12.3.2019

Les listes

app(x, [Y,L]) :- dif(x,Y), appartient(x,L).

une fois qu'il trouve x ds la liste il s'arrête (utilisé lorsque ds le cas où l'élément existe plusieurs fois ds la liste)

où

"!" empêche le moteur d'inférence de retour et chercher autres solutions.

[x|L2]

app(x, [x|_]) :- !.

[x|_]

=> ?- enfiler(4, [1,2,3], L) :- enfiler(4, [2,3], L1)

:- enfiler(4, [1|[2,3]], [1|L1]) :-

enfiler(4, [2,3], L1) :-

enfiler(4, [2|[3]], [2|L2]) L = [1,2,3,4]

enfiler(4, ~~[3]~~ [3], L2)

enfiler(4, [3|[]], [3|L3]) :-

enfiler(4, [], [4]) L3 = 4

vide(L) :- L = [].

min([2,4,1], x) :-

min([2|[4,1]], x) :-

min([4,1], x) :-

min([

[1|2|3|4|5]

[5|4|3|2|1]

1|[2,3,4,5]

2|[3,4,5]

3|[4,5]

4|[5]

min = T[0]

for(...)

{ if(T[0] < min)

{ min = T[0];

}

[3,4,2,1,5]

minimum([3|[4,2,1,5]], x) :- minimum([4,2,1,5], X1), 3 < X1,

tri_min(L, [X|L1]) :- min(L,X), enlever(X,L,L2),

tri_min(L2, L1).

4 - 3 - 2019

mon Cieé = mon instanciè

Ex 1 :

3 -

~~$x^2 - 4x + 1$~~

~~$4^2 - 4 \times 2 = 16 - 8$~~ =

$4x^2 + 2x + 1$

$\Delta = 2^2 - 4 \times \cdot 0$ $\quad x = \frac{-b}{2a} = -\frac{2}{2 \times 4} = -\frac{1}{4}$

$2^2 - 4 \times 4 \times$

$2x^2 -$

~~$8x^2 + 2x + 1$~~

$2x^2 + 4x + 1$

$x^2 + 4x + 1$

$4^2 - 4 \times 1 \times 1$

$\Delta = 16 - 4 = 12$

$16 - 4$

$x_1 = \frac{-4 - \sqrt{12}}{2}$ $\quad x_2 = \frac{4 + \sqrt{12}}{2}$ $\quad \frac{-4}{2a}$

$x^2 + 2x + 1$

$\frac{-2}{4} = -1$

writef("x1 = %d et x2 = %d ", [x1, x2]) ;

? trace .

? - predicat(,)

motrace

modebug

$A * 2 = 6$

C>0 $\quad$ C1 is 6 - 2 = 4

fois2(A1, 2, 4) ⟶ C>0 C1 is 4 - 2 = 2

fois2(A2, 2, 2) ⟶ C>0 C1 is 2 - 2

$A = 1 + 1$

$= 2$

fois(A3, 2, 0)

A2 = 0

5 - 3 - 2017

A < C < B

exemple : max(X,Y,M) :- (X>Y), M=X.
max(X,Y,M) :- M=Y.

autre manière :

max(X,Y,X) :- X>Y.
max(X,Y,Y).

tester : Y=10, X=Y
Y=10, X=Y+1
X=10, X=Y
X=10, X+1, Y
Y=10, X is Y+1

op d'affectation

structure de choix multiple :

P(X) :- condition1(X), action1
-- :- -- 2(X), -- 2

La récursivité

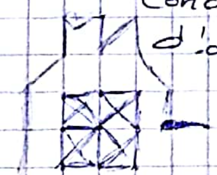
une variable en Prolog : var → instancié → une fois il est instancié on peut pas changer sa valeur.
→ libre

exp : fact(X,N)

fact(0,1).
fact(X,N) :- $\underbrace{X>0}_{\text{condition d'arrêt}}$, $\underbrace{X_1 \text{ is } X-1}$, fact($X_1$,$N_1$), N is $X*N_1$

condition d'arrêt

on peut pas faire X is X-1 car on ne peut pas changer la valeur de X.



Bnj

structure : Répéter ... tant que / jusqu'à

Répéter action tant que condition :
-- -- jusqu'à -- :

saisir_positif(X), dec(X).

date : objet composé

né(X,Y).   objet composé.

né(Ahmed, date(J,M,A)).

Si j'ajoute but 2. après : on trouve True

détrafférence

fail : exige le mécanisme de remonter pour chercher d'autres solutions

.PP particulier
↓
plus général

Prédicat fail

but1 :- p(x), writeln(x), fail.
but2 :- but1, write('fin') .

ces programme affiche
a
b
false

× car but1 → false

pour Afficher 'fin' :

but1 :- write ln(X), fail

but1.

but2 : but1, write('fin')

Prédicat cut = ≠ fail

qd on affecte une valeur à une variable on n'a pas le choix de la changer.

TP :

Exercice

faits + règles ⇒ nouvelles connaissances

; ⇔ ou
, ⇔ et
not( ) ⇔ négation

01-03-2019

write (x), nl ⇔ writeln (x)

writef(' ___ ', [_, _, _])

affiche le_fils(x, y) :- est_le_fils(x, y), writef('\n%t est le fils de %t', [x, y])

Les structures de contrôle

si cette condition est vérifiée réalisé

p(x) :- condition(x), action1

, p(x) :- action2

Sinon ∀ x

Scanned by CamScanner

?- homme(X) ; intelligent(Y)

X = ahmed , Y = ahmed

X = hamza , Y = ahmed

X = jamal , Y = ahmed

propriétaire(X, Y) :- acheter(X, Y) ; recevoir(X, Y)

si

prop(X, Y) :- acheter(X, Y)

⟺ prop(X, Y) :- recevoir(X, Y) } => les règles

Si je tape propriétaire(_, Y)

affiche just les objets

" (samir, _)

true

Mode de Pensées de Prolog

taper " ; " pour chercher autres solution.

pase : Raisonnement de Prolog

?- p(X, Y, Z)

X = a, Y = c, Z = g ; / chercher autres chemins

on commence par Z (le backtracking)

X = a, Y = d , Z = g ;

d ≠ g et d = d

Y , Z = g

e ≠ g

e ≠ d

X = b ; Y = c , Z = g (3)

X = b , Y = d , Z = g (4)

il y a un prédicat pour afficher?

Liste

but :- p(X, Y, Z) , writef("X = %t, Y = %t, Z = %t", [X, Y, Z]).

prouver P pour afficher

ds le cas où on a pas des variable liste = []

but :- p(X, Y, Z) , writef(" ______ \n", [X, Y, Z]), fail

fail: prédicat prédéfinit lorsque il rencontre "false" il affiche

(b) - $(\exists\ x,y,z)$ (Est-enfant-de(x,y,z) $\wedge$ yeux-bleu(x)) $\wedge$
(¬yeux-bleu(y) $\vee$ ¬yeux-bleu(z)))

(c) - $\forall x,y,z$ Est-enfant-de(x,y,z) $\wedge$ yeux-bruns(y) $\wedge$ yeux-bruns(z)
$\longrightarrow$ (yeux-bleu(x) $\vee$ yeux-bleu(x)))

(d) - $(\exists x,y,z)$ Est-enfant-de (x,y,z) $\wedge$ yeux-brun (x) $\longrightarrow$
(yeux-brun (y) $\vee$ yeux-brun (z)))

d - $\forall(x,y,z)$ $E(x,y,z)$ $\wedge$ $YR(x)$ $\longrightarrow$ $(YB(x) \wedge YR(y))$
$\vee(YR(x) \wedge YB(y))$ $\vee$
$(YR(x) \wedge YR(y))$

26/2/2019

P.78

Représentation de la connaissance en Prolog ← logique
programmation

terme = para d'un prédicat.

exp:
pg.pl

?- homme(ahmed).
faits homme(hamza).
homme(jamal).
femme(zaila).
femme(khadija).
intelligent(Ahmed).

Execution

?- homme (jamal)

true

?- homme (mourad)

false

?- homme(x)
x = ahmed; // click sur entrer
x = hamza; il va passer sur
x = jamal; tous les termes
qui vérifie homme(
false

?- homme (x), intelligent (x)

x = ahmed;

false

?- homme (Ahmed), intelligent(Ahmed

True

TD : Ex1 : a- $(\forall x)$ homme(x) $\longrightarrow$ primate(x) $= (\forall x)\ \neg H(x) \vee P(x)$

b- $(\forall x)$ Dauphins$(x) \longrightarrow \neg$ Primates$(x) = (\forall x)\ \neg D(x) \vee \neg P(x)$

c- $(\exists x)$ Dauphin$(x) \wedge$ intelligent$(x) = (\exists x)\ D(x) \wedge I(x)$

d- $(\exists x)$ $\neg$ Homme$(x) \wedge$ intelligent$(x) = (\exists x)\ \neg H(x) \wedge I(x)$

$\{ \neg H(x) \vee P(x),\ \neg D(x) \vee \neg P(x),\ D(x),\ I(x),\ H(x) \vee \neg I(x) \}$

x/a

$\{ \neg H(a) \vee \neg D(a),\ D(x),\ I(x),\ H(x) \vee \neg I(x) \}$

x/a

$\{ \neg H(a) \vee \cancel{D(a)},\ H(a) \} = \square$

Ex 4 :

a) ~~$\forall x$ Herbivore$(x) \wedge$ Mange$(x,y) \wedge$~~

$(\forall x)$ Herbivore$(x) \wedge$ Vigital$(y) \rightarrow$ Mange (x,y)

b) $(\forall x)(\exists y)$ Herbivore $(x) \wedge$ Vigital $(y) \longrightarrow$ Mange(x,y)

c) $(\forall y)(\exists x)$ Vigital $(x) \wedge$ Herbivore$(y) \rightarrow \neg$ Mange (x,y)

d) $(\forall x)$ Panda$(x) \longrightarrow$ Herbivores$(x) \wedge$ Bambo$(y) \longrightarrow$

$\exists$ Mange(x,y)

$\longrightarrow$ a- $(\forall x)\ (H(x) \wedge (\forall y)\ M(x,y) \longrightarrow V(y)))$

b- $(\forall x)\ H(x) \longrightarrow \exists y\, V(y) \wedge \neg$ Mange (x,y)

$(\neg \exists x)\ H(x) \wedge \forall y\, V(y) \longrightarrow M(x,y)$

L.M $(\forall x)\ \neg (H(x) \wedge ((\forall y)\ \neg V(y) \vee M(x,y)))$

$(\forall x)\ \neg H(x) \vee ((\exists y)\ V(y) \wedge \neg M(x,y))$

$(\forall x)\ H(x) \longrightarrow ((\exists y)\ V(y) \wedge \neg M(x,y)))$

c- $(\exists y)\ V(y) \wedge ((\forall x) H(x) \longrightarrow \neg M(x,y))$

d- $(\forall x)\ (P(x) \longrightarrow (H(x) \wedge (\forall y)\ (M(x,y) \longrightarrow B(y))))$

Ex 7

(a) $(\forall x,y,z)$ Est-enfant-de $(x,y,z) \wedge$ yeux-bleus$(y) \wedge$ yeux-bleus(z)

$\longrightarrow$ yeux-bleu (x)

$(\forall x)\ (\neg L(x) \vee N(x)) \wedge (\neg D(x) \vee \neg N(x)) \wedge (D(a) \wedge I(a))$

$(\neg L(x) \vee \neg D(x)) \wedge (D(a) \wedge I(a))$

x/a

$\neg L(x) \vee \neg D(x),\ D(a) \vee 0,\ I(a)$

$\neg L(a),\ I(a)$ $\quad \boxed{x/a}$

sol :

$\varphi_1 :\ (\forall x)\ (L(x) \rightarrow N(x)) \quad = (\forall x)\ \neg L(x) \vee N(x)$

$\varphi_2 :\ (\forall x)\ (D(x) \rightarrow \neg N(x)) \quad = \neg D(x) \wedge \neg N(x)$

$\varphi_3 :\ (\exists x)\ (D(x) \wedge I(x)) \quad a/x :\ D(a) \wedge I(a)$

$\psi :\ (\exists x)\ (I(x) \wedge \neg L(x)),\ \neg\psi \quad \cancel{D(x)} \wedge \neg I(x) \vee L(x)$

$\varphi_1 = \{\neg L(x) \vee N(x)\}$

$\varphi_2 : \{\neg D(x) \vee \neg N(x)\}$

$\varphi_3 : \{D(a),\ I(a)\}$

$\neg\psi : \{\neg I(x) \vee L(x)\}$

$\{\varphi_1, \varphi_2, \varphi_3, \neg\psi\} \vdash \square$

$\{\neg L(x) \vee N(x),\ \neg D(x) \vee N(x),\ D(a),\ I(a),\ \neg I(x) \vee L(x)\}$

$a/x \quad \neg D(a) \vee N(a),\ D(a)$

$\{\neg L(x) \vee N(x),\ \neg N(a),\ I(a),\ \neg I(x) \vee L(x)\}$

$a/x \quad I(a),\ \neg I(a) \vee L(a)$

$\neg L(x) \vee N(x),\ \neg N(a),\ L(a)$

a/x

$\neg L(a) \vee N(a),\ \neg N(a)$

$\{\neg L(a,\ L(a)\}$

$\square$

$\psi = \varphi_i' \vee \varphi_j'$

$C = \{\varphi_1, \varphi_2, \ldots \varphi_{i-1}, \varphi_{i+1}, \ldots \ldots \varphi_{j-1}, \varphi_{j+1}, \ldots \varphi_n, \psi\}$

Ce principe est applicable juste pour LP.

Exemple :

$\{P \vee Q, P \rightarrow Q, Q \rightarrow P\} \vdash P \wedge Q$

$\{P \vee Q, \neg P \vee Q, \neg Q \vee P\}$,

$\{Q \vee P, \neg Q \vee P\} = P$

$\{P \vee Q, \boxed{\ }\}$

$\rightarrow \{P \vee Q, \neg P \vee Q, \neg Q \vee P\} \vdash \{P \wedge Q\}$

$\{P \vee Q, \neg P \vee Q, \neg Q \vee P, \neg P \vee \neg Q\}$

8

$Q \vee \square$ $\qquad\qquad$ $\neg Q \vee \square$

$\square$

19-2-2013

Principe de résolution en LPO

F et G unifiable $\Leftrightarrow$ si $\exists\ \sigma$ tq $F\sigma = G\sigma$

exp : (64)

$F = P(x, b, z) \vee Q(f(x))$ et $G = P(a, y, g(a)) \vee Q(t)$

$\Rightarrow \quad \sigma = \{x / a, y / b, z / g(x), t / f(x)\}$

P : 67

D (x) : x est un dauphin

L (x) : x sait lire

N (x) : x est instruit

I (x) : x est Intelligent

Exemples $(\forall x)\ L(x) \longrightarrow N(x)$ $\qquad$ $(\forall x)\ \neg L(x) \vee N(x)$

$(\forall x)\ D(x) \longrightarrow \neg N(x)$ $\qquad$ $(\forall x)\ \neg D(x) \vee \neg N(x)$

$(\exists x)\ D(x) \cap I(x)$ $\qquad$ $D(a) \cap I(a)$

② : $F = (\forall x)(\exists y)(\exists z)(\forall t)(P(t) \vee R(y)) \wedge Q(z) \wedge H(t)$

$y = f(x)$ $F = (\forall x)(\exists z)(\forall t)(P(t) \vee R(f(x)) \wedge Q(z) \wedge H(t))$

$z = g(x)$ $F = (\forall x)(\forall t)(P(x) \vee R(f(x)) \wedge Q(g(x)) \wedge H(t))$

③ $F = (\exists x)(\forall y)(\forall z)(\exists t)(P(x) \vee R(y)) \wedge Q(z) \wedge H(t)$

$F = (\forall y)(\forall z)(\exists t)(P(a) \vee R(y)) \wedge Q(z) \wedge H(t)$

$t = f(y,z) \Rightarrow F = (\forall y)(\forall z)(P(a) \vee R(y)) \wedge Q(z) \wedge H(f(y,z))$

④ $F = (\forall x)(\exists y)(\forall z)(\exists t)(P(x) \vee R(y)) \wedge (Q(z) \wedge H(t))$

$F = (\forall x)(\forall z)(P(x) \vee R(f(x)) \wedge Q(z) \wedge H(g(x,z))$

Forme clausales

effet : représente les cas particulier ($\exists$)

clause : fro disjonction des littéraux

form clausal : conjonction de clauses

exemple : $F = ((\exists x)\, P(x)) \rightarrow \neg(((\exists y)\, Q(y)) \rightarrow R)$

$F = \neg((\exists x)(P(x)) \vee \neg(((\exists y)\, Q(y)) \rightarrow R)$

$= \forall x \neg P(x) \vee \neg(\neg((\exists y)\, Q(y)) \vee R)$

$= (\forall x) \neg P(x) \vee \neg(\forall y \neg Q(y) \vee R)$

$= (\forall x) \neg P(x) \vee (((\exists y)\, Q(y)) \wedge \neg R$

$= (\forall x)(\exists y)\ \neg P(x) \vee (Q(y) \wedge \neg R)$

$= (\forall x)\ \underbrace{\neg P(x) \vee (Q(f(x))}\ \underbrace{\wedge \neg R)} = \{\neg P(x) \vee Q(f(x)), \neg R\}$

$F = (\forall x)\ \underbrace{\neg(P(x)) \vee Q(f(x))} \wedge (\underline{\neg P(x) \vee \neg R})$

$F = \{\neg P(x) \vee Q(f(x)), \neg P(x) \vee \neg R\}$

p : ss $\exists$ et $\vee$ $\Rightarrow A \vee B \Leftrightarrow B \vee A$

Principe de Résolution en LP

$C = \{Q_1, Q_2, \ldots\ Q_i, \ldots\ Q_j, \ldots Q_n\}$

$Q_i = L \vee Q_i'$ $\quad Q_j = \neg L \vee Q'_j$

P.R

TD1: Exercice 1:

1 - $\forall x$ Homme$(x) \longrightarrow$ primate(x)

2 - $\forall x$ Dauphines$(x) \longrightarrow \neg$ primates(x)

3 - $\exists x$ dauphins$(x) \wedge$ intelligentes(x)

4 - $\forall$. Homme(x)

$\exists x$ ~~$\neg H \forall x$~~ $\neg$Homme$(x) \vee$ primate(x)

$\exists x$ $\neg$Homme$(x) \wedge$ intelligente(x)

$(1) \wedge (2) \wedge (3) \longrightarrow (4)$

Réponse:

1 - $(\forall x)$ $H(x) \to P(x)$

2 - $(\forall x)$ $P(x) \to \neg(P(x)) \equiv (\neg\exists x)\ D(x) \wedge (P(x))$

Exercice 2.

① 1 et 4 et 7.

② 3, 6

③ 4, 6

14-2-2018

Forme standard de Skolem

$F = (Q_1 x_1)(Q_2 x_2) \cdots (Q_{i-1} x_{i-1})(Q_i x_i)(Q_{i+1} x_{i+1}) \cdots (Q_n x_n)$

$M(x_1, \cdots x_n)$

Si $Q_1 = \exists$

$F = (Q_2 x_2) \cdots (Q_n x_n)\ M(a, \cdots x_n)$ a / x_i

Si $Q_i = \forall$

Éliment Q_i

remplacer $x_i = f(x_1 \cdots x_{i-1})$

$F = (Q_1 x_1)(Q_2 x_2) \cdots (Q_n x_n)\ F(x_1, x_2, \cdots f(x_i \cdots$

$x_{i-1} \cdots x_n)$

Exemples

$F: (\forall x)(\forall y)(\exists z)(\forall t)\ (P(x) \vee R(y)) \wedge Q(z) \wedge H(t)$

$z = f(x, y) \longrightarrow F = (\forall x)(\forall y)(\forall t)\ (P(x) \vee R(y)) \wedge$

$f(x, y)) \wedge H(t)$

faits déduits $+R+U$
$+V+\varphi$

$R_2 : S \wedge \overline{T} \to U$

$R_3 : R \wedge U \to V$

$R_4 : \varphi \wedge P \to \varphi$

Logique des prédicats du premier ordre

P.42. "travaille" est une fct
rigoureux : un prédicat.

Les paramètres d'un prédicat soit : des variables, constantes, fonctions.

un prédicat avec un seul paramètre : le prédicat représente un propriété

plus d'un param représente une relation.

12-2-2019

un terme : paramètre d'un prédicat.

p.45

$((Q_1 x) M(x)) \wedge ((Q_2 Y) N(Y)))$ erreur.

forme de Prenexe les quantificateurs + Matrice

" " Skolen généralisé les cas (élimination de $\exists$)

Exemple (48)

~~$\forall n_1 \; \forall n_2 \; \exists \; D(n) \; : \; P(x_1) \wedge P(n_2) \wedge \neg E(X, Y))$~~

$(\forall n_1)(\forall n_2)_1 (P(n_1) \wedge P(n_2) \wedge \cancel{P(n_1) \wedge P(n_2)} \; \neg E(n_1, n_2) \to$
$(\exists 3) (_2 D(3) \wedge A(n_1, 3) \wedge A(n_2, 3) \wedge \forall r (_3 D(r) \wedge$
$A(n_1, r) \wedge A(n_2, r) \to E(3, r))_3)_2)_1$

Ex 2 (71) :

« Tous les A sont B » $\to (\forall n) A(n) \to B(n)$ $A \subset B$

« seuls les A sont B » $\to$ ~~$\forall n \; A(n)$~~ $\vee$ $B \subset A$ $\forall n \; B(n) \to A(n)$

« Aucun A n'est B » $\to (\forall n) (A(n) \to \neg B(n)) = (\neg \exists n)(A(n) \wedge B(n))$

« Quelques A sont B » $\to (\exists x) A(n) \wedge B(n)$

Loi de Morgan

$\exists n \; (Lion(n) \wedge \neg boire(n, café))$

$Q \to R$	$P \wedge (Q \to R)$	$(P \wedge (Q \to R)) \to S$
0	0	1
0	0	1
1	1	0

Exercice : 3-e

Mettre sous f.n.c l'expression : $(P \vee \neg Q) \longleftrightarrow (Q \to R)$

$P \leftrightarrow Q \quad P \to Q \wedge Q \to P$

$P \to Q \equiv \neg P \vee Q$

$(P \vee \neg Q) \to (Q \to R) \wedge (Q \to R) \to P \vee \neg Q$

$\neg(P \vee \neg Q) \vee (Q \to R) \wedge \neg(Q \to R) \vee P \vee \neg Q$

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

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

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

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

$(\neg Q \vee Q) \wedge (\neg R \vee \neg Q)$

Sol

$F \equiv (P \vee \neg Q) \leftrightarrow (Q \to R)$

$\equiv (P \vee \neg Q) \leftrightarrow (\neg Q \vee R) = ((P \wedge \neg Q) \to (\neg Q \vee R)) \wedge$

$(\neg Q \vee R) \to (P \vee \neg Q))$

$\equiv (\neg(P \vee \neg Q) \vee (\neg Q \vee R)) \wedge (\neg(\neg Q \vee R) \vee (P \vee \neg Q))$

$\equiv ((\neg P \wedge Q) \vee (\neg Q \vee R) \wedge ((Q \wedge \neg R) \vee (P \wedge \vee Q))$

$\equiv ((\neg P \vee \neg Q \vee R) \wedge \underbrace{(Q \vee \neg Q \vee R)}_{\top}) \wedge (\underbrace{(Q \vee P \vee \neg Q)}_{\top} \wedge$

$(\neg R \vee P \vee \neg Q))$

f.n.c : $(\neg P \vee \neg Q \vee R) \wedge (\neg R \vee P \vee \neg Q)$

Règles d'Inférences

e.b.f : expressions bien formées

Exemple (3-f)

Base de connaissances = Base de faits + Base de Règles

faits permanents $\{P, S, T, W\}$ R1 : $P \to R$

5 - 2 - 2019

Intelligence Artificielle

forte — faible

cherche comment l'être humaine pense

Représentation de la connaissance
&
systèmes experts

Lisp (List Processing)

↓

Prolog (programmation logique) (Logiciel : swi-Prolog)

Partie "cablée" : je peux pas la changer

Représentation de la connaissance

Logique des Propositions

Logique formelle :

- Logique de Propositions (LP)
- .. des Prédicates de 1er ordre (LPO)
- .. Combinatoire
- .. floue
- .. Temporelle (utilisé pour les système temps réel)

$P \to Q \equiv \neg P \vee Q$

$P \leftrightarrow Q \equiv (P \to Q) \wedge (Q \to P)$

Exo 1 (31)

$E = (P \wedge (Q \to R)) \to S$

$P \wedge (\neg Q \vee R) \to S \to \neg(P \wedge (\neg Q \vee R)) \vee S$

$E_2 = (\neg P \vee (Q \wedge \neg R)) \vee S \equiv \neg P \vee (Q \wedge \neg R) \vee S$ f.n.d

$= ((\neg P \vee Q) \wedge (\neg P \vee \neg R)) \vee S = (\neg P \vee Q \vee S) \wedge (\neg P \vee \neg R \vee S)$ f.n.c

P	Q	R	S	¬P	¬R	¬P∨Q∨S	¬P∨¬R∨S	f.n.c
0	1	0	1	1	1	1	1	1
1	1	0	1	0	1	1	1	1

si tous les f.n.c = 1 ⇒ tautologie