Exercice 1.

Dites si la formule suivante est valide, contingente(ni valide ni insatisfiable) ou insatisfiable. Justifiez votre réponse.

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

Exercice 2

correction

A_1 et A_2 sont unifiables. Un unificateur le plus général de A_1 et A_2 est $\sigma = \sigma_5 \circ \sigma_4 \circ \sigma_3 \circ \sigma_2 \circ \sigma_1$ *soit* $\sigma = \{(x, g(h(b,v), h(b,v))); (y, h(b,v)); (z, h(b,v)); (u,v); (w,a)\}$

Correction 1. 1. $(p \vee (q \rightarrow r)) \leftrightarrow (\neg(p \wedge q) \vee \neg r)$

p	q	r	$(p \vee (q \rightarrow r))$	$(\neg(p \wedge q) \vee \neg r)$	$(p \vee (q \rightarrow r)) \leftrightarrow (\neg(p \wedge q) \vee \neg r)$
V	V	V	V	F	F
V	V	F	V	V	V
V	F	V	V	V	V
V	F	F	V	V	V
F	V	V	V	V	V
F	V	F	F	V	F
F	F	V	V	V	V
F	F	F	V	V	V

La formule est contingente.

Exercice 3

Soit l'énoncé suivant :

1. Les personnes qui ont la grippe A/H1N1 doivent prendre du Tamiflu.
2. Les personnes qui ont de la fièvre et qui toussent ont la grippe A/H1N1.
3. Ceux qui ont une température supérieure à 38° ont de la fièvre.
4. Roselyne tousse et a une température supérieure à 38°.
5. Roselyne doit prendre du Tamiflu.

a) Modélisez en logique du premier ordre l'énoncé ci-dessus en utilisant les prédicats :
-grippe(x) : x a la grippe A /H1N1.
-prendre(x,y) : x doit prendre y.
-fièvre(x) : x a de la fièvre.
-tousse(x) : x tousse.
-temp(x,t) : x a la température t.
-sup(x,y) : x est supérieur à y.

b) Prouvez à l'aide de la méthode de résolution que la dernière affirmation est une **conséquence logique de l'ensemble des autres affirmations.**

Correction 3. Modélisation Formule associée à chaque affirmation de l'énoncé $Tamiflu$, $Roselyne$ et 38 sont des constantes :

(a) $\forall x(grippe(x) \rightarrow prendre(x, Tamiflu))$

(b) $\forall x((fievre(x) \wedge tousse(x)) \rightarrow grippe(x))$

(c) $\forall x \forall t((temp(x,t) \wedge sup(t,38)) \rightarrow fievre(x))$

(d) $tousse(Roselyne) \wedge \exists t(temp(Roselyne,t) \wedge sup(t,38))$

(e) $prendre(Roselyne, Tamiflu)$

Résolution On va montrer que a ∧ b ∧ c ∧ d ∧ ¬e est insatisfaisable. Pour cela il faut mettre les formules sous forme clausale. Première étape mise sous forme prenexe :

(a) $\forall x(\neg grippe(x) \vee prendre(x, Tamiflu))$

(b) $\forall x(\neg fievre(x) \vee \neg tousse(x) \vee grippe(x))$

(c) $\forall x \forall t(\neg temp(x,t) \vee \neg sup(t,38) \vee fievre(x))$

(d) $\exists t(tousse(Roselyne) \wedge temp(Roselyne,t) \wedge sup(t,38))$

(¬e) $\neg prendre(Roselyne, Tamiflu)$

Seule d contient une variable existentielle t, on introduit la constante de Skolem a. On obtient alors la formule :

(d′) $tousse(Roselyne) \wedge temp(Roselyne,a) \wedge sup(a,38)$

Passage à la forme clausale :

$$\begin{aligned}
C_1 &= \{\neg grippe(x); prendre(x, Tamiflu)\}\\
C_2 &= \{\neg fievre(x); \neg tousse(x); grippe(x)\}\\
C_3 &= \{\neg temp(x,t); \neg sup(t,38); fievre(x)\}\\
C_4 &= \{tousse(Roselyne)\}\\
C_5 &= \{temp(Roselyne,a)\}\\
C_6 &= \{sup(a,38)\}\\
C_7 &= \{\neg prendre(Roselyne, Tamiflu)\}
\end{aligned}$$

Résolution :

$$\begin{aligned}
C_8 &= Res(C_7, C_1, \{(x, Roselyne)\}) &&= \{\neg grippe(Roselyne)\}\\
C_9 &= Res(C_8, C_2, \{(x, Roselyne)\}) &&= \{\neg fievre(Roselyne); \neg tousse(Roselyne)\}\\
C_{10} &= Res(C_9, C_3, \{(x, Roselyne)\}) &&= \{\neg tousse(Roselyne); \neg temp(Roselyne,t); \neg sup(t,38)\}\\
C_{11} &= Res(C_{10}, C_4, \{\}) &&= \{\neg temp(Roselyne,t); \neg sup(t,38)\}\\
C_{12} &= Res(C_{11}, C_5, \{(t,a)\}) &&= \{\neg sup(a,38)\}\\
C_{13} &= Res(C_{12}, C_6, \{\}) &&= \{\}
\end{aligned}$$

Exercice 4.

Forme clausale	Logique Classique	Prolog
1) ¬ t(x) ˅ c(x)	**t(x) ⇒ c(x)**	**c(x) :- t(x).**
2) ¬ z(x) ˅ a(x)	**z(x) ⇒a(x)**	**a(x) : z(x).**
3) ¬ c(x) ˅ ¬ a(y) ˅ m(x,y)	**c(x) ^ a(y) ⇒m(x,y)**	**m(x,y) :- c(x), a(y).**
¬Conc) :		
4) t(a)	**t(a)**	**t(a).**
5) z(b)	**z(b)**	**z(b).**
6) ¬m(a,b).	**m(a,b) (V ou F ?)**	**:- m(a,b).**

Exercice 5.

SOL

/********* ex.pl *********/

a_1(X,Y):-!,b(X),c(Y).

a_2(X,Y):- b(X),!,c(Y).

a_3(X,Y):-b(X),c(Y),!.
b(0).
b(1).
c(0).
c(1).
/***********************/
Donner toutes les réponses possibles de :

1. ?- a_1(X,Y). *La coupure ne concerne que les choix du prédicat : a_1 mais pas les autres (b et c)*
   X = Y, Y = 0 ;
   X = 0,
   Y = 1 ;
   X = 1,
   Y = 0 ;
   X = Y, Y = 1.
2. ?- a_2(X,Y). *La coupure ne concerne que les choix des prédicats : a_1 et b mais pas c*
   X = Y, Y = 0 ;
   X = 0,
   Y = 1
3. ?- a_3(X,Y). *La coupure concerne tous choix des prédicats : a_1, b et c*
   X = Y, Y = 0.

liste(L) :- is_list(L).
gharibe([X|Y],Z) :- gharibe(Y,[X|Z]).
gharibe(Z,Z) :-liste(Z).

?- gharibe([a,b,c],Z).
Z = [a, b, c]

Exercices 6:

1) Comment exprimer qu'on veut connaître le plus grand parmi 2 nombres ? parmi 3 ?

max2(X,Y,X) :- X>=Y.

max2(X,Y,Y) :- X<Y.

max3(X,Y,Z,M) :- max2(X,Y,MM),max2(MM,Z,M).

Que produit **max2(X,1,3)** ? Pourquoi ? Même question avec **max2(1,X,3)**.

Récursivité (sur nombres):

1) Afficher **N** fois **'bonjour'**. (Ou : **ecrit(N)** est vrai si le message **'bonjour'** est écrit **N** fois.)

ecrit(0).

ecrit(N) :- N>0, write('bonjour'), nl, N1 is N-1, ecrit(N1).

2) Dire si un nombre est pair.

pair(0).

pair(X) :- X>0, X2 is X-2, pair(X2).

3) Trouver la factorielle d'un nombre. (Ou : **fact(N,X)** est vrai si **X** vaut **N!**.)

fact(0,1).

fact(N,X) :- N>0, N1 is N-1, fact(N1,X1), X is N*X1.

Exercice 7.

Soit le programme Prolog suivant définissant les prédicats pred1 et pred2.

pred1(X, [X♣L]) :- !.

pred1(X, [Y ♣L]) :- pred1(X; L).

pred2([], L,L).

pred2([X♣L_1], L_2, L_3) :-pred1(X, L_2), !, pred2(L_1, L_2, L_3).

pred2([X♣L_1], L_2, [X♣L_3]) :- pred2(L_1, L_2, L_3).

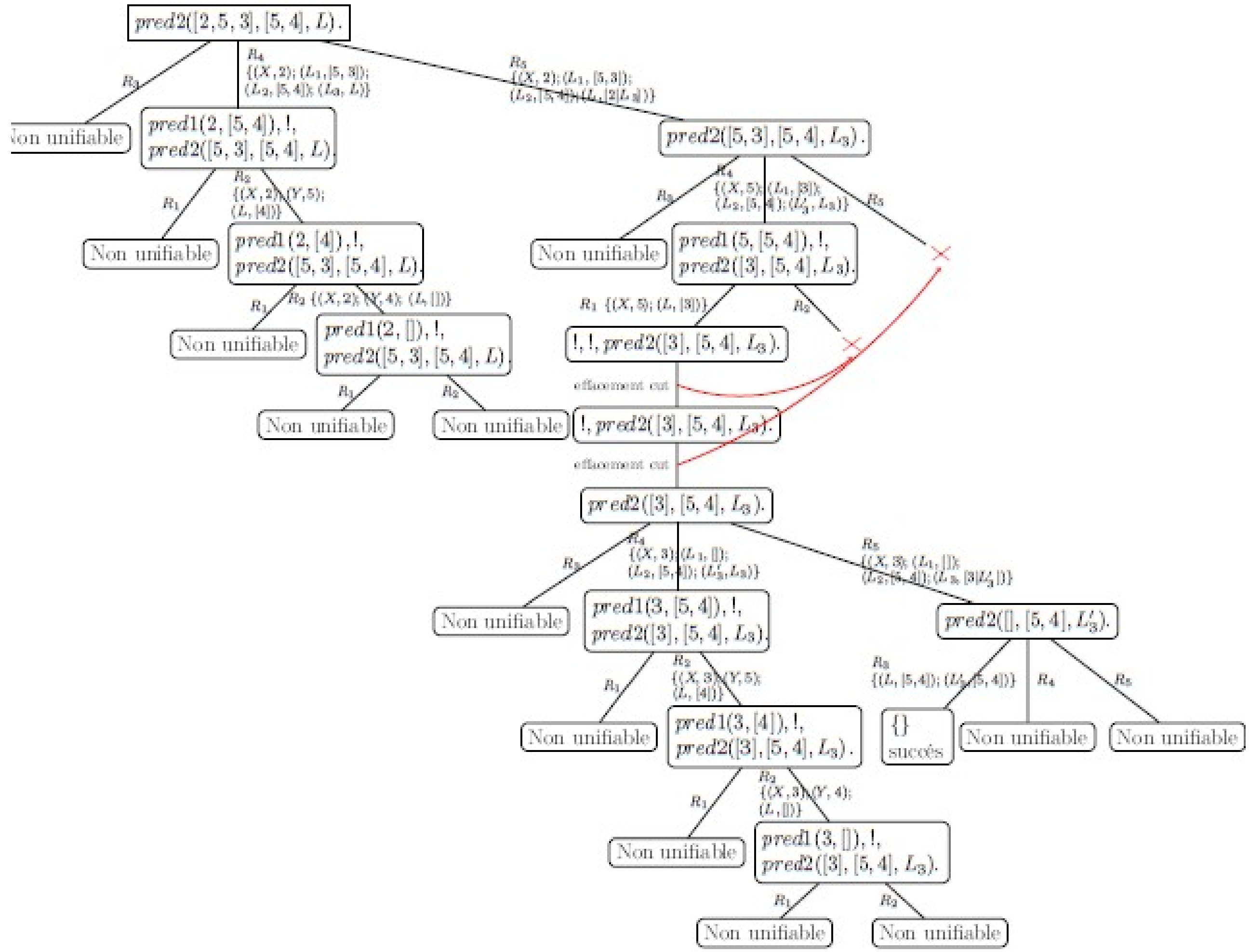
a) Que donne : pred2([2, 5,3], [5, 4], L).

b) Expliquez en quelques mots ce que font pred1 et pred2.

Correction 4.

a) Voir ci dessous.

b) $pred1(X, L)$ est vrai si X apprartient à L, et $pred2(L_1, L_2, L_3)$ est vrai si L_3 est la liste obtenue en faisant l'union des listes L_1 et L_2.


pred2([2,5,3],[5,4],L).
Non unifiable
pred1(2,[5,4]),!,
pred2([5,3],[5,4],L).
pred1(2,[4]),!,
pred2([5,3],[5,4],L).
pred1(2,[]),!,
pred2([5,3],[5,4],L).
pred2([5,3],[5,4],L3).
pred1(5,[5,4]),!,
pred2([3],[5,4],L3).
!,!,pred2([3],[5,4],L3).
effacement cut
!,pred2([3],[5,4],L3).
effacement cut
pred2([3],[5,4],L3).
pred1(3,[5,4]),!,
pred2([3],[5,4],L3).
pred1(3,[4]),!,
pred2([3],[5,4],L3).
pred1(3,[]),!,
pred2([3],[5,4],L3).
pred2([],[5,4],L'3).
{}
succès