

|2209



By
Daniel
Dalati

2023



Exercice 1 (25 points)

Soit le programme suivant :

```
p(A) :- q(A,A).  
p(C) :- q(C,D), C < D.  
q(A,B) :- between(1,3,A), between(1,3,B).
```

a) Quelles sont toutes les solutions retournées par la requête $p(X)$?

b) Même question en ajoutant une coupure (!) entre les deux prédictats `between`.

Exercice 2 (25 points)

On considère le problème suivant : « L'an dernier, mon père avait le double de mon âge (exemple 72 et 36). Cette année, nos deux âges s'expriment par les deux mêmes chiffres compris entre 1 et 9 inclus, mais écrits dans un ordre différent, c'est-à-dire le chiffre des dizaines dans l'âge du père devient le chiffre des unités dans mon âge et inversement (exemple 73 et 37). Aucun de nous deux n'a atteint les 100 ans (le maximum des âges est 99 ans). Quel est l'âge de mon père et quel est mon âge ? »

Ecrire un petit programme Prolog qui résout ce problème. On définira le prédictat `ages` qui détermine les âges du père et de l'enfant.

Aide : on a besoin d'utiliser `between(X,Y,Z)`.

L'âge X sera exprimé de la façon suivante $X \text{ is } 10^A + B$. Donc `between` sera utilisé pour A et B.

Il faut faire attention que l'âge s'ajoute de 1 entre l'an dernier et cette année. Exemple si l'an dernier j'avais 36 ans cette année j'ai 37 ans.

Exercice 3 (50 points)

Ecrire un programme qui calcule le complément-à-un (C1) d'un nombre décimal.

Pour faire, on définit les prédictats suivants :

- Convert qui convertit un nombre décimal en binaire. Le nombre binaire est stocké dans une liste L. Exemple la conversion du nombre décimal 6 donne [0,1,1]. Nous remarquons que les chiffres binaires de la liste sont stockés dans un sens inverse. Pour cela il faut inverser la liste en utilisant le prédictat prolog `reverse(L,L1)`. Exemple en utilisant le prédictat `reverse` on obtient [1,1,0].
- Le nombre binaire doit être codé en huit bits. Pour cela, écrire le prédictat complete qui complète la liste par des zéros si le nombre des éléments est plus petit que 8. Pour vous aider, vous pouvez utiliser le prédictat prolog `length(L, Long)`. Exemple, la liste [1,1,0] contient trois éléments donc elle sera complétée pour avoir une nouvelle liste [0,0,0,0,1,1,0].

final(2016-2017) :

%Exercice 1:

p(A):-q(A,A).

p(C):-q(C,D),C<D.

q(A,B):-between(1,3,A),!,between(1,3,B).

%Exercice 2:

age(X,Y):-between(1,9,A),between(1,9,B),X is A*10+B,Y is B*10+A,X1 is X-1,Y1 is Y-1,X1=:=2*Y1,X<99,Y<99.

%Exercice 3:

convert(N,L):-binaire(N,L1),reverse(L1,L).

binaire(0,[]).

binaire(N,[R|L1]):-R is N mod 2,N1 is N//2,binaire(N1,L1).

complete(L,L1):-length(L,S),S<8,! ,remplir(L,S,L1).

complete(L,L).

remplir(L,8,L).

remplir(L,S,[0|L1]):-S1 is S+1,remplir(L,S1,L1).

negatif(N,[_|L1],[1|L1]):-N<0,! .

negatif(_,L1,L1).

inverse(N,L,L):-N>0,! .

inverse(_,[],[]).

inverse(N,[0|L],[1|L1]):-inverse(N,L,L1).

inverse(N,[1|L],[0|L1]):-inverse(N,L,L1).

c1(N,X):-convert(N,L),complete(L,L1),negatif(N,L1,L2),inverse(N,L2,X).



Partie P (Partiel) : (sur 100 points pour ~ 45 minutes).

Exercice 1 (35 points)

Soit le programme suivant :

a(1). a(2). b(3). b(4). c(5). c(6).

t(R):- a(X), b(Y), c(Z), R is X*Y*Z.

- Quelles sont toutes les solutions retournées par la requête $t(R)?$
- Si on ajoute un cut (!) entre $b(Y)$ et $c(Z)$ dans la règle $t(R)$. Quelles sont toutes les solutions retournées par la requête $t(R)?$

Exercice 2 (35 points)

Traduire les énoncés suivants dans la logique des prédictats. On utilisera pour cela les prédictats `tbi/1`, `vidéoproj/1`, `amphi/1`, `salle_td/1`, `est_dans/2`.

- On trouve toujours un TBI dans une salle de TD.
- Il n'y a pas de vidéoprojecteur en salle « B301 ».
- Tous les vidéoprojecteurs sont dans des amphis.
- S'il y a un vidéoprojecteur quelque part, c'est qu'il s'agit d'un amphi.
- Il n'y a que des TBI dans les salles de TD.

Exercice 3 (30 points)

Soient les trois formules :

$f_1 : \exists x \forall y (\neg(S(x) \rightarrow S(y)) \wedge \forall x S(x))$

$f_2 : \forall x (S(x) \rightarrow \exists y R(y))$

$f_3 : R(a)$

Démontrer, à l'aide de la méthode de résolution, la validité de la formule suivante :

$f_1 \wedge f_2 \rightarrow f_3$ (avec x et y sont deux variables et a une constante)

Partie F (Final) : (sur 100 points pour ~ 105 minutes)

Exercice 1 (20 points)

On considère le problème suivant : « il ya 4 hommes : A, B, C, D. B est deux fois plus âgé que D. L'écart d'âge entre A et B est identique à celui de C et D. L'écart d'âge entre A et C est la moitié de celui de B et D. Aucun des hommes n'a strictement plus de quarante ans ni strictement moins de 20 ans. A est plus âgé que C. »

Ecrire un programme qui calcule les âges de toutes les personnes.

- c) Le nombre décimal peut être négatif. Si c'est le cas il faut écrire un prédictat negatif qui remplace le premier élément de la liste par 1. Exemple si 6 est négatif, on aura une nouvelle liste [1,0,0,0,0,1,1,0].
- d) Pour calculer le complément-à-1 il faut inverser les chiffres binaires de la liste. 1 devient 0 et inversement. Donc écrire un prédictat inverse qui calcule le C1. Ainsi la liste finale sera [0,1,1,1,1,0,0,1]. Mais il faut faire attention que le complément-à-1 s'applique uniquement aux nombres négatifs. Pour un nombre positif la liste ne change pas.
- e) Ecrire un prédictat complement qui regroupe l'ensemble des prédictats déjà définis.

Bonne chance

Partiel2018-2019

%Exercice1

a(1).

a(4).

b(3).

b(1).

c(5).

c(2).

t(R):-a(X),b(Y),!,c(Z),X>Y,R is X+Y+Z.

chanceux(N):-N>0,R1 is N//10, D is N mod 10 ,C is R1 mod 10,A is N//1000,R2 is R1//10,B is R2 mod 10, Y is A+B,Z is C+D, chanceux(N,Y,Z).

chanceux(N,Y,Z):-Y=:=Z,! ,write(N),write(""),write("est chanceux").

chanceux(N,_,_):-write(N),write(""),write("non chanceux").

%Exercice2

```
chanceu(N):-R is N mod 100,N1 is N//100,somme(R,R1),somme(N1,N2),R1=:=N2,! ,write(N),write(" est chanceu").
```

```
chanceu(N):-write(N),write(" n'est pas chanceu").
```

```
somme(N,S):-R is N mod 10,N1 is N//10,S is N1+R.
```

%Exercice 3:

```
suspect(sami).
```

```
suspect(marie).
```

```
objet(couteau,bureau).
```

```
salles(bureau,bibliothèque).
```

```
empreintes(marie,bureau).
```

```
empreintes(sami,bibliothèque).
```

```
cadavre(bureau).
```

```
coupable(Y):-cadavre(X),empreintes(Y,X).
```

```
objetdecrime(X):-cadavre(Y),objet(X,Y).
```

```
condamné(X):-suspect(X),coupable(X).
```

Final(2017-2018)

%Exercice 1:

```
age(A,B,C,D):-between(20,40,A),between(20,40,D),between(20,40,C),B is D*2,A>C,A=C+D,A=(2*C+D)//2,!.
```

%Exercice 2:

```
division(A,B,0,A):-A<B,!.
```

```
division(A,B,Q,R):-R1 is A-B,division(R1,B,Q1,R),Q is Q1+1.
```

%Exercice 3:

```
horaire(sami,lundi,8).
```

```
horaire(sami,mardi,9).
```

```
horaire(sami,mercredi,10).
```

```
horaire(houda,lundi,6).
```

```
horaire(houda,mardi,5).
```

```
heure(X,L):-findall(Y,horaire(X,_,Y),L).
```

```
paiement([],0).
```

```
paiement([T|L],S):-T<8,! ,J is T*30,paiement(L,S1),S is S1+J.
```

```
paiement([T|L],S):-J is (8*30)+(0.3*30)*(T-8),paiement(L,S1),S is S1+J.
```

```
comb(X,S):-heure(X,L),paiement(L,S).
```

%Exercice4:

```
insereR(L,L1):-length(L,S),R is S mod 2,R=0,! ,S1 is S//2,insereR(L,0,S1,L1).
```

```
insereR(L,L1):-length(L,S),S1 is ((S//2)+1),insereR(L,0,S1,L1).
```

```
insereR([],_,_,[]).
```

```
insereR([T|L],I,S1,[T|L1]):-I<S1,! ,I1 is I+1,insereR(L,I1,S1,L1).
```

```
insereR([T|L],I,S1,[T,O|L1]):-R is T mod 2,R=0,! ,I1 is I+1,insereR(L,I1,S1,L1).
```

```
insereR([T|L],I,S1,[T,1|L1]):-I1 is I+1,insereR(L,I1,S1,L1).
```



Cours : I2209

Durée : 2h

Exercice 1 (30 points)

- ✓ Ecrire un prédictat **sumDiv** qui calcule la somme des diviseurs d'un entier positif.
- ✓ Ecrire un prédictat **abondant** qui teste si un entier est abondant. Un entier N est abondant si la somme de ses diviseurs est plus grande que son double ($2 * N$).
- ✓ Ecrire un prédictat **abondList** qui donne une liste des entiers et recherche les entiers qui sont abondants.

Exercice 2 (30 points)

Voici les faits suivants :

chiffresA(sami,janvier,1000).
chiffresA(sami,mars,500).
chiffresA(houda,février,200).

chiffresA(sami,fevrier,2000).
chiffresA(houda,janvier,500).
chiffresA(houda,mars,300).

- ✓ Définir un prédictat **chiffresAffaire** qui retourne la liste des chiffres d'affaire d'une personne. Exemple, Sami a fait des chiffres d'affaires [1000, 2000, 500]
- 2) Définir un prédictat **commission** qui calcule la commission à payer à une personne en fonction des chiffres d'affaires réalisés durant un trimestre. La personne a le droit à une commission de
 - 10% du total des chiffres d'affaires si le total par trimestre est plus grand à 3000 \$.
 - 5% du total des chiffres d'affaires si le total par trimestre est plus grand que 2000\$
 - 0 \$ si le total est plus petit que 2000\$
- 3) Définir le prédictat **comChiffre** qui combine les deux prédictats précédents pour savoir si une personne a le droit à une commission et combien le montant.

Exercice 3 (20 points)

Soit deux indices, I et K, écrire un prédictat **slice** qui donne une liste L et qui recherche les éléments compris entre le Ième et Kème élément. Commence à compter à partir de 1.

Example: ?- slice([8,4,3,9,10,13,0],6,9,L).
L=[3,9,10,13,0]

Exercice 4(20 points)

Ecrire un prédictat **addHQ** qui ajoute un entier X en tête et en queue d'une liste.

Exemple, L=[1,3,9,5] X=8 L1=[8,1,3,9,5,8]

Bonne chance

Final(2018-2019)

%Exercice 1:

```
sumdiv(N,S):-sumdiv(N,2,S).  
sumdiv(N,I,1):-I>N,!.%STOP!!!  
sumdiv(N,I,S):-R is N mod I,R=0,! ,I1 is I+1,sumdiv(N,I1,S1),S is S1+I.  
sumdiv(N,I,S):-I1 is I+1,sumdiv(N,I1,S).
```

```
abondant(N):-sumdiv(N,S),N1 is N*2,S>N1,! ,write(N),write(" est abondant.").
```

```
abondant(N):-write(N),write(" n'est pas abondant.").
```

```
abondlist(L):-abondlist(L,[]).
```

```
abondlist([],_).
```

```
abondlist([T|L],L1):-member(T,L1),!,abondlist(L,L1).
```

```
abondlist([T|L],L1):-abondant(T),nl,abondlist(L,[T|L1]).
```

%Exercice 2:

```
chiffresA(sami,janvier,1000).
```

```
chiffresA(sami,fevrier,2000).
```

```
chiffresA(sami,mars,500).
```

```
chiffresA(houda,janvier,500).
```

```
chiffresA(houda,fevrier,200).
```

```
chiffresA(houda,mars,300).
```

```
chiffresA(karim,janvier,2500).
```

```
chiffresAffaire(X,L):-findall(Y,chiffresA(X,_,Y),L).
```

```
totalCA([],0).
```

```
totalCA([T|L],S):-totalCA(L,S1),S is S1+T.
```

```
comission(L,C,M):-totalCA(L,S),comission1(S,C,M).
```

```
comission1(S,0.1,M):-S>3000,! ,M is 0.1*S.
```

```
comission1(S,0.5,M):-S>2000,! ,M is 0.5*S.
```

```
comission1(_,0,0).
```

```
comChiffre(X,C,M):-chiffresAffaire(X,L),comission(L,C,M).
```

%Exercice 3:

```
slice(L,N1,N2,L1):-slice(L,N1,N2,1,L1).
```

```
slice([],_,_,_).
```

```
slice([_|L],N1,N2,I,L1):-I<N1,! ,I1 is I+1,slice(L,N1,N2,I1,L1).
```

```
slice([_|L],N1,N2,I,L1):-I>N2,! ,I1 is I+1,slice(L,N1,N2,I1,L1).
```

```
slice([T|L],N1,N2,I,[T|L1]):-I1 is I+1,slice(L,N1,N2,I1,L1).
```

%Exercice 4:

```
addHQ(L,X,[X|L1]):-addHQ1(L,X,L1).
```

```
addHQ1([],X,[X]).
```

```
addHQ1([T|L],X,[T|L1]):-addHQ1(L,X,L1).
```



Exercice 1 (20 points)

Soit le programme suivant :

```
a(X) :- b(X).  
a(3).  
a(Y) :- b(X), Y is X*X.  
b(X) :- c(X), !, X>0.  
b(X) :- c(X), X<4.  
c(3).  
c(6).
```

- Quelles sont toutes les solutions retournées par la requête `a(X)` ?
- On supprime la coupure ! dans le prédictat `b`. Quelles sont toutes les solutions de la requête `b(X)` ?

Exercice 2 (20 points)

Deux entiers `a` et `b` sont amis si la somme des diviseurs de `a` est égale à `b` et la somme des diviseurs de `b` est égale à `a`.
Ecrire un prédictat `amis` qui cherche si deux entiers sont amis ou non.

Exercice 3 (25 points)

Le code secret d'un coffre-fort est un nombre à quatre chiffres. Le chiffre des milliers, celui des dizaines, celui des unités et celui des centaines, dans cet ordre, sont consécutifs. Exemple 5243. La somme de tous les chiffres est 27. Ecrire le prédictat Prolog `codesecret` qui trouve le code secret correspondant à la description donnée ci-dessus.

Exercice 4 (35 points)

On considère deux ensembles $E1 = \{2, 8, 5, 6, 2, 9, 8, 7\}$ et $E2 = \{2, 3, 9, 4, 2, 8\}$

- Ecrire un prédictat redondant pour éliminer les éléments répétés dans un ensemble
- Ecrire un prédictat triage qui trie un ensemble en utilisant la méthode tri par insertion
- Après avoir éliminé les éléments redondants de l'ensemble $E1$ et $E2$, écrire un prédictat `différence` qui cherche les éléments qui se trouvent dans l'ensemble $E1$ et ne se trouvent pas dans l'ensemble $E2$. Exemple $E3 = \{5, 6, 7\}$

Bonne chance

session2(2016-2017)

%Exercice 1:

a(X):-b(X).

a(3).

a(Y):-b(X),Y is X*X.

b(X):-c(X),X>0.

b(X):-c(X),X<4.

c(3).

c(6).

%Exercice 2:

/*

sumdiv(N,S):-sumdiv(N,2,S).

sumdiv(N,I,1):-I>N//2,!.

sumdiv(N,I,S):-R is N mod I,R=:=0,! ,I1 is I+1,sumdiv(N,I1,S1),S is S1+I.

sumdiv(N,I,S):-I1 is I+1,sumdiv(N,I1,S). */

sumdiv(N,S):-M is N//2,sumdiv(M,N,S).

sumdiv(0,_0).

sumdiv(N,D,S):- N>0 , D mod N=:=0,! ,N1 is N-1,sumdiv(N1,D,S1), S is S1+N. /*if*/

sumdiv(N,D,S):- N>0 , N1 is N-1, sumdiv(N1,D,S).

amis(A,B):-sumdiv(A,S1),sumdiv(B,S2),S2=:=A,S1=:=B,! ,write(A),write(" et "),write(B),write(" sont des amis").

amis(A,B):-write(A),write(" et "),write(B),write(" ne sont pas des amis").

%Exercice 3:

```
codesecret(N):-between(3,9,A),C is A-1,D is A-2,B is A-3,(A+B+C+D)=:=27,N is A*1000+B*100+C*10+D.
```

%Exercice 4:

```
redondant(L,L2):-redondant(L,[],L2).
```

```
redondant([],_,[]).
```

```
redondant([T|L],L1,L2):-member(T,L1),!,redondant(L,L1,L2).
```

```
redondant([T|L],L1,[T|L2]):-redondant(L,[T|L1],L2).
```

```
insertion(X,[],[X]).
```

```
insertion(X,[T|L],[X,T|L]):-X=< T,!.
```

```
insertion(X,[T|L],[T|L1]):-insertion(X,L,L1).
```

```
/*triage(L,L2):-triage(L,[],L2).
```

```
triage([],L1,L1).
```

```
triage([T|L],L1,L2):-insertion(T,L1,L3),triage(L,L3,L2).*/
```

```
triage([],[]).
```

```
triage([T|L],LF):-triage(L,L1),insertion(T,L1,LF).
```

```
difference(L):-redondant([2,8,5,6,2,9,8,7],L1),redondant([2,3,9,4,2,8],L2),chercher(L1,L2,L).  
chercher([],_,[]).  
chercher([T|L1],L2,L):-member(T,L2),!,chercher(L1,L2,L).  
chercher([T|L1],L2,[T| L]):-chercher(L1,L2,L).
```

%COMMUN

```
commun(L):-redondant([2,8,5,6,2,9,8,7],L1),redondant([2,3,9,4,2,8],L2),chercher1(L1,L2,L).  
chercher1([],_,[]).  
chercher1([T|L1],L2,[T| L]):-member(T,L2),!,chercher1(L1,L2,L).  
chercher1([T|L1],L2,L):-chercher1(L1,L2,L).
```

Exercice 1 (20 points)

Utiliser le principe de réduction pour montrer la validité de la formule suivante :
 $((c \wedge n) \rightarrow t) \wedge (h \wedge \neg s) \wedge ((h \wedge \neg(s \vee c)) \rightarrow p) \rightarrow ((n \wedge \neg t) \rightarrow p)$

Exercice 2 (20 points)

Traduire les phrases suivantes en logique de prédictats

- 1) Un triangle est isocèle ou équilatéral ou bien rectangle.
- 2) Si un triangle est équilatéral alors il est isocèle.
- 3) Un triangle rectangle n'est jamais équilatéral.
- 4) Dans un triangle isocèle une médiane est également hauteur.
- 5) Un carré est à la fois un parallélogramme et un rectangle.
- 6) Un losange n'est ni un rectangle ni un triangle.

Exercice 3 (25 points)

On considère le problème suivant : « il ya 4 hommes : A, B, C, D. B est deux fois plus âgé que D. L'écart d'âge (la différence) entre A et B est identique à celui de C et D. L'écart d'âge entre A et C est la moitié de celui de B et D. Aucun des hommes n'a strictement plus de quarante ans ni strictement moins de 20 ans. A est plus âgé que C.

Ecrire un programme qui calcule les âges de toutes les personnes.

Exercice 4 (35 points)

- a) Ecrire un prédictat **buildNumber** qui forme un nombre à partir des éléments de la liste.
Exemple : **buildNumber([2,5,8,7], N)** donne **N= 2587**.
- b) Ecrire un prédictat **duplique** qui donne une liste L et qui duplique les éléments compris entre le lème et Kème élément. Commence à compter à partir de 1.

Exemple: ?- **duplique([8,4,3,9,10,13,0,6,9,5],3,7,L).**
L=[8,4,3,3,9,9,10,10,13,13,0,0,6,9,5]

Bonne chance

* Exercice 1:

$$F1 : ((C \wedge m) \rightarrow t) \wedge (R \wedge T_S) \wedge ((R \wedge \neg(S \vee C)) \rightarrow P) \rightarrow ((m \wedge \neg t) \rightarrow P)$$

$$\text{Supposons que } F1 \text{ est } F \Rightarrow ((C \wedge m) \rightarrow t) \wedge (R \wedge T_S) \wedge ((R \wedge \neg(S \vee C)) \rightarrow P) \text{ est } V$$

$$\Rightarrow (C \wedge m) \rightarrow t \text{ est } V \Rightarrow C \wedge m \text{ est } F \Rightarrow (C \wedge t) \text{ est } F$$

$$\Rightarrow (t \wedge V)$$

$$\Rightarrow R \wedge T_S \text{ est } V \Rightarrow (R \wedge V) \wedge T_S \text{ est } V \Rightarrow (R \wedge F)$$

$$\Rightarrow (R \wedge \neg(S \vee C)) \rightarrow P \text{ est } V \Rightarrow R \wedge \neg(S \vee C) \text{ est } F \Rightarrow (R \wedge F)$$

$$\Rightarrow \neg(S \vee C) \text{ est } F \Rightarrow S \vee C \text{ est } V$$

$$\Rightarrow (P \wedge F)$$

$$\Rightarrow (m \wedge \neg t) \rightarrow P \text{ est } F \Rightarrow m \wedge \neg t \text{ est } V \Rightarrow (m \wedge \neg t) \wedge \neg t \text{ est } V \Rightarrow (\neg t \wedge F)$$

$$\Rightarrow (P \wedge F)$$

$\Rightarrow t, m, P, R, S, C$ sont contradictoires \Rightarrow la formule est valide.

* Exercice 2:

$$1) \forall x (\text{triangle}(x) \rightarrow \text{isocèle}(x) \vee \text{équilatéral}(x) \vee \text{rectangle}(x))$$

$$2) \forall x (\text{triangle}(x) \wedge \text{équilatéral}(x) \rightarrow \text{isocèle}(x))$$

$$3) \forall x (\text{triangle}(x) \wedge \text{rectangle}(x) \rightarrow \neg \text{équilatéral}(x))$$

$$4) \forall x (\text{triangle}(x) \wedge \text{isocèle}(x) \rightarrow \exists y (\text{médiane}(y, k) \wedge \text{partie}(y, x)))$$

$$5) \forall x (\text{carre}(x) \rightarrow \text{parallélogramme}(x) \wedge \text{rectangle}(x))$$

$$6) \forall x (\text{losange}(x) \rightarrow \neg \text{rectangle}(x) \wedge \neg \text{triangle}(x))$$

* Exercice 3:

age(A, B, C, D): - between(20, 40, A), between(20, 40, B), between(20, 40, C), between(20, 40, D),
 $B \text{ is } 2^{\star} D$, $A - B \text{ is } C - D$, $A - C \text{ is } (B - D)/2$, $A > C$.



Exercice 1 (20 points)

Utiliser le principe de réduction pour montrer la validité de la formule suivante :

$$(p \rightarrow r) \rightarrow ((q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r))$$

Exercice 2 (20 points)

Nous souhaitons calculer le prix d'un nombre sachant que le prix de chaque chiffre est égal à la valeur qu'il représente sauf le chiffre 0 qui coûte 10 \$. Exemple le chiffre 5 coûte 5 \$, le chiffre 8 coûte 8 \$. Ecrire un prédictat **priceNumber** qui prend en entrée un nombre et retourne son prix. Exemple le nombre 2560 coûte $2+5+6+10 = 23 \$$

Exercice 3 (30 points)

Soit U_0 un entier positif donné par l'utilisateur et la suite :

$$U_{n+1} = U_n/2 \text{ si } U_n \text{ est pair}$$

$$U_{n+1} = 3U_n + 1 \text{ si } U_n \text{ est impair}$$

Ecrire un prédictat **calculate** qui demande à l'utilisateur d'entrer un entier (U_0) et calculer et afficher les termes de cette suite jusqu'à trouver 1.

Exercice 4 (30 points)

Définir un prédictat **remove_dup** (*List, P*) qui permet de supprimer les éléments dupliqués de la liste. *P* est la liste des éléments supprimés.

Exp : `remove_dup([1,2,3,2,3,2,1], P)` donne `P = [1,2,3]`

Bonne chance

I2209 (2019-2020(2ème)):

* Exercice 1:

$$F : (p \rightarrow r) \rightarrow ((q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r))$$

On suppose que F est fausse $\Rightarrow p \rightarrow q$ est V \Rightarrow p est F et q est V

$$\Rightarrow (q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r) \text{ est } F \Rightarrow q \rightarrow r \text{ est } V \Rightarrow \boxed{q \text{ est } F} \text{ et } \boxed{r \text{ est } V}$$

$$\Rightarrow (p \vee q) \rightarrow r \text{ est } F \Rightarrow p \vee q \text{ est } V \Rightarrow \boxed{p \text{ est } V} \text{ ou } \boxed{q \text{ est } V}$$

$$\Rightarrow \boxed{r \text{ est } F}$$

$\Rightarrow p, q, r$ sont contradictoires \Rightarrow la formule est valide.

* Exercice 2:

prix(0,0).

prix(N, P): - N > 0, N is N // 10, R is N mod 10, R = 0, !, prix(N1, P1), P is P1 + R.

prix(N, P): - N > 0, N is N // 10, prix(N1, P1), P is P1 + 10.

* Exercice 3:

calculate(1).

calculate(U): - U > 1, R is U mod 2, R = 1 = 0, !, U1 is U // 2, write(U1), write(" "), calculate(U1).

calculate(U): - U > 1, U1 is 3 * U + 1, write(U1), write(" "), calculate(U1).

Cours : I2209

Année : 2020-2021

Durée : 1h30

Examen : Final

Exercice 1 (20 points)

Des parents s'inquiètent de ce que fera leur fils qui est étudiant, la veille de son examen. Ils savent que :

- a) S'il sort, il se couchera tard ;
- b) Il révisera ou il sortira, ou même peut-être les deux ;
- c) S'il ne révise pas, il aura une mauvaise note ;
- d) Il ne révise pas et il se couche tard
- e) S'il ne sort pas et il révise, il n'aura pas une mauvaise note

- 1- Modéliser les énoncés à l'aide des formules de propositions
- 2- Prouver en utilisant le principe de résolution **qu'il aura une mauvaise note**

Exercice 2 (25 points)

Soit les énoncés suivants :

- 1) Les personnes qui ont le covid19 doivent rester à la maison
- 2) Les personnes qui ont de la fièvre et qui toussent ont soit la grippe, soit le covid19.
- 3) Les personnes qui ont une température supérieure à 38° ont de la fièvre.
- 4) Certaines personnes qui ont la grippe doivent prendre du Tamiflu.
- 5) Sami tousse et a une température supérieure à 38°.
- 6) Sami ne reste pas à la maison.
 - a) Formaliser les phrases ci-dessus en logique des prédictats
 - b) Prouver à l'aide de la méthode de résolution que Sami n'a pas le covid19

Exercice 3 (25 points)

Soit les faits suivants qui représentent les températures estimées à la ville de Tripoli pendant une semaine :

temp(lundi, 28).

temp(mardi, 29).

temp(mercredi, 31).

temp(jeudi, 31).

temp(vendredi, 30).

temp(samedi, 30).

temp(dimanche, 31).

- a) Ecrire un prédictat maxTemp qui cherche la température supérieure pour cette semaine
- b) Afficher la liste des jours qui ont la température maximum.

Exercice 4 (30 points)

- a) Ecrire un prédictat split qui divise la liste L en deux listes L1 et L2 en fonction du nombre qui représente la position où se passe la division. Exemple : split([2,5,8,7,4], 2,L1,L2) donne L1=[2,5] et L2=[8,7,4].
- b) Ecrire un prédictat range qui crée une liste des entiers à partir de deux nombres X et Y donnés par l'utilisateur. Les entiers de la liste sont compris entre X et Y inclus.

Exemple: ?- range(3,7,L) donne L=[3,4,5,6,7]

Bonne chance

T2205 (T) 10/7/2024

1pt for each clause [20]

- Ex I
- $\text{sort} \rightarrow \text{tard}$ ($\text{out} \rightarrow \text{late}$)
 - $\text{rev} \vee \text{sort}$ ($\text{Study} \vee \text{out}$)
 - $\text{rev} \rightarrow \text{mn}$ ($\text{?study} \rightarrow \text{bg}$)
 - $\text{?rev} \wedge \text{tard}$ ($\text{?study} \wedge \text{late}$)
 - $\text{?sort} \wedge \text{rev} \rightarrow \text{?mn}$ ($\text{?out} \wedge \text{?study} \rightarrow \text{?bg}$)

Conc mn (bg) 1pt

2pts each + 1 for
+1 for each clause
+1 for each clause

?out v late

Study v out

Study v bg

?study

late

out v ?study v ?bg

?bg

2pts each

$\rightarrow \text{bg} \rightarrow []$

- Ex II
- $\forall x \text{ covid}(x) \rightarrow \text{home}(x)$

c₁: ?out:d(x₁) v home(x₁) [25]

b) $\forall x (\text{fever}(x) \wedge \text{cough}(x) \rightarrow (\text{flu}(x) \vee \text{covid}(x)))$ c₂: ?fever(x₂) v ?cough(x₂) v covid(x₂)

c) $\forall x (\text{temp}(x) \rightarrow \text{fever}(x))$ 2 each + 1 clause

c₃: ?temp(x₃) v fever(x₃)

d) $\exists x (\text{flu}(x) \wedge \text{tamiflu}(x))$

c₄: flu(a)

e) $\text{cough}(\text{Sam}) \wedge \text{temp}(\text{Sam})$

c₅: tamiflu(a)

f) $\text{?home}(\text{Sam})$

c₆: cough(Sam)

Conc $\rightarrow \text{covid}(\text{Sam})$ ②

c₇: temp(Sam)

2) C₁ + C₂ $\xrightarrow{x_1 \text{ is } a} C_6: ?\text{covid}(\text{Sam})$ ③

c₈: ?home(Sam)

C₁₀ + C₉ $\rightarrow []$ ④

c₉: ?covid(Sam)

Ex III: 1) maxTemp(M) :- findall((Day, Temp), temp(Day, Temp), L), max(L, M). ⑤

max([(-, Temp)], Temp) !-!. ⑥

max([(-, Temp) | T], Max) !- max(T, Max), (Temp > M \rightarrow Max = Temp; Max = M). ⑦

2) displayDay(L) :- maxTemp(Max), findall(Day, temp(Day, Max), L). ⑧

Ex IV 1) split([H|T], 1, [H], T) !-!. ⑨

split([H|T], N, [H|L], L₂) :- N > 1, N is N-1, split(T, N, L₁, L₂). ⑩

2) range(N, N, L) :- N =₃ N, findall(X, between(H, N, X), L). ⑪

or range(H, H, [M]) !-!. ⑫

range(M, N, [M|L]) :- M < N, M is M+1, range(M, N, L). ⑬

[25]

[30]

[10]

Cours : I2209

Année : 2020-2021

Durée : 1h30

Exam : 2^{ème} session

Exercice 1 (30 points)

Soit les énoncés suivants :

- a) Si Sami a faim, il mange ;
- b) Si la maman de Sami fait du haricot et Sami mange, il tombe malade ;
- c) La maman de Sami fait du haricot et Sami a faim ;
- d) Si Sami a faim et il ne mange pas, il tombe malade ;
- e) Si Sami ne mange pas, la maman de Sami fait du haricot ;

- 1- Modéliser les énoncés à l'aide des formules de propositions
- 2- Prouver en utilisant le principe de résolution que Sami mange et tombe malade

Exercice 2 (30 points)

Soit les formules suivantes :

- (A) $\exists x (p(x) \wedge \neg q(x))$
- (B) $\forall x (p(x) \rightarrow (\exists y r(y, x)) \rightarrow q(x))$
- (C) $\forall x \forall y (r(x, y) \rightarrow \neg r(y, x))$
- (D) $\exists x (\neg \exists y r(y, x))$

Prouver, à l'aide de la méthode de résolution, la validité de la formule suivante :

$$A \wedge B \wedge C \rightarrow D$$

(Il faut détailler les étapes suivantes : mise en forme prénexe, Elimination des qualificateurs, Mise sous forme clausal)

Exercice 3 (40 points)

- a) Ecrire un prédictat **verifNumber(E, N, L)** qui teste si l'élément E se trouve N fois dans la liste L. Ce prédictat doit retourner YES ou NO.
Exemple : ?- verifNumber(2, 4, [2,5,6,2,9,2,2]) retourne YES
verifNumber(5, 3, [2,5,6,2,9,2,2]) retourne NO
- b) Ecrire un prédictat **liste_pos(E, L, L1)** qui retourne les positions de l'élément E dans une liste L1.
Exemple: ?- liste_pos(3,[1,3,5,4,3,8,3],L1) donne L1=[1,4,6]

Bonne chance

* Exercice 1:

Soit : p : faim

q : mange

r : fait de haricot.

t : tombe malade .

$$\textcircled{1} \Rightarrow p \rightarrow q$$

$$\hookrightarrow (\neg p \wedge q) \rightarrow t$$

$$\hookrightarrow r \wedge p$$

$$\textcircled{d} \Rightarrow (p \wedge r) \rightarrow t$$

$$\hookrightarrow \neg q \rightarrow r$$

$$\textcircled{2} \quad (p \rightarrow q) \wedge ((r \wedge q) \rightarrow t) \wedge (r \wedge p) \wedge ((p \wedge r) \rightarrow t) \wedge (\neg q \rightarrow r) \rightarrow (q \wedge t).$$

$$(p \rightarrow q) \wedge ((r \wedge q) \rightarrow t) \wedge (r \wedge p) \wedge ((p \wedge r) \rightarrow t) \wedge (\neg q \rightarrow r) \wedge \neg (q \wedge t).$$

$$(\neg p \vee q) \wedge (\neg r \vee \neg q \vee t) \wedge (r) \wedge (p) \wedge (\neg p \vee q \vee t) \wedge (q \vee r) \wedge (\neg q \vee t)$$

$$c_1: \neg p \vee q$$

$$c_2: \neg r \vee \neg q \vee t$$

$$c_3: r$$

$$c_4: p$$

$$c_5: \neg p \vee q \vee t.$$

$$c_6: q \vee r$$

$$c_7: \neg q$$

$$c_8: \neg t.$$

* Exercice 2:

$$\textcircled{1} \quad \exists x (p(x) \wedge \neg q(x))$$

$$p(a) \wedge \neg q(a).$$

$$\textcircled{2} \quad \forall x (p(x) \rightarrow ((\exists y r(y, x)) \rightarrow q(x)))$$

$$\forall x (p(x) \rightarrow (\neg (\exists y r(y, x)) \vee q(x)))$$

$$\forall x (p(x) \rightarrow (\forall y \neg r(y, x) \vee q(x)))$$

$$\forall x \forall y (p(x) \rightarrow \neg r(y, x) \vee q(x)).$$

$$\forall x \forall y (\neg p(x) \vee \neg r(y, x) \vee q(x))$$

$$\neg p(x) \vee \neg r(y, x) \vee q(x).$$

$$c_9: c_1 + c_4 + q.$$

$$c_{10}: c_9 + c_7; \quad \emptyset \neq$$

\Rightarrow donc la formule est valide

\Rightarrow Sami mange et tombe malade .

$$\textcircled{1} \quad \forall x \forall y (r(x, y) \rightarrow \neg r(y, x)) .$$

$$\forall x \forall y (\neg r(x, y) \vee \neg r(y, x)) .$$

$$\textcircled{2} \quad \neg (\exists x (\exists y r(y, x))).$$

$$\forall x \exists y r(y, x).$$

$$\neg r(f(x), x)$$

C 1 : P(a)

C 2 : $\neg q(a)$

C 3 : $\neg P(x) \vee \neg r(y, x) \vee a(y)$.

C 4 : $\neg r(x, y) \vee \neg r(y, x)$

C 5 : $r(f(x), x)$.

C 6 : C 1 + C 3 : $\neg r(y, a) \vee q(a)$.

C 6 : C 1 + C 3 : $\neg r(y, a)$.

C 7 : C 6 + C 2 : $\neg r(y, a) \Rightarrow$ la formule est valide.

C 8 : C 7 + C 5 : $\emptyset \Rightarrow$ la formule est valide.



Final Exam: I2209E

Duration: 1h30

2nd Year Computer Science – 2nd semester 1st session

Documents: non authorized

Date: 25/07/2023

When defining new prolog predicates, explain briefly what they are intended to do.

Exercise I

(15 pts)

We interview a logician who always tells the truth about his love life. To the question: "Is it true that if you love Marie, then you love Anne?" he answers:

- If it is true, then I love Mary.
- If I love Mart, then it is true.

Translate the answer into propositional logic, then using truth table, what can we conclude?

Exercise II

(15 pts)

Here are the curious rules of a British club. In your opinion, is the regulation coherent?

وهما القواعد الغر يه ملادى ابر طا

Voici le curieux règlement d'un club britannique. A votre avis, ce règlement est-il cohérent ?

- All non-Scottish members wear orange shoes.
- Any member wears a skirt or does not wear orange shoes.
- Married members do not go out on Sundays.
- A member goes out on Sunday if and only if he is Scottish.
- Any member that wears a skirt is Scottish and married.
- All Scottish members wear skirt.

Exercise III

(10 pts)

Using the function *chief(x)* (chief of *x*), the constant Bob, and the following predicates:

emp(x): *x* is an employee.

vac(x): *x* is on vacation.

child(x): *x* has children.

work(x, y): *x* works with *y*.

Translate the following statements in predicate logic:

- Chiefs of all employees that have children are on vacation.
- All employees without children who work with at least one employee who has children are on vacation.
- Bob does not know all the employees that are on vacation.

Exercise IV

(15 pts)

Study the validity of the following formula:

$$\left(\exists x \left(P(x) \wedge \forall y (Q(y) \rightarrow R(x, y)) \right) \wedge \forall x \left(P(x) \rightarrow \forall y (S(y) \rightarrow \neg R(x, y)) \right) \right) \rightarrow \forall x (Q(x) \rightarrow \neg S(x))$$

Exercise V

(20 pts)

Translate the following information to Prolog:

Socrates and Plato are human. Tom and Titi are animals. R2d2 is a droid. Zeus and Athena are gods. Humans and animals are living beings. Living beings and droids are mortal. Gods are immortal.

Draw the search tree for the query $\text{mortel}(R)$.

Exercise VI

(25 pts)

Consider the following Prolog notation of numerals:

- 0 is a numeral.
 - If X is a numeral, then $\text{succ}(X)$ is also a numeral.
1. Write a Prolog predicate $\text{value_of}(E, V)$ which is true if E is a numeral and V is its numerical value. Example: $\text{value_of}(\text{succ}(\text{succ}(\text{succ}(\text{succ}(0)))), V)$ returns $V = 4$
 2. Write a Prolog predicate $\text{subtract}(E1, E2, E3)$ that succeeds if $E3$ is the difference $E1 - E2$ between two numerals $E1$ and $E2$, assuming $E1$ is greater than or equal to $E2$. If $E1 < E2$, this predicate fails. Example:

$\text{subtract}(\text{succ}(\text{succ}(\text{succ}(\text{succ}(0)))), \text{succ}(\text{succ}(0)), Z)$ returns $Z = \text{succ}(\text{succ}(0))$

$\text{subtract}(\text{succ}(\text{succ}(0)), \text{succ}(\text{succ}(\text{succ}(0))), Z)$ returns No

ex III

- $\forall x (\text{empl}(x) \wedge \text{child}(x) \rightarrow \text{vac}(\text{chief}(x)))$
- $\forall x [(\text{empl}(x) \wedge \exists y (\text{child}(y) \wedge \text{work}(x, y)) \rightarrow \text{vac}(x)]$
- $\exists x [\underbrace{\text{empl}(x) \wedge \text{vac}(x)}_{\substack{\downarrow \\ \text{negation of}}} \rightarrow \text{work}(\text{Bob}, x)]$
it means: Bob knows all employees
that are on vacation

ex II:

① let $s(x)$: x is a scottish member

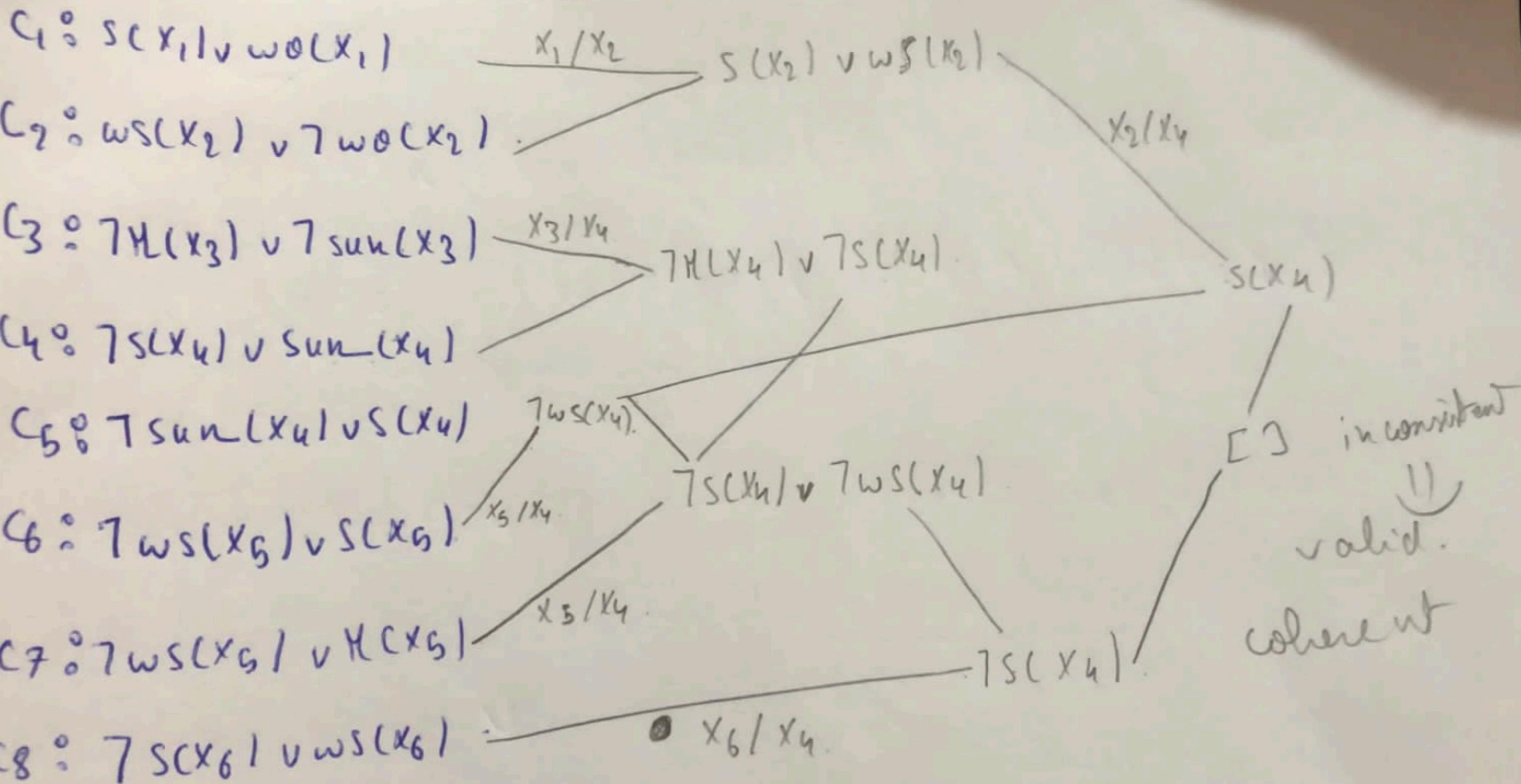
$w\theta(x)$: x wears orange shoes

$ws(x)$: x wears a skirt

$h(x)$: x is a married member

$sun(x)$: x goes out on Sundays

- $\forall x (\exists s(x) \rightarrow w\theta(x))$
 - $\forall x (ws(x_1) \vee \exists w\theta(x_1))$
 - $\forall x (h(x_1) \rightarrow \exists sun(x_1))$
 - $\forall x (s(x_1) \leftrightarrow \exists sun(x_1))$
 - $\forall x (ws(x) \rightarrow s(x) \wedge h(x))$
 - $\forall x (s(x) \rightarrow ws(x))$
-



ex 4^o

$\exists x (\exists u \forall y (P(u) \wedge P(y) \rightarrow R(x, y)))$	$\forall x (P(x) \rightarrow \forall y (S(y) \rightarrow \neg R(x, y)))$
$\exists x \forall y (P(x) \wedge (P(y) \rightarrow R(x, y)))$	$\forall x \forall y (P(x) \rightarrow (S(y) \rightarrow \neg R(x, y)))$
$(P(x) \wedge (P(y) \rightarrow R(x, y)))$	$\neg P(x) \vee (\neg S(y) \vee \neg R(x, y))$
$(P(x) \wedge (\neg Q(y) \vee R(x, y)))$	$(\neg P(x) \vee \neg Q(y)) \vee \neg R(x, y)$
$C_1 : P(a)$	$C_3 : \neg P(x_3) \vee \neg Q(y_3) \vee \neg R(x_3, y_3)$
$C_2 : \neg Q(y_2) \vee R(a, y_2)$	

B^o $\forall x (Q(x) \rightarrow \neg S(x))$

7B^o $\exists x (Q(x) \wedge S(x))$

$Q(b) \wedge S(b)$

C₄ : $Q(b)$

C₅ : $S(b)$

$\{ P(a), \neg Q(y_2) \vee R(a, y_2), \neg P(x_3) \vee \neg S(y_3) \vee \neg R(x_3, y_3), Q(b), S(b) \}$

y_2/b

$R(a, b)$

x_3/a

y_3/b

$\neg R(a, b)$

$\neg S(y_3) \vee \neg R(a, y_3)$

[] inconsistency so the formula is valid

final_ex6.pl [modified]

```
numeral(0).
numeral(succ(X)) :- numeral(X).
value_of(0, 0).
value_of(succ(X), V) :- value_of(X, W), V is W + 1.
subtract2(E1, E2, E3) :- value_of(E1, V1), value_of(E2, V2),
    V1 >= V2, V3 is V1 - V2, value_of(E3, V3).
```

SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)

File Edit Settings Run Debug Help

Action? .

```
?- subtract2(succ(succ(succ(succ(0)))), succ(succ(0)), Z).
Z = succ(succ(0)).
```

```
?- subtract2(succ(0), succ(succ(0)), Z).
false.
```

```
?- value_of(succ(succ(succ(succ(0)))), V).
V = 4.
```

```
?- ■
```

Ex :-

human (socrates).

human (pluto).

animal (titi).

animal (tom).

droid (r2d2).

god (zeus).

god (athena).

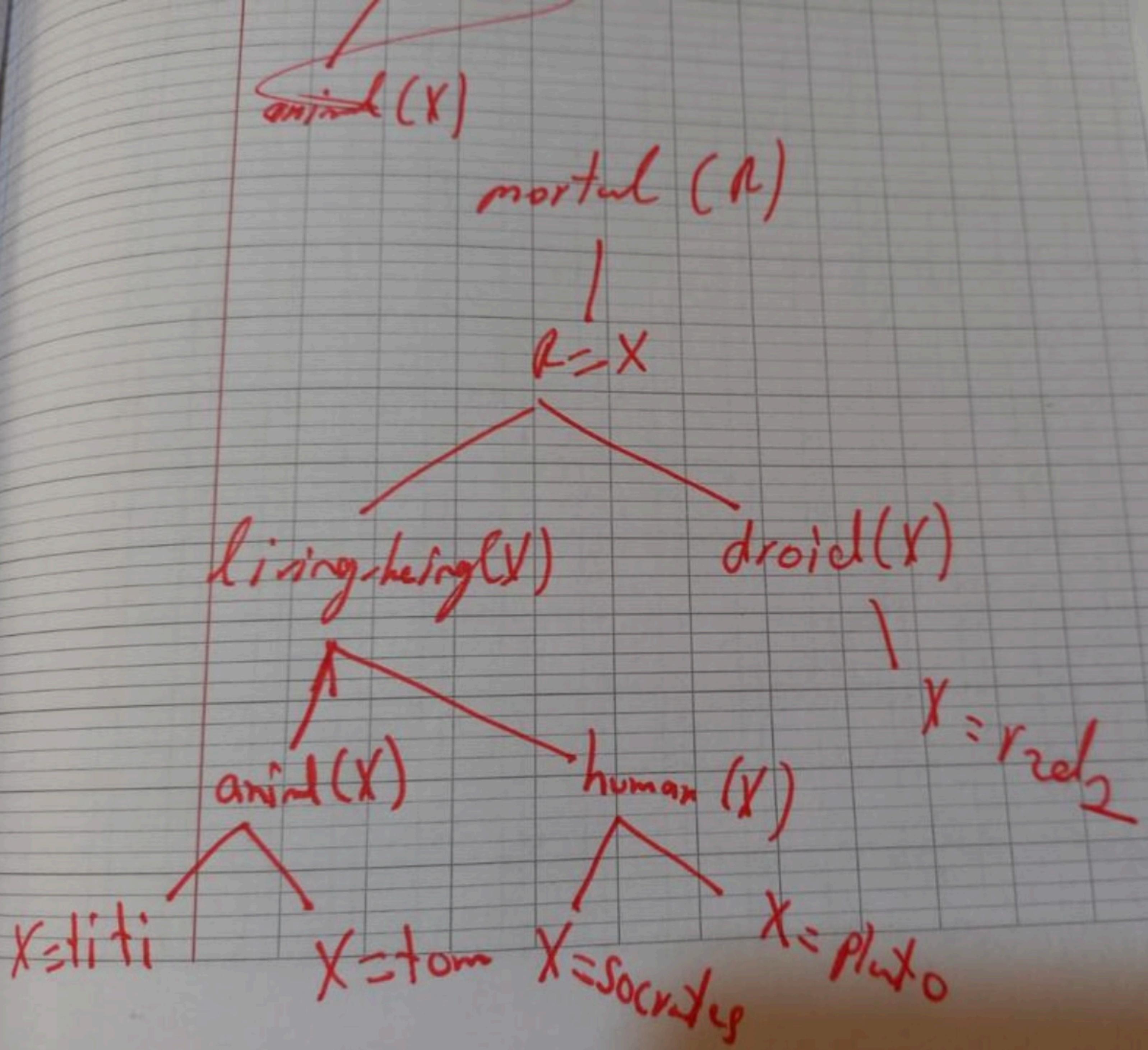
living-being (X) :- animal (X).

living-being (X) :- human (X).

mortal (X) :- living-being (X).

mortal (X) :- droid (X).

immortal (X) :- god (X).



$\text{mons}(\text{succ}(X), \text{succ}(Y), 2) :- \text{mons}(X, Y, 3).$ (9) 12

Ex I $m : j'aime Marie \models_2$ $a : j'aime Anne \models_3$ $m \rightarrow a \models_1$ $\{(m \rightarrow a) \rightarrow m\}_{F_1} 2$

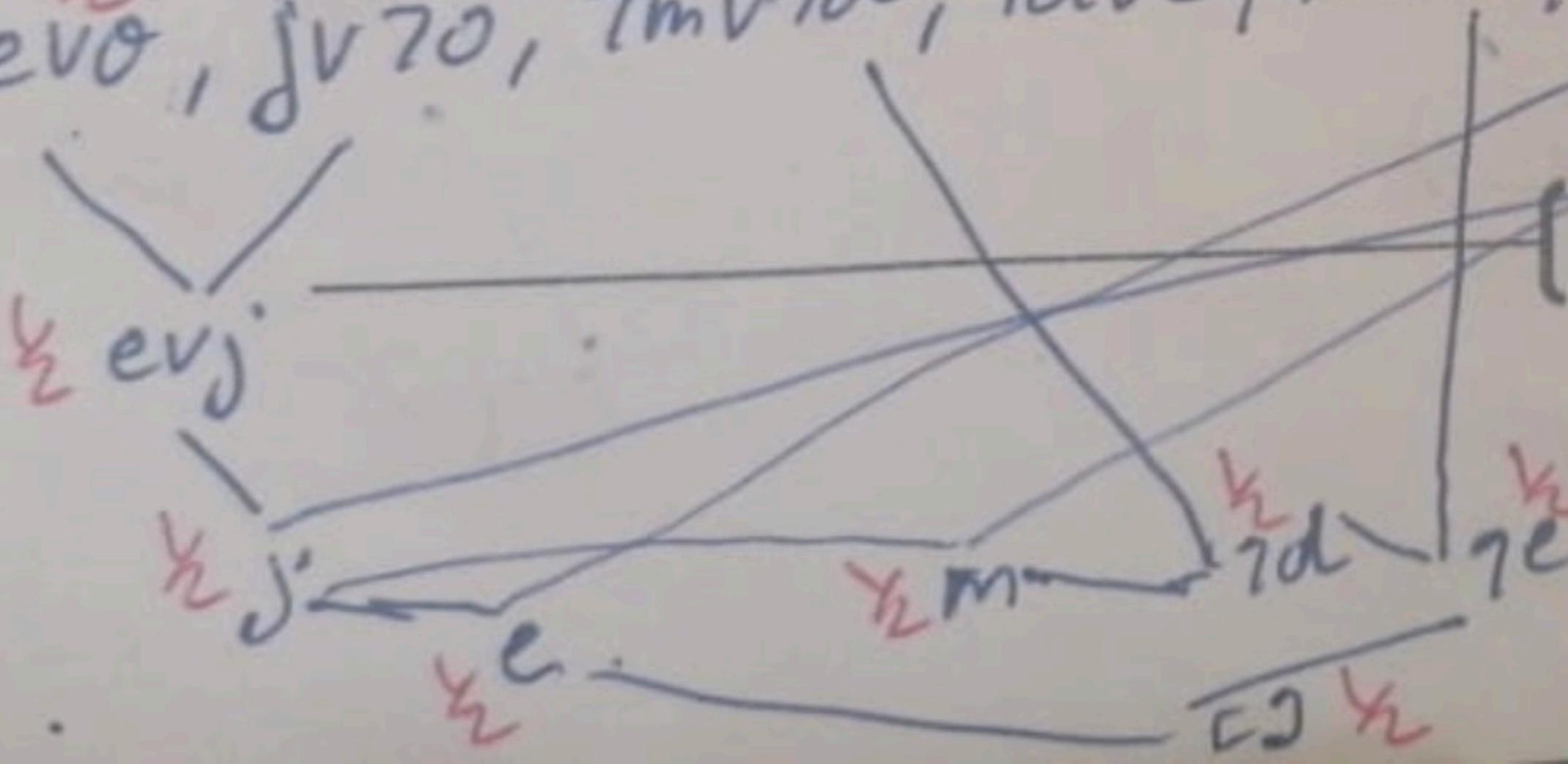
$m \rightarrow (m \rightarrow a) \models_2$ $F_2 F_1 \wedge F_2 \models_1$ 15

$\logique{aime}{Marie et Anne}$
à la fois 1

m	a	$m \rightarrow a \models_1$	$(m \rightarrow a) \rightarrow m \models_2$	$m \rightarrow (m \rightarrow a) \models_2$	$F \models_1$
F	F	T	F	T	F
F	T	T	F	T	F
T	F	F	T	F	F
T	T	T	T	T	T

Ex II $7e \rightarrow \theta$ $j \vee \theta$ $m \rightarrow 7d$ $d \Leftrightarrow e$ $j \rightarrow e \wedge m$ $e \rightarrow j$ ensemble non cohérent ou non satisfaisable

$S = \{e \vee \theta, j \vee \theta, 7m \wedge 7d, 7d \vee e, 7e \wedge d, 7j \vee e, 7d \vee m, 7e \wedge j\}$



15