CALCUL DES PREDICATS DU 1er ORDRE

EXEMPLE

$F_0 = \rceil p(a) \vee q(a,f(a))$, $F_1 = \rceil p(a) \vee r(f(a))$,

$F_2 = p(a)$, $F_3 = \rceil q(a,a)$,

$F_4 = \rceil p(f(a)) \vee q((f(a)),f((f(a)))$, $F_5 = \rceil p(f(a)) \vee r(f(f(a)))$,

$F_6 = \rceil q(a,f(a))$, $F_7 = \rceil q(f(a),a)$, etc

Soit i l'interprétation telle que :

i[p(a)] = V, i[q(a,f(a))]=V, i[r(f(a))]=V, i[q(a,a)]=F, i[p(f(a))]=F

Une fois arrivé à $F_0 \wedge F_1 \wedge F_2 \wedge \ldots \wedge F_6$, on constate qu'il est impossible de construire un modèle car F_0 ,F_2 et F_6 ne peuvent pas être satisfaites simultanément. Il n'y a pas de modèle pour A1, A2, A3 et A4. D'où :

$$\forall x\ (p(x) \rightarrow \exists\, y\ (r(y) \wedge q(x,y)))) \ ,\ \exists\, x\ p(x) \ \models\ \exists\, x\ \exists\, y\ q(x\ ,\ y))$$

CALCUL DES PREDICATS DU 1er ORDRE

EXERCICE

La formule $B = (\exists x \rceil q(x) \rightarrow \forall y\ p(y))$

est-elle conséquence des deux formules

$B1 = (\exists x\ p(x) \rightarrow \forall y\ p(y))$

$B2 = \forall x\ (p(x) \vee q(x))$?

$\{(\exists x\ p(x) \rightarrow \forall y\ p(y)), \forall x\ (p(x) \vee q(x))\} \models (\exists x \rceil q(x) \rightarrow \forall y\ p(y))$?

CALCUL DES PREDICATS DU 1er ORDRE

EXERCICE

$\{\exists x\, p(x) \rightarrow \forall y\, p(y))\ , \forall x\, (p(x) \vee q(x))\} \models (\exists x \rceil q(x) \rightarrow \forall y\, p(y))$

Mise sous forme de skolem

$\rceil q(a) \wedge \rceil p(b)$

$\rceil p(x) \vee p(y)$

$p(x) \vee q(x)$

Mise sous forme de clauses

$\rceil p(x) \vee p(y)$

$p(x) \vee q(x)$

$\rceil q(a)$

$\rceil p(b)$

Univers de Herbrand

$U_H = \{a\ , b\}$

Atomes de Herbrand

$A_H = \{q(a), p(a), p(b), q(b)\}$

Système de Herbrand

$S_H = \{\rceil p(a) \vee p(a), \rceil p(a) \vee p(b), \rceil p(b) \vee p(b), \rceil p(b) \vee p(a), p(a) \vee q(a), p(b) \vee q(b), \rceil q(a), \rceil p(b)\}$

Soit i telle que : $i(q(a))=F$, $i(p(a))=V$, $i(q(b))=V$, $i(p(b))=F$

On a $i(F_1 \wedge F_2 \wedge F_3 \wedge F_4 \wedge F_5 \wedge F_6 \wedge F_7 \wedge F_8 = F$ Donc : B1, B2 |== B

INFÉRENCE LOGIQUE

<u>Définition</u>

➢Nous avons une série de propositions qui sont vraies (prémisses)

➢Nous déduisons de nouveaux faits vrais de celles-ci (conclusion)

Exemple :

Prémisses (propositions vraies)

A→ B : si Ali est sage alors il ira au cinéma

<u>Prémisse</u>	<u>Conclusion</u>
A : Ali est sage	B : Ali ira au cinéma

On note l'inférence par |-- : A→B , A |-- B

INFÉRENCE LOGIQUE

- ➢ Grâce à l'inférence, on déduit de nouvelles propositions qui sont des conséquences logiques des propositions initiales

- ➢L'inférence est l'utilisation des faits et de règles pour trouver de nouveaux faits : soit de nouvelles propositions

INFERENCE LOGIQUE

REGLES D'INFERENCE

Modus ponens : $A, A \rightarrow B \mid\!-\!- B$

Modus tollens : Si $\rceil B$ et $A \rightarrow B \mid\!-\!- \rceil A$

Enchaînement : $A \rightarrow B , B \rightarrow C \mid\!-\!- A \rightarrow C$

Résolution : $A \vee B , \neg B \vee C \mid\!-\!- A \vee C$

INFERENCE LOGIQUE

INSTANTIATION

Instanciation universelle :

On peut substituer une variable par une constante

Exemple : $\forall x$ Aime(x, fruit)

La substitution de x par Ali donne Aime(Ali, fruit)

Instanciation Existentielle :

$\exists x$ Aime(x,fruits) donne après substitution Aime(Ali, fruit)

CHAINAGE AVANT

Définition:

Pour déduire un fait particulier, on déclenche les règles dont les prémisses sont connues jusqu'à ce que le fait à déduire soit également connu ou qu'aucune règle ne soit plus déclenchable

INFÉRENCE LOGIQUE

CHAINAGE AVANT

Existe-t-il une règle applicable ? : on cherche parmi toutes les règles celles dont la partie condition est vraie et à en choisir une à l'aide d'une fonction de choix.

Appliquer cette règle : une fois la règle choisie, le programme exécute sa partie action ou conclusion.

Désactiver cette règle : en logique des propositions, il est inutile d'appliquer plus d'une fois la même règle : les règles utilisées sont rendues inactives.

Le but souhaité est-il démontré ? : Si ce fait vient d'être obtenu, il est inutile de poursuivre le travail.

Dans le cas où aucun but particulier n'est demandé, le moteur fonctionne jusqu'au moment où aucune règle n'est applicable (condition d'arrêt). On dit alors que le moteur **fonctionne par saturation**.

INFERENCE LOGIQUE

ALGORITHME DU CHAINAGE AVANT

ENTREES : BF , BR , F

DEBUT

<u>TANT QUE</u>

(F n'est pas dans BF) ET (il existe dans BR une règle applicable)

<u>FAIRE</u>

- ✓ choisir une règle applicable R
- ✓ BR = BR - R (désactivation de R)
- ✓ BF = BF U {conclusion(R)}

<u>FIN DU TANT QUE</u>

<u>SI</u> F appartient à BF <u>ALORS</u> F est établi

<u>SINON</u> F n'est pas établi

FIN

INFERENCE LOGIQUE

EXEMPLE

Base de Faits : {B, C}

But à démontrer : H

Bases de règles :

(1) B , D , E → F
(2) G , D → A
(3) C , F → A
(4) B → X
(5) D → E
(6) X , A → H
(7) C → D
(8) X , C → A
(9) X , B → D

INFÉRENCE LOGIQUE

EXEMPLE

Démonstration du But H

$\{B, C\} \xrightarrow{(4)} \{B, C, X\} \xrightarrow{(7)} \{B,C,X,D\} \xrightarrow{(5)} \{B,C,X,D,E\} \xrightarrow{(1)} \{B,C,X,D,E,F\} \xrightarrow{(3)} \{B,C,X,D,E,F,A\} \xrightarrow{(6)} \{B,C,X,D,E,F,A,H\}$

Chainage avant jusqu'à saturation

$\{B, C\} \xrightarrow{(4)} \{B, C, X\} \xrightarrow{(7)} \{B,C,X,D\} \xrightarrow{(5)} \{B,C,X,D,E\} \xrightarrow{(1)} \{B,C,X,D,E,F\} \xrightarrow{(3)} \{B,C,X,D,E,F,A\} \xrightarrow{(6)} \{B,C,X,D,E,F,A,H\} \xrightarrow{(8)} \{B,C,X,D,E,F,A,H\} \xrightarrow{(9)} \{B,C,X,D,E,F,A,H\}$

INFERENCE LOGIQUE

EXEMPLE

Base de Faits : {A, G}

But à démontrer : I

Bases de règles :
(1) A , B , C → D
(2) G , D → F
(3) A→ H
(4) B → C
(5) F , H → I
(6) E → B
(7) H , A → B
(8) H , G → F

INFERENCE LOGIQUE

(1) A , B , C → D
(2) G , D → F
(3) A → H
(4) B → C
(5) F , H → I
(6) E → B
(7) H , A → B
(8) H , G → F

EXEMPLE

- On prend comme règle déclenchable la 1ère règle déclenchable dans l'ordre de la numérotation

{A,G} (3)→ {A,G,H} (7) → {A,G,H,B} (4) → {A,G,H,B,C} (1) → {A,G,H,B,C,D} (2) → {A,G,H,B,C,D,F} (5) → {A,G,H,B,C,D,F,I}

- On prend comme règle déclenchable la 1ère règle parmi celles qui comportent le maximum de prémisses

{A,G} (3)→ {A,G,H} (7) → {A,G,H,B} (8) → {A,G,H,B,F} (5) → {A,G,H,B,F,I}

INFERENCE LOGIQUE

CHAINAGE ARRIERE

Définition:

Le mécanisme de chaînage arrière consiste à:

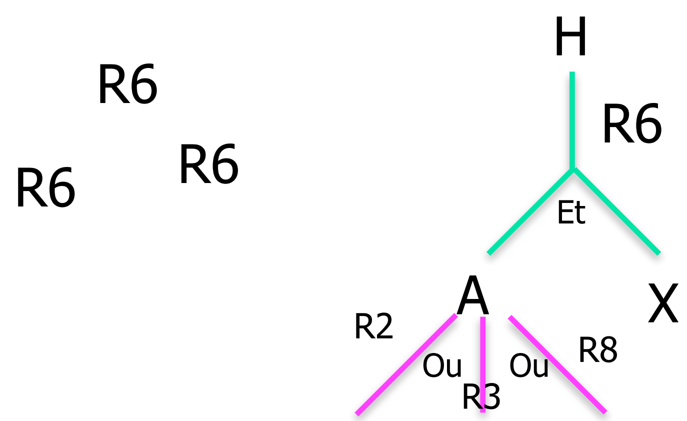
partir du fait que l'on souhaite établir, rechercher toutes les règles qui concluent sur ce fait, établir la liste des faits qu'il suffit de prouver pour qu'elles puissent se déclencher puis appliquer récursivement le même mécanisme aux faits contenus dans ces listes.

un arbre dont les noeuds sont étiquetés soit par un : fait ,et , ou
Arborescence et-ou

Base de Faits : {B, C}

But à démontrer : H

Bases de règles :



(1) B , D , E → F
(2) G , D → A
(3) C , F → A
(4) B → X
(5) D → E
(6) X , A → H
(7) C → D
(8) X , C → A
(9) X , B → D

INFERENCE LOGIQUE

CHAINAGE ARRIERE

On peut enrichir l'algorithme de chaînage arrière en tenant compte de l'interactivité

Dans ce cas, lorsqu'un fait demandable n'a pas encore été établi, le système le demandera à l'utilisateur avant d'essayer de le déduire à partir d'autres faits connus

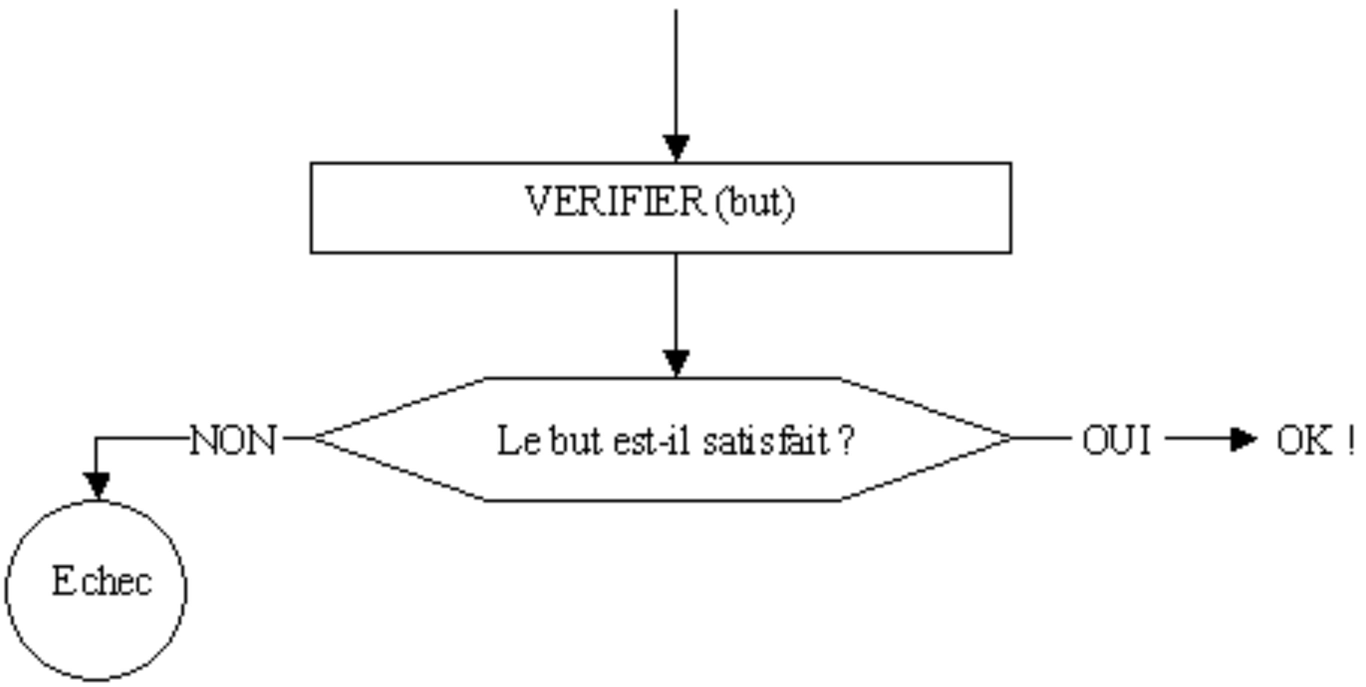
⇓

Il faut poser des questions pertinentes

INFERENCE LOGIQUE

CHAINAGE ARRIERE

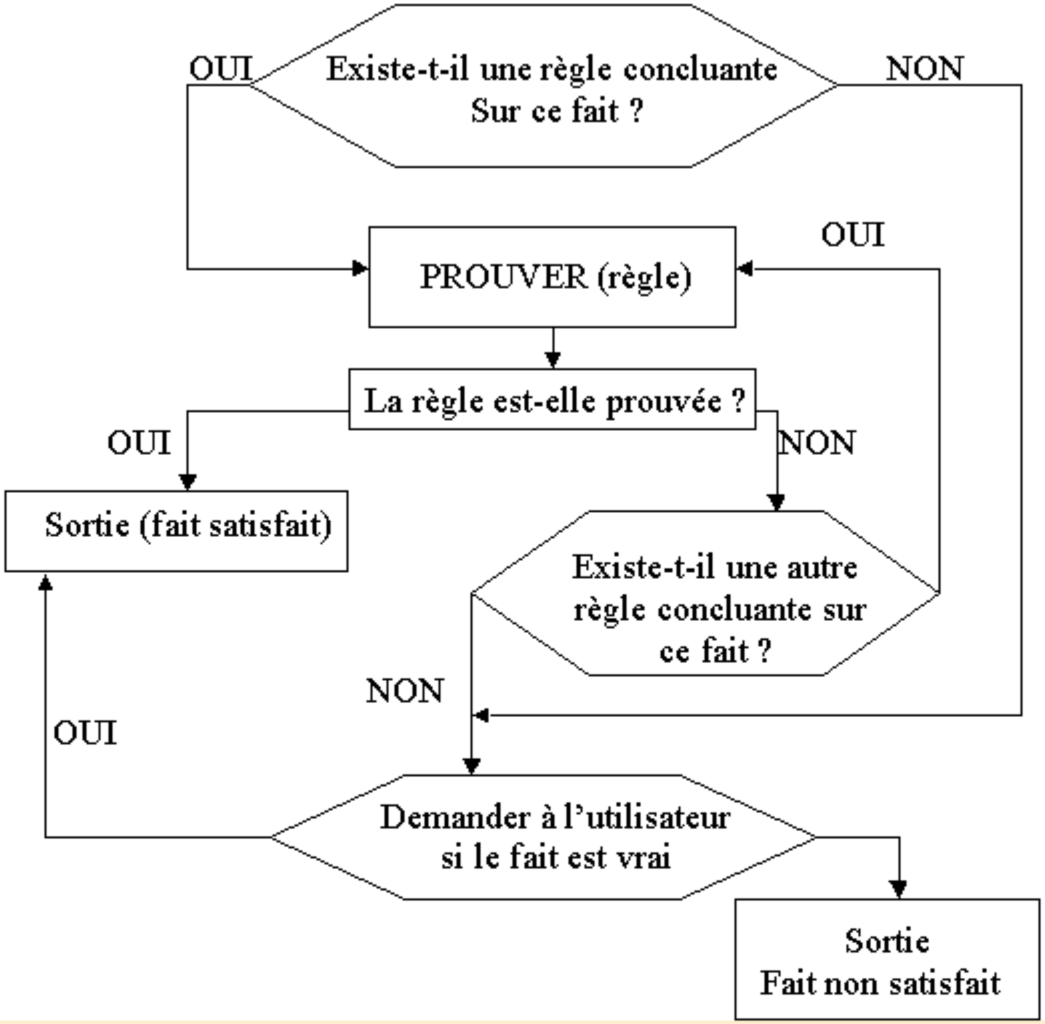
ORGANIGRAMME :



INFERENCE LOGIQUE

CHAINAGE ARRIERE

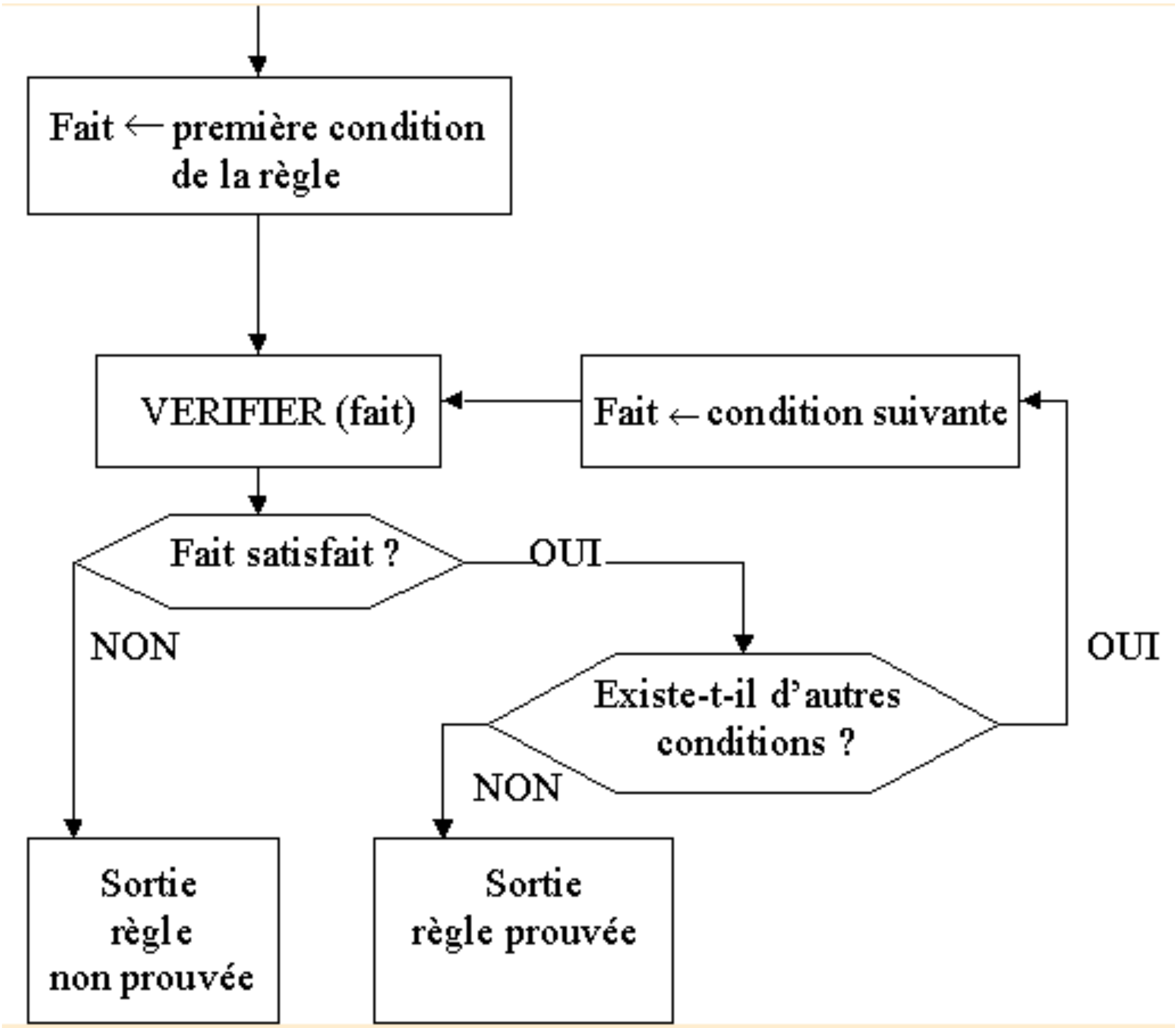
VERIFIER :



INFERENCE LOGIQUE

CHAINAGE ARRIERE

PROUVER :



INFERENCE LOGIQUE

EXEMPLE

Base de Faits : {A, G}

But à démontrer : I

Bases de règles :
(1) A , B , C → D
(2) G , D → F
(3) A → H
(4) B → C
(5) F , H → I
(6) E → B
(7) H , A → B
(8) H , G → F

INFERENCE LOGIQUE

EXEMPLE

Base de Faits : {B, C}

But à démontrer : H

Bases de règles :
(1) B , D , E → F
(2) G , D → A
(3) C , F → A
(4) B → X
(5) D → E
(6) X , A → H
(7) C → D
(8) X , C → A
(9) X , B → D

INFERENCE LOGIQUE

EXEMPLE

Base de Faits : J est vrai

Question : A est-il vrai ?

Faits demandables :
B
D
F
I

Base de règles :
(1) B , C → A
(2) D , E → A
(3) F , G → A
(4) I , J → G
(5) J → $\rceil$E

INFERENCE LOGIQUE

CHAINAGE MIXTE

L'algorithme de chaînage mixte combine les algorithmes de chaînage avant et de chaînage arrière.

ENTREE : F (à déduire)
DEBUT
TANT QUE F n'est pas déduit mais peut encore l'être **FAIRE**

- ✓ Saturer la base de faits par chaînage AVANT(i.e. déduire tout ce qui peut être déduit)
- ✓ Chercher quels sont les faits encore déductibles
- ✓ Déterminer une question pertinente à poser à l'utilisateur et ajouter sa réponse à la base de faits

FIN DU TANT QUE
FIN

RESOLUTION

EXERCICE

Soit le système formel RSV défini par :

- Σ_{RSV} = {p_0, p_1, ….., p_n,….} ∪ {⌉ , v}
- F_{RSV} =ensemble de toutes les clauses construites à partir des pi, toutes les clauses sont supposées sans répétitions de littéraux. La clause n'ayant aucun littéral est notée ? et n'est satisfaite par aucune interprétation.

Résolution : A V B , ¬B V C |-- A V C

- $A_{RSV} = \varnothing$
- R_{RSV} = {cut}

$$f \text{ v } p_i,\ g \text{ v } \rceil\, p_i \vdash_{cut} f \text{ v } g$$

RESOLUTION

EXERCICE

1°/ $p_0 \vee p_1 \vee p_2 \, , \, \neg p_0 \vee p_1 \vee p_2 , \, \neg p_1 \vee p_2 \; |\!\!--- \; p_2$?

2°/ $p_0 \vee p_1 \, , \, p_0 \vee \neg p_1 \, , \neg p_0 \vee p_1 , \, \neg p_0 \vee \neg p_1 \; |\!\!--- \; \boxed{?}$?

3°/ Y a-t-il une relation entre la relation cut et les chainages avant et arrière

4°/ Comparer le système formel Σ_{RSV} et le calcul propositionnel

5°/ Montrer que si $F_1 \; |\!\!--_{cut} \; C$ alors $F_1 \models C$

RESOLUTION

EXERCICE

$p_0 v p_1 v p_2$, $\neg p_0 v p_1 v p_2$, $\neg p_1 v p_2$ |--- p_2 ?

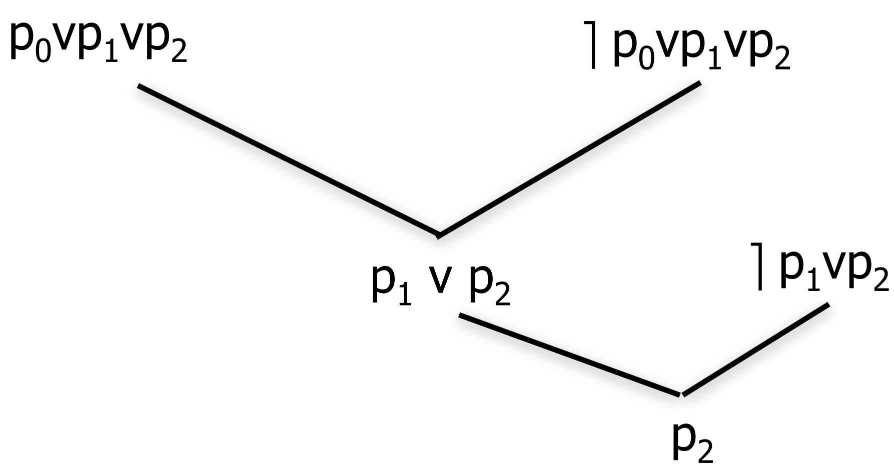
f1 : $p_0 v p_1 v p_2$ (Hypothèse)

f2 : $\neg p_0 v p_1 v p_2$ (Hypothèse)

f3 : p_1 v p_2 (cut avec f1 et f2)

f4 : $\neg p_1 v p_2$ (Hypothèse)

f5 : p_2 (cut avec f3 et f4)



Arbre de dérivation associé à la déduction

RESOLUTION

EXERCICE

$p_0 \vee p_1 \,,\; p_0 \vee \neg p_1 \,,\; \neg p_0 \vee p_1 \,,\; \neg p_0 \vee \neg p_1 \;\vdash\!\text{---}\; \boxed{?}$

f1 : $p_0 \vee p_1$ (Hypothèse)

f2 : $p_0 \vee \neg p_1$ (Hypothèse)

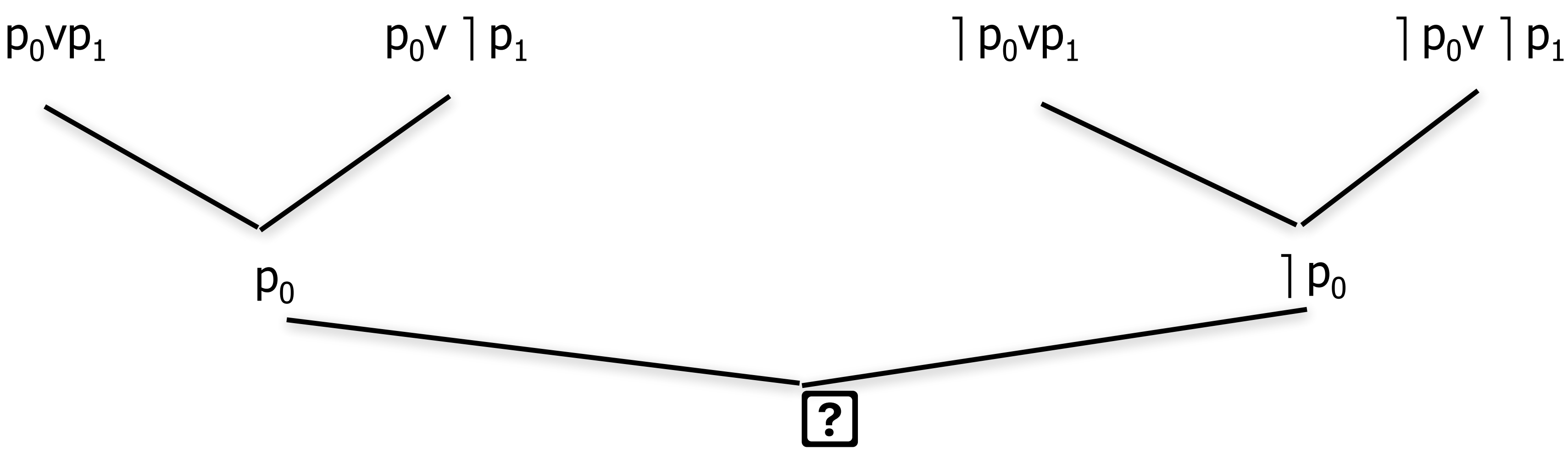
f3 : p_0 (cut avec f1 et f2)

f4 : $\neg p_0 \vee p_1$ (Hypothèse)

f5 : $\neg p_0 \vee \neg p_1$ (Hypothèse)

f6 : $\neg p_0$ (cut avec f4 et f5)

f7 : $\boxed{?}$ (cut avec f3 et f6)



Arbre de dérivations associé à la déduction