Université Bretagne Sud
L3 Informatique

Logique

Travaux Dirigés - Partie 6
Corrigés

Ce sixième TD est consacré à des exercices et compléments pour la Logique Propositionnelle (LP0).
Les exercices sont de difficultés diverses et sont à traiter en se basant sur les notions introduites en cours (chapitre 3) et en annexe de ce TD.
J'ai choisi de présenter deux méthodes d'analyse pour la LP0 dans l'annexe du TD pour ne pas surcharger le cours, déjà riche en concepts.

Bon travail !

Rappelons qu'on appelle **clause** *une disjonction de littéraux de la LP0.*

Exercice 1 (clauses de Horn)

Les *clauses de Horn* sont particulièrement importantes en programmation. Elles sont définies de la façon suivante.

Si L, P, L_i ($A \leq i \leq n$) sont des littéraux positifs, alors une *clause de Horn* ou *formule conditionnelle* a une des formes suivantes :

1) L
2) $\bigvee_{i=1}^{n} \neg L_i$
3) $P \vee \bigvee_{i=1}^{n} \neg L_i$ ou $\bigvee_{i=1}^{n}(L_i \Rightarrow P)$ ou $(\bigwedge_{i=1}^{n} L_i) \Rightarrow P$

Questions :

a) Prouver le théorème suivant : *Si H est un ensemble satisfaisable de clauses de Horn, alors l'intersection de tout ensemble non vide de modèles de H et aussi un modèle de H.*

On dit que les clauses de Horn ont la *propriété d'intersection de modèles.*

b) Si l'on remplace *clauses de Horn* par *clauses* dans l'énoncé précédent, a-t-on toujours un théorème ? Justifier.

a) On suppose que les modèles sont représentés par l'ensemble des symboles propositionnels évalués à V dans les modèles.

Évidemment, le théorème ne dépend pas de la représentation des modèles adoptée.

On note :

$$H = \{C_1, \ldots, C_n\}$$

$Mod(H)$: l'ensemble des modèles de H.

$\mathcal{N} \subseteq Mod(H)$ ($\mathcal{N} \neq \emptyset$).

$M_\cap = \cap M_k, \; M_k \in \mathcal{N}$.

Preuve :

Par l'absurde. On suppose :

$\not\models_{M_\cap} H$ donc pour au moins une clause $C_j \in H$: $\not\models_{M_\cap} C_j$.

Deux cas sont a considérer ((correspondant aux différentes formes des clauses de Horn).

i) $C_j : A \vee \neg B_1 \vee \neg B_2 \vee \cdots \vee \neg B_m$, $m \geq 0$ ($m = 0$ correspond à une clause positive unitaire).

Pour évaluer à F une clause, il faut évaluer à F tous ses ittéraux, c'est-à-dire :

$$A \notin M_\cap \text{ et } B_1, B_2, \ldots, B_m \in M_\cap$$

donc, par définition de l'intersection, il existe $M_k \in \mathcal{N}$ tel que :

$$A \notin M_k \text{ et } B_1, B_2, \ldots, B_m \in M_k$$

mais alors M_k n'est pas un modèle de H (puisqu'il falsifie l'une de ses clauses : C_j). Contradiction.

ii) $C_j : \neg B_1 \vee \neg B_2 \vee \cdots \vee \neg B_m$, $m > 0$.

Pour évaluer C_j à F :

$$B_1, B_2, \ldots, B_m \in M_\cap$$

donc, par définition de l'intersection, pour tout $M_k \in \mathcal{N}$:

$$B_1, B_2, \ldots, B_m \in M_k$$

mais alors les M_k ne sont pas des modèles de H (puisqu'ils falsifient l'une de leurs clauses : C_j). Contradiction.

b) Non. Prendre $H = \{A \vee B\}$, $Mod(H) = \{\{A\}, \{B\}, \{A,\ B\}\}$ et $\mathcal{N} = \{\{A\}, \{B\}\}$. Alors, $M_\cap = \emptyset$, qui n'est pas un modèle de H.

Les exercices 2 et 3 portent sur la méthode de Davis et Putnam*. Celle-ci est présentée dans l'annexe 1 de ce TD.*

Exercice 2

On considère le système formel sans axiomes logiques ($\mathcal{A} = \emptyset$) :

$$\mathcal{S}_{DP} =< \mathcal{L}, \mathcal{R} >$$

où $\mathcal{L}$ est composé des ensembles de clauses et où $\mathcal{R} = \{$R-0, R-1, R-2, R-3, R-4$\}$.

Prouver que la méthode de Davis et Putnam est *correcte* et *complète*.

Ici, on peut traduire *correcte* par : **si** $S \vdash_{\mathcal{S}_{DP}}$ (S insatisfaisable) **alors** S est insatisfaisable, autrement dit si on peut faire confiance à la méthode.

On peut traduire *complète* par : **si** S est insatisfaisable **alors** $S \vdash_{\mathcal{S}_{DP}}$ (S insatisfaisable), autrement dit si on peut détecter *tous* les ensembles de clauses insatisfaisables par la méthode.

R-0)

Sans perte de généralité on suppose que S contient une seule clause tautologique T.

Soit $S_1 = S \setminus \{T\}$.

S insat si et seulement si S_1 insat.

si)

On prouve la contraposée :

S sat.

Par définition, tout sous-ensemble d'un ensemble sat est sat, donc S_1 sat.

seulement si)

On prouve la contraposée :

S_1 sat, avec pour modèle, disons, M.

Comme T est satisfaite par toute interprétation, elle l'est aussi par M, donc S sat .

R-1a) Triviale. Aucune interprétation ne peut rendre V à la fois L et L^c.

R-1b) (Pour simplifier la notation on suppose une seule clause unitaire $\{L\}$ et une seule clause de la forme $L^c \cup \alpha$. Évidemment, le raisonnement peut être répété).

$S_1 = ((S \setminus \{L\}) \setminus (\{L^c\} \cup \alpha)) \cup \{\alpha\}$.

S insat si et seulement si S_1 insat.

seulement si) On prouve la contraposée.

S_1 sat, soit M_1 un modèle de S_1. $L^c \notin M_1$ (si $L^c \in M_1$, on peut l'enlever sans que ceci ait une quelconque conséquence). Il existe $K \in \alpha$ tel que $K \in M_1$ (toutes les clauses de S_1 doivent être évaluées à V) .

$M = \{L\} \cup M_1$ est un modèle de S. Donc S sat.

si) On prouve la contraposée : S sat, soit M un modèle de S. Nécéssairement $L \in M$ (et donc $L^c \notin M$). Il existe $K \in \alpha$ tel que $K \in M$, donc donc S_1 sat.

R-2)

On suppose, sans perte de généralité, qu'il a un seul littéral pur : $L \in C \in S$ et L : pur.

$S_1 = S \setminus \{C\}$

S insat si et seulement si S_1 insat.

On prouve la contraposée.

S_1 sat, soit M_1 un modèle de S_1, $L^c \notin M_1$ (puisque L est pur).

$M_1 \cup \{L\}$ est un modèle de S. S est donc sat.

si)

On prouve la contraposée.

S sat et $S_1 \subsetneq S$. S_1 ne peut pas être insat, parce que tout modèle de S est modèle de *toutes* les clauses de S. Donc S_1 sat.

Cette règle est connue comme *principe de pureté.*

R-3)

$S_1 = \{C \setminus \{L\} | C \in S \text{ et } L^c \notin C\}$

$S_2 = \{C \setminus \{L^c\} | C \in S \text{ et } L \notin C\}$

S insat si et seulement si (S_1 insat et S_2 insat).

seulement si)

On prouve la contraposée.

S_1 sat ou S_2 sat.

Sans perte de généralité, on suppose S_1 sat (on ne traite pas le cas S_2 sat, la preuve étant la même).

S_1i sat, soit M_1 un modèle de S_1. $L \notin M_1$ et il existe $K \in M_1$, $K \in C \setminus \{L\}$.

$M = M_1 \cup \{L^c\}$ est un modèle de S. Donc S sat.

si)

On suppose sans perte de généralité qu'il y a une seule clause contenant L (disons la clause $L \vee \alpha$) et une seule clause contenant L^c (disons $L^c \vee \beta$).

On prouve la contraposée.

S sat, soit M un modèle de S. Supposons $L \in M$, donc $L^c \notin M$. Comme M est un modèle de *toutes* les clauses de S, il existe $K \in \beta$ tel que $K \in M$, donc S_2 sat donc S_1 sat ou S_2 sat. Même raisonnement si $L^c \in M$.

Si $L \notin M$ et $L^c \notin M$, il existe $K_1 \in M$, $K_2 \in M$ tel que $K_1 \in \alpha$, $K_2 \in \beta$. Donc S_1

sat et S_2 sat.

R-4)

$L \vee \alpha \in S$, $L \vee \alpha \vee \beta \in S$

$S_1 = S \setminus \{L \vee \alpha \vee \beta\}$

S insat si et seulement si S_1 insat.

si)

On prouve la contraposée.

S sat. Comme $S_1 \subsetneq S$, alors S_1 sat.

seulement si)

On prouve la contraposée.

S_1 sat, soit M_1 un modèle de S_1. Évidemment M_1 est modèle de $L \vee \alpha$, donc (définition de clause) M_1 est aussi modèle de $L \vee \alpha \vee \beta$. Donc S est sat.

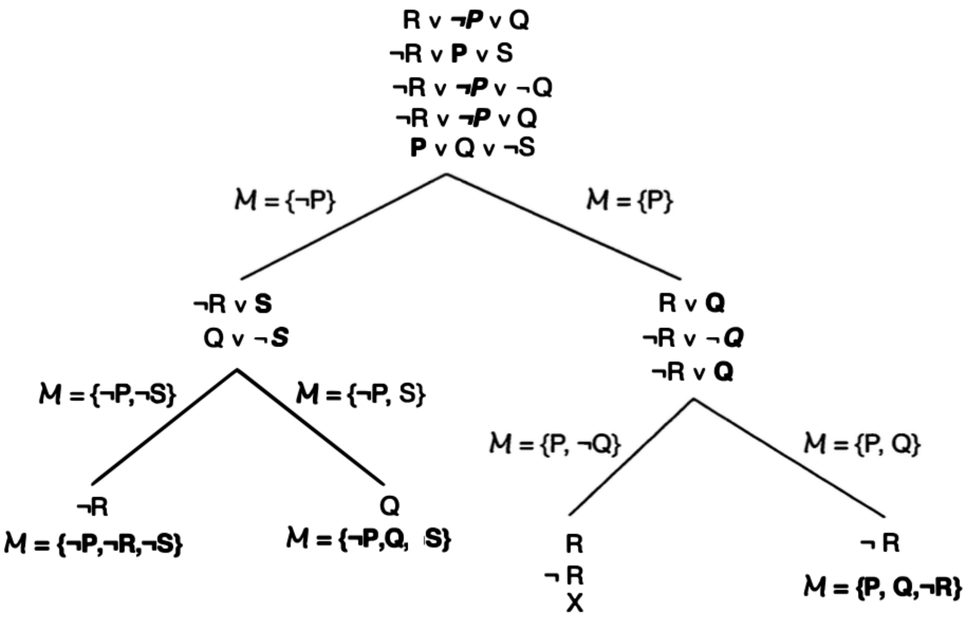
Exercice 3

Peut-on, pour l'ensemble de clauses S de l'exemple 2 de l'annexe 1, trouver d'autres modèles en appliquant la méthode de Davis et Putnam avec une autre stratégie ?

La réponse est oui, comme le montre l'application de l'algorithme avec la stratégie ci-dessous.

Dans l'exemple, on a implicitement appliqué la stratégie « trouver un(des) modèle(s) le plus tôt possible ».

Nous appliquons maintenant la stratégie « essayer de trouver le plus de modèles possible »(en fait dans ce cas particulier on les trouve tous).



Nous avons donc trouvé les six modèles suivants :

$$\begin{aligned}
\mathcal{M}_1 &= \{\neg P, Q, \neg R, \neg S\} \\
\mathcal{M}_2 &= \{\neg P, \neg Q, \neg R, \neg S\} \\
\mathcal{M}_3 &= \{\neg P, Q, R, S\} \\
\mathcal{M}_4 &= \{\neg P, Q, \neg R, S\} \\
\mathcal{M}_5 &= \{P, Q, \neg R, S\} \\
\mathcal{M}_6 &= \{P, Q, \neg R, \neg S\}
\end{aligned}$$

Les exercices 4 et 5 font appel à la notion d'arbre sémantique. *Celle-ci est présentée dans l'annexe 2 de ce TD.*

Exercice 4

a) Donner un *arbre sémantique* (il y en a plusieurs, dépendants de l'ordre choisi pour $B(S)$) pour l'ensemble de clauses :

$$S = \{C_1, C_2, C_3, C_4, C_5, C_6\}$$

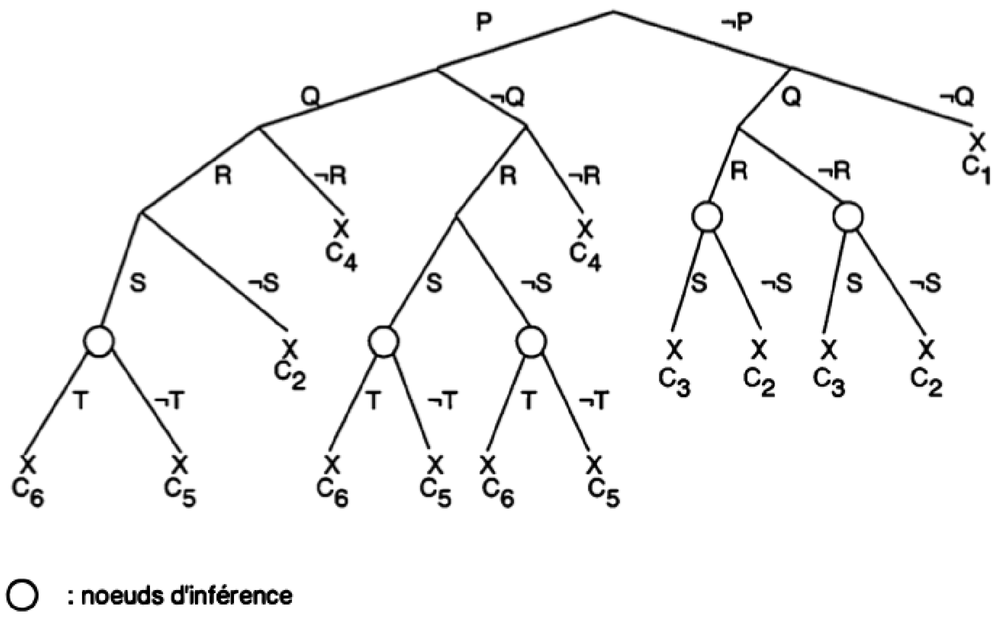
avec

$$C_1 \; : \; P \vee Q$$
$$C_2 \; : \; \neg Q \vee S$$
$$C_3 \; : \; P \vee \neg S$$
$$C_4 \; : \; \neg P \vee R$$
$$C_5 \; : \; \neg P \vee \neg R \vee T$$
$$C_6 \; : \; \neg R \vee \neg T$$

b) Marquez tous les noeuds d'inférence. Quelle signification pouvez-vous donner à ces noeuds ?

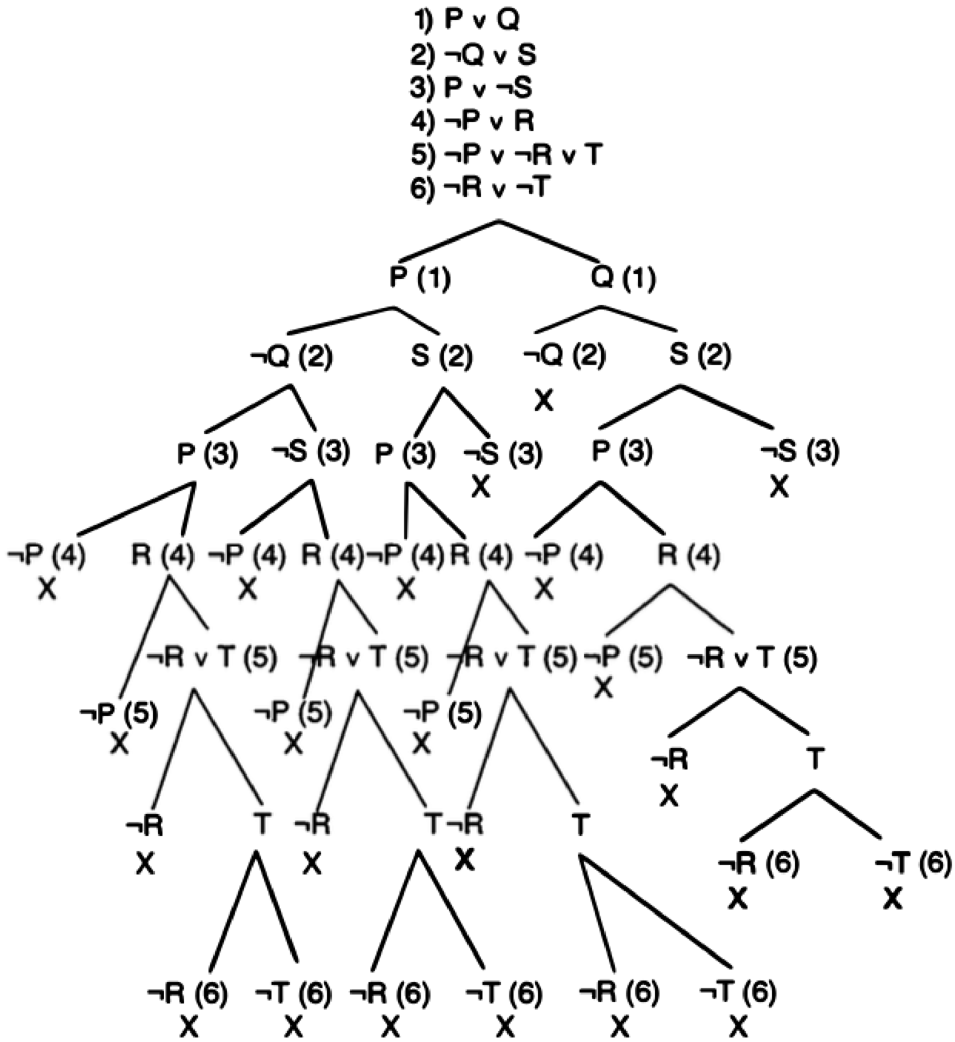
c) Donner un *tableau sémantique* pour S.

a) On prend l'ordre suivant (arbitraire) pour les formules de base : $P < Q < R < S < T$.



b) L'interprétation que l'on peut donner des nœuds d'inférence est la suivante. Les interprétations dénotées par les deux branches passant par un nœud d'inférence falsifient les clauses $C_i : \alpha \vee L$ et $C_j : \beta \vee L^c$ (où α et β sont des disjonctions de littéraux). Par définition de clause, l'interprétation dénotée par la branche finissant dans le nœud d'inférence falsifie tous les littéraux de α et tous les littéraux de β. Elle falsifie donc la *résolvante* de C_i et C_j (cf. TD 7).

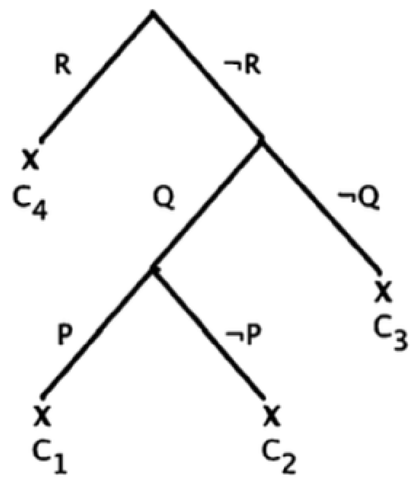
c)



Exercice 5

Donner un *arbre sémantique* pour l'ensemble de clauses ci-dessous :

$$
\begin{aligned}
C_1 &: \neg P \vee \neg Q \vee R \\
C_2 &: P \vee R \\
C_3 &: Q \vee R \\
C_4 &: \neg R
\end{aligned}
$$



ANNEXE 1 : METHODE DE DAVIS ET PUTNAM

L'importance de cette méthode reflète celle du problème SAT (problème de **sat**isfaisabilité d'un ensemble de clauses).

Elle s'applique à des **ensembles de clauses** S ou, ce qui est équivalent, à des formules sous **forme normale conjonctive** (fnc).

Elle permet de décider si S est insatisfaisable ou satisfaisable et, dans ce dernier cas, d'en fournir des modèles.

L'idée sous-jacente est assez simple. Si l'on veut détecter si un ensemble de clauses a des modèles, on considère les littéraux un à un. Un littéral L peut être évalué soit à V soit à F.
S'il est évalué à V, la clause le contenant peut être ignorée.
S'il est évalué à F, on peut effacer L de la clause le contenant.
Évidemment les mêmes règles s'appliquent à L^c.

La méthode utilise les règles suivantes :

R-0 Éliminer les clauses contenant des tautologies (c'est-à-dire les clauses de la forme $L \vee \neg L \vee \alpha$).

R-1 a) Si S contient deux clauses unitaires complémentaires alors S est insatisfaisable.
b) (*règle de la clause unitaire*) Si a) ne s'applique pas et si S contient une clause unitaire L, éliminer toutes les clauses contenant L et toutes les occurrences de L^c dans les clauses.

R-2 (*règle du littéral pur*) Si L est présent mais L^c ne l'est pas, toutes les clauses dans lesquelles L apparaît peuvent être éliminées.

R-3 (*règle de la division*) S'il y a dans S des clauses non unitaires contenant L et L^c, remplacer S par S_1 et S_2 :

S_1 est l'ensemble de clauses où toutes les occurrences de L ont été éliminées ainsi que toutes les clauses contenant L^c.

S_2 est l'ensemble de clauses où toutes les occurrences de L^c ont été éliminées ainsi que toutes les clauses contenant L .

R-4 (*règle de la subsomption*) Si S contient une clause de la forme $L \vee \alpha$, éliminer toutes les clauses de la forme $L \vee \alpha \vee \beta$ (α et β étant des disjonctions de littéraux).

L'algorithme **DP** ci-après applique les règles R-1 à R-4 et permet de détecter la satisfaisabilité (ou insatisfaisabilité) d'un ensemble de clauses de la LP0.

Il est facile de vérifier que la règle R-0 peut être appliquée comme un prétraitement puisque les autres règles ne peuvent pas engendrer de tautologies (elles divisent les clauses ou en éliminent).

Algorithme $DP(S)$

entrée : un ensemble fini de clauses de la LP0 : S

sortie : *insat* ou *sat*

début

cas

• **si** $S = \emptyset$ retourner *sat*

• **si** R-1 a) s'applique (c.a.d. $L \in S$ et $L^c \in S$) retourner *insat*

• **si** R-2 s'applique donnant S_1 **alors**

si $DP(S1) = sat$ alors retourner *sat* sinon retourner *insat*

• **si** R-1 b) s'applique donnant S_1 **alors**

si $DP(S_1) = sat$ alors retourner *sat* sinon retourner *insat*

• **si** R-3 s'applique donnant S_1 et S_2 **alors**

si $DP(S_1) = sat$ ou $DP(S_2) = sat$ alors retourner *sat* **sinon**

retourner *insat*

fin-cas

fin

Exemple 1 : Soit l'ensemble de clauses

$$S = \{P \vee Q, \neg Q \vee S, \neg S \vee P, \neg P \vee R, \neg R \vee \neg P \vee T, \neg T \vee \neg R\}$$

correspondant à la fbf sous fnc :

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

La déduction montrant que S est contradictoire est la suivante :

$$
\begin{array}{c}
\mathbf{P} \vee Q \\
\neg Q \vee S \\
\neg S \vee \mathbf{P} \\
\neg \mathbf{P} \vee R \\
\neg R \vee \neg \mathbf{P} \vee T \\
\neg T \vee \neg R
\end{array}
$$

$$
\begin{array}{cccc}
\swarrow & & \searrow & \text{(R} - 3\text{)}
\end{array}
$$

$$
\begin{array}{cc|cc}
Q & & \neg \mathbf{Q} \vee \mathbf{S} & \\
\neg Q \vee S & & R & \\
\neg S & & \neg R \vee T & \\
\neg \mathbf{T} \vee \neg \mathbf{R} & & \neg T \vee \neg R & \\
\downarrow & \text{(R} - 2\text{)} & \downarrow & \text{(R} - 2\text{)} \\
\mathbf{Q} & & \mathbf{R} & \\
\neg \mathbf{Q} \vee S & & \neg \mathbf{R} \vee T & \\
\neg S & & \neg T \vee \neg \mathbf{R} & \\
\downarrow & \text{(R} - 1b\text{)} & \downarrow & \text{(R} - 1b\text{)} \\
S & & T & \\
\neg S & & \neg T & \\
\times & \text{(R} - 1a\text{)} & \times & \text{(R} - 1a\text{)}
\end{array}
$$

Dans la littérature cette méthode est souvent présentée dans le but de résoudre le problème SAT (problème de *satisfaisabilité*).

Il est très facile, en s'inspirant de la preuve de correction et complétude d'obtenir l'algorithme pour la construction de modèles d'ensembles de clauses satisfaisables.

L'exemple suivant montre clairement les étapes de l'algorithme de construction de modèles.

Les deux propriétés suivantes, dont la justification est immédiate, sont très utiles dans la conception de l'algorithme (la première n'est pas utilisée dans l'exemple).

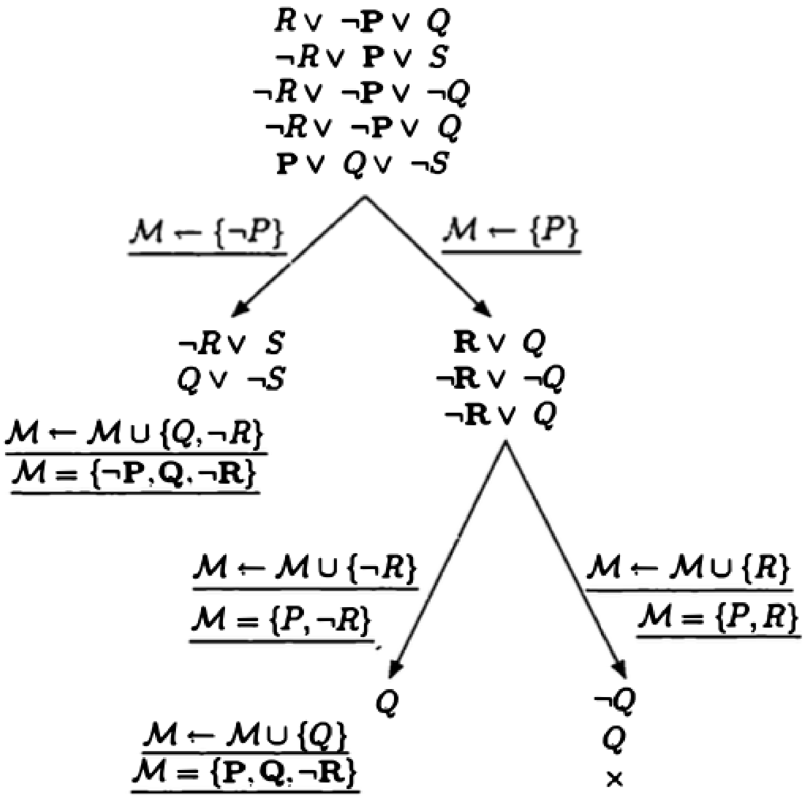
Soit S l'ensemble de clauses à étudier, $\mathcal{M}$ est l'ensemble spécifiant le modèle possible que l'on est en train de construire.

– Si $C \in S$ et C unitaire (c'est-à-dire si C contient un seul littéral L), alors nécessairement $L \in \mathcal{M}$.

– Si $L \in C \in S$ et L pur, alors $\mathcal{M} \leftarrow \mathcal{M} \cup \{L\}$ (au début, $\mathcal{M} \leftarrow \{L\}$) est un modèle de C.

Si une clause devient unitaire et/ou un littéral devient pur par application des règles et $L^c \in \mathcal{M}$, alors le modèle n'est pas viable.

Exemple 2 : Déterminer si l'ensemble de clauses $\mathcal{E}$ ci-dessous est insatisfaisable ou satisfaisable. Dans le dernier cas en donner des modèles.



Nous avons donc construit quatre modèles de $\mathcal{E}$:

$$\{\neg P, Q, \neg R, S\},\ \{\neg P, Q, \neg R, \neg S\},\ \{P, Q, \neg R, S\},\ \{P, Q, \neg R, \neg S\}.$$

On aurait pu arrêter la recherche lors de l'obtention du premier modèle (par exemple $\{\neg P, Q, \neg R, S\}$).

ANNEXE 2 : ARBRES SEMANTIQUES

Avant tout, notons que la méthode des *arbres sémantiques* est différente de la méthode des *tableaux sémantiques* :

– La méthode des *tableaux sémantiques* sert à énumérer les *modèles* (modèles partiels dans la LP0) d'un ensemble de *fbf*.

– La méthode des *arbres sémantiques* sert à énumérer les *interprétations* (interprétations partielles dans la LP0) d'un ensemble de *clauses*.

Définition 1. *Soit S un ensemble de clauses. La* **base** *de S, notée $B(S)$ est définie par :*

$$B(S) = \{L | L \textit{ littéral positif et } [(L \in C \in S) \textit{ ou } (L^c \in C \in S)]\}$$

Définition 2. *Soit S un ensemble de clauses. Etant donnée une énumération des éléments de $B(S)$:*

$$B(S) = \{L_1, \ldots, L_n\}$$

ou $B(S) = \{L_1, L_2, \ldots, L_n, \ldots\}$ si S est infini, un **arbre sémantique** *pour S est un arbre binaire dont les branches sont étiquetées comme suit :*

$$f_g(n_i^j) = L_{i+1} \quad f_d(n_i^j) = \neg L_{i+1}$$

où $L_i \in B(S)$ $(0 \leq i \leq n-1)$, f_g désigne le fils gauche, f_d le fils droit, $i \geq 0$ la profondeur du noeud et j tel que $1 \leq j \leq 2^{i+1}$ la place de gauche à droite.

Il est clair d'après la définition que :

– une branche ne peut pas contenir $L \in C \in S$ et $L^c \in D \in S$

– l'ensemble des branches correspond à l'ensemble des interprétations de S.

Le noeud n_i (pour le plus petit i) dont la branche (i.e. l'interprétation) qui passe par n_i est un *contre-modèle* d'une clause de S est appelé un *noeud d'échec* (noté $\times$).

Une branche contenant un noeud d'échec est dite *fermée*.

Une branche non fermée est dite *ouverte* et correspond à un modèle de S (quand S est infini, une branche ouverte est nécessairement infinie).

Un arbre sémantique est dit *fermé* si et seulement si toutes ses branches sont fermées, c'est-à-dire si toutes ses feuilles sont des noeuds d'échec. Autrement il est dit *ouvert*.

Un noeud est dit *noeud d'inférence* si et seulement si ses descendants immédiats sont des noeuds d'échec.

Théorème. *Soit S un ensemble fini de clauses. Alors S est insatisfaisable si et seulement s'il existe un arbre sémantique fermé T pour S.*

Preuve.
Si)
Chaque branche fermée est un contre-modèle d'une clause de S, donc de S. Comme l'arbre sémantique T énumère *toutes* les interprétations et T est fermé, on en déduit que S est insatisfaisable.

Seulement si)
Supposons S insatisfaisable. Alors, aucune interprétation ne peut être un modèle de S, donc il ne peut pas y avoir de branche ouverte B_0 dans T. Autrement, B_0 ne falsifierait aucune clause de S et serait donc un modèle de S : contradiction. T est donc fermé. $\blacksquare$

Exemple. Soit l'ensemble de clauses :

$$S = \{C_1, C_2, C_3, C_4\}$$

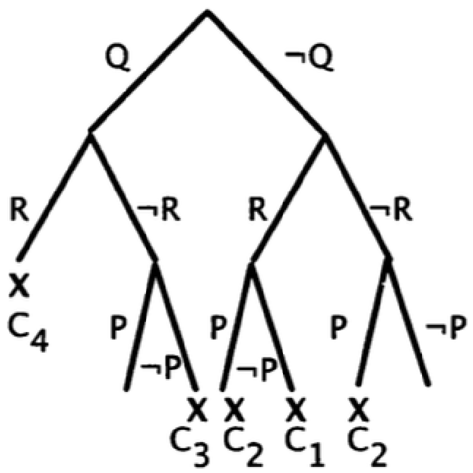
avec :

$$\begin{aligned} C_1 &: \quad P \vee Q \vee \neg R \\ C_2 &: \quad \neg P \vee Q \\ C_3 &: \quad P \vee \neg Q \vee R \\ C_4 &: \quad \neg Q \vee \neg R \end{aligned}$$

Alors,

$$B(S) = \{P, Q, R\}$$

et un arbre sémantique pour S est donné par :



On a donc trouvé parmi les huit interprétations possibles deux modèles :

$$\{Q, \neg R, P\}, \quad \{\neg Q, \neg R, \neg P\}$$

et six contre-modèles :

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