

## RAONAMENT AUTOMÀTIC

**Regla d'inferència:** Patrons estàndards de comportament lògic, permeten derivar conclusions lògiques a partir de sentències que són certes.

**Exemple:** la regla del modus ponens, és:  $a \rightarrow b$  és cert  
 $a$  és cert  
-----  
 $b$  és cert

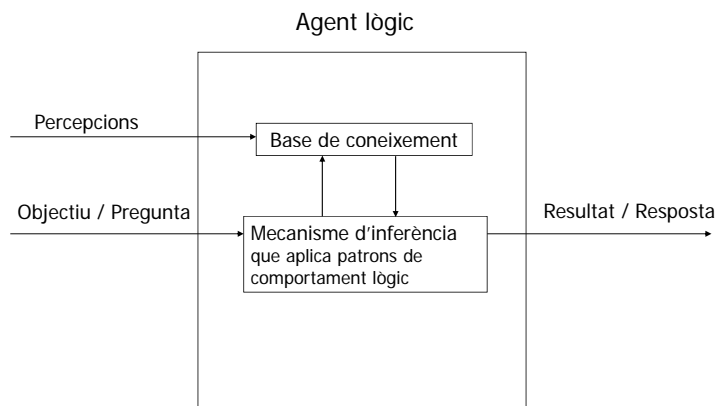
Aquesta regla ens permet demostrar que:  $[a \rightarrow b, a] \vdash b$  és cert

**Notació:** Expressió lògica que es vol demostrar, es denota com

$$F \vdash A$$

i es llegeix, "*F* permet deduir *A*"

és a dir, a partir de  $F = \{\text{coneixement a priori, percepcions}\}$ , hem de poder deduir lògicament que *A* és cert.



## Lògica Proposicional (LP)

**Lèxic:** sentència, àtom, proposició, connectiva lògica ( $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ ), àtoms especials: cert i fals.

**Estructural:** L'estructura en LP ve definida per la següent gramàtica:

Àtom :- cert | fals | Proposició.  
Proposició :- <cadena de caràcters>.  
Sentència :- Àtom |  $\neg$ Sentència |  
Sentència  $\wedge$  Sentència |  
Sentència  $\vee$  Sentència |  
Sentència  $\rightarrow$  Sentència |  
Sentència  $\leftrightarrow$  Sentència.

**Semàntic:** les proposicions representen els objectes i les accions, les sentències agafen el significat segons les connectives lògiques. Aquestes tenen el significat que els donen les taules de veritat.

**Procediments:** proc. que implementen les taules de veritat, proc. que apliquen regles d'inferència, proc. que comproven si dues proposicions són iguals.

**La semàntica de les connectives lògiques** queda totalment especificada amb les seves taules de veritat

**Taules de veritat:**

A	B	$\neg A$	$A \leftrightarrow B$	$A \rightarrow B$	$A \wedge B$	$A \vee B$
F	F	V	V	V	F	F
F	V	V	F	V	F	V
V	F	F	F	F	F	V
V	V	F	V	V	V	V

**Cas particular:** La tautologia

**Tautologia** és una expressió lògica amb una taula de veritat que és sempre certa, independentment de la veritat o la falsedat de les seves parts.

Per tant, si l'objectiu  $[ ] \vdash T$  és cert, aleshores, T és una tautologia

**Exemple:**  $(P \rightarrow Q) \leftrightarrow (\neg P \vee Q)$

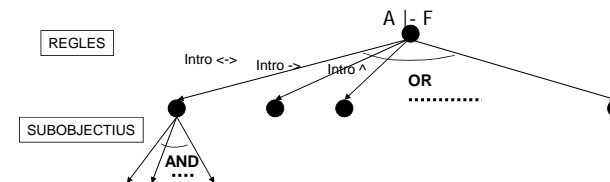
P	Q	$\neg P$	$P \rightarrow Q$	$\neg P \vee Q$	$(P \rightarrow Q) \leftrightarrow (\neg P \vee Q)$
F	F	V	V	V	V
F	V	V	V	V	V
V	F	F	F	F	V
V	V	F	V	V	V

### Regles de la deducció natural

Regla	Nom	Objectiu	Sub-Objectius
1	Introducció de $\leftrightarrow$	$F \vdash A \leftrightarrow B$	$F \vdash A \rightarrow B$ $F \vdash B \rightarrow A$
2	Introducció de $\rightarrow$	$F \vdash A \rightarrow B$	$[A F] \vdash B$
3	Introducció de $\wedge$	$F \vdash A \wedge B$	$F \vdash A$ $F \vdash B$
4	Introducció de $\vee$ (esq.)	$F \vdash A \vee B$	$F \vdash A$
5	Introducció de $\vee$ (dre.)	$F \vdash A \vee B$	$F \vdash B$
6	Introducció de $\neg$	$F \vdash \neg A$	$[A F] \vdash \text{fals}$
7	Eliminació de $\rightarrow$	$F \vdash C$	$A \rightarrow B \in F$ $F \vdash A$ $[B F] \vdash C$
8	Eliminació de $\wedge$	$F \vdash C$	$A \wedge B \in F$ $[A, B F] \vdash C$
9	Eliminació de $\vee$	$F \vdash C$	$A \vee B \in F$ $[A F] \vdash C$ $[B F] \vdash C$
10	Eliminació de $\neg$	$F \vdash C$	$\neg A \in F$ $F \vdash A$
11	Contradició	$F \vdash C$	$F \vdash \text{fals}$
12	Negació Doble	$F \vdash A$	$F \vdash \neg(\neg A)$
13	Immediata	$F \vdash A$	$A \in F$

### Mecanisme de la deducció natural: (Arbre OR-AND)

**Demostració d'un objectiu** = Cercar un camí complet en un arbre (OR-AND) que demostrï tots els subobjectius.



**Problemes:** (a l'hora de fer-ho computacional)

- Factor de ramificació molt alt (Exemple:  $[a, a \rightarrow b] \vdash b$ ) (Cost alt)
- Existència de camins infinits a l'arbre (Exemple:  $[a \rightarrow b, a \vee c] \vdash b \vee c$ ) (No complet)

### Computació de la Deducció Natural:

```
Funció Demostrar (<Objectiu>,<substitucio>)
  i:=1;
  Repetir
    Resultat=Crida(<Regla_i>(<Objectiu>));
    i++;
  Mentre(Resultat <> "Demostrat" o i<14);
  Si(Resultat="Demostrat") llavors Retorna(<Substitució>);
  Sinó Retorna("No es pot demostrar");
  FSi
Ffunció

Funció <Nom_de_la_regla> (<Objectiu>)
  Resultat=Demostrar(<Subobjectiu_1>,[,]);
  Si (Resultat<>"Demostrat") llavors Retorna("Fals"); Fsi
  Resultat=Demostrar(Aplicar_substitucio(<Subobjectiu_2>,Resultat));
  Si (Resultat<>"Demostrat") llavors Retorna("Fals"); Fsi
  ...
  Resultat=Demostrar(Aplicar_substitucio(<Subobjectiu_N>,Resultat));
  Si (Resultat<>"Demostrat") llavors Retorna("Fals"); Fsi
Ffunció
```

### Problemes de l'algorisme de la deducció natural:

- Factor de ramificació molt alt per demostrar coses simples

(Exercici: Demostreu amb deducció natural que  $[a, a \rightarrow b] \vdash b$ )

- Existència de camins infinits a l'arbre

(Exercici: Demostreu amb deducció natural que  $[a \rightarrow b, a \vee c] \vdash b \vee c$ )

### Lògica proposicional és insuficient,

Exemple de raonament lògic:

Coneixement a priori: "Qualsevol animal amb plomes és un ocell" en LP:  $ap \rightarrow ao$

Percepció: "El canari té plomes" en LP: cp

Deducció: "El canari és un ocell" en LP: co

En LP no és possible demostrar:  $[ap \rightarrow ao, cp] \vdash co$

necessitem variables i termes per representar conceptes generals i instàncies per separat.

### Lògica de predicats,

Coneixement a priori: "Qualsevol animal amb plomes és un ocell" en LP1:  $\forall x(p(x) \rightarrow o(x))$

Percepció: "El canari té plomes" en LP1:  $p(\text{canari})$

Deducció: "El canari és un ocell" en LP:  $o(\text{canari})$

En LP1 sí que es pot demostrar:  $[\forall x p(x) \rightarrow o(x), p(\text{canari})] \vdash o(\text{canari})$

### Lògica de predicats (LP1)

**Lèxic**: predicats, termes, constants, variables, connectiva lògica ( $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ ), quantificadors ( $\forall, \exists$ )  
predicats especials: cert i fals.

**Estructural**: L'estructura en LP ve definida per la següent gramàtica:

Predicat :- fals | cert | Terme.  
Terme :- Constant | Variable | Funtor(Terme,...).  
Constant :- <cadena de caràcters en minúscula>.  
Variable :- <cadena de caràcters en majúscula>.  
Funtor:- Constant.  
Sentència :- Predicat |  $\neg$ Sentència |  
Sentència  $\wedge$  Sentència |  
Sentència  $\vee$  Sentència |  
Sentència  $\rightarrow$  Sentència |  
Sentència  $\leftrightarrow$  Sentència |  
Quantificador Variable Sentència  
Quantificador :-  $\forall$  |  $\exists$

**Semàntic**: les constants representen objectes concrets, les variables objectes indefinits, els termes representen relacions o propietats dels objectes, el significat de les connectives lògiques ve donat per les taules de veritat i els quantificadors especifiquen l'abast de les variables.

**Procediments**: proc. que implementen les taules de veritat, proc. que apliquen regles d'inferència, proc. que comproven si dues constants són iguals, proc. que unifiquen parelles de termes, proc. que passen expressions a forma clausal.

### Procediment d'Unificació:

**En LP**, dues proposicions són iguals si venen donades per la mateixa tira de caràcters,

**En LP1**, dos termes són iguals si existeix una substitució de les variables que fa que siguin iguals,

**Objectiu de l'unificació:** trobar la substitució que fa que dos termes siguin iguals, entenem per substitució un conjunt de parelles de la forma variable igual a valor.

#### Exemple:

Terme 1: p(X)                      unifiquen per la substitució S,  
Terme 2: p(canari)                      S={ X=canari }

### Algorisme d'Unificació: per a dos termes donats, P i Q

```
Funció Unificar(P,Q)
1. Cas ( P i Q constants iguals) Retornar(Cert)
2. Cas (P variable i Q no variable) Retornar(Cert, P=Q)
3. Cas (Q variable i P no variable) Retornar(Cert, Q=P)
4. Cas (P i Q variables) Retornar(Cert, P i Q són la mateixa variable)
5. Cas (P i Q són termes i P=p(A1,...,An) i Q=q(B1,...,Bm))
   • Si (n i m distints) Retornar (Fals)
   • Si (p i q no són constants iguals) Retornar(Fals)
   • En altre cas
     • i=0;
     • Repetir
       • i++;
       • Ri= Unificar(Ai,Bi);
       • Actualitzar_Variables(Ri)
     • Fins que (Ri=fals o i=n)
     • Si Ri=fals llavors Retornar(Fals)
     • Sinó Retornar(Cert, {Ri}i=1..n)
     • Fsi
   • Fsi
6. Fcas
7. Retornar(Fals)
FFunció
```

**Mecanisme de la resolució**, intenta solucionar els problemes de la deducció natural. Es basa en l'aplicació d'una sola regla:

**Regla de la resolució:** Donades, P i Q en forma clausal  
P=  $\neg A1 \vee B$  és cert  
Q=  $A2 \vee C$  és cert  
A1 i A2 unifiquen amb S  
-----  
 $(B \vee C)_S$  és cert

**Demostració d'un objectiu** = Demostrar que F |- A és cert, passa per demostrar que  $\neg A$  és inconsistent amb F.

**Demostració d'inconsistència** = Un conjunt és inconsistent si es pot deduir el conjunt buit després de l'aplicació iterativa de la regla de la resolució.

**Avantatges:** El mecanisme és complet i només aplica una sola regla

**Problemes:** les assumpcions han d'estar en forma clausal, complexitat alta

Per poder aplicar el mecanisme de la resolució, necessitem que l'enunciat estigui en una forma normal, la forma clausal, vegem com ho podem aconseguir

**FORMES NORMALS:** expressions lògiques amb un número restringit de connectives lògiques.

#### Alguns exemples:

- Forma clausal ( $\wedge, \vee, \neg$ )
- Forma de Kowalski ( $\wedge, \vee, \rightarrow$ )
- Clausula de Horn ( $\wedge, \rightarrow$ )
- ...

**FORMA CLAUSAL:** expressió lògica que ve donada per la conjunció de disjuncions de termes negats i no negats.

Exemple:  $(A \vee B \vee \neg C) \wedge (\neg M \vee C \vee \neg H) \wedge (\neg C \vee K)$

**Existeix un algorisme** que assegura el pas de qualsevol expressió en **LP1** a la seva **Forma Clausal** (LP12FC) :

### Algorisme de Pas a Forma clausal

- Pas 1. Eliminació de  $\rightarrow$  i  $\leftrightarrow$ .
- Pas 2. Passar a forma prenex (quantificadors a l'esquerra).
- Pas 3. Eliminar quantificadors existencials.
- Pas 4. Eliminar quantificador universals.
- Pas 5. Dirigir negacions cap als termes.
- Pas 6. Posar en forma de conjunció de disjuncions.

### Algorisme de Pas a Forma clausal (I)

- Pas 1. Eliminació de  $\rightarrow$  i  $\leftrightarrow$ .  
 $(P \leftrightarrow Q) \leftrightarrow ((P \rightarrow Q) \wedge (Q \rightarrow P))$   
 $(P \rightarrow Q) \leftrightarrow (\neg P \vee Q)$

- Pas 2. Passar a forma prenex (quantificadors a l'esquerra).

$(A \wedge (\forall X B)) \leftrightarrow \forall X (A \wedge B)$	$A \wedge (\exists X B) \leftrightarrow \exists X (A \wedge B)$
$(A \vee (\forall X B)) \leftrightarrow \forall X (A \vee B)$	$A \vee (\exists X B) \leftrightarrow \exists X (A \vee B)$
$((\forall X A) \vee B) \leftrightarrow \forall X (A \vee B)$	$((\exists X A) \vee B) \leftrightarrow \exists X (A \vee B)$
$((\forall X A) \wedge B) \leftrightarrow \forall X (A \wedge B)$	$((\exists X A) \wedge B) \leftrightarrow \exists X (A \wedge B)$
$\neg(\forall X A) \leftrightarrow \exists X \neg A$	$\neg(\exists X A) \leftrightarrow \forall X \neg A$

- Pas 3. Eliminar quantificadors existencials.

- Cas 1. Les variables quantificades existencialment que no estan dins l'àmbit de quantificadors universals es substitueixen per constants d'Skolem.
- Cas 2. Les variables quantificades existencialment que estan dins l'àmbit de quantificadors universals es substitueixen per funcions d'Skolem.

### Algorisme de Pas a Forma clausal (II)

- Pas 4. Eliminar quantificador universals, ja que en aquest punt totes les variables estan quantificades universalment.

- Pas 5. Dirigir negacions cap als termes, fent servir:

$$\begin{aligned} \neg(P \wedge Q) &\leftrightarrow (\neg P \vee \neg Q) \\ \neg(P \vee Q) &\leftrightarrow (\neg P \wedge \neg Q) \\ \neg(\neg P) &\leftrightarrow P \end{aligned}$$

- Pas 6. Posar en forma de conjunció de disjuncions, fent servir:

$$(P \vee (Q \wedge R)) \leftrightarrow ((P \vee Q) \wedge (P \vee R))$$

### Algorisme per l'aplicació del Mecanisme de la Resolució:

Funció **Mecanisme\_Resolucio**(F|-C)

1.  $F = \text{LP12FC}(F)$ ;
2.  $R = \text{LP12FC}(\neg C)$ ;
3. Repetir
  - $F = [R|F]$ ;
  - Seleccionar dues clausules de F;
  - Aplicar la Regla de Resolució entre elles;
  - $R = \text{Resultat del pas anterior}$ ;
4. Fins que (R és el conjunt buit);
5. Si (R és buit) llavors Retornar ("C és cert");
6. Sinó Retornar ("Objectiu no demostrable");
7. Fsi

Ffunció

**Exercici:** Demostració d'objectius pel mecanisme de la resolució

Coneixement a priori: "Qualsevol animal amb plomes és un ocell" en LP1:  $\forall p(X) \rightarrow o(X)$

Percepció: "El canari té plomes" en LP1:  $p(\text{canari})$

Dedució: "El canari és un ocell" en LP:  $o(\text{canari})$

$[\forall p(X) \rightarrow o(X), p(\text{canari})] \vdash o(\text{canari})$

**Aplicació de l'algorisme:**

Pas 1.  $F = \text{LP12FC}([\forall p(X) \rightarrow o(X), p(\text{canari})]) \rightarrow F = [\neg p(X) \vee o(X), p(\text{canari})]$

Pas 2.  $R = \text{LP12FC}(\neg o(\text{canari})); \rightarrow R = \neg o(\text{canari})$

Pas 3. Iteració 1.  $F = [\neg p(X) \vee o(X), p(\text{canari}), \neg o(\text{canari})]$

Pas 4. Iteració 1. Resolució entre:  $\neg p(X) \vee o(X)$  i  $p(\text{canari})$ ;  $R = o(\text{canari})$ ;

Pas 5. Iteració 2.  $F = [\neg p(X) \vee o(X), p(\text{canari}), \neg o(\text{canari}), o(\text{canari})]$

Pas 5. Iteració 2. Resolució entre:  $o(\text{canari})$  i  $\neg o(\text{canari})$ ;  $R = \text{buit}$ ;

Pas 6. Retornar "o(canari) és cert";

**Exercici:** Expresses en LP1 el següent enunciat en llenguatge natural:

*"Qualsevol persona que té un salari alt i no és morosa, aleshores se li concedeix un crèdit. En Joan té un salari alt i no se li concedeix un crèdit."*

Demostreu amb el mecanisme de la resolució que:

*"En Joan és morós"*

**Aplicació del mecanisme de la resolució:** Resolució sistemàtica

Aplicació sòlida i completa del mecanisme de la resolució.

**Idea:** Els resolvents s'agrupen per nivells (clausules inicials = Nivell 0), i es va iterant per nivells.

**Avantatges:** Completa, permet millores importants.

**Problemes:** Complexitat Alta

**Mecanisme de la resolució en PROLOG**, és una simplificació del mecanisme de la resolució per reduir la complexitat de la seva aplicació, per això s'introdueixen 3 restriccions:

1. Només fer servir expressions que són CLAUSULES DE HORN.

$$\neg A_1 \vee \dots \vee \neg A_n \vee P \rightarrow P : \neg A_1, \dots, A_n.$$

2. Només es poden demostrar CONJUNCIONS DE TERMES NO NEGATS.

$$F \vdash P_1 \wedge P_2 \wedge \dots \wedge P_p \rightarrow P_1, P_2, \dots, P_p ?$$

3. L'ordre d'aplicació ve determinat per l'ordre de les clausules de horn.

$$\begin{array}{l} \neg A_1 \vee \dots \vee \neg A_n \vee P_1 \rightarrow P_1 : \neg A_1, \dots, A_n. \\ \vdots \\ \neg A_1 \vee \dots \vee \neg A_n \vee P_n \rightarrow P_n : \neg A_1, \dots, A_n. \end{array}$$

**FORMES NORMALS: De FORMA CLAUSAL a FORMA DE KOWALSKI:**

1. Donada una expressió en forma clausal:

$$P = P_1 \vee P_2 \vee \dots \vee P_N$$

2. Separem termes negats i no negats:

$$P = \underbrace{\neg P_1 \vee \dots \vee \neg P_k}_{\text{Termes negats}} \vee \underbrace{P_{k+1} \vee \dots \vee P_N}_{\text{Termes No negats}}$$

3. Eliminem la múltiple negació:

$$\downarrow (\neg A \vee \neg B) \leftrightarrow \neg(A \wedge B)$$

$$P = \underbrace{\neg(P_1 \wedge \dots \wedge P_k)}_{\text{Termes negats}} \vee \underbrace{P_{k+1} \vee \dots \vee P_N}_{\text{Termes No negats}}$$

4. Introduim la implicació i obtenim

$$\downarrow (\neg P \vee \neg Q) \leftrightarrow (P \rightarrow Q)$$

**FORMA DE KOWALSKI**

$$P = (P_1 \wedge \dots \wedge P_k) \rightarrow (P_{k+1} \vee \dots \vee P_N)$$

**FORMES NORMALS: CLAUSULA DE HORN**

**Definició:** Una clausula de Horn és una expressió en forma de Kowalski que només té un terme després de la implicació:

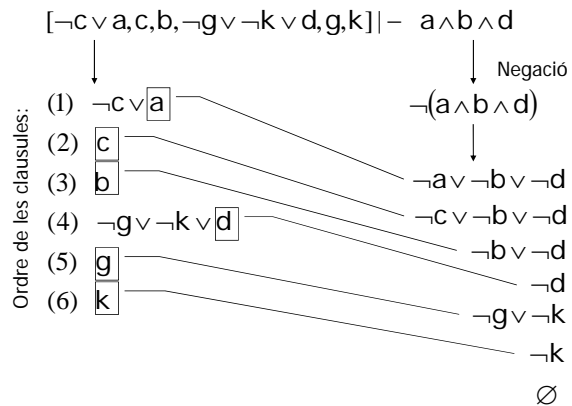
$$P = \neg P_1 \vee \dots \vee \neg P_k \vee Q$$

o bé és una expressió en forma clausal que te totes les clausules negades menys una:

$$P = (P_1 \wedge \dots \wedge P_k) \rightarrow Q$$

Les clausules de Horn són només un subconjunt de totes les expressions que es poden escriure en LP1

**Exercici:** Aplicació del mecanisme de la resolució en el PROLOG



**Algorisme per l'aplicació del Mecanisme de la Resolució en Prolog**

Funció **Mecanisme\_Resolucio\_Prolog**(F|-C)

1. F=LP12FC(F); /\* llista de clausules de Horn \*/
2. R=LP12FC(-C); /\* Disjunció de termes negats \*/
3. Repetir
  - Seleccionar el primer terme de R;
  - Seleccionar la primera clausula de F amb terme no negat que unifiqui amb el seleccionat de R;
  - Aplicar la Regla de Resolució entre elles;
  - R=Resultat del pas anterior;
4. Fins\_que(R és el conjunt buit);
5. Si (R és buit) llavors Retornar("C és cert");
6. Sinó Retornar ("Objectiu no demostrable");
7. Fsi

Ffunció

**Mecanisme del modus ponens generalitzat**, encadenament de regles)

**Demostració d'un objectiu** = passa per l'aplicació del modus ponens generalitzat de manera iterativa sobre el conjunt d'assumpcions. S'anomena **ENCADENAMENT DE REGLES**.

**Regla del Modus Ponens generalitzat:**

P1 és cert  
 P2 és cert  
 ...  
 Pn és cert  
 $P1 \wedge P2 \wedge \dots \wedge Pn \rightarrow Q$  és cert  
 i existeix una substitució, S, que fa que  $(P1 \wedge P2 \wedge \dots \wedge Pn)_S = (P1 \wedge P2 \wedge \dots \wedge Pn')_S$

-----  
 Q<sub>S</sub> és cert

**Problemes:** No està demostrat que sigui complet.  
**Avantatges:** És molt senzill d'aplicar en problemes reals.

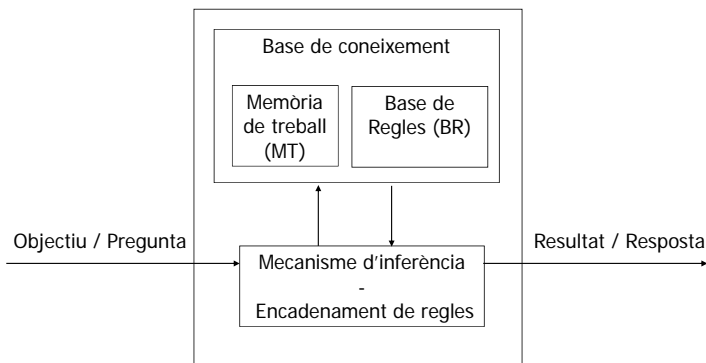
**Mecanisme del modus ponens generalitzat:**

**Regla**=Expressió lògica que conté una implicació

**Encadenament de regles:** consisteix en l'aplicació iterativa del modus ponens generalitzat. En funció de la direcció en que s'aplica el modus ponens tenim dos tipus d'encadenament:

- Encadenament endavant: Des dels antecedents cap als conseqüents.
- Encadenament enrera: Des dels conseqüents cap als antecedents, es requereix partir d'una hipòtesi.

Agent lògic = Sistema basat en regles



**Exemple:** Sistema basat en regles

**Memòria de Treball:**

a, b, d, e, g, m

**Base de Regles:**

R1:  $a \wedge b \rightarrow c$   
 R2:  $c \wedge d \wedge e \rightarrow f$   
 R3:  $f \wedge g \wedge m \rightarrow o$

**Inferència:**

Encadenament Endavant:

R1 -> MT={a, b, d, e, g, m, c}  
 R2 -> MT={a, b, d, e, g, m, c, f}  
 R3 -> MT={a, b, d, e, g, m, c, f, o}

Encadenament Enrera:

o? -> R3 -> f, g, m ?  
 f? -> R2 -> c, d, e ?  
 c? -> R1 -> a, b ?  
     a? -> MT  
     b? -> MT  
     d? -> MT  
     e? -> MT  
 g? -> MT  
 m? -> MT

### Implementació de l'encadenament de regles:

- **Aplicació seqüencial i ordenada**, es du a terme fent servir estratègies de resolució de conflictes.
- **Aplicació paral·lela o massiva**, es du a terme fent servir operacions d'àlgebra relacional en bases de dades

### Existència de conflictes en l'encadenament seqüencial:

**Regla aplicable:** Regla que té tots els seus antecedents satisfets pels fets que existeixen a la memòria de treball.

**Conflicte:** Existència de més d'una regla aplicable i necessitat de seleccionar-ne una.

#### Exemple:

##### Base de Regles:

R1:  $a \wedge b \wedge c \rightarrow \text{esborrar}(d)$   
R2:  $a \wedge m \wedge k \rightarrow \text{esborrar}(q)$   
R3:  $p \wedge d \rightarrow t$   
R4:  $p \wedge q \rightarrow z$

##### Memòria de treball:

a, b, c, m, k, p, d, q

**Regles aplicables:** R1, R2

### Estratègia de resolució de conflictes:

**Refractorietat.** Consisteix en prioritzar només les regles que no han estat aplicades cap vegada.

**Especificitat.** Consisteix en prioritzar les regles més específiques, és a dir, regles amb un major nombre d'antecedents.

**Recència.** Consisteix en prioritzar les regles que tinguin antecedents més recentment deduïts.

**Ordre de les regles.** Consisteix en prioritzar les regles d'acord amb una assignació prèvia de prioritats.

**Ordre de les dades.** Consisteix en prioritzar els fets de la MT d'acord amb una assignació prèvia de prioritats, s'aplicaran primer aquelles regles que tinguin antecedents més prioritzats.

### Estratègies per evitar conflictes:

**Limitació del context.** Consisteix en dividir la base de regles en mòduls segons criteris de significació, i aleshores només es mantenen actius certs grups al mateix temps.

**Exercici 1:** Construir un sistema basat en regles que permeti diagnosticar panes de cotxe amb el següent coneixement expert d'un mecànic

"Si el cotxe es para de sobte, s'hauria de mirar el subministrador de potència. Si no hi ha corrent des de la bateria, aleshores mirar els cables de la bateria i estrènyer-los si és necessari.

Alternativament, si hi ha corrent des de la bateria mirar i estrènyer qualsevol cable que estigui lliure.

Alternativament, si no es troba cap d'aquests símptomes, aleshores s'hauria d'examinar el distribuïdor. Si els punts estan bruts, netejar-los pot solucionar el problema però si les connexions de dins del distribuïdor són les que fallen, aleshores tu mateix no ho podràs fixar, i hauries d'anar al garatge.

Alternativament, si cap d'aquests remeis són aplicables, llavors mirar la bugia i ajustar les obertures si és que són massa grans.

Alternativament, cal mirar el subministrador de combustible. Si l'indicador del combustible diu BUIT, llavors posar gasolina, sinó buscar pèrdues de gasolina..."

A l'enunciat anterior hi ha tres tipus de coneixement:

**Síntomes:**

- s1: símptoma(cotxe-parat)
- s2: símptoma(hi-ha-corrent-bateria)
- s3: símptoma(connexions-bateria-lliures)
- s4: símptoma(cables-lliures)
- s5: símptoma(punts-bruts)
- s6: símptoma(fallen-connexions-distribuidor)
- s7: símptoma(obertures-bugies-grans)
- s8: símptoma(subministrador-combustible-buit)

**Objectius:**

- o(1): objectiu(examinar-suministrador-potencia)
- o(2): objectiu(examinar-distribuidor)
- o(3): objectiu(examinar-bugies)
- o(4): objectiu(examinar-connexions-bateria)
- o(5): objectiu(examinar-cables)
- o(6): objectiu(examinar-suministrador-gasolina)

**Solucions:**

- I(1): solucio(ajustar-connexions-bateria)
- I(2): solucio(ajustar-cables)
- I(3): solucio(netejar-punts)
- I(4): solucio(trucar-garatge)
- I(5): solucio(ajustar-bugies)
- I(6): solucio(posar-gasolina)
- I(7): solucio(eliminar-pèrdues-gasolina)

**La base de regles quedarà com:**

- R1: símptoma(cotxe-parat) -> objectiu(examinar-suministrador-potencia)
- R2: símptoma(cotxe-parat) -> objectiu(examinar-distribuidor)
- R3: símptoma(cotxe-parat) -> objectiu(examinar-bugies)
- R4: objectiu(examinar-suministrador-potencia) ^ not(símptoma(hi-ha-corrent-bateria)) -> objectiu(examinar-connexions-bateria)
- R5: objectiu(examinar-suministrador-potencia) ^ símptoma(hi-ha-corrent-bateria) -> objectiu(examinar-cables)
- R6: objectiu(examinar-connexions-bateria) ^ símptoma(connexions-bateria-lliures) -> esborrar(objectiu(examinar-suministrador-potencia)) ^ solucio(ajustar-connexions-bateria)
- R7: objectiu(examinar-cables) ^ símptoma(cables-lliures) -> esborrar(objectiu(examinar-cables)) ^ solucio(ajustar-cables)
- R8: objectiu(examinar-distribuidor) ^ símptoma(punts-bruts) -> esborrar(objectiu(examinar-distribuidor)) ^ solucio(netejar-punts)
- R9: objectiu(examinar-distribuidor) ^ símptoma(fallen-connexions-distribuidor) -> esborrar(objectiu(examinar-distribuidor)) ^ solucio(trucar-garatge)
- R10: objectiu(examinar-bugies) ^ símptoma(obertures-bugies-grans) -> esborrar(objectiu(examinar-bugies)) ^ solucio(ajustar-bugies)
- R11: objectiu(examinar-bugies) ^ not(símptoma(obertures-bugies-grans)) -> esborrar(objectiu(examinar-bugies)) ^ objectiu(examinar-suministrador-gasolina)
- R12: objectiu(examinar-suministrador-gasolina) ^ símptoma(subministrador-combustible-buit) -> esborrar(objectiu(examinar-suministrador-gasolina)) ^ solucio(posar-gasolina)
- R13: objectiu(examinar-suministrador-gasolina) ^ not(símptoma(subministrador-combustible-buit)) -> solucio(eliminar-pèrdues-gasolina)

**Exercici:** Aplicar un encadenament endavant amb la base de regles anterior, amb la següent memòria de treball i fent servir les estratègies de resolució de conflictes amb l'ordre que es donen:

**Base de Regles:**

- R1: s1 -> o1
- R2: s1 -> o2
- R3: s1 -> o3
- R4: o1 ^ not(s2) -> o4
- R5: o1 ^ s2 -> o5
- R6: o4 ^ s3 -> esborrar(o1) ^ I1
- R7: o5 ^ s4 -> esborrar(o5) ^ I2
- R8: o2 ^ s5 -> esborrar(o2) ^ I3
- R9: o2 ^ s6 -> esborrar(o2) ^ I4
- R10: o3 ^ s7 -> esborrar(o3) ^ I5
- R11: o3 ^ not(s7) -> esborrar(o3) ^ o6
- R12: o6 ^ s8 -> esborrar(o6) ^ I6
- R13: o6 ^ not(s8) -> I7

**Memòria de treball:**

- s1
- s2
- s4

**Estratègies de resolució de conflictes:**

- Refractorietat
- Especificitat
- Recència
- Ordre de les regles

**Exemple:** Definir un sistema basat en regles que es comporti com un sistema reactiu, és a dir, que les seves deduccions es tradueixen en accions.

En aquest cas volem fer un robot empaquetador, que a la sortida d'un supermercat ajuda a posar les coses que compra un client en bosses  
(Exemple extret de [Winston 92])

El robot té encomenades **4 tasques:**

Tasca 1. Fase d'ordenació.

Ha de mirar què ha comprat el client i en funció d'això, suggerir-li que compri alguna cosa més.

Tasca 2. Fase d'empaquetar coses grans, ha de tenir en compte dues coses:

- 2.1. Posar primer a les bosses les botelles.
- 2.2. Tenir en compte que a cada bossa hi caben 6 objectes grans.

Tasca 3. Fase d'empaquetar coses mitjanes, ha de tenir en compte dues coses:

- 3.1. Posar els productes congelats en bosses especials.
- 3.2. Tenir en compte que a cada hi caben 12 objectes mitjans.

Tasca 4. Fase d'empaquetar coses petites.

Ha de tenir en compte que a cada bossa hi caben 18 productes petits.

#### Base de regles (I)

- B1 Si (pas=fase-ordenacio) i (Hi ha patates fregides) i (No hi ha refrescs) llavors  
(Demandar al client si no li agradaria una refresc)
- B2 Si (pas=fase-ordenacio) llavors  
esborrar-mt(pas=fase-ordenacio) i afegir-mt(pas=empaquetar-coses-grans)
- 
- B3 Si (pas=empaquetar-coses-grans) i (Hi ha X=objecte-gran) i (X=botella) i  
(num-objectes(bossa-actual)< 6 objectes-grans) llavors  
esborrar-mt(Hi ha X) i afegir-mt(X es a bossa-actual)
- B4 Si (pas=empaquetar-coses-grans) i (Hi ha X=objecte-gran) i  
(num-objectes(bossa-actual)< 6 objectes-grans) llavors  
esborrar-mt(Hi ha X) i afegir-mt(X es a bossa-actual)
- B5 Si (pas=empaquetar-coses-grans) i (Hi ha X=objecte-gran) i (Hi ha Y=bossa-buida) llavors  
esborrar-mt(bossa-actual) i afegir-mt(bossa-actual=Y)
- B6 Si (pas=empaquetar-coses-grans) llavors  
esborrar-mt(pas=empaquetar-coses-grans) i afegir-mt(pas=empaquetar-coses-mitges)
- 

#### Base de regles (II)

- B7 Si (pas=empaquetar-coses-mitges) i (Hi ha X=objecte-mig) i (Hi ha X=objecte-congelat) i  
(X no esta bossa-especial) llavors  
esborrar-mt(X no esta bossa-especial) i afegir-mt(X es a bossa-especial)
- B8 Si (pas=empaquetar-coses-mitges) i (Hi ha X=objecte-mig) i ((bossa-actual=bossa-buida) o  
(bossa-actual no te objectes-grans)) i (num-objectes(bossa-actual)< 12 objectes-mitjans) llavors  
esborrar-mt(Hi ha X) i afegir-mt(X es a bossa-actual)
- B9 Si (pas=empaquetar-coses-mitges) i (Hi ha X=objecte-mig) i (Hi ha Y=bossa-buida) llavors  
esborrar-mt(bossa-actual) afegir-mt(bossa-actual=Y)
- B10 Si (pas=empaquetar-coses-mitges) llavors  
esborrar-mt(pas=empaquetar-coses-mitges) i afegir-mt(pas=empaquetar-coses-petites)
- 
- B11 Si (pas=empaquetar-coses-petites) i (Hi ha X=objecte-petit) i ((bossa-actual no te objectes-grans) o  
(bossa-actual no te objectes-mitjans)) i (num-objectes(bossa-actual)< 18 objectes-petits) llavors  
esborrar-mt(Hi ha X) i afegir-mt(X es a bossa-actual)
- B12 Si (pas=empaquetar-coses-petites) i (Hi ha X=objecte-petit) i (Hi ha Y=bossa-buida) llavors  
esborrar-mt(bossa-actual) afegir-mt(bossa-actual=Y)
- B13 Si (pas=empaquetar-coses-petites) llavors  
esborrar-mt(pas=empaquetar-coses-petites) i afegir-mt(pas=tot-fet)

#### Selecció del tipus d'encadenament de regles

Factors que poden intervenir a l'hora de seleccionar l'encadenament que pot ser millor en un problema concret:

**De quins conjunts de partida disposem.** Anirem en les direccions que els conjunts de partida permetin.

**Tamany de l'estat de partida i de l'estat objectiu.** Interessa moure's en la direcció que vagi d'estats petits cap a grans.

**Direcció de creixement del factor de ramificació.** Interessa moure's en la direcció que el factor de ramificació sigui més petit.

**Justificació del raonament.** Interessa moure's en la mateixa direcció que va l'usuari del sistema, i això depèn del disseny de la base de regles.

#### Implementació de l'encadenament de regles:

- **Aplicació seqüencial i ordenada**, es du a terme fent servir estratègies de resolució de conflictes.
- **Aplicació paral·lela o massiva**, es pot dur a terme fent servir operacions d'àlgebra relacional en bases de dades

**Exercici:** Aplicació de l'encadenament endavant de manera massiva fent servir l'algorisme RETE basat en operacions d'àlgebra relacional.

**Base de Regles:**

R1: Si (X és un cavall)  $\wedge$  (X és pare de Y)  $\wedge$  (Y és ràpid) llavors (X és valuós) ...

**Memòria de Treball:**

c és un cavall      p és ràpid      c és pare de d  
 p és un cavall      t és ràpid      c és pare de p  
 t és un cavall                           d és pare de t  
 d és un cavall

**Pas 1.** Representació de la Memòria de treball amb una taula o relació

Observació: en aquest cas n'hi ha prou amb tres atributs

MT

Primer	Segon	Tercer
c	es-un	cavall
p	es-un	cavall
t	es-un	cavall
d	es-un	cavall
p	es	rapid
t	es	rapid
c	es-pare-de	d
c	es-pare-de	p
d	es-pare-de	t

**Pas 2.** Selecció dels fets que verifiquen cada antecedent de la regla R1

R1: Si (X es-un cavall)  $\wedge$  (X es-pare-de Y)  $\wedge$  (Y es rapid) llavors (X és valuós)

- Antecedent 1: (X es-un cavall)

S1=SELECT MT WHERE Segon=es-un AND Tercer=cavall  
 A1=PROJECT S1 OVER Primer

X
c
p
t
d

- Antecedent 2: (X es-pare-de Y)

S2=SELECT MT WHERE Segon=es-pare-de  
 A2=PROJECT S2 OVER Primer AND Tercer

X	Y
c	d
c	p
d	t

- Antecedent 3: (Y es rapid)

S3=SELECT MT WHERE Segon=es AND Tercer=rapid  
 A3=PROJECT S3 OVER Primer

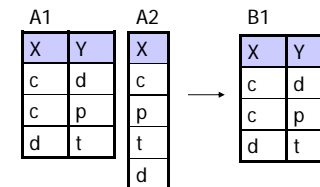
Y
p
t

**Pas 3.** Selecció dels fets que verifiquen tots els antecedents de la regla R1

R1: Si (X es-un cavall)  $\wedge$  (X es-pare-de Y)  $\wedge$  (Y es rapid) llavors (X és valuós)

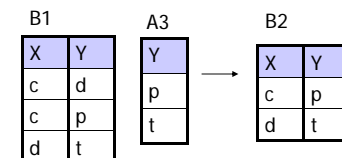
- Antecedent 1 i Antecedent 2

B1=JOIN A1 AND A2 WITH A1.X=A2.X



- Resultat anterior i Antecedent 3

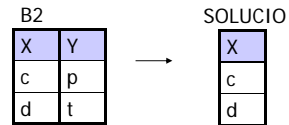
B2=JOIN B1 AND A3 WITH B1.Y=A3.Y



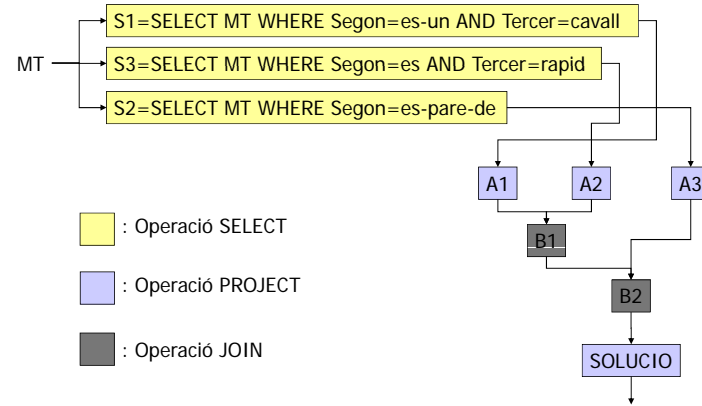
**Pas 4.** Projecció sobre la variable de la solució

R1: Si (X es-un cavall) ^ (X es-pare-de Y) ^ (Y es rapid) llavors (X és valuos)

SOLUCIO= PROJECT B2 OVER X



**RESUM:** L'aplicació d'una regla es pot veure com una xarxa d'operacions d'àlgebra relacional (cas de R1) :



**Resum/Anàlisi dels mecanismes de raonament vists:**

Mecanismes	Algorisme	Algorisme Complet	General (en la representació)
Deducció Natural	Arbre OR-AND	NO	SÍ
Resolució	Resolució Sistemàtica	SÍ	SÍ
Resolució amb estr. PROLOG	Estratègia PROLOG (Backtracking / Encadenament ençera)	SÍ	NO (només clausules de HORN)
Modus Ponens generalitzat	Encadenament de regles	SÍ	NO (només implicacions)

Classe del Javier Vazquez Salceda

Classificació de tipus de sistemes experts:

- Sistemes de diagnòs
- Sistemes de predicció
- ...

segons [Hayes]