals per a implementar un tipus de dades que s'anomena **cua de prioritat** i que suporta operacions com obtenció/eliminació del mínim, inserció de nous elements, canvis de prioritat o eliminacions.

Los montículos son en general estructuras arborescentes que son ideales para implementar un tipo de datos que se denomina **cola de prioridad** *y que soporta operaciones como obtención/eliminación del mínimo, inserción de nuevos elementos, cambios de prioridad o eliminaciones.*

Perquè unes operacions o altres puguen ser eficients, cal restringir fortament l'estructura arborescent, i això dona lloc a diferents tipus de monticles.

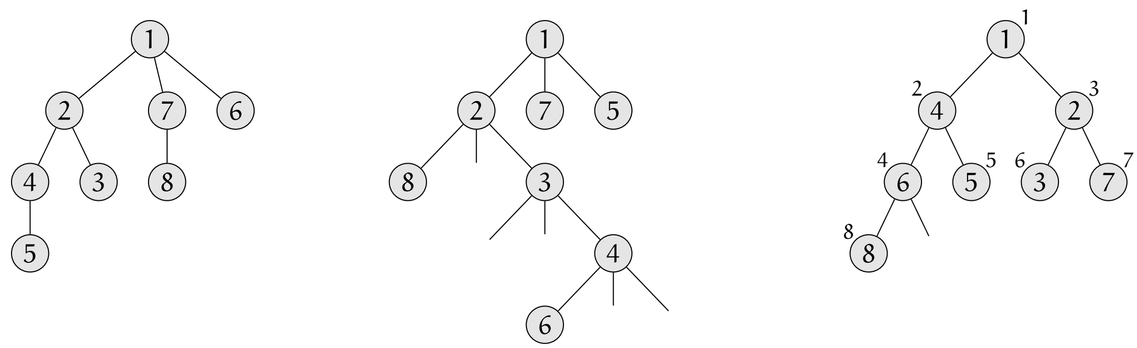
Para que unas operaciones u otras puedan ser eficientes, hace falta restringir fuertemente la estructura arborescente lo que da lugar a diferentes tipos de montículos.

Un **monticle binari (de mínims)** és un arbre binari **complet** que és monticle.

Un **montículo binario (de mínimos)** *es un árbol binario* **completo** *que es montículo.*

El fet de ser complet permet identificar un monticle binari d'ordre n amb una n-tupla de valors, la qual cosa fa que moltes operacions es puguen fer de manera molt eficient.

El hecho de ser completo permite identificar un montículo binarlo de orden n con una n-tupla de valores lo que hace que muchas operaciones se puedan hacer de manera muy eficiente.



En la figura es mostren tres arbres que compleixen la condició de monticle (de mínims). El de l'esquerra és un arbre amb arrel, el del mig és un arbre ternari i el de la dreta és un arbre binari complet i, per tant és un **monticle binari.**

En la figura se muestran tres árboles que cumplen la condición de montículo (de mínimos). El de la izquierda es un árbol con raiz, el del medio es un árbol ternario y el de la derecha es un árbol binario completo y, por tanto es un **montículo binario***.*

La representació d'aquest monticle binari com a tupla seria

La representación de este montículo binario como tupla sería

1	2	3	4	5	6	7	8
1	4	2	6	5	3	7	8

4.4 Problemes resolts i comentats

Problemas resueltos y comentados

Problema 4.1: [minmaxArcs]

Si un graf dirigit té n vèrtexs, quin és el nombre mínim i màxim d'arestes que pot tenir? I en el cas particular que siga connex? Raona les respostes.

Si un grafo dirigido tiene n vértices, ¿Cual es el número mínimo y màximo de aristas que puede tener? ¿Y en el caso particular de que sea conexo? Razona las respuestas.

El nombre mínim d'arestes o arcs de qualsevol graf és clarament zero i es correspon amb el graf buit d'ordre n, $\emptyset_V$.

El número mínimo de aristas o arcos de cualquier grafo es claramente cero y se corresponde con el grafo vacío de orden n, $\emptyset_V$.

Com que es tracta d'un graf dirigit (sense bucles, ni arcs paral·lels) tot arc ha d'unir dos dels n nodes. El nombre màxim d'arestes es correspondrà, doncs, al nombre de possibles parells (ordenats) sense repeticions, que és

Como se trata de un grafo dirigido (sin bucles, ni arcos paralelos) todo arco tiene que unir dos de los n nodos. El número máximo de aristas se corresponderá pues al número de posibles pares (ordenados) sin repeticiones, que es

$$|V_n^2| = n^{\underline{2}} = n(n-1).$$

També es pot raonar que cada un dels n nodes pot estar unit amb tots els $n-1$ restants. I la quantitat anterior s'obté en aplicar el principi del producte.

También se puede razonar que cada uno de los n nodos puede estar unido con todos los $n-1$ restantes. Y la cantidad anterior se obtiene al aplicar el principio del producto.

En el cas particular que el graf siga connex el nombre màxim no canvia, ja que es correspon amb un graf on tots els nodes estan connectats amb tots (o graf complet) que serà, doncs, connex.

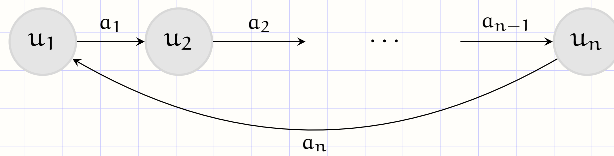
En el caso particular en que el grafo sea conexo, el número máximo no cambia, ya que se corresponde con un grafo donde todos los nodos están conectados con todos (o grafo completo) que será, pues, conexo.

Quant al nombre mínim en grafs connexos, cal tenir en compte que hi ha d'haver camins entre els $n(n-1)$ parells de nodes, tot i que aquests camins compartisquen arcs.

En cuanto al número mínimo en grafos conexos, hay que tener en cuenta que tiene que haber caminos entre los $n(n-1)$ pares de nodos, aunque estos caminos compartan arcos.

Una manera de connectar tots els parells és la següent: Podem recórrer els n nodes (en un ordre donat) mitjançant $n-1$ arcs dirigits i afegir un altre arc per a tornar a l'origen. És a dir, podem construir un cicle hamiltonià i eulerià de n arcs que contindrà de fet camins entre tots els parells (ordenats) de nodes.

Una manera de conectar todos los pares es la siguiente: Podemos recorrer los n nodos (en un orden dado) mediante $n-1$ arcos dirigidos y añadir otro arco para volver al origen. Es decir, podemos construir un ciclo hamiltoniano y euleriano de n arcos que contendrá de hecho caminos entre todos los pares (ordenados) de nodos.



De l'anterior es dedueix que el nombre mínim d'arcs en un graf dirigit connex de n nodes ha de ser menor o igual que n. Però pot ser inferior? Ho demostrarem per reducció a l'absurd.

De lo anterior se deduce que el número mínimo de arcos en un grafo dirigido conexo de n nodos tiene que ser menor o igual que n. Pero ¿puede ser inferior? Lo demostraremos por reducción al absurdo.

Suposem que es pogueren connectar n nodes amb $k < n$ arcs. En aquest cas, com que la suma total de graus és $2k$, es pot assegurar que hi ha d'haver nodes amb grau menor que 2. (Si tots els nodes tingueren grau major o igual que 2 la seua suma seria major o igual que $2n$).

Supongamos que se pudieran conectar n nodos con $k < n$ arcos. En este caso, como la suma total de grados es $2k$, se puede asegurar que tiene que haber nodos con grado menor que 2. (Si todos los nodos tuviesen grado mayor o igual que 2 su suma sería mayor o igual que $2n$).

Això implica que a aquests nodes, o bé no es pot arribar, o bé no se'n pot eixir, i per això hi ha d'haver parells de nodes sense connectar i el graf no pot ser connex.

Esto implica que a estos nodos, o bien no se puede llegar, o bien no se puede salir, y por esto tiene que haber pares de nodos sin conectar y el grafo no puede ser conexo.

Com a conseqüència, podem estar segurs que el nombre mínim d'arcs en un graf dirigit connex és n

Como consecuencia, podemos estar seguros de que el número mínimo de arcos en un grafo dirigido conexo es n.

Problema 4.2: [caseta]

Considera el graf $G = (V, A)$, de la figura.

Considera el grafo $G = (V, A)$, de la figura.

a) Es pot trobar un recorregut eulerià en aquest graf? Per què? Escriu-ne un si és possible.

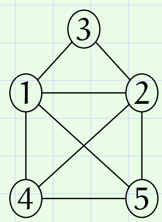
a) ¿Se puede encontrar un recorrido Euleriano en este grafo? ¿por qué? Escribe uno si es posible.

b) Es pot trobar un circuit que siga eulerià i hamiltonià al mateix temps? Per què? I en algun subgraf de G de 5 nodes? Digues en quants i en quins.

b) ¿Se puede encontrar un circuito que sea Euleriano y Hamiltoniano al mismo tiempo? ¿por qué? ¿Y en algún subgrafo de G de 5 nodos? Di en cuántos y en cuáles.

c) Hi ha algun arbre generador de G que es puga veure com a arbre binari? Dibuixa'n alguns o digues per què no n'hi ha.

c) ¿Hay algun árbol generador de G que se pueda ver como árbol binario? Dibuja algunos o di por qué no hay.



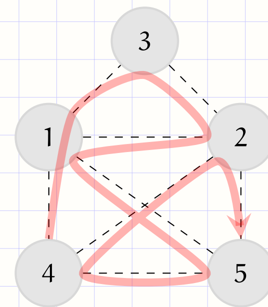
a) Sí que es pot trobar un recorregut eulerià a causa del teorema d'Euler (pàgina 178). O més concretament del seu corol·lari que afirma que si hi ha exactament dos nodes amb grau imparell, ha d'haver-hi un recorregut eulerià (no circular), que començarà i acabarà en aquests dos nodes.

a) Sí que se puede encontrar un recorrido euleriano a causa del teorema de Euler (página 178). O más concretamente de su corolario que afirma que si hay exactamente dos nodos con grado impar, tiene que haber un recorrido euleriano (no circular), que comenzará y acabará en estos dos nodos.

Un d'aquests recorreguts eulerians seria

Uno de estos recorridos eulerianos sería

$$(4, 1, 3, 2, 1, 5, 4, 2, 5)$$



b) Segons també el teorema d'Euler només hi pot haver un circuit eulerià si tots els nodes tenen grau parell. Per tant, menys encara pot haver-n'hi un que siga eulerià i hamiltonià.

b) Según también el teorema de Euler sólo puede haber un circuito euleriano si todos los nodos tienen grado par. Por tanto, menos aún puede haber uno que sea euleriano y hamiltoniano.

La mateixa pregunta sobre subgrafs d'ordre 5 requereix primer calcular tots els subgrafs connexos de 5 nodes.

La misma pregunta sobre subgrafos de orden 5 requiere primero calcular todos los subgrafos conexos de 5 nodos.

Afortunadament, dels 256 subgrafs d'ordre 5 (ja que hi ha 8 arcs i el nombre total de subconjunts serà $|VR_2^8| = 2^8$), només cal tenir en compte els que són connexos.

Afortunadamente, de los 256 subgrafos de orden 5 (ya que hay 8 arcos y el número total de subconjuntos será $|VR_2^8| = 2^8$), sólo hay que tener en cuenta los que son conexos.

Però això tampoc no és tan fàcil. En el seu lloc considerarem directament grafs connexos que tinguen exactament 5 arcs, que és la longitud (en arcs) que haurà de tenir qualsevol circuit eulerià i hamiltonià (i per tant, cicle).

Pero eso tampoco es tan fácil. En su lugar consideraremos directamente grafos conexos que tengan exactamente 5 arcos, que es la longitud (en arcos) que tendrá que tener cualquier circuito euleriano y hamiltoniano (y por tanto, ciclo).

En aquest cas tampoc cal considerar tots els subconjunts de 8 arcs de cardinalitat 5, ja que algunes configuracions donen lloc a grafs no connexos. Efectivament, si el cicle ha de passar pel node 3, això implica que els seus 2 arcs incidents hi han d'estar. I com a conseqüència, l'arc $(1, 2)$ no cal considerar-lo perquè es formaria un cicle de 3 nodes (i el recorregut resultant no podria ser hamiltonià).

En este caso tampoco hay que considerar todos los subconjuntos de 8 arcos de cardinalidad 5, ya que algunas configuraciones dan lugar a grafos no conexos. Efectivamente, si el ciclo tiene que pasar por el nodo 3, esto implica que sus 2 arcos incidentes han de estar.

En altres paraules, qualsevol cicle hamiltonià haurà d'estar format d'una banda pels arcs $(1,3)$ i $(3,2)$. I de l'altra per un camí que vaja (indirectament) des de 1 a 2 o al revés. I les úniques possibilitats són:

En otras palabras, cualquier ciclo hamiltoniano habra de estar formado por una parte por los arcos $(1,3)$ i $(3,2)$. Y por otra por un camino que vaya (indirectamente) desde 1 a 2 o al revés. Y las únicas posibilidades son:

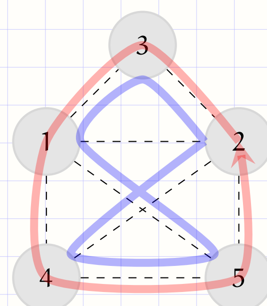
$$(1,4,5,2) \qquad (1,5,4,2).$$

Per tant, els únics cicles hamiltonians que hi podria haver en algun subgraf (de 5 nodes) de G serien

Por tanto, los únicos ciclos hamiltonianos que podría haber en algún subgrafo (de 5 nodos) de G serían

$$(4,1,3,2,1,5,4,2,5)$$

$$(4,1,3,2,1,5,4,2,5)$$

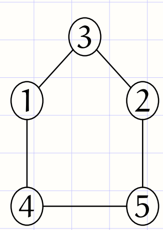


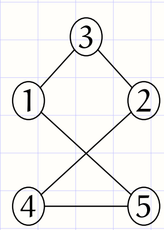
Com que ens demanen que els cicles siguen hamiltonians i eulerians, això vol dir que els subgrafs que els continguen no poden tenir cap arc més a banda dels que formen el cicle.

Como nos piden que los ciclos sean hamiltonianos y eulerianos, esto quiere decir que los subgrafos que los contengan no pueden tener ningún arco más aparte de los que forman el ciclo.

Per tant, els únics subgrafs de G d'ordre 5 que contenen un circuit (cicle) hamiltonià i eulerià són:

Por tanto, los únicos subgrafos de G de orden 5 que contienen un circuito (ciclo) hamiltoniano y euleriano son:





En realitat, hauríem pogut arribar a la mateixa conclusió més directament de la següent manera. Els nodes d'un subgraf que continga un circuit eulerià han de tenir grau parell. Però si a més a més ha de ser hamiltonià, el grau de **tots** els nodes ha de ser 2 (perquè només es pot entrar i eixir de cada node una vegada).

En realidad, habríamos podido llegar a la misma conclusión más directamente de la siguiente manera. Los nodos de un subgrafo que contenga un circuito euleriano han de tener grado par. Pero si además tiene que ser hamiltoniano, el grado de **todos** *los nodos debe ser 2 (porque sólo se puede entrar y salir de cada nodo una vez).*

Per a obtenir subgrafs de G on tots els nodes tinguen grau 2, caldrà anar eliminant arcs considerant totes les possibilitats. Ja hem raonat que l'arc $(1,2)$ s'ha d'eliminar per a desfer el cicle no desitjat $(1,3,2,1)$.

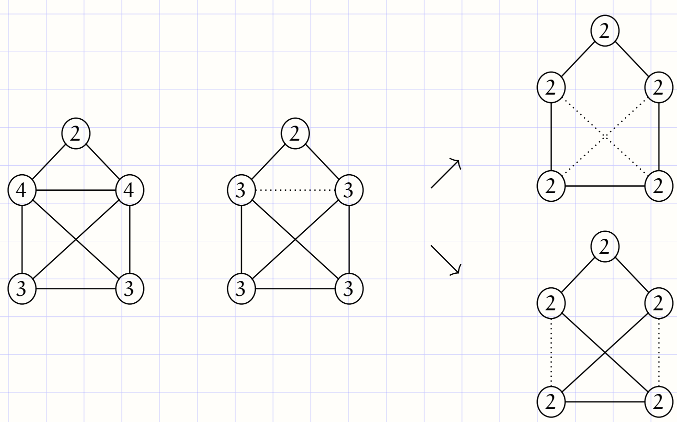
Para obtener subgrafos de G donde todos los nodos tengan grado 2, habrá que ir eliminando arcos considerando todas las posibilidades. Ya hemos razonado que el arco $(1,2)$ *se tiene que eliminar para deshacer el ciclo no deseado* $(1,3,2,1)$.

A partir d'ací tenim els nodes $1,2,4$ i 5 amb grau igual a 3. Per a aconseguir que els quatre acaben amb grau igual a 2 només hi ha dues possibilitats. O eliminem $(1,5)$ i $(2,4)$, o eliminem $(1,4)$ i $(2,5)$. I això porta a les dues solucions d'abans.

A partir de aquí tenemos los nodos $1,2,4$ *y* 5 *con grado igual a 3. Para conseguir que los cuatro acaben con grado igual a 2 sólo hay dos posibilidades. O eliminamos* $(1,5)$ *y* $(2,4)$, *o eliminamos* $(1,4)$ *y* $(2,5)$. *Y esto lleva a las dos soluciones anteriores.*

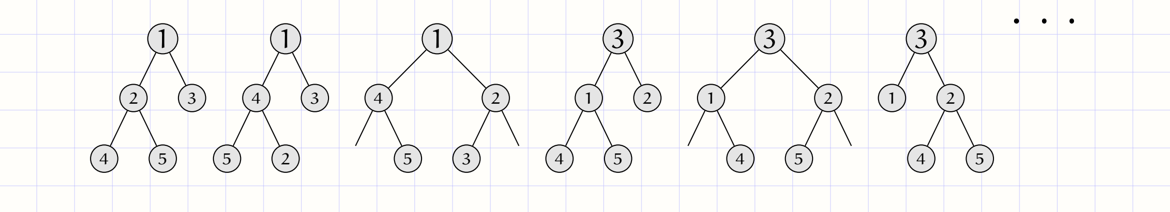
Les seqüències de subgrafs que s'obtindrien amb indicació del grau de cada node seria:

Las secuencias de subgrafos que se obtendrían con indicación del grado de cada nodo sería:



c) Sí que n'hi ha arbres generadors que es poden veure com a binaris. I són molts.

c) Sí que hay árboles generadores que se pueden ver como binarios. Y son muchos.



Problema 4.3: [grausSpan]

És possible que en un arbre generador d'un graf de n nodes els graus de tots els nodes siguen parells? I que siguen tots imparells? Justifica les respostes i posa'n exemples si es pot.

¿Es posible que en un árbol generador de un grafo de n nodos los grados de todos los nodos sean pares? ¿Y que sean todos impares? Justifica las respuestas y da ejemplos si se puede.

Un arbre generador d'un graf de n nodes és un arbre dels mateixos n nodes. Per tant, ha de ser **connex** i **acíclic**.

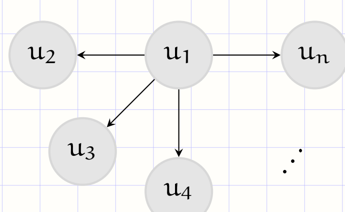
Un árbol generador de un grafo de n nodos es un árbol de los mismos n nodos. Por tanto, ha de ser **conexo** *y* **acíclico**.

Raonarem per **reducció a l'absurd**. Suposem que tots els nodes tenen grau parell. Aleshores, pel teorema d'Euler, hauria d'existir un circuit eulerià, cosa que implicaria que el graf (arbre) és **cíclic**. I això és impossible. Per tant, **no** poden ser tots els nodes de grau imparell.

Razonaremos por **reducción al absurdo**. *Supongamos que todos los nodos tienen grado par. Entonces, por el teorema de Euler, tendría que existir un circuito euleriano, lo que implicaría que el grafo (árbol) es* **cíclico**. *Y eso es imposible.*

La pregunta de si poden ser tots els nodes de grau imparell es pot reformular com a: Existeix un graf connex i acíclic de n nodes tots ells de grau imparell? I aquesta pregunta la podem contestar amb un exemple.

La pregunta de si pueden ser todos los nodes de grado impar se puede reformular como: ¿Existe un grafo conexo y acíclico de n nodos todos ellos de grado impar? Y esta pregunta la podemos contestar con un ejemplo.



A partir de l'exemple podem dir que **sí** que poden ser tots els nodes de grau imparell, però sempre que n siga **parell** (i major que zero, clar). Això és perquè en l'exemple el grau del node u_1 és exactament $n-1$.

*A partir del ejemplo podemos decir que **sí** que pueden ser todos llos nodos de grado impar, pero siempre que n sea **par** (y mayor que cero, claro). Esto es porque en el ejemplo el grado del nodo u_1 es exactamente $n-1$.*

I què passa si n és imparell? En aquest cas és impossible que tots els nodes tinguen grau imparell perquè aleshores la suma dels graus de tots els nodes seria un nombre imparell. I sabem que aquesta suma ha de ser igual a dues vegades el nombre d'arcs ($2n-2$ en el nostre cas), que és un nombre parell.

¿Y qué pasa si n es impar? En este caso es imposible que todos los nodos tengan grado impar porque entonces la suma de los grados de todos los nodos sería un número impar. Y sabemos que esta suma tiene que ser igual a dos veces el número de arcos ($2n-2$ en nuestro caso), que es un número par.

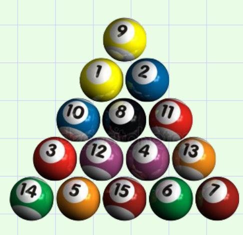
De fet, hauríem pogut simplement aplicar la proposició de la pàgina 173 i el seu corol·lari que diu que no pot haver-hi un nombre imparell de nodes amb grau imparell.

De hecho, habríamos podido simplemente aplicar la proposición de la página 173 y su corolario que dice que no puede haber un número impar de nodos con grado impar.

Problema 4.4: [billar]

Considera el graf $G = (V, A)$, on els nodes són les boles (numerades de la 1 a la 15) i els arcs es corresponen amb parells de boles que es toquen entre elles, com es mostra en la figura.

Considera el grafo $G = (V, A)$, donde los nodos son las bolas (numeradas de la 1 a la 15) y los arcos se corresponden con pares de bolas que se tocan entre ellas, como se muestra en la figura.



a) Defineix formalment el graf G.

b) Calcula el grau de cada node i la suma dels graus de tot el graf.

c) Dona, si existeix, un recorregut eulerià. Justifica la resposta.

d) Dona, si existeix, un camí $\mathbf{i/o}$ cicle hamiltonià. Justifica la resposta.

e) Què podem deduir del teorema d'Euler aplicat a G?

f) Siguen els subgrafs $G_i = (V_i, A_i)$, $i = 1, 2$, de manera que V_1 són les boles de l'1 a la 7 i V_2 les boles de la 9 a la 15. I els corresponents A_i estan formats per les arestes de A que relacionen els corresponents nodes. Descriu clarament els dos subgrafs i digues quantes i quines components connexes té cada un.

a) Define formalmente el grafo G.

b) Calcula el grado de cada nodo y la suma de los grados de todo el grafo.

c) Da, si existe, un recorrido Euleriano. Justifica la respuesta.

d) Da, si existe, un camino $\mathbf{y/o}$ ciclo Hamiltoniano. Justifica la respuesta.

e) ¿Qué podemos deducir del teorema de Euler aplicado a G?

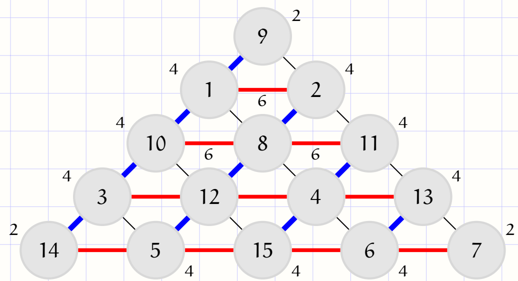
f) Sean los subgrafos $G_i = (V_i, A_i)$, $i = 1, 2$, de manera que V_1 son las bolas de la 1 a la 7 y V_2 les bolas de la 9 a la 15. Y los correspondientes A_i están formados por las aristas de A que relacionan los correspondientes nodos. Describe claramente los dos subgrafos y di cuántas y qué componentes conexas tiene cada uno.

a) Es tracta d'un graf no dirigit, $G = (V, A)$, on el conjunt de nodes és $V = \{1, 2, \ldots, 15\}$, i el conjunt d'arcs el separarem en 3 subconjunts disjunts, $A = H \cup D \cup B$, per a distingir entre arcs que connecten boles en horitzontal (H), en diagonal (D) i en antidiagonal (B).

a) Se trata de un grafo no dirigido, $G = (V, A)$, donde el conjunto de nodos es $V = \{1, 2, \ldots, 15\}$, y el conjunto de arcos lo separaremos en 3 subconjuntos disjuntos, $A = H \cup D \cup B$, para distinguir entre arcos que conectan bolas en horizontal (H), en diagonal (D) y en antidiagonal (B).

$$H = \{(1,2), (10,8), (8,11), (3,12), (12,4), (4,13), (14,5), (5,15), (15,6), (6,7)\}$$
$$D = \{(14,3), (3,10), (10,1), (1,9), (5,12), (12,8), (8,2), (15,4), (4,11), (6,13)\}$$
$$B = \{(3,5), (10,12), (12,15), (1,8), (8,4), (4,6), (9,2), (2,11), (11,13), (13,7)\}.$$

Gràficament,



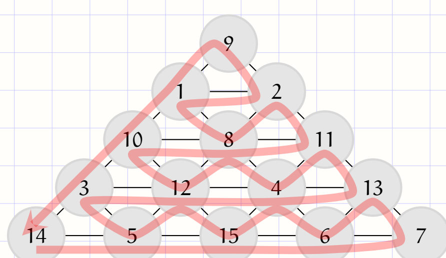
b) El grau de cada node s'ha indicat en la figura anterior mitjançant nombres petits prop de cada node. La suma dels graus serà

b) El grado de cada nodo se ha indicado en la figura anterior mediante números pequeños cerca de cada nodo. La suma de los grados será

$$S = 3 \cdot 3 \cdot 4 + 3 \cdot 2 + 3 \cdot 6 = 3(12 + 2 + 6) = 60.$$

c) Segons el teorema d'Euler, com que tots els nodes tenen grau parell, necessàriament hi ha d'haver un circuit eulerià. Per tant, la resposta és que sí. Existeix un recorregut (circular) eulerià. Per a trobar-ne un cal procedir per inspecció. Se'n mostra un en la figura següent.

c) Según el teorema de Euler, como todos los nodos tienen grado par, necesariamente tiene que haber un circuito euleriano. Por tanto, la respuesta es que sí. Existe un recorrido (circular) euleriano. Para encontrar uno hay que proceder por inspección. Se muestra uno en la figura siguiente.

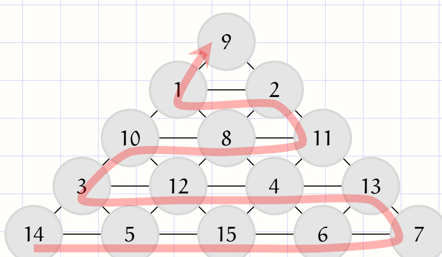


c) Per a veure si existeix o no algun recorregut hamiltonià només podem procedir per inspecció. És relativament fàcil trobar un recorregut hamiltonià si recorrem el graf per files (en sentits alternats) començant pel node 14 i acabant en el 9. També podem recórrer aquest mateix camí de manera simètrica (començant pel 7), i cada un dels dos de manera inversa (començant pel 9). I també podríem intercanviar el costat horitzontal del triangle per qualsevol dels altres dos en tots els recorreguts anteriors.

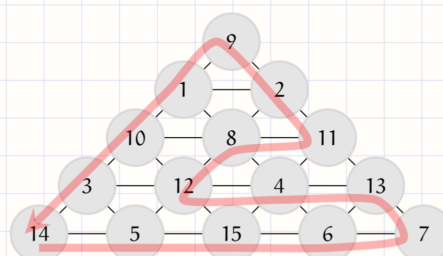
c) Para ver si existe o no algún recorrido hamilotniano sólo podemos proceder por inspección. Es relativamente fácil encontrar un recorrido hamiltoniano si recorremos el grafo por filas (en sentidos alternos) comenzando por el nodo 14 y acabando en el 9. También podemos recorrer este mismo camino de manera simétrica (comenzando por el 7), y cada uno de los dos de manera inversa (comenzando por el 9). Y también podríamos intercambiar el lado horizontal del triángulo por cualquiera de los otros dos en todos los recorridos anteriores.

Trobar un cicle hamiltonià és un poc més entretingut. En la figura següent es mostren un camí i un cicle hamiltonians sobre el graf.

Encontrar un ciclo hamiltoniano es un poco más entretenido. En la figura siguiente se muestran un camino y un ciclo hamiltonianos sobre el grafo.



Camí hamiltonià:
Camino hamiltoniano:
(14, 5, 15, 6, 7, 13, 4, 12, 3, 10, 8, 11, 2, 1, 9)



Ciclo hamiltonià:
Ciclo hamiltoniano:
(14, 5, 15, 6, 7, 13, 4, 12, 8, 11, 2, 9, 1, 10, 3, 14)

Problema 4.5:

[maxFulles]

Demostra per inducció que el nombre màxim de fulles en un arbre binari de n nodes és com a molt $\lceil \frac{n}{2} \rceil$. Quin seria el mínim?

Demuestra por inducción que el número máximo de hojas en un árbol binario de n nodos es como mucho $\lceil \frac{n}{2} \rceil$. ¿Cuál sería el mínimo?

BI

Un arbre binari buit té zero fulles i un arbre d'un node en té una. Es compleix, doncs, que

Un árbol binario vacío tiene cero hojas y un árbol de un nodo tiene una. Se cumple pues que

$$f_0 \le \left\lceil \frac{0}{2} \right\rceil = 0,$$
$$f_1 \le \left\lceil \frac{1}{2} \right\rceil = 1,$$

on f_n és el nombre màxim de fulles d'un arbre binari de n nodes.

donde f_n es el número máximo de hojas de un árbol binario de n nodos.

HI

$$\left[f_k \le \left\lceil \frac{k}{2} \right\rceil, \; \forall k < n \right].$$

PI

Qualsevol arbre binari de n nodes ($n > 0$), A_n, és format per dos subarbres de talles estrictament menors que n de manera que entre tots dos sumen $n-1$ nodes.

Cualquier árbol binario de n nodos ($n > 0$), A_n, està formado por dos subárboles de tallas estrictamente menores que n de manera que entre los dos suman $n-1$ nodos.

Siga $f(A)$ el nombre de fulles de qualsevol arbre A. Aleshores,

Sea $f(A)$ el número de hojas de cualquier árbol A. Entonces,

$$f_n = \max_{\forall A_n} f(A_n) = \max_{\forall A_n} \left(f(esq(A_n)) + f(dre(A_n)) \right) \le$$
$$\le \max_{0 \le \ell \le n-1} \left(f_\ell + f_{n-\ell-1} \right) \overset{\text{(HI)}}{\le} \max_{0 \le \ell \le n-1} \left(\left\lceil \frac{\ell}{2} \right\rceil + \left\lceil \frac{n-\ell-1}{2} \right\rceil \right).$$

En el cas que ℓ siga parell tenim que

En el caso que ℓ sea par tenemos que

$$\left\lceil \frac{\ell}{2} \right\rceil + \left\lceil \frac{n-\ell-1}{2} \right\rceil = \cancel{\frac{\ell}{2}} + \left\lceil \frac{n-1}{2} \right\rceil - \cancel{\frac{\ell}{2}} \le \left\lceil \frac{n}{2} \right\rceil.$$

I si és senar tenim també que

Y si es impar tenemos también que

$$\left\lceil\frac{\ell}{2}\right\rceil + \left\lceil\frac{n-\ell-1}{2}\right\rceil = \left\lceil\cancel{\frac{\ell}{2}}\right\rceil + \left\lceil\frac{n}{2}\right\rceil - \cancel{\frac{\ell+1}{2}} = \left\lceil\frac{n}{2}\right\rceil.$$

O siga, que tenim una fita superior per a f_n independent de ℓ, i per això podem assegurar que

O sea, que tenemos una cota superior para f_n independiente de ℓ, y por eso podemos asegurar que

$$f_n \leq \left\lceil\frac{n}{2}\right\rceil.$$

Una vegada completat el pas d'inducció on s'ha considerat qualsevol arbre binari d'un o més nodes, podem concloure que l'únic cas base estrictament necessari en aquesta demostració és $n = 0$.

Una vez completado el paso de inducción donde se ha considerado cualquier árbol binario de uno o más nodos, podemos concluir que el único caso base estrictamente necesario en esta demostración es $n = 0$.

El nombre mínim de fulles en un arbre binari de n nodes és 1, que es correspon amb un arbre binari degenerat on cap dels nodes té dos fills.

El número mínimo de hojas en un árbol binario de n nodos es 1, que se corresponde con un árbol binario degenerado donde ningún nodo tiene dos hijos.

Problema 4.6:

[arbresCoixos]

Un arbre binari és o està **coix** si és format per una arrel d'on pengen dos arbres coixos de manera que la profunditat de l'esquerre és la meitat de la del dret.

Un árbol binario es o está **cojo** *si está formado por una raíz de la que cuelgan dos árboles cojos de manera que la profundidad del izquierdo es la mitad que la del derecho.*

a) Converteix l'anterior descripció informal en una definició recursiva formal utilitzant lògica i la notació que cregues oportuna.

a) Convierte la anterior descripción informal en una definición recursiva formal utilizando lógica y la notación que creas oportuna.

b) Posa exemples d'arbres coixos d'almenys 3 profunditats diferents.

b) Da ejemplos de árboles cojos de al menos 3 profundidades diferentes.

Primer contestarem a) de manera informal (però precisa), donant els exemples que es demanen en b). I després donarem les definicions formals que es demanen en a).

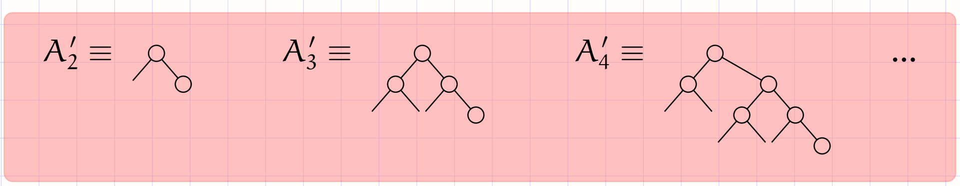
Primero contestaremos a) de manera informal (pero precisa), dando los ejemplos que se piden en b). Y después daremos las definiciones formales que se piden en a).

b) Ens ajudarà a decidir primer quin és l'arbre més menut que pot ser coix. L'opció més lògica és considerar l'**arbre buit**, $\emptyset$, com a coix, perquè aleshores un arbre d'un sol node (del qual pengen 2 subarbres buits, ja que són arbres binaris) serà evidentment coix, ja que 0 és la meitat de 0. Amb això ja tindríem 2 dels exemples que demana b):

b) Nos ayudará a decidir primero cuál es el árbol más pequeño que puede ser cojo. La opción más lógica es considerar el **árbol vacío**, *$\emptyset$, como cojo, porque entonces un árbol de un sólo nodo (del cual cuelgan 2 subárboles vacíos, ya que son árboles binarios) será evidentemente cojo, ya que 0 es la mitad de 0. Con esto ya tendríamos 2 de los ejemplos que pide b):*

Per a poder continuar cal decidir si interpretem **meitat** com a divisió entera, $\lfloor \frac{p}{2} \rfloor$, o com a divisió arredonida per dalt, $\lceil \frac{p}{2} \rceil$. En el primer cas, els següents arbres coixos que podríem formar serien

Para poder continuar hay que decidir si interpretamos **mitad** *como división entera, $\lfloor \frac{p}{2} \rfloor$, o como división redondeada por arriba, $\lceil \frac{p}{2} \rceil$. En el primer caso, los siguientes árboles cojos que podríamos formar serían*



que complirien la condició de ser coixos, ja que les profunditats dels subarbres de les arrels compleixen que

que cumplirían la condición de ser cojos, ya que las profundidades de los subárboles de las raíces cumplen que

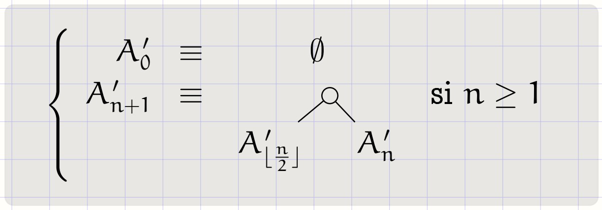
$$0 = \lfloor \tfrac{1}{2} \rfloor, \qquad 1 = \lfloor \tfrac{2}{2} \rfloor, \qquad 1 = \lfloor \tfrac{3}{2} \rfloor,$$

i tots els subarbres implicats són coixos.

y todos los subárboles implicados son cojos.

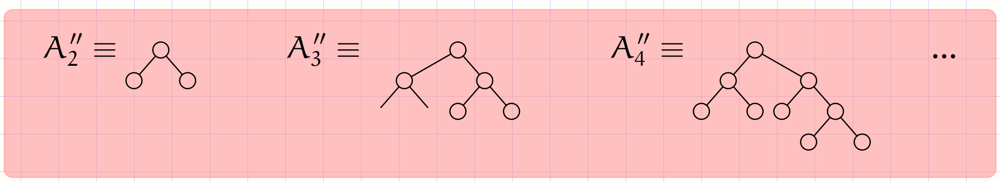
La definició recursiva en aquest primer cas només requereix un cas base i es podria escriure com

La definición recursiva en este primer caso sólo requiere un caso base y se podría escribir como



En canvi, en el segon cas en què la meitat s'arredoneix per dalt els següents arbres serien

En cambio, en el segundo caso en que la mitad se redondea por arriba los siguientes árboles serían



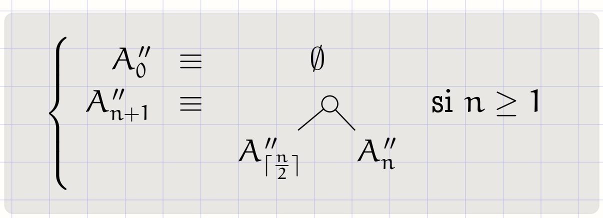
i en les seues arrels es compliria ara

y en sus raíces se cumpliría ahora

$$1 = \lceil \tfrac{1}{2} \rceil, \qquad 1 = \lceil \tfrac{2}{2} \rceil, \qquad 2 = \lceil \tfrac{3}{2} \rceil.$$

En el segon cas la definició recursiva seria aleshores

En el segundo caso la definición recursiva sería entonces



Tant en un cas com en l'altre hi ha un únic arbre coix per cada enter no negatiu: $A_0, A_1, A_2, \ldots$. I resulta que el subíndex es correspon amb la profunditat de cada arbre.

Tanto en un caso com en el otro hay un único árbol cojo por cada entero no negativo: $A_0, A_1, A_2, \ldots$. Y resulta que el subíndice se corresponde con la profundidad de cada árbol.

Què passaria si començàrem amb A_1 com a cas base?

¿Qué pasaría si comenzáramos con A_1 como caso base?

En el segon cas, res. Només caldria explicitar com a (únic) cas base

En el segundo caso, nada. Sólo habría que explicitar como (único) caso base

$A_1'' \equiv$ 

i la seqüència d'arbres seria ara $A_1'', A_2'', \ldots$

y la secuencia de árboles sería ahora $A_1'', A_2'', \ldots$

En canvi, en el primer cas necessitaríem definir **tant A_1' com A_2' com a casos base**. Això és perquè en el primer cas A_2' depèn de A_0' que ara no seria coix.

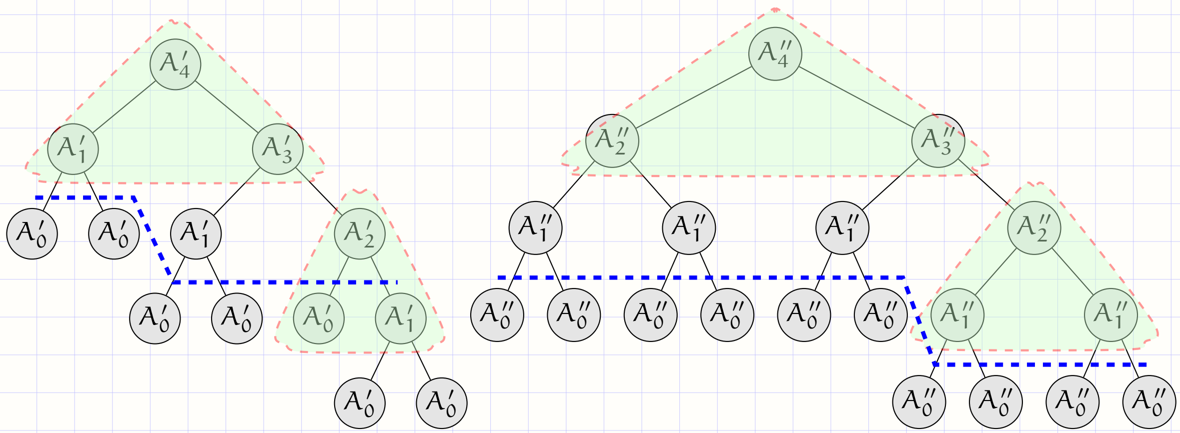
En cambio, en el primer caso necesitaríamos definir **tanto A_1' como A_2' como casos base**. *Esto es porque en el primer caso A_2' depende de A_0' que ahora no sería cojo.*

Aleshores tenim 2 opcions per a definir el cas base. I altres 2 per al cas recursiu. En total (variacions amb repetició de 2 elements agafats de 2 en 2), 4 definicions possibles lleugerament diferents.

Entonces tenemos 2 opciones para definir el caso base. Y otras 2 para el caso recursivo. En total (variaciones con repetición de 2 elementos tomados de 2 en 2), 4 definiciones posibles ligeramente diferentes.

Podem visualitzar gràficament les diferències entre les diferents definicions recursives si dibuixem els corresponents arbres de recursió per al cas d'arbres coixos de 4 nodes.

Podemos visualizar gráficamente las diferencias entre las diferentes definiciones recursivas si dibujamos los correspondientes árboles de recursión para el caso de árboles cojos de 4 nodos.



En la figura es mostren els dos arbres de recursió segons cada una de les definicions recursives en funció de com s'interprete la meitat d'un enter. S'han marcat les parts dels arbres que són diferents.

En la figura se muestran los dos árboles de recursión según cada una de la definiciones recursivas en función de como se interprete la mitad de un entero. Se han marcado las partes de los árboles que son diferentes.

També s'ha indicat per on s'hauria de retallar cada arbre en el cas que no es considerara el cas $n = 0$. En el cas de la dreta totes les fulles es correspondrien amb l'arbre A_1''. Però en el de l'esquerra, tenim fulles amb A_1 i amb A_2, els dos possibles casos base.

También se ha indicado por dónde se tendría que recortar cada árbol en el caso de que no se considerara el caso $n = 0$. En el caso de la derecha todas las hojas se corresponderían con el árbol A_1''. Pero en el de la izquierda, tenemos hojas con A_1' y con A_2', los dos posibles casos base.

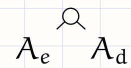
Considerarem a partir d'ací només la primera definició que té en compte l'arbre buit ($n = 0$) i arredoneix per baix. És a dir, arbres de recursió com el de l'esquerra de la figura anterior al complet.

Consideraremos a partir de aquí sólo la primera definición que tiene en cuenta el árbol vacío ($n = 0$) y redondea por abajo.

La definició formal podria constar de dues afirmacions (cas base i cas recursiu):

La definición formal podría constar de dos afirmaciones (caso base y caso recursivo):

C.B. $A_0 \equiv \emptyset$ és coix *es cojo* ,

C.R. $\forall A,\ A$ és coix (*es cojo*) sii
- A és de la forma (*es de la forma*) 

,
- A_e, A_d són coixos (*son cojos*),
- $p(A_e) = \lfloor \frac{p(A_d)}{2} \rfloor$,

on $p : \mathcal{A} \to \mathbb{Z}^+$ és una funció que ens dona la profunditat de qualsevol arbre, $A \in \mathcal{A}$.

donde $p : \mathcal{A} \to \mathbb{Z}^+$ es una función que nos da la profundidad de cualquier árbol, $A \in \mathcal{A}$.

Observem que no es diu explícitament en cap lloc que A siga binari. Però és que l'anterior definició sense la condició de les profunditats és **en realitat una definició recursiva d'arbre binari!**

Observamos que no se dice explícitamente en ningún sitio que A sea binario. Pero es que la anterior definición sin la condición de las profundidades es en realidad una definición recursiva de árbol binario!

Per això, es pot donar també una definició alternativa sabent que tot arbre binari (llevat del buit) ha de tenir un subarbre esquerre i un subarbre dret:

Por eso, se puede dar también una definición alternativa sabiendo que todo árbol binario (excepto el vacío) tiene que tener un subárbol izquierdo y un subárbol derecho:

C.B. $A_0 \equiv \emptyset$ és coix (*es cojo*),

C.R. $\forall A \in \mathcal{A}_B \backslash \{\emptyset\},\ A$ és coix (*es cojo*) sii
- $e(A), d(A)$ són coixos (*son cojos*),
- $p(e(A)) = \lfloor \frac{p(d(A))}{2} \rfloor$,

on $\mathcal{A}_B$ és el conjunt de tots els arbres binaris i

donde $\mathcal{A}_B$ es el conjunto de todos los árboles binarios y

$$e, d : \mathcal{A}_B \to \mathcal{A}_B$$

són funcions que ens donen cada subarbre de qualsevol arbre binari no buit, $A \in \mathcal{A}_B \backslash \{\emptyset\}$.

son funciones que nos dan cada subárbol de cualquier árbol binario no vacío, $A \in \mathcal{A}_B \backslash \{\emptyset\}$.

Si volguérem utilitzar explícitament predicats, podríem escriure:

Si quisiéramos utilizar explícitamente predicados, podríamos escribir:

$$\mathrm{coix}(\emptyset) \;\; \wedge$$

$$\forall A \neq \emptyset, \;\; \mathrm{coix}(A) \;\; \textbf{sii} \; \left[\mathrm{binari}(A) \;\; \wedge \right.$$

$$\left. \left(\exists E, D, P_d, \;\; \mathrm{esq}(A, E) \wedge \mathrm{dre}(A, E) \wedge \mathrm{pr}(D, P_d) \wedge \mathrm{pr}(E, \lfloor \frac{P_d}{2} \rfloor) \right) \right],$$

on ara $\mathrm{esq}, \mathrm{dre}$ i pr són predicats equivalents a les funcions anteriors. I A, E, D, P_d són variables que representaran arbres i enters, respectivament.

donde ahora $\mathrm{esq}, \mathrm{dre}$ y pr son predicados equivalentes a las funciones anteriores. Y A, E, D, P_d son variables que representarán árboles y enteros, respectivamente.

Alternativament, també podem expressar la mateixa definició amb molt poca notació però igual de formalment de la següent manera:

Alternativamente, también podemos expresar la misma definición con muy poca notación pero igual de formalmente de la siguiente manera:

- L'arbre buit és coix.

- *El árbol vacío es cojo.*

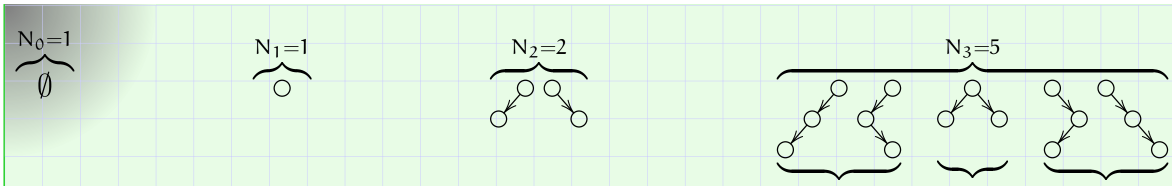
- Tot arbre no buit és coix **sii**
  - és binari,
  - la profunditat del seu subarbre esquerre és $\lfloor \frac{p}{2} \rfloor$, on p és la profunditat del seu subarbre dret,
  - I a més a més tots dos subarbres són coixos.

- *Todo árbol no vacío es cojo* **sii**
  - *es binario,*
  - *la profundidad de su subárbol izquierdo es $\lfloor \frac{p}{2} \rfloor$, donde p es la profundidad de su subárbol derecho,*
  - *Y además los dos subárboles son cojos.*

Problema 4.7: [arbresDiferents]

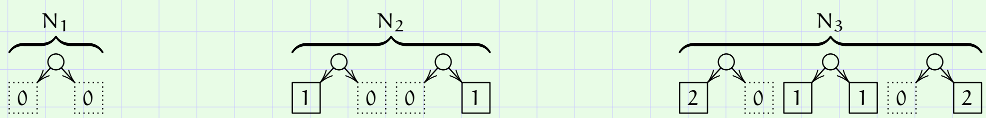
Siga N_n el nombre d'arbres binaris diferents de $n \geq 0$ nodes. Tenim per a $n \leq 3$ que

Sea N_n el número de árboles binarios diferentes de $n \geq 0$ nodos. Tenemos para $n \leq 3$ que



I podem establir les següents relacions on $\boxed{n}$ representa tots els arbres de n nodes.

Y podemos establecer las siguientes relaciones donde $\boxed{n}$ representa todos los árboles de n nodos.

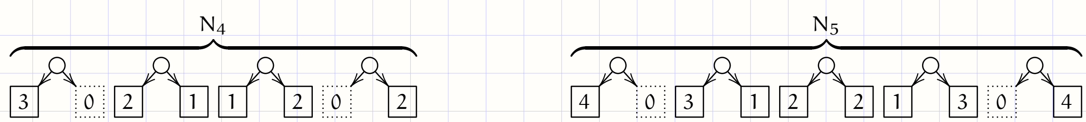


a) Descriu i explica el raonament recursiu anterior i arriba a una definició recursiva per a N_n,
b) Calcula els valors N_4 i N_5.

a) Describe y explica el razonamiento recursivo anterior y llega a una definición recursiva para N_n,
b) Calcula los valores N_4 y N_5.

Només observant la sèrie hauria de ser fàcil arribar a:

Sólo observando la serie debería ser fácil llegar a:



i fins i tot arribar a calcular els corresponents valors com a

e incluso llegar a calcular los correspondientes valores como

$$N_4 = 5 + 2 + 2 + 5 = 14,$$
$$N_5 = 14 + 5 + 2 \cdot 2 + 5 + 14 = 42.$$

Però és molt més important descobrir per què:

Pero es mucho más importante descubrir por qué:

Tot arbre binari de $n \geq 1$ nodes (llevat del cas $n = 0$) és necessàriament format per **una** arrel i **dos** subarbres que, entre tots dos, han de contenir els restants $n - 1$ nodes.

Todo árbol binario de $n \geq 1$ nodos (excepto en el caso $n = 0$) está necesariamente formado por **una** *raíz y* **dos** *subárboles que, entre los dos, tienen que contener los restantes $n - 1$ nodos.*

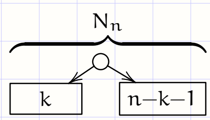
I de quantes maneres poden els 2 subarbres (que són distingibles en un arbre binari) repartir-se els $n-1$ nodes? Això són parells ordenats d'enters diferents menors que n (incloent-hi el zero). En total, n.

¿Y de cuantas maneras pueden los 2 subárboles (que son distinguibles en un árbol binario) repartirse los $n-1$ nodos? Eso son pares ordenados de enteros diferentes menores que n (incluyendo el cero). En total, n,

$$(n-1,0),(n-2,1),(n-3,2),\ldots,(1,n-2),(0,n-1).$$

O, en general

O, en general



$$k = 0,1,\ldots,n-1.$$

I quants arbres corresponen a cada parell d'enters?

¿Y cuántos árboles corresponden a cada par de enteros?

Si el subarbre esquerre té k nodes, hi haurà N_k opcions per a completar-lo. I altres N_{n-k-1} opcions per al dret. Pel **principi del producte**, per a cada valor de k tindrem aleshores $N_k \times N_{n-k-1}$ opcions. I el total serà la suma per a tot valor de k (pel **principi de la suma**),

Si el subárbol izquierdo tiene k nodos, habrá N_k opciones para completarlo. Y otras N_{n-k-1} opciones para el derecho. Por el **principio del producto***, para cada valor de k tendremos entonces $N_k \times N_{n-k-1}$ opciones. Y el total será la suma para todo valor de k (por el* **principio de la suma***),*

$$N_n = \sum_{k=0}^{n-1} N_k N_{n-k-1},$$

que juntament amb el cas base, $N_0 = 1$, ens duu a la definició recursiva,

que junto con el caso base, $N_0 = 1$, nos lleva a la definición recursiva,

$$\begin{cases} N_0 = 0, \\ N_n = \sum_{k=0}^{n-1} N_k N_{n-k-1}, \ \forall\, n \geq 1. \end{cases}$$

En aquest cas, la definició recursiva la podem expressar també simplement com a

En este caso, la definición recursiva la podemos expresar también simplemente como

$$N_n = \sum_{k=0}^{n-1} N_k N_{n-k-1}, \ \forall\, n \geq 0,$$

ja que el cas base ($n = 0$) es correspondria amb un sumatori buit que és igual a zero.

ya que el caso base ($n = 0$) se correspondería con un sumatorio vacío que es igual a cero.

Els successius valors de la funció es poden calcular com a:

Los sucesivos valores de la función se pueden calcular como:

$$
\begin{aligned}
N_0 &= 0,\\
N_1 &= \cancelto{1}{N_0} \cdot \cancelto{1}{N_0} = 1,\\
N_2 &= \cancelto{1}{N_1} \cdot \cancelto{1}{N_0} + \cancelto{1}{N_0} \cdot \cancelto{1}{N_1} = 2,\\
N_3 &= \cancelto{2}{N_2} \cdot 1 + 1 \cdot 1 + 1 \cdot \cancelto{2}{N_2} = 5,\\
N_4 &= \cancelto{5}{N_3} \cdot 1 + \cancelto{2}{N_2} \cdot 1 + 1 \cdot \cancelto{2}{N_2} + 1 \cdot \cancelto{5}{N_3} = 14,\\
N_5 &= \cancelto{14}{N_4} \cdot 1 + \cancelto{5}{N_3} \cdot 1 + 2 \cdot 2 + 1 \cdot \cancelto{5}{N_3} + 1 \cdot \cancelto{14}{N_4} = 42.
\end{aligned}
$$

Els valors d'aquesta successió es coneixen en la literatura com els **nombres de Catalan**.

Los valores de esta sucesión se conocen en la literatura como los **números de Catalan**.

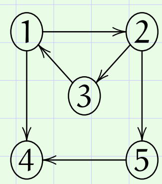
4.5 Problemes proposats

Problemas Propuestos

Problema 4.8: [grafSobre]

Considera el graf de la figura i calcula: a) els graus de tots els nodes, b) la matriu d'adjacència, c) un recorregut que passe pel màxim d'arcs sense repetir-ne cap, d) un recorregut que passe pel màxim de nodes sense repetir-ne cap, e) un arbre d'extensió que siga binari i un altre que siga ternari, i f) digues quants cicles diferents conté el graf i quins són.

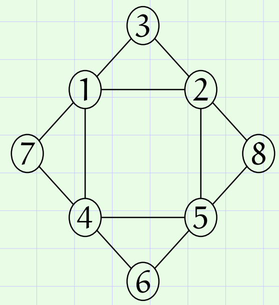
Considera el grafo de la figura y calcula: a) los grados de todos los nodos, b) la matriz de adyacencia, c) un recorrido que pase por el máximo de arcos sin repetir ninguno, d) un recorrido que pase por el máximo de nodos sin repetir ninguno, e) un árbol de extensión que sea binario y otro que sea ternario, y f) di cuántos ciclos diferentes contiene el grafo y cuáles son.



Problema 4.9: [grafRomb]

Siga G el graf de la figura. a) Digues quants cliques (grafs totalment connexos) diferents conté G. b) Calcula un recorregut hamiltonià en G (o explica per què no n'hi ha). c) Calcula un recorregut eulerià en G (o explica per què no n'hi ha). d) Digues quants cicles té el recorregut anterior. e) Calcula un recorregut eulerià sense cicles en G (o explica per què no n'hi ha).

Sea G *el grafo de la figura. a) Di cuantos cliques (grafos totalmente conexos) diferentes contiene* G*. b) Calcula un recorrido Hamiltoniano en* G *(o explica por qué no hay ninguno). c) Calcula un recorrido Euleriano en* G *(o explica por qué no hay ninguno). d) Di cuantos ciclos tiene el recorrido anterior. e) Calcula un recorrido Euleriano sin ciclos en* G *(o explica por qué no hay ninguno).*

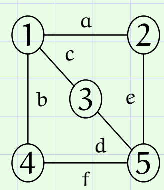


Problema 4.10:

[grafZ]

Considera el graf de la figura i respon a les qüestions: a) La matriu d'adjacència. b) Un recorregut eulerià, si existeix; i si no, digues per què. c) Un camí eulerià, si existeix; i si no, digues per què. d) Hi ha algun subgraf que continga algun cicle eulerià? Quants? Quins?

Considera el grafo de la figura y contesta: a) su matriz de adyacencia, b) un recorrido Euleriano, si existe. Y si no, di por qué, c) un camino Euleriano, si existe. Y si no, di por qué, d) hay algún subgrafo que contenga algún ciclo Euleriano? Cuántos? Cuáles?



Problema 4.11:

[components]

Considera la proposició de la pàgina 181 que relaciona els nombres de nodes, arcs i components connexes d'un graf i demostra-la per inducció **sobre el nombre de nodes**.

Considera la proposición de la página 181 que relaciona los números de nodos, arcos y componentes conexas de un grafo y demuéstrala por inducción **sobre el número de nodos**.

Problema 4.12: [sumaGrausInd]

Demostra per inducció que la suma dels graus d'un arbre ternari és un nombre parell. Enuncia formalment el que es vol demostrar (usant lògica de predicats), justifica el tipus d'inducció i raona clarament els diferents passos.

Demuestra por inducción que la suma de los grados de un árbol ternario es un número par. Enuncia formalmente lo que se quiere demostrar (usando lógica de predicados), justifica el tipo de inducción i razona claramente los diferentes pasos.

Problema 4.13: [arbresEquilibrats]

Demostra per inducció que en un arbre binari equilibrat de n nodes, la profunditat, h, és logarítmica i com a molt $2\lg n$. Equivalentment,

Demuestra por inducción que en un árbol binario equilibrado de n nodos, la profundidad, h, es logarítmica y como mucho $2\lg n$. Equivalentemente,

$$h \leq 2\lg n \quad \equiv \quad n \geq 2^{\frac{h}{2}}.$$

A. El llenguatge PROLOG--

El lenguaje PROLOG--. *A*

A.1 Definicions

Definiciones

Per tal d'il·lustrar conceptes relacionats amb la lògica i la recursió considerarem un llenguatge de programació que és un subconjunt (estricte) del llenguatge de programació Prolog. Per aquest motiu anomenarem aquest llenguatge `PROLOG--`.

Para ilustrar conceptos relacionados con la lógica y la recursión consideraremos un lenguaje de programación que es un subconjunto (estricto) del lenguaje de programación Prolog. Por este motivo llamaremos a este lenguaje `PROLOG--`*.*

És important subratllar que no es tracta d'un llenguatge de programació complet ja que només es pretén utilitzar la part purament lògica de Prolog. Per això exclourem explícitament aquells aspectes no lògics de Prolog com ara l'entrada/sortida i sobretot l'operador **tall** (*cut*).

Es importante subrayar que no se trata de un lenguaje de programación completo puesto que sólo se pretende utilizar la parte puramente lógica de Prolog. Por este motivo excluiremos explícitamente aquellos aspectos no lógicos de Prolog como por ejemplo la entrada/salida y sobre todo el operador **corte** *(*cut*).*

La implementació que s'ha fet servir per als exemples i problemes i que es recomana és SWI-Prolog†. En particular s'ha usat la versió 7.6.4. No obstant això, tots els exemples de `PROLOG--` haurien de funcionar perfectament amb qualsevol altra implementació de Prolog.

La implementación que se ha usado para los ejemplos y problemas y que se recomienda es SWI-Prolog†. En particular se ha usado la versión 7.6.4. Sin embargo, todos los ejemplos de `PROLOG--` *deberían funcionar perfectamente sobre cualquier otra implementación de Prolog.*

A.1.1 Elements

Elementos

Els components d'un programa en PROLOG-- són bàsicament els mateixos que s'han definit en la lògica de predicats.

Los componentes de un programa en PROLOG-- son básicamente los mismos que se han definido en la lógica de predicados.

D'una banda tenim les **constants** que poden ser de tipus **numèric** (enter o real) o **atòmic**. Els nombres enters o reals s'escriuen usant les notacions habituals.

Por una parte tenemos las **constantes** *que pueden ser de tipo* **numérico** *(entero o real) o* **atómico**. *Los números enteros o reales se escriben usando las notaciones habituales.*

```
1024          3.14          2.4e-3
```

Els **àtoms** representen elements de qualsevol domini i són cadenes de caràcters alfanumèrics (incloent-hi el símbol subratllat, "_") que comencen amb una lletra minúscula.

Los **átomos** *representan elementos de cualquier dominio y son cadenas de caracteres alfanuméricos (incluyendo el símbolo subrayado, "_") que comienzan con una letra minúscula.*

```
a          aaaa          aB_c123_Ab
```

D'altra banda, les **variables** en PROLOG-- s'escriuran com a cadenes de caràcters alfanumèrics (incloent "_") que comencen amb una lletra majúscula.

Por otra parte, las **variables** *en PROLOG-- se escribirán como cadenas de caracteres alfanuméricos (incluyendo "_") que comiencen con una letra mayúscula.*

```
A          X1          Ab_102_X
```

Existeix una variable especial que és la **variable anònima**, "_", que representarà una variable diferent cada vegada que aparega en una expressió.

Existe una variable especial que es la **variable anónima**, *"_", que representará una variable diferente cada vez que aparezca en una expresión.*

†Aquesta implementació és de llicència pública i es pot trobar en https://www.swi-prolog.org
Esta implementación es de licencia pública y se puede encontrar en

Els àtoms, nombres i variables són els **termes** bàsics o simples en PROLOG--. Però a banda hi ha dues maneres d'agrupar termes per a formar-ne de nous. D'una banda els **functors** ja introduïts en definir les fórmules ben formades en lògica de predicats (pàg. 82), i d'altra les **llistes**, també introduïdes en la secció 3.1.1 (pàg. 131).

Los átomos, números y variables son los **términos** *básicos o simples en* PROLOG--. *Pero aparte existen dos maneras de agrupar términos para crear otros nuevos. Por un lado los* **functores** *ya introducidos al definir la fórmulas bien formadas en lógica de predicados (pág. 82), y por otro las* **listas**, *también introducidas en la sección 3.1.1 (pág. 131).*

Els functors tenen un nom com el dels àtoms i un nombre variable d'arguments que han de ser necessàriament **termes**. També és possible agrupar termes formant tuples la qual cosa es pot entendre com un functor sense nom.

Los functores tienen un nombre como el de los átomos y un número variable de argumentos que han de ser necesariamente **términos**. *También es posible agrupar términos formando tuplas en cuyo caso se puede entender como un functor sin nombre.*

```
a(b,X)        f(a,X,_)        f(a,g(X,b))        (a,g(X,b),c)
```

Una **llista** és una seqüència de zero o més termes separats per comes i delimitat per claudàtors.

Una **lista** *es una secuencia de cero o más términos separados por comas y delimitado por corchetes.*

```
[b,X]        [a,X,_]        [a,[X,b],[]]
```

Una variable o un functor o llista que conté variables és un terme **genèric**, que pot representar diversos elements d'un determinat domini. En un moment donat una variable es pot instanciar a qualsevol altre terme, fins i tot a una altra variable.

Una variable o un functor o lista que contiene variables es un término **genérico**, *que puede representar diversos elementos de un determinado dominio. En un momento dado una variable se puede instanciar a cualquier otro término, incluso a otra variable.*

A banda dels termes, l'altre component bàsic dels programes en PROLOG-- són els **predicats**. Un **predicat** té associat un nom (cadena de caràcters alfanumèrics que comença amb minúscula) i una **aritat** o nombre d'arguments associats. Ho escriurem resumidament com `nom/#`, on # és l'aritat del predicat `nom`. En el següent exemple es mostren els predicats `aa/3`, `aa/2` i `aa/1`.

Aparte de los términos, el otro componente básico de los programas en PROLOG-- son los **predicados**. *Un* **predicado** *tiene asociado un nombre (cadena de caracteres alfanuméricos que comienza con minúscula) y una* **aridad** *o número de argumentos asociados. Lo escribiremos resumidamente como* `nombre/#`, *donde # es la aridad del predicado* `nombre`. *En el siguiente ejemplo se muestran los predicados* `aa/3`, `aa/2` *y* `aa/1`.

```
aa(a,X,_)          aa(a,g(X,b))        aa(a)
```

En lògica de primer ordre, els arguments d'un predicat han de ser necessàriament termes per la qual cosa no pot haver cap confusió en expressions com

En lógica de primer orden, los argumentos de un predicado tienen que ser necesariamente términos por lo que no puede haber ninguna confusión en expresiones como

```
aa(aa,aa(_),aa(_,_))
```

on el mateix identificador, `aa`, és alhora predicat amb aritat 3 , àtom, functor amb aritat 1 i functor amb aritat 2. De la mateixa manera, predicats amb idèntic nom i aritat diferent poden coexistir sense confusió.

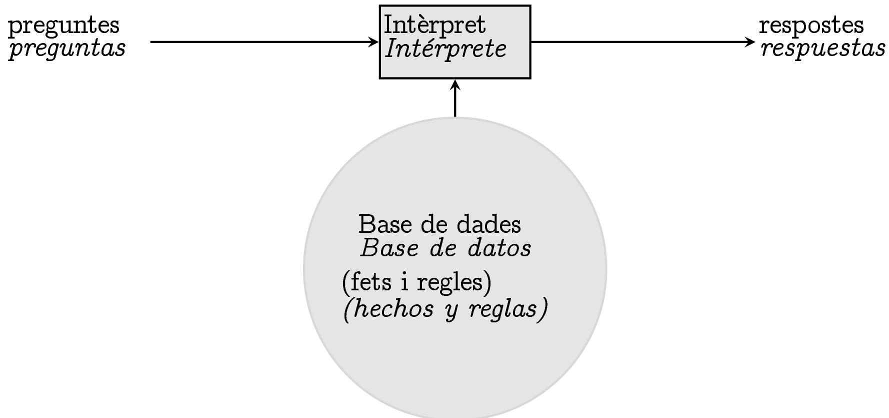
donde el mismo identificador, `aa`, *es al mismo tiempo predicado con aridad 3, átomo, functor con aridad 1 y functor con aridad 2. De la misma manera, predicados con idéntico nombre y aridad diferente pueden coexistir sin confusión.*

A.1.2 Estructura

Estructura

Un programa en PROLOG-- consisteix en la definició de zero o més predicats juntament amb una o més **qüestions** o **preguntes**. El resultat del programa és format aleshores per les respostes a les qüestions, donats els predicats definits. Un esquema general es mostra en la següent figura.

Un programa en PROLOG-- consiste en la definición de cero o más predicados junto con una o más **preguntas** *o* **cuestiones**. *El resultado del programa está formado entonces por las respuestas a las preguntas, dados los predicados definidos. Un esquema general se muestra en la siguiente figura.*



Cada predicat es defineix mitjançant una o més clàusules (de Horn, pàg. 65) que poden ser de dos tipus, **fets** o **regles**, i totes dues han d'acabar amb el símbol "." (punt).

Cada predicado se define mediante una o más cláusulas (de Horn, pág. 65) que pueden ser de dos tipos, **hechos** *o* **reglas***, y las dos tienen que acabar con el símobolo "." (punto).*

Un **fet** és una afirmació sobre la veracitat d'alguna cosa i que pot contenir o no variables.

Un **hecho** *es una afirmación sobre la veracidad de algo y que puede contener o no variables.*

En `PROLOG--` els fets només poden ser **fórmules atòmiques**. És a dir, han d'estar formats per un únic predicat (de qualsevol aritat) juntament amb tots els seus arguments i sense cap connectiva lògica. Això vol dir que en `PROLOG--` no podem afirmar coses com

En `PROLOG--` *los hechos sólo pueden ser* **fórmulas atómicas***. Es decir, tienen que estar formados por un único predicado (de cualquier aridad) junto con todos sus argumentos y sin ninguna conectiva lógica. Esto significa que en* `PROLOG--` *no se pueden afirmar cosas como*

$$P \vee Q, \quad \neg(P \wedge Q)$$

Si que és possible en canvi afirmar la conjunció de diverses coses simplement en afirmar separadament cada una d'elles.

Sí que es posible en cambio afirmar la conjunción de varias cosas simplemente afirmando separadamente cada una de ellas.

Una **regla** en `PROLOG--` és una implicació l'antecedent de la qual és una expressió formada a partir de predicats usant les connectives "," (conjunció) i ";" (disjunció). El conseqüent en canvi ha de ser una fórmula atòmica.

Una regla en `PROLOG--` *es una implicación cuyo antecedente es una expresión formada a partir de predicados usando las conectivas "," (conjunción) y ";" (disyunción). El consecuente en cambio tiene que ser una fórmula atómica.*

En les implicacions s'escriu primer el conseqüent seguit dels símbols ":-" que es llegeixen "si" i a continuació l'antecedent, normalment en línies diferents i indentat respecte del conseqüent. A continuació es mostren exemples tant de fets com de regles en PROLOG--.

En las implicaciones se escribe primero el consecuente seguido de los símbolos ":-" que se leen "si" y a continuación el antecedente, normalmente el líneas diferentes y indentado respecto al consecuente. A continuación se muestran ejemplos tanto de hechos como de reglas en PROLOG--.

```
1.  plou.

2.  color(roig).
3.  color(gris).

4.  trist :-      color(roig).
5.  trist :-      plou.

6.  trist :-      color(roig); plou.
```

En la línia 1 es mostra la definició del predicat `plou/0`. Les línies 2 i 3 il·lustren la definició de `color/1`. En els 3 casos es té una fórmula atòmica. En canvi, les línies 4 i 5 constitueixen una definició de `trist/0` mitjançant regles. En realitat, l'enumeració de regles (o fets) és equivalent a la seua conjunció per la qual cosa (i en aplicar la pseudodistributivitat per l'esquerra de la implicació) les línies 4 i 5 són equivalents a la regla de la línia 6 on l'antecedent és format per una disjunció.

En la línea 1 se muestra la definición del predicado `plou/0`. *Las líneas 2 y 3 ilustran la definición de* `color/1`. *En los 3 casos se tiene una fórmula atómica. En cambio, las líneas 4 y 5 constituyen una definición de* `trist/0` *mediante reglas. En realidad, la enumeración de reglas (o hechos) es equivalente a su conjunción por lo que (aplicando la pseudodistributividad por la izquierda de la implicación) las líneas 4 y 5 son equivalentes a la regla de la línea 6 cuyo antecendente está formado por una disyunción.*

En PROLOG-- sempre evitarem l'ús de la disjunció fins on siga possible i sempre serà preferible definir predicats usant (la conjunció de) diverses regles en lloc d'utilitzar disjuncions en l'antecedent.

En PROLOG-- *siempre evitaremos el uso de la disyunción hasta donde sea posible y siempre será preferible definir predicados usando (la conjunción de) varias reglas en lugar de utilizar disyunciones en el antecedente.*

En PROLOG-- es pot escriure la negació d'un predicat o expressió amb predicats mitjançant el pseudopredicat `not/1` de manera que l'expressió `not(expressioLogica)` és certa si la veritat de `expressioLogica` no es pot demostrar. Cal tindre la precaució d'envoltar les expressions que continguen connectives amb parèntesis per tal que el compilador interprete correctament l'únic argument de `not/1`.

En PROLOG-- se puede escribir la negación de un predicado o expresión con predicados mediante el pseudopredicado `not/1` de manera que la expresión `not(expresionLogica)` es cierta si la verdad de `expresionLogica` no se puede demostrar. Hay que tener la precaución de rodear las expresiones que contengan conectivas con paréntesis para que el compilador interprete correctamente el único argumento de `not/1`.

En PROLOG-- també evitarem l'ús explícit de la negació ja que aquesta no es correspon exactament amb la negació lògica.

En PROLOG-- también evitaremos el uso explícito de la negación ya que ésta no se corresponde exactamente con la negación lógica.

Una **qüestió** o **pregunta** en PROLOG-- és qualsevol expressió formada amb predicats, parèntesis, conjuncions i disjuncions.

*Una **cuestión** o **pregunta** en PROLOG-- es cualquier expresión formada con predicados, paréntesis, conjunciones y disyunciones.*

Si l'expressió és **tancada**, la resposta serà vertader o fals. Si l'expressió conté variables (fórmula **oberta**) la resposta, que podrà ser única o no, vindrà donada pels valors de les variables que fan vertadera l'expressió.

*Si la expresión es **cerrada**, la respuesta será verdadero o falso. Si la expresión contiene variables (fórmula **abierta**) la respuesta, que podrá ser única o no, vendrá dada por los valores de las variables que hacen verdadera la expresión.*

En qualsevol cas, quan la contestació és fals, no significa necessàriament que l'expressió corresponent a la pregunta siga lògicament falsa, sinó que PROLOG-- no ha sigut capaç de demostrar la seua veracitat. Això és el que s'anomena **negació per fallada**.

*En cualquier caso, cuando la contestación es falso, no significa necesariamente que la expresión correspondiente a la pregunta sea lógicamente falsa, sino que PROLOG-- no ha sido capaz de demostrar su veracidad. Esto es lo que se conoce como **negación por fallo**.*

Sintàcticament, les preguntes en PROLOG-- es poden afegir a les definicions dels predicats o es poden introduir de manera interactiva. Només en el primer cas s'escriuen precedides dels símbols ":-". És a dir, com una regla sense conseqüent.

Sintácticamente, las preguntas en PROLOG-- se pueden añadir a las definiciones de los predicados o se pueden introducir de manera interactiva. Sólo en el primer caso se escriben precedidas de los símbolos ":-". Es decir, como una regla sin consecuente.

A.1.3 Unificació i comparació de termes

Unificación y comparación de términos

La igualtat de dos termes qualssevol es pot comprovar usant l'operador "==". L'expressió `terme1 == terme2` constitueix un predicat i és cert sii els 2 termes són lògicament idèntics.

La igualdad de dos términos cualesquiera se puede comprobar usando el operador "==". La expresión `termino1 == termino2` *constituye un predicado y es cierto sii los 2 términos son lógicamente idénticos.*

La **unificació** de dos termes que en general contenen variables s'escriu com a `terme1 = terme2`. Aquesta expressió és certa sii existeixen valors per a les variables dels termes que facen que els dos termes siguen idèntics.

La **unificación** *de dos términos que en general contienen variables se escribe como* `termino1 = termino2`. *Esta expresión es cierta sii existen valores para las variables de los términos que hagan que los dos términos sean idénticos.*

L'anomenat algorisme d'unificació és el que s'encarrega de comprovar-la i també de calcular els conjunts de valors per a les variables. De la mateixa manera que en el cas de la comparació de termes, la expressió `terme1 = terme2` constitueix un predicat.

El llamado algoritmo de unificación es el que se encarga de comprobarla y también de calcular los conjuntos de valores para las variables. De la misma manera que en el caso de la comparación de términos, la expresión `termino1 = termino2` *constituye un predicado.*

Existeixen també versións negades tant per a la comparació de termes com per a la unificació que s'escriuen com a "\==" i "\=", respectivament.

Existen también versiones negadas tanto para la comparación de términos como para la unificación que se escriben como "\==" y "\=", respectivamente.

A continuació es mostren alguns exemples de unificació i comparació de termes.

A continuación se muestran algunos ejemplos de unificación y comparación de términos.

```
?- f(a,b)==f(a,b).
true.

?- f(X,b)==f(a,b).
false.

?- f(X,b)=f(a,b).
X = a.
```

```
?- f(X,Y,Z)=f(Y,X,X).
X = Y, Y = Z.

?- f(X,b,Z)=f(a,b,1);f(a,Y,Z)=f(a,b,2).
X = a, Z = 1;
Y = b, Z = 2.
```

Si hi ha més d'un conjunt de valors per a les variables, l'intèrpret donarà diverses respostes que caldrà obtindre en introduir el caràcter ";" o també polsant la barra d'espai del teclat (segons implementacions). Aquest caràcter es mostra en pantalla o no també segons implementacions.

Si hay más de un conjunto de valores para las variables, el intérprete dará varias respuestas que habrá que obtener introduciendo el carácter ";" o también pulsando la barra espaciadora del teclado (según implementaciones).

En l'últim exemple del quadre anterior, la primera resposta diu que la variable `X` ha de valdre (o s'ha d'instanciar a) `a` i la variable `Z` a 1, i que la variable `Y` pot valdre qualsevol cosa (o pot no estar instanciada a cap valor) ja que no es diu res d'ella. En la segona resposta la variable `Z` ha d'estar instanciada a 2 i les variables `X` i `Y` han intercanviat els seus papers.

En el último ejemplo del cuadro anterior, la primera respuesta dice que la variable `X` *tiene que valer (o se tiene que instanciar a)* `a` *y la variable* `Z` *a 1, y que la variable* `Y` *puede valer cualquier cosa (o puede no estar instanciada a ningún valor) ya que no se dice nada de ella. En la segunda respuesta la variable* `Z` *tiene que estar instanciada a 2 y las variables* `X` *e* `Y` *han intercambiado sus papeles.*

A.1.4 Resolució

Resolución

En els exemples anteriors hem vist programes (interactius) sense cap predicat definit, on les preguntes es donen com a entrada a l'intèrpret (identificat pel *prompt* "?-"). No obstant això, la situació més normal és que el programa continga diversos predicats la definició dels quals es trobe en un arxiu amb extensió "`.pl`", com per exemple

En los ejemplos anteriores hemos visto programas (interactivos) sin ningún predicado definido, donde las preguntas se dan como entrada al intérprete (identificado por el prompt "?-"). Sin embargo, la situación más normal es que el programa contenga varios predicados cuya definición se encuentra en algún archivo con extensión "`.pl`*", como por ejemplo*

```
% Arxiu ‘‘prova.pl’’
cotxe(bmv).
moto(ducati).
vehicle(reliant).
vehicle(X) :-
        cotxe(X).
vehicle(X) :-
        moto(X).
```

Aquest arxiu es pot carregar en PROLOG-- equivalentment mitjançant qualsevol de les quatre preguntes següents.

Este archivo se puede cargar en PROLOG-- *de manera equivalente mediante cualquiera de las cuatro preguntas siguientes.*

```
?-  [prova].
    true.
?-  ['prova.pl'].
    true.
?-  consult(prova).
    true.
?-  consult('prova.pl').
    true.
```

La contestació és la mateixa encara que pot dependre de diferents implementacions. Si el nom de l'arxiu (sense extensió) no és (sintàcticament) un àtom, aleshores és necessari envoltar-lo amb cometes simples.

La contestación es la misma aunque puede depender de diferentes implementaciones. si el nombre del archivo (sin extensión) no es (sintácticamente) un átomo, entonces es necesario encerrarlo entre comillas simples.

A banda del predicat `consult/1`, també es poden fer servir els predicats predefinits `ls/0`, `pwd/0` i `cd/1` per llistar, comprovar i canviar de directori.

Aparte del predicado `consult/1`, *también se pueden usar los predicados predefinidos* `ls/0`, `pwd/0` *y* `cd/1` *para listar, comprobar y cambiar de directorio.*

En el cas que l'arxiu continga també preguntes, aquestes no han de contindre variables no lligades i el seu resultat ha de ser vertader. Altrament, l'intèrpret emitirà un avís (*warning*)

*En el caso que el archivo contenga también preguntas, éstas no deben contener variables no ligadas y su resultado debe ser verdadero. En otro caso, el intérprete emitirá un aviso (*warning*).*

Al conjunt de definicions format per fets i regles que es carrega en l'intèrpret se'l coneix com a base de dades. Una vegada carregats, fets i regles esdevenen axiomes del sistema i es poden fer preguntes com, per exemple

Al conjunto de definiciones formado por hechos y reglas que se carga en el intérprete se lo conoce como base de datos. Una vez cargados, hechos y reglas se convierten en axiomas del sistema y se pueden hacer preguntas como, por ejemplo

```
?-  cotxe(bmv).
    true.
?-  moto(bmv).
    false.
?-  vehicle(Y).
    Y=reliant ;
    Y=bmv ;
    Y=ducati.
?-  vehicle(X), not(moto(X);cotxe(X)).
    X=reliant
    false.
```

Quan les preguntes consisteixen en predicats, PROLOG-- recorre la base de dades de manera seqüencial i intenta unificar el predicat de la pregunta amb algun fet o amb el cap (conseqüent) d'alguna regla. Si es troba primer un fet, PROLOG-- respon cert, o amb els valors de les variables que fan cert el predicat, com en els exemples anteriors sobre unificació.

Cuando las preguntas consisten en predicados, PROLOG-- *recorre la base de datos de manera secuencial e intenta unificar el predicado de la pregunta con algún hecho o con la cabeza (consecuente) de alguna regla. Si se encuentra primero un hecho,* PROLOG-- *responde cierto, o con los valores de las variables que hacen cierto el predicado, como en los ejemplos anteriores sobre unificación.*

Si el que es troba és (el conseqüent o cap d') una regla, aleshores tot l'antecedent esdevé una nova pregunta i el procés continua recursivament fins que no queden possibilitats. Cada vegada que ocorre una unificació, algunes variables poden ser instanciades a diferents elements. Si hi ha més d'una possibilitat, l'intèrpret anirà provant-les totes si s'obté com a resultat fals o si s'introdueix la tecla ";" després d'una contestació positiva.

Si lo que se encuentra es (el consecuente o cabeza de) una regla, entonces todo el antecedente se convierte en una nueva pregunta y el proceso continua recursivamente hasta que no queden más posibilidades. Cada vez que ocurre una unificación, algunas variables pueden ser instanciadas a diferentes elementos. Si hay más de una posibilidad, el intérprete irá probándolas todas si se obtiene como resultado falso o si se introduce la tecla ";" después de una contestación positiva.

Per exemple, la pregunta "`vehicle(Y).`" donaria lloc a una primera unificació amb el fet de la línia 3 que faria que la variable s'instanciara al valor "`reliant`" i provocaria la primera resposta.

Por ejemplo, la pregunta "`vehicle(Y).`" daría lugar a una primera unificación con el hecho de la línea 3 que haría que la variable se instanciara al valor "`reliant`" y provocaría la primera respuesta.

Si aleshores polsem "`;`" l'intèrpret torna enrere (desfà la instanciació de la variable) i troba una nova unificació en la línia 3 (la variable `Y` s'instanciaria a la variable `X`, que continuaria sent una variable lliure).

Si entonces pulsamos "`;`" el intérprete vuelve atrás (deshace la instanciación de la variable) y encuentra una nueva unificación en la línea 3 (la variable `Y` se instanciaría a la variable `X`, que continuaría siendo una variable libre).

Això donaria lloc a la nova pregunta "`cotxe(X).`" que trobaria una única unificació en la línia 1 que donaria lloc al seu temps a la segona resposta.

Esto daría lugar a la nueva pregunta "`cotxe(X).`" que encontraría una única unificación en la línea 1 que daría lugar a su vez a la segunda respuesta.

Si tornem a polsar "`;`" l'intèrpret tornarà enrere i aquesta vegada trobarà una unificació en la línia 5 que de la mateixa manera que abans dispararà una nova pregunta que donarà lloc a una darrera unificació en la línia 2 i a la tercera i última resposta.

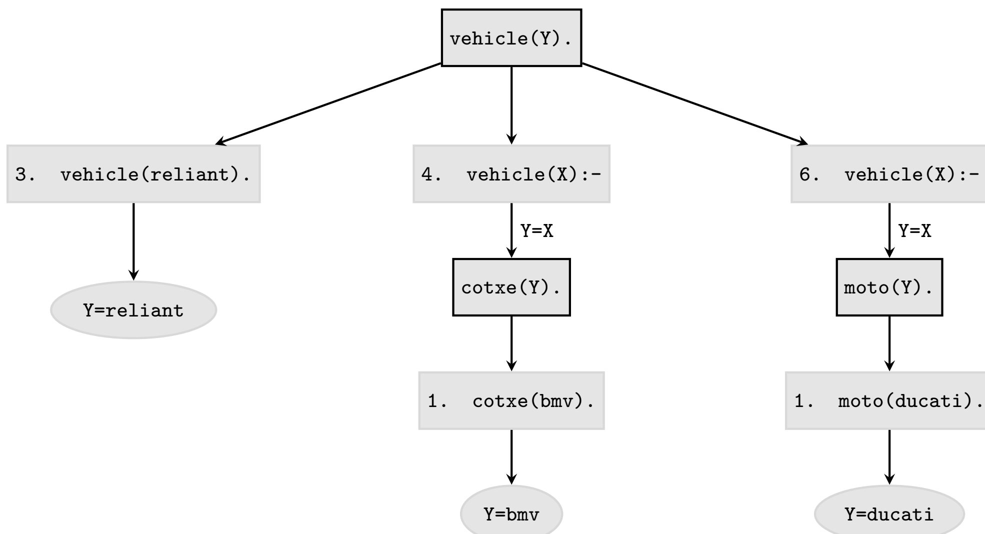
Si volvemos a pulsar "`;`" el intérprete volverá atrás y esta vez encontrará una unificación en la línea 5 que de la misma manera que antes disparará una nueva pregunta que dará lugar a una última unificación en la línea 2 y a la tercera y última respuesta.

Aquest procés és el que es coneix com resolució que és en realitat un algorisme de retrocés el funcionament del qual es pot representar de manera arborescent.

Este proceso es lo que se conoce como resolución que es en realidad un algoritmo de vuelta atrás cuyo funcionamiento se puede representar de manera arborescente.

Per exemple, el càlcul anterior es pot representar mitjançant el següent arbre on les preguntes es mostren en rectangles emmarcats amb línies i les respostes en caixes ovalades.

Por ejemplo, el cálculo anterior se puede representar mediante el siguiente árbol donde las preguntas se muestran en rectángulos enmarcados con líneas y las respuestas en cajas ovaladas.



A.2 Predicats recursius en PROLOG--
Predicados recursivos en PROLOG--

La potència de PROLOG-- com a llenguatge de programació rau en la unificació, la resolució i en la recursió. Considerem el següent predicat i les contestacions de l'intèrpret a algunes preguntes.

La potencia de PROLOG-- *como lenguaje de programación recae en la unificación, la resolución y en la recursión. Consideramos el siguiente predicado y las contestaciones del intérprete a algunas preguntas.*

```
simple(a).
simple(X-a) :-
   simple(X).
```

```
?- simple(a).
   true.
?- simple(b).
   false.
?- simple(a-a-a).
   true.
?- simple(Y).
   Y = a ;
   Y = a-a ;
   Y = a-a-a
   ...
```

Es tracta d'un predicat recursiu unari l'argument del qual són expressions formades amb l'àtom `a` i l'operador "-". Quan es fa la pregunta "`simple(a-a-a).`", l'intèrpret la unificarà amb la regla de la línia 2 la qual cosa porta a que `X=a-a` i a que es dispare la nova pregunta "`simple(a-a).`". A continuació es tornaria a repetir el mateix i aleshores la nova pregunta seria "`simple(a).`", que ja no s'unificaria amb la regla sinó amb el fet de la línia 1.

Se trata de un predicado recursivo unario cuyo argumento son expresiones formadas con el átomo `a` *y el operador "-". Cuando se hace la pregunta "*`simple(a-a-a).`*", el intérprete la unificará con la regla de la línea 2 lo que lleva a que* `X=a-a` *y a que se dispare la nueva pregunta "*`simple(a-a).`*". A continuación se volvería a repetir lo mismo y entonces la nueva pregunta sería "*`simple(a).`*", que ya no se unificaría con la regla sino con el hecho de la línea 1.*

En canvi si li fem a l'intèrpret la pregunta "`simple(Y).`", es trobaria la unificació en la línia 1 la qual cosa provocaria la primera resposta: `Y=a`. Si polsem ";" l'intèrpret trobarà a continuació la unificació amb la regla de la línia 2 que donarà lloc a que `Y=X-a` i a que es llance l'antecedent, "`simple(X).`", com a (nova) pregunta la resposta de la qual serà `X=a`. Combinant la unificació amb la resposta es té que `Y=a-a`, que seria la segona resposta. De la mateixa manera es produiria la tercera resposta, la quarta, etc.

En cambio si le hacemos al intérprete la pregunta "`simple(Y).`*", se encontraría la unificación en la línea 1 lo que provocaría la primera respuesta:* `Y=a`*. Si pulsamos ";" el intérprete encontrará a continuación la unificación con la regla de la línea 2 que dará lugar a que* `Y=X-a` *y a que se lance el antecedente, "*`simple(X).`*", como (nueva) pregunta cuya respuesta será* `X=a`*. Combinando la unificación con la respuesta se tiene que* `Y=a-a`*, que sería la segunda respuesta. De la misma forma se produciría la tercera respuesta, la cuarta, etc.*

A.2.1 Predicats recursius sobre enters

Predicados recursivos sobre enteros

L'exemple anterior funcionaria exactament igual si canviàrem l'àtom `a` per l'enter 1 (o qualsevol altre). I també si canviem el símbol "–" per "+", per exemple. Aquestes expressions no signifiquen en principi res per a PROLOG--. Si volem utilitzar expressions aritmètiques i fer càlculs amb nombres cal avaluar-los.

El ejemplo anterior funcionaría exactamente igual si cambiáramos el átomo `a` *por el entero 1 (o cualquier otro). Y también si cambiamos el símbolo "–" por "+", por ejemplo. Estas expresiones no significan en principio nada para* PROLOG--. *Si queremos utilizar expresiones aritméticas y hacer cálculos con números es necesario* **evaluarlos**.

El mecanisme bàsic per avaluar expressions en PROLOG-- és l'operador "`is`", que avalua l'expressió de la part dreta a un valor (enter o real) i desprès intenta unificar aquest valor amb la part esquerra.

El mecanismo básico para evaluar expresiones en PROLOG-- *es el operador "*`is`*", que evalua la expresión de la parte derecha a un valor (entero o real) y después intenta* **unificar** *este valor con la parte izquierda.*

```
?-  2 is 1+1.
    true.
?-  X is 1+1.
    X = 2.
?-  2 is X+1.
    ERROR: Arguments are not sufficiently instantiated ...
?-  X=1,2 is X+1.
    X = 1.
?-  X=a,2 is X+1.
    ERROR: Arithmetic: 'a/0' is not a function
```

Com es dedueix dels exemples, la part dreta no pot contindre variables sense instanciar (a un valor numèric). Considerem el predicat `simple1/3` com una extensió de l'últim exemple.

Como se deduce de los ejemplos, la parte derecha no puede contener variables sin instanciar (a un valor numérico). Consideremos el predicado `simple1/3` *como una extensión del último ejemplo.*

```
simple1(a,1,a+a).
simple1(X-a,N,Y+a+a) :-
   simple1(X,N1,Y),
   N is N1+1.
```

```
?-  simple1(a-a,N,R).
    N = 2,
    R = a+a+a+a.
?-  simple1(Z,N,a+a+a+a+a+a).
    Z = a-a-a,
    N = 3.
?-  simple1(Z,2,R).
    Z = a-a,
    R = a+a+a+a ;
    ERROR!!!!
```

L'esquema recursiu és el mateix que abans però ara després de cada "crida" recursiva s'incrementa en 1 el valor del segon argument. En la primera pregunta, el primer argument és una entrada i `PROLOG--` calcula com a sortida els valors de `N` i `R`. En la segona pregunta els arguments segon i tercer intercanvien els seus papers.

El esquema recursivo es el mismo que antes pero ahora después de cada "llamada" recursiva se incrementa en 1 el valor del segundo argumento. En la primera pregunta, el primer argumento es una entrada y `PROLOG--` *calcula como salida los valores de* `N` *y* `R`*. En la segunda pregunta los argumentos segundo y tercero intercambian sus papeles.*

En la tercera pregunta, només el segon argument té un valor. Com es pot veure, `PROLOG--` contesta i roman a l'espera. I si polsem ";" es produeix un bucle infinit. Això és per que les preguntes que dispara la regla sempre poden unificar-se amb la mateixa regla, "`X=X-a, N1=N,Y=Y+a+a.`". Per tant, no s'instancia cap variable i el procés continua indefinidament. La primera resposta es produeix per que la primera vegada que es dispara, s'obté la unificació "`X=a, N1=1, R=a+a.`" en la línia 1. Si en l'arxiu escriguérem primer la regla i després el fet obtindríem directament el bucle infinit.

En la tercera pregunta, sólo el segundo argumento tiene un valor. Como se puede ver, `PROLOG--` *contesta y queda a la espera. Y si pulsamos ";" se produce un bucle infinito. Esto es por que las preguntas que dispara la regla siempre pueden unificarse con la misma regla, "*`X=X-a, N1=N,Y=Y+a+a.`*". Por tanto, no se instancia ninguna variable y el proceso continua indefinidamente. La primera respuesta se produce por que la primera vez que se dispara, se obtiene la unificación "*`X=a, N1=1, R=a+a.`*" en la línia 1. Si en el archivo escribiésemos primero la regla y después el hecho obtendríamos directamente el bucle infinito.*

És possible modificar el predicat per a que responga correctament a preguntes com l'anterior.

Es posible modificar el predicado para que responda correctamente a preguntas como la anterior.

```
1. simple2(a,1,a+a).
2. simple2(X-a,N,Y+a+a) :-
3.    N>1.
4.    N1 is N-1.
5.    simple2(X,N1,Y),
```

```
?- simple2(Z,2,R).
   Z = a-a,
   R = a+a+a+a.
```

La diferència entre els dos predicats és subtil. La identificació entre les variables `N` i `N1` és exactament la mateixa. Però abans es calculava `N` en funció de `N1` després de la recursió, i ara es calcula `N1` en funció de `N` abans de la recursió. D'aquesta manera, ara és el segon argument el que controla la recursió la qual cosa obliga a que aquest sempre estiga instanciat (si no, es produirà sempre un error en la línia 4). L'altra diferència important és que ara cal afegir la condició de la línia 3 per a que el fet i la regla siguen casos mútuament exclusius.

La diferencia entre los dos predicados es sutil. La identificación entre las variables `N` *y* `N1` *es exactamente la misma. Pero antes se calculaba* `N` *en función de* `N1` *después de la recursión, y ahora se calcula* `N1` *en función de* `N` *antes de la recursión. De esta manera, ahora es el segundo argumento el que controla la recursión lo cual obliga a que éste siempre esté instanciado (si no, se producirá siempre un error en la línea 4). La otra diferencia importante es que ahora hay que añadir la condición de la línea 3 para que el hecho y la regla sean casos mútuamente exclusivos.*

És interessant comentar que aquesta necessitat d'instanciació no s'aplica al predicat `simple1/3` ja que una pregunta com "`simple(Z,N,R).`" és perfectament vàlida com ho era la pregunta "`simple(Y).`". Totes dues donen lloc a infinites respostes però no a un bucle infinit.

Es interesante comentar que esta necesidad de instanciación no se aplica al predicado `simple1/3` *ya que una pregunta como "*`simple(Z,N,R).`*" es perfectamente válida como lo era la pregunta "*`simple(Y).`*". Las dos dan lugar a infinitas respuestas pero no a un bucle infinito.*

Hi ha molts exemples de predicats recursius on la recursió depén d'un argument enter. Sempre s'han d'afegir les condicions que calga per a assegurar que els diferents casos, recursius i no recursius siguen mútuament exclusius. També s'ha de parar atenció a fer els càlculs relacionats amb la variable de la que depén la recursió abans de qualsevol referència recursiva.

Hay muchos ejemplos de predicados recursivos donde la recursión depende de un argumento entero. Siempre hay que añadir las condiciones que sean necesarias para asegurar que los diferentes casos, recursivos y no recursivos sean mútuamente exclusivos. También hay que fijarse en hacer los cálculos relacionados con la variable de la que depende la recursión antes de cualquier referencia recursiva.

Tots els exemples del capítol 3 corresponen a aquest cas. Normalment, hi ha un argument enter que controla la recursió, i els altres arguments poden ser d'entrada (dades que es necessiten per poder fer els càlculs) o de sortida (resultats del procediment recursiu que o bé s'instanciaran a una variable o bé es compararan mitjançant unificació amb un valor donat.

Todos los ejemplos del capítulo 3 corresponden a este caso. Normalmente, hay un argumento entero que controla la recursión, y los otros argumentos pueden ser de entrada (datos que se necesitan para poder hacer los cálculos) o de salida (resultados del prodedimiento recursivo que o bien se instanciarán a una variable o bien se compararán mediante unificación con un valor dado.

A.2.2 Predicats recursius sobre llistes

Predicados recursivos sobre listas

Les llistes són la manera en què `PROLOG--` pot representar informació estructurada. D'entrada les llistes tenen ja una definició recursiva:

Las listas son la manera en que `PROLOG--` *puede representar información estructurada. De entrada las listas tienen ya una definición recursiva:*

1. La llista buida, `[]`, és una llista.
2. Qualsevol terme t seguit d'una llista `L` és una llista. Això s'escriu com a `[t|L]`.

1. *La lista vacía,* `[]`*, es una lista.*
2. *Cualquier término* `t` *seguido de una lista* `L` *es una lista. Esto se escribe como* `[t|L]`*.*

Les notacions següents són equivalents per a tot terme t_k on $1 \leq k \leq n$.

Las notaciones siguientes son equivalentes para todo término t_k *donde* $1 \leq k \leq n$.

$$[t_1, t_2, \ldots, t_k, \ldots, t_n] \qquad [t_1, t_2, \ldots, t_k \,|\, [t_{k+1}, \ldots, t_n]] \qquad [t_1 \,|\, [t_2 \,|\, [\ \cdots [t_n \,|\, [\,]\ ]]]]$$

A mode d'exemple s'inclouen a continuació i sense comentaris, versions dels predicats anteriors `simple/1` i `simple1/3` amb llistes.

A modo de ejemplo se incluyen a continuación y sin comentarios, versiones de los predicados anteriores `simple/1` *y* `simple1/3` *con listas.*

```
lsimple([]).
lsimple([a|L]) :-
   lsimple(L).

lsimple1([],0,[]).
lsimple1([a|L],N,[a,a|R]) :-
   lsimple1(L,N1,R),
   N is N1+1.
```

És relativament fàcil plantejar recursions sobre llistes si el cas recursiu es correspon amb la llista d'entrada menys un nombre fix dels seus **primers** elements. En el cas d'un sol element l'esquema seria alguna cosa com

Es relativamente fácil plantear recursiones sobre listas si el caso recursivo se corresponde con la lista de entrada menos un número fijo de sus **primeros** *elementos. En el caso de un solo elemento el esquema sería algo como*

```
pred([],...).        % Cas Base (normalment)
                       Caso base (normalmente)
pred([C|L],...) :-
   ...
   pred(L,...),
   ...
```

Però també podríem pensar en una relació recursiva entre la llista d'entrada i algun element de la llista que no fóra el cap. En aquestos casos es pot utilitzar el predicat predefinit `select/3` juntament amb alguna altra condició la qual cosa donaria lloc a un esquema com

Pero también podríamos pensar en una relación recursiva entre la lista de entrada y algún elemento de la lista que fuese la cabeza. En estos casos se puede utilizar el predicado predefinido `select/3` *junto con alguna otra condición lo que daría lugar a un esquema como*

```
pred([],...).       % Cas Base (normalment)
                      Caso base (normalmente)
pred(L,...) :-
   select(X,L,R),
   ...
   pred(R,...),
   ...
```

El predicat select(E,L,R) és cert si E és un element de la llista L i R és el resultat d'eliminar E en la llista L. Qualsevol dels arguments de select pot ser d'entrada o de sortida.

El predicado select(E,L,R) *es cierto si* E *es un elemento de la lista* L *y* R *es el resultado de eliminar* E *en la lista* L. *Cualquiera de los argumentos de* select *puede ser de entrada o de salida.*

Una altra possibilitat és que la relació recursiva no siga amb la llista d'entrada sense un element (o sense un nombre fix d'elements), sinó amb el resultat de dividir la llista en dues subllistes. En aquestos casos es pot utilitzar el predicat predefinit append/3 juntament amb altres condicions la qual cosa donaria lloc a un esquema com

Otra posibilidad es que la relación recursiva no sea con la lista de entrada sin un elemento (o sin un número fijo de elementos), sino con el resultado de dividir la lista en dos sublistas. En estos casos se puede utilizar el predicado predefinido append/3 *junto con otras condiciones lo que daría lugar a un esquema como*

```
pred([],...).          % Cas Base (ara es pot complicar!)
                         Caso base (ahora se puede complicar!)
pred(L,...) :-
   append(L1,L2,L),
   ...
   pred(L1,...),       % o L2 o les dues
                         o L2 o las dos
   ...
```

El predicat append(L1,L2,L12) és cert si L12 és la concatenació de les llistes L1 i L2. Qualsevol dels arguments de append pot ser d'entrada o de sortida.

El predicado append(L1,L2,L12) *es cierto si* L12 *es la concatenación de las listas* L1 *y* L2. *Cualquiera de los argumentos de* append *puede ser de entrada o de salida.*

A banda de poder-se usar per a plantejar relacions recursives, molts problemes sobre llistes es poden resoldre de manera fàcil i compacta usant els predicats `append` i/o `select`.

Aparte de poderse usar para plantear relaciones recursivas, muchos problemas sobre listas se pueden resolver de manera fácil y compacta usando los predicados `append`, *y/o* `select`.

A.3 Operadors, funcions i predicats predefinits
Operadores, funciones y predicados predefinidos

A.3.1 Operadors lògics i pseudo-predicats
Operadores lógicos y pseudopredicados

Els operadors lògics es corresponen amb les connectives principals de la lògica proposicional i els seus arguments són per tant predicats. Anomenem pseudo-predicats aquells predicats que poden tindre algun argument que també és un predicat. Els operadors lògics són també pseudo-predicats ja que tot operador es pot utilitzar també com a predicat o com a functor. Aquesta forma que s'anomena canònica es pot visualitzar mitjançant el predicat `write_canonical/1` l'argument del qual és qualsevol expressió. El resultat d'aquestes expressions és sempre un predicat.

Los operadores lógicos se corresponden con las conectivas principales de la lógica proposicional y sus argumentos son por tanto predicados. Llamamos pseudopredicados a aquellos predicados que pueden tener algún argumento que es también un predicado. Los operadores lógicos son también pseudopredicados ya que todo operador se puede utilizar también como predicado o como functor. Esta forma que se la llama canónica se puede visualizar mediante el predicado `write_canonical/1` *cuyo argumento es cualquier expesión. El resultado de estas expresiones es siempre un predicado.*

nom/aritat ***nombre/aridad***	**descripció** ***descripción***	**associatiu** ***asociativo***
`:-/2`	definició de regles *definición de reglas*	no
`:-/1`	pregunta	no
`,/2`	conjunció lògica *conjunción lògica*	si
`;/2`	disjunció lògica *disyunción lògica*	si
`not/1`	negació *negación*	
`findall/3`	totes les respostes a una pregunta *todas las respuestas a una pregunta*	

El pseudo-predicat `findall/3` permet recollir totes les respostes a una pregunta utilitzant una llista. En particular, `findall(X,P,L)` és cert si `X` és una variable, `P` és una pregunta (que depén de `X`) i `L` és una llista que conté tots els valors de `X` per als quals la resposta a `P` és certa.

El pseudopredicado `findall/3` permite recoger todas las respuestas a una pregunta utilizando una lista. En particular, `findall(X,P,L)` es cierto si `X` es una variable, `P` es una pregunta (que depende de `X`) y `L` es una lista que contiene todos los valores de `X` para los cuales la respuesta a `P` es cierta.

A.3.2 Operadors de comparació i unificació
Operadores de comparación y unificación

Aquests operadors ja s'han vist abans. Els seus arguments són termes i el resultat és un predicat.

Estos operadores ya se han visto antes. Sus argumentos son términos y el resultado es un predicado.

nom/aritat *nombre/aridad*	descripció *descripción*	associatiu *asociativo*
`=/2`, `\=/2`	unificació de termes *unificación de términos*	no
`==/2`, `\==/2`	comparació de termes *comparación de términos*	no

A.3.3 Operadors de comparació i unificació aritmètica
Operadores de comparación y unificación aritmética

L'operador "`is/2`" ja s'ha vist abans. Tots dos arguments de la resta d'operadors aritmètics han de ser expressions que puguen ser avaluades a un valor numèric. En altres paraules, si les expressions contenen variables aquestes hauran d'estar instanciades. Els valors numèrics són unificats o comparats fent els corresponents canvis de tipus si escau.

El operador "`is/2`" ya se ya visto antes. Los dos argumentos del resto de operadores aritméticos tienen que ser expresiones que puedan ser evaluadas a un valor numérico. En otras palabras, si las expresiones contienen variables éstas deberán estar instanciadas. Los valores numéricos son unificados o comparados haciendo los correspondientes cambios de tipo si es necesario.

nom/aritat *nombre/aridad*	descripció *descripción*	associatiu *asociativo*
`is/2`	avaluació/unificació *evaluación/unificación*	no
`=:=/2`, `=\=/2`	comparació aritmètica *comparación aritmética*	no
`>=/2`, `=</2`	comparació aritmètica *comparación aritmética*	no

A.3.4 Operadors aritmètics

Operadores aritméticos

Es tracta dels operadors que també estan presents en la majoria de llenguatges de programació. Tant els seus arguments com el seu resultat són expressions aritmètiques que poden contindre variables. És a dir, a diferència dels anteriors, les expressions resultants són termes qualssevol.

Se trata de los operadores que también están presentes en la mayoría de lenguajes de programación. Tanto sus argumentos como su resultado son expresiones aritméticas que pueden contener variables. Es decir, a diferencia de los anteriores, las expresiones resultantes son términos cualesquiera.

nom/aritat *nombre/aridad*	**descripció** *descripción*	**associatiu** *asociativo*
`+/2,  -/2`	addició/substracció *adición/sustracción*	esquerra
`+/1,  -/1`	signe *signo*	
`*/2,  //2`	multiplicació/divisió *multiplicación/división*	esquerra
`///2`	divisió entera *división entera*	esquerra
`mod/2`	residu de la divisió entera *resto de la división entera*	esquerra
`^/2`	potència *potencia*	dreta

A.3.5 Funcions aritmètiques

Funciones aritméticas

Les funcions aritmètiques són functors predefinits que avaluen els seus arguments i donen com a resultat un valor numèric. Per tant, els seus arguments han de ser expressions que puguen ser avaluades.

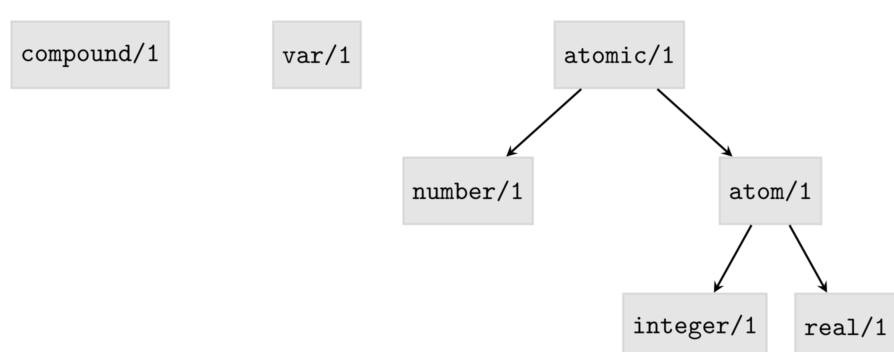
Las funciones artiméticas son functores predefinidos que evaluan sus argumentos y dan como resultado un valor numérico. Por tanto, sus argumentos han de ser expresiones que puedan ser evaluadas.

`abs/1`	valor absolut *valor absoluto*	`sign/1`	funció signe *función signo*
`floor/1`	funció sol *función suelo*	`ceiling/1`	funció sostre *función techo*
`exp/1`	exponenciació *exponenciación*	`log/1`	logaritme natural *logaritmo natural*
`mod/2`	funció mòdul *función módulo*	`round/1`	enter més pròxim *entero más próximo*
`max/2`	funció màxim *función máximo*	`min/2`	funció mínim *función mínimo*
`e/0, pi/0`	nombres e i π *números e y* π	`nan/0`	no és un nombre *no es un número*
`inf/0`	infinit *infinito*	`epsilon/0`	valor real més petit *valor real más pequeño*

A.3.6 Predicats sobre termes

Predicados sobre términos

`compound/1`	és compost (no atòmic) *es compuesto (no atómico)*	`ground/1`	sense variables lliures *sin variables libres*
`var/1`	és variable *es variable*	`atomic/1`	és atòmic: àtom o nombre *es atómico: átomo o número*
`atom/1`	és àtom *es átomo*	`number/1`	és nombre: enter o real *es número: entero o real*
`integer/1`	és enter *es entero*	`real/1`	és real *es real*



A.3.7 Predicats sobre llistes

Predicados sobre listas

append/3	append(L1,L2,L12)	L12 és la concatenació de L1 i L2 *L12 es la concatenación de L1 y L2*
select/3	select(E,L,R)	R és la llista L sense l'element E *R es la lista L sin el elemento E*
member/2	member(E,L)	E és un element de L *E es un elemento de L*
is_list/1	is_list(L)	L és una llista *L es una lista*
length/2	length(L,N)	la llista L té N elements *la lista L tiene N elementos*

A.3.8 Predicats sobre enters

Predicados sobre enteros

between/3	between(N,M,K)	$N \leq K \leq M$
succ/2	succ(N,M)	N, M consecutius i no negatius *consecutivos y no negativos*
plus/3	plus(N,M,S)	$S = N + M$
divmod/4	divmod(N,M,Q,R)	$N = M * Q + R$

A.4 Algunes relacions entre lògica i PROLOG--

Algunas relaciones entre lógica y PROLOG--

A.4.1 Fets

Hechos

Els fets en PROLOG-- són fórmules atòmiques, encara que es pot afirmar la conjunció de diverses coses simplement en afirmar separadament cada una d'elles.

Los hechos en PROLOG-- *son fórmulas atómicas, aunque se puede afirmar la conjunción de varias cosas simplemente afirmando separadamente cada una de ellas.*

Els fets també es poden veure com (conseqüents de) regles l'antecedent de les quals és cert. De fet es poden escriure així en PROLOG-- encara que no té cap sentit fer-ho.

Los hechos también se pueden ver como (consecuentes de) reglas cuyo antecedente es cierto. De hecho se pueden escribir así en PROLOG-- *aunque no tiene ningún sentido hacerlo.*

`plou.`	P	`plou :- true`	$T \Rightarrow P \equiv P$
`plou.` `neva.`	$P \wedge N$	`p(f(a,f(b))).`	$P(f(a, f(b)))$
`animal(nemo).`	$A(w)$	`oblida(dorothy,nemo).`	$O(d, n)$

Els fets que contenen variables duen un quantificador universal implícit per cada variable diferent.

Los hechos que contienen variables llevan un cuantificador universal implícito por cada variable diferente.

`animal(X).`	$\forall x, A(x)$	`oblida(dorothy,X).`	$\forall x, O(d, x)$
`oblida(X,Y).`	$\forall x, \forall y, O(x, y)$	`oblida(X,X).`	$\forall x, O(x, x)$

Els fets anteriors són perfectament vàlids en PROLOG--. Però si s'inclou en un programa algun dels tres primers, l'intèrpret emetrà un avís (*warning*) per que hi ha variables que apareixen una única vegada i per tant no és necessari donar-los un nom. La forma en què caldria expressar aquests fets en PROLOG-- requereix l'ús de la variable anònima. Per exemple, els fets anteriors s'escriurien com

Los hechos anteriores son perfectamente válidos en PROLOG--. *Pero si se incluye en un programa alguno de los tres primeros, el intérprete emitirá un aviso (*warning*) por que hay variables que aparecen una sola vez y por tanto no es necesario darles un nombre. La forma en que habría que expresar estos hechos en* PROLOG-- *requiere el uso de la variable anónima. Por ejemplo, los hechos anteriores se escribirían como*

`animal(_).`	$\forall x, A(x)$	`oblida(dorothy,_).`	$\forall x, O(d, x)$
`oblida(_,_).`	$\forall x, \forall y, O(x, y)$	`oblida(X,X).`	$\forall x, O(x, x)$

A.4.2 Regles

Reglas

Les regles en PROLOG-- s'han de poder escriure com a clàusules de Horn. En la pràctica sempre s'ha d'intentar escriure regles amb un conseqüent i un antecedent format per la conjunció d'una o més fórmules atòmiques fent servir preferentment línies diferents convenientment indentades.

Las reglas en PROLOG-- tienen que poder escribirse como cláusulas de Horn. En la práctica siempre se tiene que intentar escribir reglas con un consecuente y un antecedente formado por la conjunción de una o más fórmulas atómicas usando preferentemente líneas diferentes convenientemente indentadas.

```
oblida(dorothy,X) :-
   peix(X),
   amic(X,dorothy),
   neva.
```

$$\forall x, \Big(P(x) \wedge A(x,d) \wedge N\Big) \Rightarrow O(d,x)$$

En les regles s'intenta minimitzar l'ús de la disjunció. Per això se solen descomposar antecedents formats per disjuncions en diverses regles fent servir la pseudodistributivitat de la implicació per la dreta.

En las reglas se intenta minimizar el uso de la disyunción. Por eso se suelen descomponer antecedentes formados por disyunciones en varias reglas usando la pseudodistributividad de la implicación por la derecha.

```
oblida(dorothy,X) :-
   peix(X);
   animal(X).
```

$$\forall x, (P(x) \vee A(x)) \Rightarrow O(d,x)$$

```
oblida(dorothy,X) :-
   peix(X).
oblida(dorothy,X) :-
   animal(X).
```

$$\Big(\forall x, P(x) \Rightarrow O(d,x)\Big) \wedge \Big(\forall x, A(x) \Rightarrow O(d,x)\Big)$$

De fet, en els antecedents de les regles es poden combinar arbitràriament conjuncions i disjuncions ja que sempre es pot trobar una fórmula equivalent formada per clàsules de Horn. No obstant això, en la pràctica habitual es prefereix utilitzar més regles més senzilles i evitar si es pot la disjunció.

De hecho, en los antecedentes de las reglas se pueden combinar arbitrariamente conjunciones y disyunciones ya que siempre se puede encontrar una fórmula equivalente formada por cláusulas de Horn. No obstante, en la práctica habitual se prefiere utilizar más reglas más sencillas y evitar si se puede la disyunción.

```
oblida(dorothy,X) :-
   peix(X),
   ( amic(X,dorothy);
      neva ),
   amic(dorothy,X).
```

$$\forall x, \Big(P(x) \wedge (A(x,d) \vee N) \wedge A(d,x)\Big) \Rightarrow O(d,x) \equiv$$

$$\equiv \Big(\forall x, \big(P(x) \wedge A(x,d) \wedge A(d,x)\big) \Rightarrow O(d,x)\Big) \wedge \Big(\forall x, \big(P(x) \wedge N \wedge A(d,x)\big) \Rightarrow O(d,x)\Big)$$

Com ja s'ha explicat, per cada variable diferent que apareix en el conseqüent hi ha un quantificador universal sobre tota la regla. Per cada variable diferent que apareix en l'antecedent i no en el conseqüent, es pot considerar que hi ha un quantificador existencial que s'aplica sobre l'antecedent. Per exemple,

Como ya se ha explicado, por cada variables diferente que aparece en el consecuente hay un cuantificador universal sobre toda la regla. Por cada variable diferente que aparece en el antecedente y no en el consecuente, se puede considerar que hay un cuantificador existencial que se aplica sobre el antecedente. Por ejemplo,

```
oblida(dorothy,X) :-
   peix(Y),
   ( amic(X,Y);
      neva ).
```

$$\forall x, \Big(\exists y : \big(P(y) \wedge (A(x,y) \vee N)\big)\Big) \Rightarrow O(d,x)$$

Alternativament i de manera equivalentment, es pot considerar que les variables que apareixen només en l'antecedent tenen associat un quantificador universal aplicat sobre tota la regla. Aquesta equivalència és deguda a la pseudodistributivitat de la implicació per la dreta respecte dels quantificadors (conjuncions/disjuncions).

Alternativamente y de manera equivalente, se puede considerar que las variables que aparecen sólo en el antecedente tienen asociado un cuantificador universal aplicado sobre toda la regla. Esta equivalencia se debe a la pseudodistributividad de la implicación por la derecha respecto de los cuantificadores (conjunciones/disyunciones).

```
oblida(dorothy,X) :-
   peix(Y),
   ( amic(X,Y);
      neva ).
```

$$\forall x, \forall y, \Big(P(y) \wedge \big(A(x,y) \vee N\big)\Big) \Rightarrow O(d,x)$$

A.4.3 Preguntes

Preguntas

Les preguntes en PROLOG-- són expressions formades mitjançant predicats, conjuncions i disjuncions exactament igual que en els antecedents de les regles.

Las preguntas en PROLOG-- son expresiones formadas mediante predicados, conjunciones y disyunciones exactamente igual que en los antecedentes de las reglas.

`?- plou,neva.`	$P \wedge N$	`?- plou;neva.`	$P \vee N$
`?- animal(nemo).`	$A(w)$	`?- oblida(dorothy,nemo).`	$O(d,n)$

Per cada variable que apareix en una pregunta es pot considerar que hi ha un quantificador existencial.

Por cada variable que aparece en una pregunta se puede considerar que hay un cuantificador existencial.

`?- animal(X),peix(X).`	$\exists x : \big(A(x) \wedge P(x)\big)$
`?- oblida(X,Y);peix(Y).`	$\exists x, \exists y : \big(O(x,y) \vee P(y)\big)$
`?- oblida(_,_);peix(_).`	$\exists x, \exists y, \exists z : \big(O(x,y) \vee P(z)\big)$

A.4.4 La negació en PROLOG--

***La negación en* PROLOG--**

La diferència més notable entre el que es pot expressar en PROLOG-- en relació a la lògica és la negació ja que aquesta no existeix en PROLOG--. El més pròxim que es té és el pseudopredicat `not/1` que es pot usar en preguntes i en antecedents de regles de manera que el seu resultat és cert o fals en funció de que PROLOG-- puga demostrar o no la seua veracitat. D'aquesta manera es poden expressar algunes fórmules amb negacions.

La diferencia más notable entre lo que se puede expresar en PROLOG-- ane relación a la lógica es la negación ya que esta no existe en PROLOG--. Lo más próximo que se tiene es el pseudopredicado `not/1` que se puede usar en preguntas y en antecedentes de reglas de manera que su resultado es cierto o falso en función de que PROLOG-- pueda demostrar o no su veracidad. De esta manera se pueden expresar algunas fórmulas con negaciones.

```
oblida(dorothy,X) :-
   peix(X),
   not((
      amic(X,dorothy),
      neva )).
```

$$\forall x, (P(x) \wedge \neg(A(x, d) \wedge N)) \Rightarrow O(d, x)$$

En la pràctica és convenient evitar la negació tant com siga possible ja siga reestructurant la fórmula o fins i tot definint predicats explícits per a alguna cosa i la contraria.

En la práctica es conveniente evitar la negación tanto como sea posible ya sea reestructurando la fórmula o incluso definiendo predicados explícitos para algo y su contrario.

A.4.5 Alguns exemples

Algunos ejemplos

Germanastres: Defineix la relació `esGermanastreDe/2` en funció de la relació `esFillDe/2`.

Hermanastros: Define la relación `esGermanastreDe/2` (ser hermanastro de) en función de la relación `esFillDe/2` (ser hijo de).

La relació/predicat que suposem donat, `esFillDe/2`, l'escriurem en lògica com un predicat binari, F, de manera que $F(x, y)$ siga cert si x és fill de y i fals en cas contrari.

La relación/predicado que suponemos dado, `esFillDe/2`, la escribiremos en lógica como un predicado binario, F, de manera que $F(x, y)$ sea cierto si x es hijo de y y falso en caso contrario.

Dues persones són germanes si comparteixen els dos progenitors. I són germanastres si només en comparteixen un dels dos. El més fàcil és escriure una fórmula que (per a tot x i y) siga certa sii dues persones son germanes o germanastres. És a dir,

Dos personas son hermanas si comparten los dos progenitores. Y son hermanastros si sólo comparten uno de los dos. Lo más fácil es escribir una fórmula que (para todo x e y) sea cierta sii *dos personas son hermanas o hermanastras. Es decir,*

$$P_1(x, y) \equiv \Big(\exists z : \; F(x, z) \wedge F(y, z) \wedge (x \neq y)\Big)$$

La desigualtat és necessària per que normalment no diem mai que algú és germà o germanastre d'ell mateix.

La desigualdad es necesaria por que normalmente no decimos nunca que alguien es hermano o hermanastro de si mismo.

De la mateixa manera que abans, es pot escriure una fórmula que siga certa **sii** dues persones tenen almenys un progenitor diferent.

De la misma manera que antes, se puede escribir una fórmula que sea cierta sii dos persoanas tienen almenos un progenitor diferente.

$$P_2(x,y) \equiv \Big(\exists z: \ F(x,z) \land \neg F(y,z)\Big)$$

Una primera solució per a la relació germanastre seria definir el nostre predicat, G, com la conjunció dels dos anteriors.

Una primera solución para la relación hermanastro sería definir nuestro predicado, G*, como la conjunción de los dos anteriores.*

$$G(x,y) \equiv P_1(x,y) \land P_2(x,y)$$

Però també podem simplificar l'expressió anterior si canviem el nom de la variable en P_2 i apliquem la propietat distributiva sobre els quantificadors existencials (disjuncions) i les conjuncions. Al final podem arribar a

Pero también podemos simplificar la expresión anterior si cambiamos el nombre de la variable en P_2 *y aplicamos la propiedad distributiva sobre los cuantificadores existenciales (disyunciones) y conjunciones. Al final podemos llegar a*

$$G(x,y) \equiv \exists z, \exists t: \quad \Big(F(x,z) \land F(y,z)\Big) \land \Big(F(x,t) \land \neg F(y,t)\Big)$$

on hem eliminat a més a més la desigualtat de P_1 per que no pot passar a la vegada que $x = y$ i que $P_2(x,x)$ siga cert.

donde hemos eliminado además la desigualdad de P_1 *por que no puede pasar a la vez que* $x = y$ *i que* $P_2(x,x)$ *siga cert.*

Com a conclusió, la definició anterior en forma de regla en lògica i PROLOG-- podria ser

Como conclusión, la definición anterior en forma de regla en lógica y PROLOG-- *podría ser*

```
esGermanastreDe(X,Y) :-
   esFillDe(X,Z),
   esFillDe(Y,Z),
   esFillDe(X,T),
   not((
      esFillDe(Y,T))).
```

$$\forall x, \forall y, \quad \Big(\exists z, \exists t: \quad (F(x,z) \land F(y,z) \land F(x,t) \land \neg F(y,t))\Big) \Rightarrow G(x,y)$$

La fórmula lògica anterior és també equivalent, com ja s'ha explicat abans, a

La fórmula lógica anterior es también equivalente, como ya se ha explicado antes, a

$$\forall x, \forall y, \forall z, \forall t, \quad \Big(\big(F(x,z) \wedge F(y,z) \wedge F(x,t) \wedge \neg F(y,t) \big) \Rightarrow G(x,y) \Big)$$

Com a exercici, raona si la següent definició alternativa del predicat G és o no correcta.

Como ejercicio, razona si la siguiente definición alternativa del predicado G es o no correcta.

$$G'(x,y) \equiv \exists z, \exists t, \exists v : \quad \Big(F(x,z) \wedge F(x,t) \wedge \neg F(x,v) \wedge F(y,z) \wedge F(y,v) \wedge \neg F(y,t) \Big)$$

A.5 Problemes resolts i comentats

Problemas resueltos y comentados

Problema A.1: [kami]

En Takamagahara viuen els kamis. En particular, els amatsukamis són deus bons que sempre diuen la veritat. En canvi, els shinigamis són els deus de la mort i sempre diuen mentides. Tres kamis tenen la següent conversació:

En Takamagahara viven los kamis. En particular, los amatsukamis son dioses buenos que siempre dicen la verdad. En cambio, los shinigamis son los dioses de la muerte y siempre dicen mentiras. Tres kamis tienen la siguiente conversación:

a) Agyo: Bepo menteix!

a) Agyo: Bepo miente!

b) Bepo: Calicarcha és un amatsukami.

b) Bepo: Calicarcha es un amatsukami.

c) Calicarcha: Agyo y Bepo són la mateixa cosa.

c) Calicarcha: Agyo y Bepo son lo mismo.

Quin tipus de kami són Agyo, Bepo i Calicarcha, respectivament? Podries resoldre el problema usant `PROLOG--`? Fes-ho.

¿Qué tipo de kami son Agyo, Bepo y Calicarcha, respectivamente? Podrías resolver el problema usando `PROLOG--`? Hazlo.

Hi ha diverses maneres de representar aquest problema mitjançant lògica. D'entrada identifiquem el fet de ser amatsukami o shinigami amb el fet de dir sempre la veritat o no. Aleshores el més senzill és definir variables proposicionals, a, b i c, per a cada un dels 3 kamis.

Hay varias formas de representar este problema usando la lógica. De entrada identificamos el hecho de ser amatsukami o shinigami con el hecho de decir siempre la verdad o no. Entonces lo más sencillo es definir variables proposicionales, a, b y c, para cada uno de los 3 kamis.

Les 3 afirmacions es podrien escriure aleshores usant lògica proposicional com:

Las 3 afirmaciones se podrían escribir entonces usando lògica proposicional como:

$$a \Leftrightarrow \neg b$$

$$b \Leftrightarrow c$$

$$c \Leftrightarrow (a \Leftrightarrow b)$$

Però també podríem definir un predicat `amatsukami/1` que siga cert o fals per a cada un dels kamis i tindríem una representació equivalent en lògica de predicats. Encara que es poden aplicar regles d'inferència estàndar prenent com a premisses aquestes representacions, no es pot arribar a un programa en PROLOG-- que reproduesca el raonament corresponent.

Pero también podríamos definir un predicado `amatsukami/1` *que sea cierto o falso para cada uno de los kamis y tendríamos una representación equivalente en lógica de predicados. Aunque se pueden aplicar reglas de inferencia estándar tomando como premisas estas representaciones, no se puede llegar a un programa en* PROLOG-- *que reproduzca el razonamiento correspondiente.*

Una alternativa senzilla consisteix a definir un predicat ternari, `conversa(A,B,C)`, de manera que els seus tres arguments sobre el domini $\mathcal{K} = \{amatsukami, shinegami\}$, representen els kamis que tenen la conversa. El predicat serà cert si els arguments prenen valors compatibles amb el que s'afirma.

Una alternativa sencilla consiste en definir un predicado ternario, `conversa(A,B,C)`, *de manera que sus tres argumentos sobre el dominio* $\mathcal{K} = \{amatsukami, shinegami\}$, *representen los kamis que tienen la conversación. El predicado será cierto si los argumentos toman valores compatibles con lo que se afirma.*

Cada afirmació, es representarà també mitjançant un predicat ternari `diuquees(X,Y,Z)` amb variables també sobre $\mathcal{K}$, de manera que `X` representa el (tipus de) kami que parla, `Y` representa el (tipus de) kami a qui es refereix, i `Z` representa el (tipus de) kami amb qui s'equipara `Y`.

Cada afirmación, se representará también mediante un predicado ternario `diuquees(X,Y,Z)` *con variables también sobre* $\mathcal{K}$, *de manera que* `X` *representa el (tipo de) kami que habla,* `Y` *representa el (tipo de) kami al que se refiere, y* `Z` *representa el (tipo de) kami con quien se equipara* `Y`.

Per fer que les variables prenguen els (s'instancien als) possibles valors dins de $\mathcal{K}$ es requereix també un predicat `kami/1` que es definirà amb dos fets, un per cada valor del domini. El resultat es mostra a continuació.

Para hacer que las variables tomen (se instancien a) los posibles valores dentro de $\mathcal{K}$ *se requiere también un predicado* `kami/1` *que se definirá con dos hechos, uno por cada valor del dominio. El resultado se muestra a continuación.*

```
kami(amatsukami).
kami(shinigami).

% diuquees(X,Y,Z) : X diu que   Y és com (o és)  Z
                      dice que    es como (o es)
diuquees(amatsukami ,K,K).
diuquees(shinigami , K,Q) :-
   K≡Q.

conversa(X,Y,Z) :-   % X,Y,Z representen a  Agyo,Bepo,Calicarcha
                             representan a
   kami(X),kami(Y),kami(Z),
   diuquees(X,Y,shinigami),                         % a)
   diuquees(Y,Z,amatsukami),                        % b)
   diuquees(Z,X,Y).                                 % c)
----------------------------------------------------------------
?- conversa(A,B,C).
   A = amatsukami, B = C, C = shinigami.
```

Problema A.2: [bessones]

Anna, Bea i Carme són tres bessones idèntiques. Però Anna sempre diu la veritat, Bea sempre menteix, i Carme de vegades diu la veritat i de vegades no. Sa mare les fa seure en tres cadires una al costat de l'altra i comença a preguntar d'esquerra a dreta ...

Anna, Bea y Carme son tres gemelas idénticas. Pero Anna siempre dice la verdad, Bea siempre miente, y Carme a veces dice la verdad y a veces no. Su madre las sienta en tres sillas una al lado de la otra y empieza a preguntar de izquierda a derecha ...

a) Qui és la que seu al mig? Anna, contesta la primera.

a) ¿Quién es la que se sienta en el centro? Anna, contesta la primera.

b) Qui eres tu? Sóc Carme, contesta la segona.

b) ¿Quién eres tú? Soy Carme, contesta la segunda.

c) Qui seu a la teua esquerra? Bea, diu la tercera

c) ¿Quién se sienta a tu izquierda? Bea, dice la tercera.

Sabries endevinar qui és la que seu en cada cadira? Podries resoldre el problema usant PROLOG--? Fes-ho.

¿Sabrías adivinar quién se sienta en cada silla? ¿Podrías resolver el problema usando PROLOG--*? Hazlo.*

Representarem el domini format per les tres bessones mitjançant una llista amb l'ajuda del predicat `germanes/1` definit com

Representaremos el dominio formado por las trillizas mediante una lista con la ayuda del predicado `germanes/1` *definido como*

```
germanes([anna,bea,carme]).
```

El comportament a l'hora de contestar cada germana el representarem mitjançant el predicat `contesta/3` definit com a

El comportamiento a la hora de contestar cada hermana lo representaremos mediante el predicado `contesta/3` *definido como*

```
contesta(anna,X,X).
contesta (bea,X,Y) :-
   X\==Y.
contesta(carme,_,_).
```

Representarem també la manera en què seuen les tres bessones mitjançant una llista de tres variables que podran seleccionar el seu valor del domini anterior sense repeticions.

Representaremos también la manera en que se sientan las trillizas mediante una lista de tres variables que podrán seleccionar su valor del dominio anterior sin repeticiones.

Per a això, farem servir el predicat predefinit `select/3` de manera que quan una variable s'instancie a un valor, podrem obtindre el subdomini romanent per a poder seleccionar valors no repetits per a les variables.

Para eso, usaremos el predicado predefinido `select/3` *de manera que cuando una variables se instancie a un valor, podremos obtener el subdominio restante para poder seleccionar valores no repetidos para las variables.*

El predicat `bessones/1` que resol el problema es mostra a continuació juntament amb la pregunta i la contestació corresponent que indica com seuen les tres bessones.

El predicado `bessones/1` *que resuelve el problema se muestra a continuación junto con la pregunta y la contestación correspondiente que indica como se sientan las tres gemelas.*

```
bessones([Esq,Centr,Dre]) :-
   domini(L),
   select(Esq,L,L1), select(Centr,L1,[Dre]),
   contesta(Esq, Centr,anna),
   contesta(Centr,Centr,carme),
   contesta(Dre, Centr,bea).
```

```
?- bessones(L).
   L = [carme, bea, anna] .
```

Problema A.3: [llop]

Un pagès ha comprat en un mercat, un llop, una cabra i una col, i per a tornar a casa ha de travessar un riu amb una barca en què només cap ell i una de les tres coses que ha comprat. El llop es menjarà la cabra si es queda amb ella en qualsevol de les vores del riu sense la supervisió del pagès. I el mateix passarà amb la cabra i la col. De quina manera podrà el pobre home tornar a casa amb tota la seua compra? Quants viatges en barca necessitarà? És possible obtindre la solució al problema usant PROLOG--? Com?

Un labrador ha comprado en un mercado, una lobo, una cabra y una col, y para volver a casa tiene que atravesar un rio con una barca en la que sólo cabe él y una de las tres cosas que ha comprado. El lobo se comerá a la cabra si se queda solo con ella en cualquiera de las orillas del rio sin la supervisión del labrador. Y lo mismo pasará con la cabra y la col. ¿De qué manera podrá el pobre hombre volver a casa con toda su compra? ¿Cuántos viajes en barca necesitará? ¿Es posible obtener la solución al problema usando PROLOG--*? ¿Cómo?*

Representarem cada vora del riu mitjançant una llista que podrà contindre elements del domini

Representaremos cada lado del rio mediante una lista que podrá contener elementos del dominio

$$\{\texttt{barca,llop,cabra,col}\},$$

on l'àtom `barca` representa tant la barca com el pagès. Inicialment tots els elements estan en una vora i es tracta de que tots passen a l'altra.

donde el átomo `barca` *representa tanto a la barca como al labrador. Inicialmente todos los elementos están en un lado y se trata de que todos pasen al otro.*

Definirem aleshores un predicat, `segur/1`, que siga cert si ningú es pot menjar a ningú **quan no està la barca** (i el pagès). Una possibilitat seria,

Definiremos entonces un predicado, segur/1, que sea cierto si nadie se puede comer a nadie **cuando no está la barca** *(y el labrador). Una posibilidad sería,*

```
menja(L) :- member(cabra,L), member(col,L).
menja(L) :- member(llop,L), member(cabra,L).
segur(L) :-
   not(menja(L)).
```

I una altra possibilitat equivalent seria

Y otra posibilidad equivalente sería

```
segur([]).
segur([_]).
segur([llop,col]).
segur([col,llop]).
```

Definirem ara un predicat, `viatge/5`, de manera que

Definiremos ahora un predicado, viatge/5, de manera que

`viatge(L,R,LL,RR,Elem)`

siga cert si, donat el contingut de les dues vores en `L` i `R`, `Elem` és un element en la mateixa vora on està la barca i `LL` i `RR` és el contingut de les dues vores una vegada tant la barca com `Elem` passen d'una vora a l'altra.

sea cierto si, dado el contenido de los dos lados en `L` *y* `R`, `Elem` *es un elemento en el mismo lado donde está la barca y* `LL` *y* `RR` *es el contenido de los dos lados una vez tanto la la barca como* `Elem` *pasan de un lado al otro.*

Alternativament, la barca sempre pot passar d'una vora a l'altra de buit. En eixe cas, el cinquè argument s'instanciarà a l'àtom "o".

Alternativamente, la barca siempre puede pasar de un lado al otro vacía. En ese caso, el quinto argumento se instanciará al átomo "o".

Suposarem, sense pèrdua de generalitat, que en la llista que representa el contingut de cada vora, la barca, si hi és, es troba en la primera posició. El ordre dels altres elements és en canvi irrelevant.

Supondremos, sin pérdida de generalidad, que en la lista que representa el contenido de cada lado, la barca, si está, se encuentra en la primera posición. El orden de los otros elementos es en cambio irrelevante.

Aleshores, per a definir el predicat viatge/5, hi ha cuatre casos en funció d'on es troba la barca i si el viatge és de buit o no.

Entonces, para definir el predicado viatge/5, hay cuatro casos en función de dónde se encuentra la barca y si el viaje es vacío o no.

```
viatge([barca|E],D,EE,[barca,A|D],A) :-
   select(A,E,EE),
   segur(EE).
viatge(E,[barca|D],[barca,A|E],DD,A) :-
   select(A,D,DD),
   segur(DD).
viatge([barca|E],D,E,[barca|D],o) :-
   segur(E).
viatge(E,[barca|D],[barca|E],D,o) :-
   segur(D).
```

Finalment, definirem un altre predicat recursiu, viatges/6, de manera que

Finalmente, definiremos otro predicado recursivo, viatges/6, de manera que

viatges(N,L,R,LL,RR,L_Elem)

siga cert si, després de N viatges els continguts indicats per L i R han passat a ser LL i RR i L_Elems és una llista amb tots els elements que s'han mogut i en quin ordre, incloent-hi viatges buits.

sea cierto si, después de N viajes los contenidos indicados por L y R han pasado a ser LL y RR y L_Elems es una lista con todos los elementos que se han movido y en qué orden, incluyendo viajes vacíos.

Si considerem que la variable N ha d'estar instanciada, una definició possible seria

Si consideramos que la variable N tiene que estar instanciada, una definición posible sería

```
viatges(N,E,D,EEE,DDD,[Mov|Movs]) :-
   N>0,
   N1 is N-1,
   viatge(E,D,EE,DD,Mov),
   viatges(N1,EE,DD,EEE,DDD,Movs).

viatges(0,E,D,E,D,[]).
```

Considerant totes les definicions anteriors, podríem definir un últim predicat `solucio/2` que inicialitzara la vora esquerra amb tots els elements, exigira que aquesta vora esquerre acabe buida i intentara diferents nombres de moviments dins d'un rang, per exemple fins a 8. El resultat es mostra a continuació.

Considerando todas las definiciones anteriores, podríamos definir un último predicado `solucio/2` *que inicializara el lado izquierdo con todos los elementos, exigiera que este lado izquierdo acabe vacío e intentara diferentes números de movimientos dentro de un rango, por ejemplo hasta 8. El resultado se muestra a continuación.*

```
solucio(N, Movs) :-
   between(1,8,N),
   Ini = [barca,llop,cabra,col],
   viatges(N,Ini,[],[],_,Movs).
```

```
?-  solucio(N,Movs).
    N = 7,
    Movs = [cabra, o, llop, cabra, col, o, cabra]
    N = 7,
    Movs = [cabra, o, col, cabra, llop, o, cabra]
    false.
```

El pagès necessita per tant fer almenys set viatges dos dels quals correspondran a tornades de buit.

El labrador necesita por tanto hacer al menos siete viajes de los cuales dos corresponderán a volver con la barca vacía. En el prim

En el primer i últim viatge transportarà la cabra, i per als tres viatges centrals té les dues possibilitats que ilustren les anteriors respostes de l'intèrpret.

En el primero y último viajes transportará la cabra, y para los tres viajes centrales tiene las dos posibilidades que ilustran las anteriores respuestas del intérprete.

Problema A.4: [minizebra]

Hi ha 3 cases, en línia, de 3 colors diferents on viuen persones de 3 diferents nacionalitats, que els agraden 3 tipus d'animals diferents i juguen a 3 esports diferents. Considera les pistes següents:

Hay 3 casas, en línea, de 3 colores diferentes donde viven personas de 3 diferentes nacionalidades, a las que les gustan 3 tipos de animales diferentes y juegan a 3 deportes diferentes. Considera las pistas siguientes:

a) El brasiler no viu en la segona casa.

a) El brasileño no vive en la segunda casa

b) A qui li agraden els gossos juga a bàsquet.

b) Al que le gustan los perros juega a basquet.

c) Hi ha una casa entre la casa on es juga a futbol i la casa roja.

c) Hay una casa entre la casa donde se juega a futbol y la casa roja.

d) A qui li agraden els peixos viu a l'esquerra de qui li agraden els gats.

d) El que le gustan los peces vive a la izquierda del que le gustan los gatos.

e) La casa on els agraden els gossos està a la dreta de la verda.

e) La casa donde les gustan los perros está a la derecha de la verde.

f) L'alemà viu en la tercera casa.

f) El alemán vive en la tercera casa.

Quin animal li agrada a l'angles? Qui juga a golf? Qui viu en la casa blava? Resol el problema també mitjançant `PROLOG--`.

¿Qué animal le gusta al inglés? ¿Quién juega a golf? ¿Quién vive en la casa azul? Resuleve el problema también mediante `PROLOG--`*.*

Com que cada casa té associats quatre atributs (nacionalitat, color, animals i esport), considerarem un functor `casa/4`,

Como cada casa tiene asociados cuatro atributos (nacionalidad, color, animales y deporte), consideraremos un functor `casa/4`*,*

```
casa(Nacionalitat, Color, Animal, Esport)
```

per representar cada casa. I com que hi ha 3 cases alineades, considerarem una terna de functors com l'anterior,

para representar cada casa. Y como hay 3 casas alineadas, consideraremos una terna de functores como el anterior,

`Cases=(C1,C2,C3)` : `Ci=casa(Ni,Ci,Ai,Ei)` $i \in \{1,2,3\}$

Per resoldre el nostre problema definirem aleshores un predicat `problema/1` que prenga com a argument la terna de cases i comprove que es compleixen totes i cada una de les sis afirmacions.

Para resolver nuestro problema definiremos entonces un predicado `problema/1` *que tome como argumento la terna de casas y compruebe que se cumplen todas y cada una de las seis afirmaciones.*

```
problema(CASES) :-
   pista1(CasaBrasiler,CASES),
   pista2(CasaGossosBasquet,CASES),
   pista3(CasaFutbol,CasaRoja,CASES),
   pista4(CasaPeixos,CasaGats,CASES),
   pista5(CasaGossos,CasaVerda,CASES),
   pista6(CasaAlema,CASES),

   CasaBrasiler =casa(brasiler,_,_,_),
   CasaGossosBasquet=casa(_,_,gossos,basquet),
   CasaFutbol =casa(_,_,_,futbol),
   CasaRoja =casa(_,roja,_,_),
   CasaPeixos =casa(_,_,peixos,_),
   CasaGats =casa(_,_,gats,_),
   CasaGossos =casa(_,_,gossos,_),
   CasaVerda =casa(_,verda,_,_),
   CasaAlema =casa(alema ,_,_,_).
```

En l'anterior codi s'haguera pogut canviar la variable `CasaGossos` per la variable `CasaGossosBasquet` ja que és evident que es tracta de la mateixa casa. Però s'ha preferit escriure una traducció literal de l'enunciat.

En el código anterior se hubiese podido cambiar la variable `CasaGossos` *por la variable* `CasaGossosBasquet` *ya que es evidente que se trata de la misma casa. Pero se ha preferido escribir una traducción literal del enunciado.*

Per a convertir les sis afirmacions o pistes en codi observem que totes elles fan referència a les posicions de les cases dins la terna. Per això definirem un predicat general `estala/3` de manera que

Para convertir las seis afirmaciones o pistas en código observamos que todas ellas hacen referencia a las posiciones de las casas dentro de la terna. Por eso definiremos un pedicado general `estala/3` *de manera que*

```
estala(N,C,Cs)
```

se satisfà si una casa `C` està en una posició `N` en la terna `Cs`.

se satisface si una casa `C` está en una posición `N` en la terna `Cs`.

```
estala(1,A,(A,_,_)).
estala(2,A,(_,A,_)).
estala(3,A,(_,_,A)).
```

A partir d'aquest i per millorar la legibilitat definirem també el predicat `esta/2` que se satisfà si una casa està en qualsevol posició.

A partir de éste y para mejorar la legibilidad definiremos también el predicado `esta/2` que se satisface si una casa está en cualquier posición.

```
esta(C,Cs) :-
   between(1,3,N),
   estala(N,C,Cs).
```

I dos nous predicats que consideren la posició relativa (a la dreta o a l'esquerra) de dues cases.

Y dos nuevos predicados que consideran la posición relativa (a derecha o izquierda) de dos casas.

```
aladreta(A,B,(B,A,_)).
aladreta(A,B,(_,B,A)).
alesquerra(A,B,Cs) :- aladreta(B,A,Cs).
```

Amb els predicats anteriors, les definicions dels sis predicats corresponents a les afirmacions podrien ser

Con los predicados anteriores, las definiciones de los seis predicados correspondientes a las afirmaciones podrían ser

```
pista1(C,Cs) :-                                // No està la 2a
   member(N,[1,3]), estala(N,C,Cs).               No está la 2a.

pista2(C,Cs) :-                                // Hi ha una casa que...
   esta(C,Cs).                                    Hay una casa que ...

pista3(C1,C2,Cs) :-                            // No estan al mig.
   estala(1,C1,Cs),estala(3,C2,Cs).               No están en el centro.
pista3(C1,C2,Cs) :-
   estala(3,C1,Cs),estala(1,C2,Cs).

pista4(C1,C2,Cs) :-                            // A l'esquerra
   alesquerra(C1,C2,Cs).                          A la izquierda.

pista5(C1,C2,Cs) :-                            // A la dreta
   aladreta(C1,C2,Cs).                            A la derecha.

pista6(C,Cs) :-                                // Està la 3a
   estala(3,C,Cs).                                Está la 3a.
```

En realitat, la tercera s'haguera pogut expressar com la primera (per a cada una de les cases implicades), però s'ha fet de manera diferent.

En realidad, la tercera se hubiese podido expresar como la primera (para cada una de las casas implicadas), pero se ha hecho de manera diferente.

Si fem ara la pegunta `problema(CASES).` observarem quina és la disposició de les cases que compleix totes les condicions imposades per les sis afirmacions.

Si hacemos ahora la pregunta `problema(CASES).` *observaremos cuál es la disposición de las casas que cumplen todas las condiciones impuestas por las seis afirmaciones.*

```
?-  problema(CASES).
    CASES = (casa(brasiler, _16890, peixos, futbol),
       casa(_16938, verda, gats, _16944),
       casa(alema, roja, gossos, basquet)) ;
    false.
```

Vegem que en la resposta no apareix res sobre l'anglès ni sobre el golf ni sobre la casa blava. Això és perque aquesta informació només es troba en les preguntes. La forma correcta de contestar la primera pregunta de l'enunciat seria

Vemos que en la respuesta no aparece nada sobre el inglés ni sobre el golf ni sobre la casa azul. Esto es porque esta información sólo se encuentra en las preguntas. La forma correcta de contestar la primera pregunta del enunciado sería

```
?-  problema(CASES),esta(casa(angles,_,X,_),CASES).
    CASES = (casa(brasiler, _18390, peixos, futbol),
       casa(angles, verda, gats, _17984),
       casa(alema, roja, gossos, basquet)),
    X = gats ;
    false.
```

En canvi si intentem obtenir la contestació de la segona de la mateixa manera (i independentment de la primera) veurem que no obtenim resposta.

En cambio si intentamos obtener la contestación de la segunda de la misma manera (e independientemente de la primer) veremos que no obtenemos respuesta.

En altres paraules, no podem respondre (correctament) la segona pregunta sense saber que hi ha algú que és anglès. Tenim aleshores dues opcions, o fem les dues preguntes al mateix temps, o indiquem explicitament que en alguna casa viu un anglès.

En otras palabras, no podemos responder (correctamente) la segunda pregunta sin saber que hay alguien que es inglés. Tenemos entonces dos opciones, o hacemos las dos preguntas al mismo tiempo, o indicamos explícitamente que en alguna casa vive un inglés.

En canvi, la tercera pregunta es pot fer independentment de les altres. No obstant això, el més natural seria fer les tres preguntes alhora.

En cambio, la tercera pregunta se puede hacer independientemente de las otras. Sin embargo, lo más natural sería hacer las tres preguntas a la vez.

```
?-  problema(CASES),esta(casa(angles,_,X,_),CASES),
       esta(casa(Y,_,_,golf),CASES),
       esta(casa(Z,blava,_,_),CASES).
    CASES = (casa(brasiler, blava, peixos, futbol),
       casa(angles, verda, gats, golf),
       casa(alema, roja, gossos, basquet)),
    X = gats,
    Y = angles,
    Z = brasiler ;
    false.
```

Problema A.5: [ultim]

Dissenya un predicat recursiu en PROLOG--, `ultim/3`, de manera que `ultim(L,E,R)` siga cert si la llista `L` és igual a la llista `R` afegint l'element `E` al final.

Diseña un predicado recursivo en PROLOG--, `ultim/3`, *de manera que* `ultim(L,E,R)` *sea cierto si la lista* `L` *es igual a la lista* `P` *añadiendo el elemento* `E` *al final.*

Com que ens demanen un predicat recursiu especificarem primer la idea recursiva. Considerarem el primer argument com a entrada, i es tracta de calcular quin és l'últim element de `L` i quina és la llista romanent quan aquest s'elimina.

Como nos piden un predicado recursivo especificaremos primero la idea recursiva. Consideraremos el primer argumento como entrada, y se trata de calcular cuál es el último elemento de `L` *y cuál es la lista restante cuando éste se elimina.*

La idea recursiva per calcular l'últim element és molt senzilla: l'últim d'una llista (de dos o més elements) és el mateix que l´últim de la subllista sense el seu cap. En el cas base, l'últim d'una llista d'un element és eixe element. Això mateix, en PROLOG--,

La idea recursiva para calcular el último elemento es muy sencilla: el último de una lista (de dos o más elementos) es el mismo que el último de la sublista sin su cabeza. En el caso base, el último de una lista de un elemento es ese mismo elemento. Esto mismo, en PROLOG--,

```
1. ultim([_|L],E) :-
2.    ultim(L,E).
3. ultim([H],H).

?- ultim([a,b,c],E).
   E=c.
```

Tal i com s'ha escrit en el codi anterior, els casos recursiu i base no són estrictament exclusius ja que la cua `L` en la línia 1 pot ser buida. No obstant això, en eixe cas tindríem en la segona línia `ultim([],E)` que seria fals. O siga que la línia 2 està fent el mateix paper que la condició que caldria afegir per a comprovar que `L` **no** siga buida.

Tal como se ha escrito en el código anterior, los casos recursivo y base no son estrictamente exclusivos ya que la cola `L` *en la línea 1 puede estar vacía. Sin embargo, en ese caso tendríamos en la segunda línea* `ultim([],E)` *que sería falso. O sea que la línea 2 está haciendo el mismo papel que la condición que habría que añadir para comprobar que* `L` ***no*** *sea vacía.*

Per completar el disseny de `ultim/3` afegirem el tercer argument que haurà de contindre tots els elements del primer argument (i en el mateix ordre) llevat de l'últim. És a dir,

Para completar el diseño de `ultim/3` *añadiremos el tercer argumento que tendrá que contener todos los elementos del primer argumento (y en el mismo orden) excepto el último. Es decir,*

```
ultim([H|L],E,[H|R]) :-
   ultim(L,E,R).
ultim([H],H,[]).

?- ultim([a,b,c],E,R).
   E=c, R=[a,b].
```

Podem ara comparar aquesta versió recursiva amb la no recursiva, més natural, que faria ús del predicat estàndar `append/3`.

Podemos ahora comparar esta versión recursiva con la no recursiva, más natural, que usaría el predicado estándard `append(3)`*.*

```
ultim(L,E,R) :-
   append(R,[E],L).

?- ultim([a,b,c],E,R).
   E=c, R=[a,b].
```

Problema A.6: [longitudLlista]

a) Dissenya una funció recursiva que donada una llista i un element qualsevol torne el nombre de vegades que apareix aquest element en la llista. b) Expressa mitjançant lògica de predicats un predicat recursiu equivalent. c) Dissenya el corresponent predicat recursiu en PROLOG--.

a) Diseña una función recursiva que dada un alista y un elemento cualquiera devuelva el número de veces que aparece este elemento en la lista. b) Expresa mediante lógica de predicados un predicado recursivo equivalente. c) Diseña el correspondiente predicado recursivo en PROLOG--.

La funció que se'ns demana és clarament una aplicació exhaustiva entre el conjunt de tots els possibles parells de llistes i elements, $\mathbb{L} \times \mathcal{U}$ (domini), i els enters no negatius, $\mathbb{Z}^+$ (codomini).

La función que se pide es claramente una aplicación exhaustiva entre el conjunto de todos los posibles pares de listas y elementos, $\mathbb{L} \times \mathcal{U}$ (dominio), y los enteros no negativos, $\mathbb{Z}^+$ (codominio).

$$p : \mathbb{L} \times \mathcal{U} \longrightarrow \mathbb{Z}^+$$

Primer caldrà relacionar la funció per a un cas general (una llista qualsevol) i un cas més senzill (la mateixa llista sense el seu cap) la qual cosa és relativament directa en aquest cas.

Primero habrá que relacionar la función para un caso general (una lista cualquiera) y un caso más sencillo (la misma lista sin su cabeza) lo que es relativamente directo en este caso.

Si l'element apareix n vegades en la llista sense el cap, a aquesta quantitat caldrà sumar 1 o 0 en funció de què el cap de la llista siga igual o no l'element.

Si el elemento aparece n veces en la lista sin la cabeza, a esta cantidad habrá que sumar 1 ó 0 en función de que la cabeza de la lista sea igual o no al elemento.

I com en tota definició recursiva caldrà també definir els casos base que ara només serà un i molt senzill: en la llista buida qualsevol element apareix zero vegades.

Y como en toda definición recursiva habrá también que definir los casos base que ahora sólo será uno y muy sencillo: en la lista vacía cualquier elemento aparece cero veces.

Aleshores la definició recursiva que se'ns demana es pot escriure com a

Entonces la definición recursiva que se pide se puede escribir como

$$p(L,e)=\begin{cases}p(cua(L),e), & \text{si } L\neq\emptyset\wedge cap(L)\neq e\\ 1+p(cua(L),e), & \text{si } L\neq\emptyset\wedge cap(L)=e\\ 0, & \text{si } L=\emptyset\end{cases}$$

O simplement com a

O simplemente como

$$p(L,e)=\begin{cases}0, & \text{si } L=\emptyset\\ p(cua(L),e)+(cap(L)=e), & \text{si no}\end{cases}$$

si entenem $(cap(L)=e)$ com a funció característica.

si entendemos $(cap(L)=e)$ como función característica.

Per al predicat corresponent considerarem només la implicació en un sentit (deducció a partir de casos més senzills). Aleshores podem escriure,

Para el predicado correspondiente consideraremos sólo la implicación en un sentido (deducción a partir de caso más sencillos). Entonces podemos escribir,

$$\forall L\in\mathbb{L},\forall e\in U,\forall v\in\mathbb{Z}^+,\ p(L,e,v)\Leftarrow\begin{cases}(L\neq\emptyset\wedge(cap(L)\neq e)\wedge(p(cua(L),e,v))\\ \vee\\ (L\neq\emptyset\wedge(cap(L)=e)\wedge(p(cua(L),e,v-1))\\ \vee\\ (L=\emptyset\wedge v=0)\end{cases}$$

O equivalentment d'una manera més usual en lògica de predicats,

O equivalentemente de una manera más usual en lógica de predicados,

$$\begin{cases}\forall L\in\mathbb{L},\forall e\in U,\forall v\in\mathbb{Z}^+,(L\neq\emptyset\wedge(cap(L)\neq e)\wedge(p(cua(L),e,v))\Rightarrow p(L,e,v)\\ \wedge\\ \forall L\in\mathbb{L},\forall e\in U,\forall v\in\mathbb{Z}^+,(L\neq\emptyset\wedge(cap(L)=e)\wedge(p(cua(L),e,v-1))\Rightarrow p(L,e,v)\\ \wedge\\ \forall e\in U,p(\emptyset,e,0)\end{cases}$$

I ara podem fer un canvi de variable per a escriure

Y ahora podemos hacer un cambio de variable para escribir

$$\begin{cases} \forall L \in \mathbb{L}, \forall e, h \in U, \forall v \in \mathbb{Z}^+, (h \neq e) \wedge (p(cua(L), e, v)) \Rightarrow p(h|L, e, v) \\ \wedge \\ \forall L \in \mathbb{L}, \forall e \in U, \forall v \in \mathbb{Z}^+, p(cua(L), e, v-1) \Rightarrow p(e|L, e, v) \\ \wedge \\ \forall e \in U, p(\emptyset, e, 0) \end{cases}$$

Aquesta darrera definició es tradueix a PROLOG-- d'una manera quasi literal.

Esta última definición se traduce a PROLOG-- *de manera casi literal.*

```
p([H|L],E,V) :-  % Cas recursiu 1
                   Caso recursivo 1
   H\==E,
   p(L,E,V).

p([E|L],E,V) :-  % Cas recursiu 2
                   Caso recursivo 2
   p(L,E,V1),
   V is 1+V1.

p([],_,0).       % Cas base
                   Caso base
```

Problema A.7: [esMembre]

Dissenya un predicat recursiu que donat un element qualsevol i una llista siga cert o fals si l'element està o no està en la llista. Escriu diferents definicions si les trobes i les corresponents traduccions a PROLOG--.

Diseña un predicado recursivo que dado un elemento cualquiera y una lista sea cierto o falso si el elemento está o no está en la lista. Escribe diferentes definiciones si la encuentras y las correspondientes traducciones a PROLOG--.

La relació entre el predicat per a una llista qualsevol i la mateixa sense el seu cap depén de si el cap de la llista és o no igual a l'element. En particular, podem escriure

La relación entre el predicado para una lista cualquiera y la misma sin su cabeza depende de si la cabeza de la lista es o no igual a el elemento. En particular, podemos escribir

$$\forall e \in \mathsf{U}, \forall L \in \mathbb{L},\ m(e,L) \Leftarrow \begin{cases} ((L \neq \emptyset) \wedge (cap(L) = e)) \\ \qquad \vee \\ ((L \neq \emptyset) \wedge (cap(L) \neq e) \wedge (m(e, cua(L))) \end{cases}$$

ja que només interessa poder demostrar quan el predicat és cert. En cas que es volguera demostrar també quan el predicat és fals caldria afegir

ya que sólo nos interesa poder demostrar cuándo el predicado es cierto. En el caso de que se quisiera demostrar también cuándo el predicado es falso habría que añadir

$$\forall e \in \mathsf{U}, \forall L \in \mathbb{L},\ \neg m(e,L) \Leftarrow \begin{cases} (L = \emptyset) \\ \qquad \vee \\ ((L \neq \emptyset) \wedge (cap(L) \neq e) \wedge (\neg m(e, cua(L))) \end{cases}$$

Si ens restringim només al primer cas com és usual, podem reescriure la primera definició com a

Si nos restringimos sólo al primer caso como es usual, podemos reescribir la primera definición como

$$\begin{cases} \forall e \in \mathsf{U}, \forall L \in \mathbb{L}, & m(e, e|L) \\ & \wedge \\ \forall e, h \in \mathsf{U}, \forall L \in \mathbb{L}, & ((h \neq e) \wedge m(e, cua(L))) \Rightarrow m(e, h|L) \end{cases}$$

En totes les definicions anteriors, les regles (el dos casos de la definició) són mútuament exclusives. És a dir, només un dels dos casos és cert al mateix temps.

En todas las definiciones anteriores, las reglas (los dos casos de la definición) son mútuamente exclusivas. Es decir, sólo unmo de los casos es cierto al mismo tiempo.

Però si ho pensem be, la implicació

Pero si lo pensamos bien, la implicación

$$m(e, cua(L)) \Rightarrow m(e, h|L)$$

és certa independentment de què el cap de la llista, h, siga o no igual a e. En altres paraules, podem escriure una definició equivalent més senzilla on les regles no són mútuament exclusives.

es cierta independientemente de que la cabeza de la lista, h, sea o no igual a e. En otras palabras, podemos escribir una definición equivalente más sencilla donde las reglas no son mútuamente exclusivas.

$$\begin{cases} \forall e \in \mathbb{U}, \forall L \in \mathbb{L}, & m(e, e|L) \\ & \wedge \\ \forall e, h \in \mathbb{U}, \forall L \in \mathbb{L}, & m(e, cua(L)) \Rightarrow m(e, h|L) \end{cases}$$

La traducció a PROLOG-- d'aquestes definicions és la mateixa.

La traducción a PROLOG-- de estas definiciones es la misma.

```
m(E,[H|L]) :-    % Cas recursiu
                   Caso recursivo
   H\==E, % (*)
   m(E,L).

m(E,[E|_]).      % Cas base
                   Caso base
```

L'única diferència entre les dues versions consistiria a incloure o no la línia marcada amb (*). Tot i que les dues definicions són lògicament equivalents, el funcionament dels corresponents programes en PROLOG-- seria diferent.

La única diferencia entre las dos versiones consistiría en incluir o no la línea marcada con (*). Aunque las dos definiciones son lógicamente equivalentes, el funcionamiento de los correspondientes programas en PROLOG-- sería diferente.

En la segona definició (sense la línia (*)), la recursió sempre esgotaria la llista d'entrada, i l'intèrpret de PROLOG-- contestaria per cada vegada que l'element apareguera en la llista.

En la segunda definición (sin la línea (*)), la recursión siempre agotaría la lista de entrada, y el intérprete de PROLOG-- contestaría por cada una de las veces en que el elemento apareciese en la lista.

En canvi, en la primera on les regles són excloents, l'intèrpret només contestaria la primera vegada que es trobara l'element en la llista i la recursió acabaria.

En cambio, en la primera donde las reglas son excluyentes, el intérprete sólo contestaría la primera vez que se encontrara el elemento en la lista y la recursión acabaría.

Per cert, la implementació del predicat predefinit `member/2` es correspon amb la segona de les definicions.

Por cierto, la implementación del predicado predefinido `member/2` se corresponde con la segunda de las definiciones.

Problema A.8: [esMembreBis]

Trobar una definició no recursiva en PROLOG--, a) per al predicat `member/2` en funció dels predicats `select/3` i/o `append/3`. b) per al predicat `select/3` en funció de `append/3`.

Encontrar una definición no recursiva en PROLOG--, *a) para el predicado* `member/2` *en función de los predicados* `select/3` *y/o* `append/3`. *b) para el predicado* `select/3` *en función de* `append/3`.

La definició en funció de `select/3` és molt senzilla ja que els dos predicats són molt similars.

La definición en función de `select/3` *es muy sencilla ya que los dos predicados son muy similares.*

```
m(E,L) :-
   select(E,L,_).
```

El mateix fent servir `append/3` és relativament similar.

Lo mismo usando `append/3` *es relativamente similar.*

```
m(E,L) :-
   append(_,[E|_],L).
```

I finalment, la definició de `select/3` a partir de `append/3` és una generalització de l'anterior.

Y finalmente, la definición de `select/3` *a partir de* `append/3` *es una generalización de lo anterior.*

```
s(E,L,R) :-
   append(L1,[E|L2],L),
   append(L1,L2,R).
```

En realitat, la implementació anterior no és exactament equivalent a la implementació del predicat estàndar `select/3` ja que quan la primera de les dues llistes no està instanciada es produeix una recursió infinita.

En realidad, la implementación anterior no es exactamente equivalente a la implementación del predicado estándar `select/3` *ya que cuando la primera de las dos listas no está instanciada se produce una recursión infinita.*

Per a obtenir un comportament idèntic en aquest cas, caldria invertir les dues línies de què consta la definició anterior.	*Para obtener un comportamiento idéntico en este caso, habría que invertir las dos líneas de que consta la definición anterior.*
Des del punt de vista lògic, les dues definicions són la mateixa ja que la conjunció és commutativa.	*Desde el punto de vista lógico, las dos definiciones son la misma ya que la conjunción es conmutativa.*

A.6 Problemes proposats

Problemas Propuestos

Problema A.9: [bufons]

En un regne molt, molt llunyà, tothom és cavaller, bufó o espia. Els cavallers sempre diuen la veritat, els bufons sempre diuen mentides, i els espies diuen el que els interessa.

En un reino muy, muy lejano, todos son caballeros, bufones o espías. Los caballeros siempre dicen la verdad, los bufones siempre dicen mentiras, y los espías dicen lo les interesa.

Un cavaller, un bufó i un espia tenen una conversació:

Un caballero, un bufón y un espía tienen una conversación:

a) Alleyn: Carvalho és un bufó.

a) Alleyn: Carvalho es un bufón.

b) Bond: Alleyn es un cavaller.

b) Bond: Alleyn es un caballero.

c) Carvalho: Jo sóc un espia.

c) Carvalho: Yo soy un espía.

Sabries endevinar qui és qui en la conversa? Podries resoldre el problema usant `PROLOG--`? Fes-ho.

¿Sabrías adivinar quién es quién? ¿Podrías resolver el problema usando `PROLOG--`*? Hazlo.*

Problema A.10: [paraigua]

Quatre amics arriben a una festa un dia plujós i han de travessar el pati.

Cuatro amigos llegan a una fiesta un día lluvioso y han de atravesar el patio.

El primer problema és que només tenen un paraigua que com a molt tapa a dues persones.

El primer problema es que sólo tienen un paraguas que como mucho cubre a dos personas.

L'altre problema és que, per diferents problemes físics, no tots ells són igual de ràpids i per travessar el patí necessiten 1, 2, 5 i 8 minuts, respectivament. Òbviament, quan dos van junts han d'anar a la velocitat del més lent.

El otro problema es que, por diferentes problemas físicos, no todos ellos son igual de rápidos y para atravesar el patio necesitan 1, 2, 4 y 8 minutos, respectivamente. Óbviamente, cuando dos van juntos deben ir a la velocidad del más lento.

El més ràpid proposa immediatament una solució: jo, que sóc el més ràpid, aniré i tornaré les vegades que faça falta i vos acompanyaré a tots. En total tardarem 17 minuts en 5 viatges: 2, 5 i 8 minuts per acompanyar-vos als tres i 2 viatges de tornada de 1 minut.

El más rápido propone inmediatamente una solución: yo, que soy el más rápido, iré y volveré las veces que haga falta y os acompañaré a todos. En total tardaremos 17 minutos en 5 viajes: 2, 5 y 8 minutos para acompañaros a los tres y 2 viajes de vuelta de 1 minuto.

És possible que entren tots a la festa sense mullar-se en menys de 17 minuts? Com? És possible obtindre la solució al problema usant PROLOG--? Com?

¿Es posible que entren todos a la fiesta sin mojarse en menos de 17 minutos? ¿Cómo? ¿Es posible obtener la solución al problema usando PROLOG--? ¿Cómo?

Problema A.11: [llopbis]

Considera el problema del llop, la cabra i la col de la pàgina 265 i imagina que el pagès també ha comprat un lleó que només li agrada menjar-se llops. De quantes i quines maneres diferents pot arribar el pagès a casa? Amb quantes i quines coses arribaria en el pitjor cas? Amb quants viatges?

Considera el problema del lobo, la cabra y la col de la página 265 e imagina que el labrador también ha comprado un león que sólo le gusta comer lobos. ¿De cuántas y de qué maneras diferentes puede llegar el labrador a casa? ¿Con cuántas y con qué cosas llegaría en el peor caso? ¿Con cuántos viajes?

Per exemple, si transporta primer la col, el llop es podria menjar la cabra. I si en tornar transportara el lleó i després el llop, arribaria a casa amb el lleó, el llop i la col.

Por ejemplo, si transporta primero la col, el lobo se podría comer a la cabra. Y si al volver transportara al león y después al lobo, llegaría a casa con el león, el lobo y la col.

Suposa, com en l'exemple, que en absència del pagès només pot ocórrer una consumició.

Supón, como en el ejemplo, que en ausencia del labrador sólo puede darse una consumición.

Problema A.12: [zebra]

Hi ha 5 cases, en línia, de 5 colors diferents on viuen persones de 5 diferents nacionalitats, que tenen 5 tipus d'animals diferents, beuen 5 begudes diferents i fumen 5 marques diferents de cigarrets. Considera les pistes següents:

Hay 5 casas, en línea, de 5 colores diferentes donde viven personas de 5 diferentes nacionalidades, que tienen 5 tipos de animales diferentes, beben 5 bebidas diferentes y fuman 5 marcas diferentes de cigarrillos. Considera las pistas siguientes:

a) L'anglès viu a la casa roja.

a) El inglés vive en la casa roja.

b) L'espanyol té un gos.

b) El español tiene un perro.

c) En la casa verda es beu cafè.

c) En la casa verde se bebe café.

d) L'ucraïnès beu te.

d) El ucraniano bebe te.

e) La casa verda està al costat de la blanca.

e) La casa verde está al lado de la blanca.

f) Qui fuma *Old Gold* té caragols.

f) El que fuma Old Gold *tiene caracoles.*

g) En la casa groga es fuma *Kools*.

g) En la casa amarilla se fuma Kools.

h) En la casa del mig beuen llet.

h) En la casa del medio beben leche.

i) El noruec viu en la primera casa.

i) El noruego vive en la primera casa.

j) Qui fuma *Chesterfields* viu al costat de qui té una rabosa.

j) Quien fuma Chesterfields *vive al lado de quien tiene un zorro.*

k) Fumen *Kools* al costat d'on tenen un cavall.

k) Fuman Kools *al lado de donde tienen un caballo.*

l) Qui fuma *Lucky Strike* beu suc de taronja.

l) Quien fuma Lucky Strike *bebe zumo de naranja.*

m) El japonès fuma *Parliaments*.

m) El japonés fuma Parliaments.

n) El noruec viu al costat de la casa blava.

n) El noruego vive al lado de la casa azul.

Contesta: Qui beu aigua? Qui té una zebra? Resol el problema també mitjançant PROLOG--.

Contesta: ¿Quién bebe agua? ¿Quién tiene una cebra? Resuelve el problema también mediante PROLOG--.

Problema A.13: [invertir]

Dissenya un predicat recursiu en PROLOG-- de 2 arguments que siguen llistes i que siga cert si una és igual a l'altra en ordre invers dels seus elements.

Diseña un predicado recursivo en PROLOG-- *de 2 argumentos que sean listas y que sea cierto si una es igual a la otra en orden inverso de sus elementos.*

Problema A.14: [selectAppendRec]

Dissenya una versió recursiva en PROLOG-- per als predicats `select/3` i `append/3`.

Diseña una versión recursiva en PROLOG-- *para los predicados* `select/3` *i* `append/3`.

```
?- select(2,[1,2,3,2],X). --> X=[1,3,2]
?- append([1,2,3,2],[4,3,2],X). --> X=[1,2,3,2,4,3,2]
```

Problema A.15: [unioRec]

Dissenya un predicat recursiu en PROLOG-- que donades tres llistes, siga cert si la tercera llista conté els elements que estiguen en alguna de les dues primeres sense cap repetició.

Diseña un predicado recursivo en PROLOG-- *que dadas tres listas, sea cierto si la tercera lista contiene los elementos que estén en alguna de las dos primeras sin ninguna repetición.*

```
?- unio([1,2,3,2],[4,3,2],X). --> X=[1,2,3,4]
```

Problema A.16: [interseccioRec]

Dissenya un predicat recursiu en PROLOG-- que donades tres llistes, siga cert si la tercera llista conté només els elements que estan en les dues primeres sense cap repetició.

Diseña un predicado recursivo en PROLOG-- *que dadas tres listas, sea cierto si la tercera lista contiene sólo los elementos que estén en las dos primeras sin ninguna repetición.*

```
?- interseccio([1,2,3,2],[4,3,2],X). --> X=[2,3]
```

Índex analític

Historial de versions

1.0 (15 de gener, 2020):	versió enviada al Servei de Política Lingüística.
1.1 (25 de febrer, 2020):	versió revisada i acabada.
2.0 (8 de març, 2020):	primera versió bilingüe.
2.1 (24 de març, 2020):	publicació en Roderic[‡].
2.2 (18 de gener, 2021):	afegit apèndix sobre Prolog
2.3 (19 d'agost, 2021):	versió bilingüe completa
2.4 (4 d'agost de 2022):	codificació UTF, errades, addicions, més problemes

[‡] https://roderic.uv.es/handle/10550/73645