Chapitre 2: SYSTEME DE REFUTATION PAR RESOLUTION

1) Introduction : Utilisé dans les problèmes ayant un ensemble de fbf S à partir duquel nous voulons démontrer une fbf but W. Certains systèmes reposant sur la résolution sont conçus pour produire une démonstration par l'absurde ou <u>Réfutation</u>. La fbf but W est d'abord niée puis ajoutée à l'ensemble S. Ce nouvel ensemble de fbf $\{S, \neg W\}$ est transformé en clauses puis on utilise la résolution dans l'espoir de dériver une contradiction représentée par la clause vide Nil. (Toutes ces notions ont été vues dans le chapitre précédent). Rappelez vous de la démonstration par l'absurde vue au Lycée.

2) Exemple:

Soient les assertions suivantes et leurs fbf correspondantes:

1) Quiconque sait lire est instruit. $(\forall x)\ [lire(x) \Rightarrow ins(x)]$

2) Les dauphins ne sont pas instruits. $(\forall x)\ [d(x) \Rightarrow \neg\, ins(x)]$

3) *Certains dauphins sont intelligents.* $(\exists x)[d(x) \wedge int(x)]$

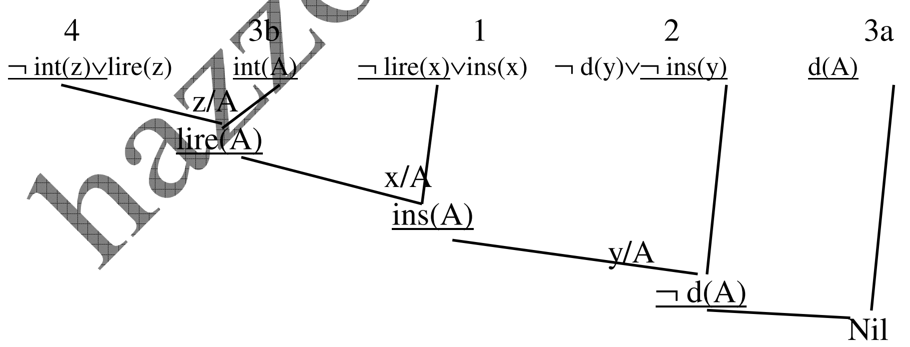
et on veut démontrer:

4) Certains êtres intelligents ne savent pas lire. $(\exists x)[int(x) \wedge \neg\, lire(x)]$

L'ensemble des clauses obtenu après transformations de $\{1, 2, 3, \neg 4\}$ (processus chapitre précédent):

1) $\neg$ lire(x) $\vee$ ins(x) 2) $\neg$ d(y) $\vee$ $\neg$ ins(y)
3a) d(A) 3b) int(A) A est une constante de Skolem
La négation du but à démontrer donne: 4) $\neg$ int(z) $\vee$ lire(z).

Pour montrer le but par réfutation, il faut générer des résolvantes à partir de l'ensemble des clauses 1, 2, 3a, 3b et 4 jusqu'à ce que Nil (contradiction) soit produit.



Un arbre de Réfutation utilisant la résolution

L'algorithme de base pour ce système peut être le suivant:

procédure résolution;

1) Clauses := ensemble des clauses de {S, $\neg$ But}

2) <u>Jusqu'à</u> ce que Nil $\in$ Clauses

3) <u>faire début</u>

*4) **<u>Sélectionner</u>** dans Clauses 2 clauses $\neq C_i$ et C_j qui peuvent être résolues. /* sur lesquelles on peut appliquer la résolution*/*

*5) **<u>calculer une</u>** résolvante r_{ij} de C_i et C_j*

6) Clauses := Clauses $\vee$ r_{ij}

7) <u>fin</u>

3) Stratégies de Contrôle pour les méthodes de résolution.

En regardant l'algorithme précèdent on remarque qu'il y a un choix à faire au niveau 4 et 5. Plusieurs stratégies de sélection de clauses ont été développées. Afin de se rappeler quelles résolutions ont été sélectionnées et éviter ainsi éviter de boucler, la stratégie de contrôle peut utiliser ce qu'on appelle: ***Graphe de dérivation.*** Une réfutation par résolution est représentée comme un arbre de réfutation (à l'intérieur du graphe) qui a une racine associé à Nil. La stratégie de contrôle construit un graphe de dérivation jusqu'à ce qu'un arbre de racine Nil soit produit.

Definition: On dit qu'une stratégie de contrôle pour un système de réfutation est **complète,** si son utilisation aboutit à une procédure qui découvre systématiquement toute contradiction nil (lorsqu'elle existe).

a) Stratégie en largeur d'abord: dans cette stratégie, toutes les résolvantes du 1er niveau son calculées d'abord puis on passe à ceux du 2eme etc .....

Une telle stratégie est **complète** mais inefficace.

b) Stratégie de l'ensemble support: au moins un parent de chaque résolvante est sélectionné parmi les **clauses résultantes de la négation du but ou parmi ses descendants. Stratégie complète** (à condition de produire toutes les résolvantes d'un niveau avant de passer à un autre). Chaque résolvante a un caractère de raisonnement en chaînage Arrière car elle utilise une clause qui vient du but.

c) Stratégie des unités sélectionner une clause à un **littéral unique** (appelé unité) comme parent au cours de la résolution.

d) Stratégie par rapport aux données: chaque résolvante a au moins un parent qui appartient à l'ensemble de **clauses initiales**. Cette stratégie est **non complète**.

e) Stratégie linéaire. Chaque résolvante à un parent qui: Soit se trouve dans l'ensemble initial, Soit est un ancêtre de l'autre parent.

f) Combinaison des stratégies: On peut combiner les différentes stratégies. Les combinaisons ensemble support + stratégie linéaire ou stratégie de résolution par rapport aux données sont courantes.

4) **Stratégie de simplification**

On peut parfois simplifier un ensemble de clauses en éliminant certaines (ou certain littéraux) des clauses. Ces simplifications sont telles que l'ensemble simplifié soit insatisfiable ssi l'ensemble original est insatisfiable :

a) Elimination des tautologies: Toute clause contenant un littéral et sa négation peut être éliminée.

Exemple: $p(x) \vee b(y) \vee \neg b(y) \vee$....... $p(f(a)) \vee \neg p(f(a)) \vee$ ....

b) Attachement procédural: il est possible et plus commode d'évaluer les valeurs de vérité des littéraux que d'inclure ces littéraux (ou leur négation) dans l'ensemble initial lorsque ceci est possible.

Exemple: *egal(7,3)* il suffit d'associer une interprétation c-a-d associer au prédicat *egal*, le programme informatique *egal*.

5) Systèmes de Question-Réponses utilisant la réfutation par résolution

Des problèmes consistent à démontrer que des formules contenant des variables existentielles sont conséquence logique d'un ensemble de fbf et découvrir les valeurs de ces variables.

Exemple 1: Soit le problème suivant: $(\exists x)$ W(x)} découle logiquement de S ?
Si oui trouver une instance de x

Exemple 2: Existe-t-il une séquence solution pour un certain Taquin à 9 cases.

On va décrire un processus qui permet d'extraire une instance adéquate d'une variable existentielle dans une fbf à partir d'une réfutation par résolution de cette fbf.

Exemple: Soit le problème: *« Si Farid va partout où Djamel va et si Djamel est à l'école, où est Farid ».* Ce problème spécifie 2 faits et une question:

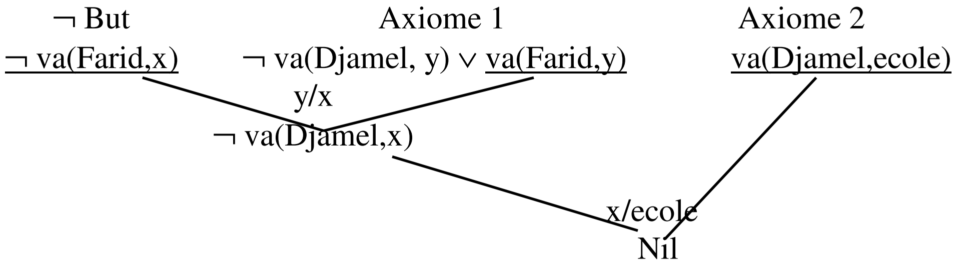
$(\forall x)$ *[va(Djamel,x)* $\Rightarrow$ *va(Farid, x)]* *va(Djamel,ecole)*

Pour la question « *où est Farid* »? on démontre que la fbf $(\exists x)$ *va(Farid,x)* découle logiquement des 2 fbfs précédentes et on trouve ensuite une instance de x.

La réfutation par résolution est obtenue de façon habituelle. Les fbfs (ensemble de départ et négation du but) sont transformées en clauses, ce qui nous donne:

Axiome 1: $\neg$ va(Djamel, y) $\vee$ va(Farid,y) Axiome 2: va(Djamel,ecole)
$\neg$ but: $\neg$ va(Farid,x).

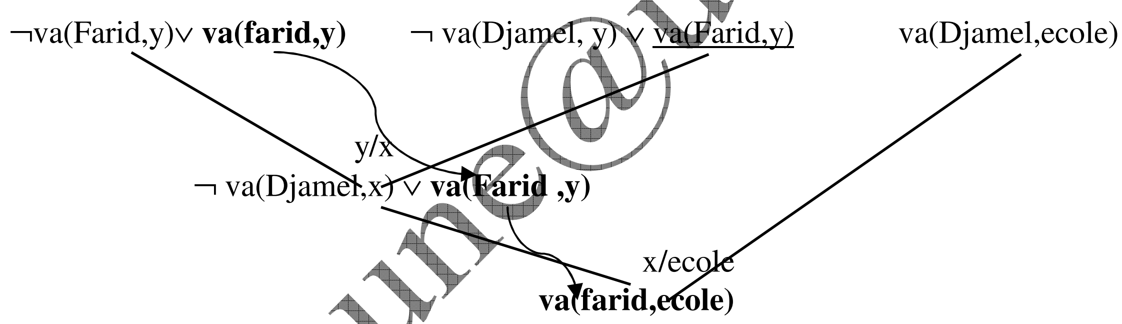
Un arbre de réfutation pour cet exemple est le suivant:



Nous devons maintenant extraire la valeur de la variable x. Le processus d'extraction est le suivant:

1) Ajouter à chaque clause qui découle de la négation du but, sa propre négation.
2) Accomplir les mêmes résolutions qu'auparavant jusqu 'à ce qu'une clause soit produite à la racine.
3) Utiliser la clause à la racine comme réponse.

Pour notre exemple, la négation de la clause résultante de la négation du but est en gras) (But est va(farid,y), sa négation est ¬va(farid,y) et sa négation est ¬¬va(farid,y)), c'est ce qui est gras sur l'arbre suivant :



Et donc en comparant le but *va(Farid,x)* avec **va(farid,ecole)** de la racine de cet arbre, on remarque que la valeur de x vaut « ecole ».

Le processus d'extraction de réponses.

La procédure précédente est une simple transformation de l'arbre de dérivation ayant Nil à la racine en un arbre de démonstration ayant une clause à la racine, en transformant chaque clause résultant de la négation du but en une tautologie.

Résume: *On peut résumer le processus d'extraction comme suit:*
1) Un arbre de réfutation par résolution est découvert par le processus de recherche.
2) De nouvelles variables sont substituées à toute fonction de Skolem apparaissant dans les ***clauses qui résultent de la négation de la fbf but. (uniquement les fonctions de skolem des clauses de ¬But).***
*3) Les clauses qui résultent de la négation de la fbf but sont transformées en tautologies en leur ajoutant leurs propres négations. (**uniquement les clauses de ¬But).***

4) Un arbre de démonstration modifié modélisant la structure de l'arbre de réfutation original est produit. Chaque résolution dans l'arbre modifié utilise un ensemble d'unification déterminé par l'ensemble d'unification utilisé par la résolution correspondante dans l'arbre de réfutation.
5) La clause à la racine de l'arbre modifié est la formule réponse qui contient les valeurs des variables recherchées.