Université Bretagne Sud
L3 Informatique

Logique
Travaux Dirigés - Partie 7

Ce septième TD est consacré à des compléments pour la Logique Propositionnelle.

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 **la méthode de résolution** *pour la LP0 dans l'annexe 1 du TD pour ne pas surcharger le cours.*

L'annexe 2 contient des développement autour de la notion de **stratégie** *(de preuve).*

Il est nécessaire d'étudier ces annexes avant d'aborder les exercices du TD.

Rappelons qu'on appelle **clause** *une disjonction de littéraux de la LP0 et que si tous les littéraux d'une clause sont positifs (respectivement négatifs) la clause est dite* **positive** *(respectivement* **négative***).*
Étant donné un ensemble de clauses, un littéral apparaissant dans une clause et dont la négation n'apparaît pas dans les autres clauses, est appellé un **littéral pur.**

Bon travail !

Exercice 1

Prouver la complétude pour la réfutation de la méthode de résolution.

Exercice 2

Prouver que dans une réfutation par résolution l'on peut éliminer les tautologies sans perdre la complétude réfutationnelle.

Exercice 3

Comment détecter par résolution qu'un ensemble de clauses est satisfaisable ?
Donner un exemple.

Exercice 4

Utiliser la méthode de résolution pour répondre aux questions suivantes.
a) Prouver que $\mathcal{S}$ est insatisfaisable :

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

b) Prouver que $\mathcal{S}$ est insatisfaisable :

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

c) $\mathcal{S}$ est-elle satisfaisable ou insatisfaisable ?

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

d) Prouver que $\mathcal{S}$ est insatisfaisable :

$$\mathcal{S} = \{\neg P, \neg R \Rightarrow W, Q \vee (\neg T \Rightarrow \neg P \wedge \neg Q), \neg P \Rightarrow (S \wedge \neg R), \neg Q, \neg S, \neg T, \neg R \Rightarrow Y\}$$

e) Prouver, d'abord comme vous auriez fait en ignorant la règle de résolution, et après par résolution, que le raisonnement suivant est correct :

$$A \wedge B \Rightarrow C \wedge D$$
$$E \wedge F \Rightarrow G$$
$$G \wedge D \Rightarrow H$$
$$A$$
$$B$$
$$F$$
$$E$$
$$\hline$$
$$H$$

Exercice 5

a) Donner une réfutation de l'ensemble de clauses S ci-dessous :

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

en utilisant la *stratégie d'entrée*.

b) La stratégie d'entrée est-elle *complète* ? Justifier.

Exercice 6

Comme l'objectif dans la méthode de résolution est de détecter une contradiction élémentaire (c'est-à-dire entre deux clauses unitaires) une bonne stratégie serait d'appliquer *toujours* la règle de résolution avec au moins l'une des clauses parent unitaire (si les deux sont unitaires et la résolution s'applique, on peut arrêter, on a produit $\square$).

On appellera cette stratégie la *stratégie unitaire*.

Est-elle complète ? Justifier.

Exercice 7

Prouver que tout ensemble S contenant les 2^n clauses (différentes), de longueur n, que l'on peut former à partir d'un ensemble de n symboles propositionnels est

insatisfaisable.

Exercice 8

a) Un ensemble de clauses ne contenant pas de clause positive ni de clause négative, peut-il être insatisfaisable ?

b) Comme corollaire de la réponse, prouver qu'un ensemble de clauses insatisfaisable contient (au moins) *une clause positive et une clause négative*.

Exercice 9

Spécifier, en utilisant la logique propositionnelle que l'on peut colorier une carte où figurent N pays en utilisant trois couleurs différentes, de telle façon qu'en utilisant une seule couleur par pays, deux pays frontaliers n'ont jamais la même couleur (il y aura des cartes où ce coloriage n'est pas possible).

Comme $N \in \mathbb{N}$ n'est pas spécifié, on donnera un *schéma* de la spécification.

Il s'agit d'une réduction au problème SAT. Le problème de la 3-coloriabilité est donc aussi en NP.

Quelle est la *taille* de la spécification ?

Exercice 10

Un ensemble de clauses *insatisfaisable* S est dit *minimalement insatisfaisable* si et seulement si pour tout $R \subset S$ (c'est-à-dire $R \subseteq S$ et $R \neq S$), R est satisfaisable.

a) Montrer que tout sous-ensemble d'un ensemble de clauses (et plus généralement d'un ensemble de fbf) satisfaisable est satisfaisable.

b) Montrer que tout sur-ensemble d'un ensemble de clauses (et plus généralement d'un ensemble de fbf) insatisfaisable est insatisfaisable.

c) Montrer qu'un ensemble de clauses minimalement insatisfaisable ne contient pas de littéral pur.

d) Un ensemble de clauses insatisfaisable ne contenant pas de littéral pur est-il nécessairement minimalement insatisfaisable ?

Exercice 11

a) Montrer que si un ensemble de clauses S est insatisfaisable, alors *il n'existe pas* d'interprétation falsifiant *toutes* les clauses de S.

b) Soit S un ensemble de clauses insatisfaisable et $\mathcal{I}$ une interprétation pour S. Montrer qu'il existe $S_1 \subset S$, $S_2 \subset S$, $S_1 \neq \emptyset$, $S_2 \neq \emptyset$, $S_1 \cap S_2 = \emptyset$ tels que $\mathcal{I}$ est un modèle de S_1 et $\mathcal{I}$ est un contre-modèle de S_2.

ANNEXE 1 : LA METHODE DE RESOLUTION DANS LA LP0

La méthode de résolution est l'une des plus utilisée en pratique (surtout dans sa version pour la logique du premier ordre).

Elle utilise une seule règle d'inférence ce qui fait qu'elle est particulièrement facile à implémenter, mais elle a besoin d'une forme normale : la forme clausale (ou fnc).

Nous commençons par quelques remarques :

– La méthode exige comme format d'entrée des ensembles de clauses (fnc). Ceci n'est pas une limitation parce que toute fbf de la LP0 peut être transformée en une fbf équivalente sous fnc.

– Une clause (ou un ensemble contenant une clause) est, par définition, satisfaisable.

– $P \wedge \neg P$ n'est pas une clause, bien que cette contradiction soit représentée par ce que l'on appelle la clause vide (notée $\square$). P est une clause unitaire et $\neg P$ en est une autre.

– On considère une fbf sous forme clausale indifféremment comme une fbf sous fnc ou comme un *ensemble* de clauses et les clauses comme des *ensembles* de littéraux.

Définition 1. *Soit $S = \{C_1, \ldots, C_n\}$ un ensemble de clauses. Un ensemble de littéraux M est un modèle de S si et seulement si :*
Si $L \in M$, alors $L^c \notin M$ et :

$$C_k \cap M \neq \emptyset \quad (0 \leq k \leq n)$$

Cette définition peut être exprimée en français en disant : pour évaluer à V un ensemble de clauses il faut évaluer à V toutes ses clauses. On ne peut pas évaluer un littéral à V et à F simultanément. Pour évaluer à V une clause il faut et il suffit d'évaluer à V au moins l'un de ses littéraux.
Comme conséquence, si tous les littéraux d'une clause sont évalués à F , la clause sera évaluée à F.

Définition 2. *Soient données deux clauses contenant des littéraux complémentaires :* $C_1 : L \vee \alpha$ *et* $C_2 : L^c \vee \beta$ *(α et β sont des clauses c'est-à-dire des disjonctions de littéraux.*
La règle d'inférence nommée **règle de résolution** *est définie comme suit :*

$$R : \frac{L \vee \alpha \quad L^c \vee \beta}{\alpha \vee \beta}$$

On écrira aussi :

$$R(C_1, C_2) = C$$

ou, si l'on veut mettre en évidence les littéraux complémentaires :

$$R(C_1, C_2, L, L^c) = C$$

où

$$C = (C_1 \setminus \{L\}) \cup (C_2 \setminus \{L^c\})$$

La clause $C : \alpha \vee \beta$ est appelée **la résolvante** de C_1 et C_2.
C_1 et C_2 sont appelées **clauses parentes**.

C est une *conséquence logique* de $\{C_1, C_2\}$ (de $C_1 \wedge C_2$) mais *n'est pas* équivalente à $C_1 \wedge C_2$: tout modèle de α (resp. β) est un modèle de la résolvante, mais pas nécessairement des deux clauses parentes.

Dans le cas où α et β ne contiennent pas de littéraux :

$$\frac{L \quad L^c}{\square}$$

où $\square$, dénotant la *contradiction*, est appelé *la clause vide*.

On utilisera également la règle dite *d'absorption* :

$$Abs : \frac{\alpha \vee L \vee \beta \vee L \vee \beta \vee \gamma}{\alpha \vee L \vee \beta \vee \gamma}$$

où α, β, γ sont des clauses.

Cette règle revient à considérer les clauses comme des ensembles de littéraux ou, ce qui est équivalent, à utiliser les propriétés d'associativité, commutativité et d'absorption du $\vee$: $(\alpha \vee \beta) \vee \gamma \Leftrightarrow \alpha \vee (\beta \vee \gamma)$, $L \vee \alpha \Leftrightarrow \alpha \vee L$, $L \vee L \Leftrightarrow L$ respectivement.

Remarque importante. On ne doit pas confondre la clause vide avec un ensemble de clauses vide. La première est insatisfaisable et le second est satisfaisable (il ne contient rien, il ne peut donc pas contenir une contradiction).
On peut en donner une preuve plus formelle. Par l'absurde : si $\emptyset$ est insatisfaisable, alors (par exemple) $\{A \vee B\} = \emptyset \cup \{A \vee B\}$ serait insatisfaisable, puisque tout sur-ensemble d'un ensemble contradictoire es lui-même contradictoire. Mais $\{A \vee B\}$ est satsifaisable (modèles $\{A\},\ \{B\},\ \{A, B\}$) : contradiction. Donc $\emptyset$ est satisfaisable.

Pour une règle non-déterministe comme la résolution il est utile de définir un *opérateur* permettant de capturer toutes les résolvantes que l'on peut obtenir en **appliquant la règle de résolution de toutes les façons possibles**.

Définition 3 (opérateur $\mathcal{R}$). *Soit S un ensemble fini de clauses. On définit :*

$$\begin{aligned}
\mathcal{R}(S) &= S \cup \{R(C_1, C_2) | C_1, C_2 \in S\} \\
\mathcal{R}^0(S) &= S \\
\mathcal{R}^{n+1}(S) &= \mathcal{R}(\mathcal{R}^n(S)) \quad (n \geq 0) \\
\mathcal{R}^*(S) &= \bigcup_{n \geq 0} \mathcal{R}^n(S)
\end{aligned}$$

Remarque. Pour un ensemble fini satisfaisable de clauses propositionnelles S, il existe n tel que

$$\mathcal{R}^*(S) = \mathcal{R}^n(S)$$

(voir l'exercice 3 de ce TD).

Définition 4 (un système déductif pour la résolution).

$\mathcal{S}_R =< \mathcal{L}, \mathcal{R}, \mathcal{A} >$

$\mathcal{L}$: clauses et ensembles de clauses

$\mathcal{R} = \{R, Abs\}$

$\mathcal{A} = \emptyset$

On dit que la clause C **se déduit par résolution** de l'ensemble de clauses S, noté :

$$S \vdash_{\mathcal{S}_R} C$$

ou

$$S \vdash_{\mathcal{R}} C$$

si et seulement s'il existe une suite finie $C_1, \ldots, C_k$ telle que :

$$\begin{aligned}
C_k &= S \\
C_{i+1} &= R(C_m, C_n) \quad (0 \leq i \leq k-1) \\
C_m, C_n &\in S \cup \{C_1, \ldots, C_i\}
\end{aligned}$$

La suite $C_1, \ldots, C_k$ s'appelle une **déduction** à partir de S.

Si $C = \square$, elle s'appelle une **réfutation** de S.

Pour la méthode de résolution, la correction et la complétude (pour la réfutation) s'énoncent :

– Correction : si $S \vdash_{\mathcal{R}} \square$ alors S est insatisfaisable (contradictoire) ;

– Complétude pour la réfutation : Si S est insatisfaisable (contradictoire), alors $S \vdash_{\mathcal{R}} \square$.

L'expression *complétude pour la réfutation* appliquée à la méthode de résolution s'explique facilement en constatant, par exemple, que :

$$A \nvdash_{\mathcal{R}} A \vee B$$

bien que $A \vee B$ soit une conséquence logique de A.

En revancje, si l'on nie $A \vee B$, on obtient l'ensemble de clauses $\{A, \neg A, \neg B\}$ et immédiatement la clause $\square$ en appliquant la résolution entre A et $\neg A$.

Théorème 1. *Soit S un ensemble satisfaisable de clauses et M un modèle de S. Si $S \vdash_{\mathcal{R}} C$, alors $M \cap C \neq \emptyset$.*

Preuve. Si C est obtenu par application de la règle de résolution, alors il existe $C_1 \in S$, $C_2 \in S$ et un littéral $L \in C_1$ tel que $L^c \in C_2$ et

$$\begin{aligned} C &= R(C_1, C_2, L, L^c) \\ &= (C_1 \setminus \{L\}) \cup (C_2 \setminus \{L^c\}) \end{aligned}$$

Puisque M est un modèle de S, M est un modèle de toutes les clauses de S, donc :

$$M \cap C_1 \neq \emptyset \text{ et } M \cap C_2 \neq \emptyset$$

Trois cas sont à considérer :

i) Si $L \in M$ et puisque M est un modèle de C_1 et de C_2, il existe $K \in C_2 \setminus \{L^c\}$ tel que $K \in M$. Donc, par définition de la règle R : $K \in C$, donc $M \cap C \neq \emptyset$.

ii) Si $L^c \in M$, alors il existe $N \in C_1 \setminus \{L\}$ et $N \in M$.

Par définition de la règle R, $N \in C$, donc $M \cap C \neq \emptyset$.

iii) Si $L \notin M$ et $L^c \notin M$, alors M ne dépend pas des valeurs données à L et à L^c, donc $M \cap C \neq \emptyset$.
La preuve est complétée en appliquant la définition de déduction et par induction sur la longueur de la déduction. ■

On peut énoncer ce théorème avec une autre notation : *Si* $C \in \mathcal{R}^*(S)$ *et* $\models_{\mathcal{I}} S$ *alors* $\models_{\mathcal{I}} C$.
Nous avons utilisé la contraposée, c'est-à-dire :

$$\text{si } \not\models_{\mathcal{I}} C \text{ alors } \not\models_{\mathcal{I}} S$$

pour fermer des branches dans les arbres sémantiques. Elle est aussi utilisée dans des SAT-solvers (c'est-à-dire des logiciels permettant de résoudre le problème SAT) très performants pour élaguer l'espace de r echerche. En effet, quand on vérifie que l'interprétation (en général partielle, mais qui peut suffire à évaluer certaines clauses) proposée falsifie une clause (d'entrée ou déduite par résolution) , ce n'est pas la peine de continuer dans cette voie.

Corollaire 1 (correction de la résolution). *La méthode de résolution est correcte.*

Preuve. Si $S \vdash_{\mathcal{R}} \square$ et S satisfaisable, alors $\square$ serait satisfaisable d'après le théorème 1. Or c'est impossible. Donc, S est insatisfaisable. ■

Exemple. Il s'agit de trouver, en utilisant la méthode de résolution que l'ensemble de clauses :

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

est insatisfaisable et ceci en vue de concevoir dans le futur un *programme* pour le faire.

Le problème principal qui se présente à nous est celui de gérer le *non-déterminisme* (c'est-à-dire les *choix* pour l'application de la règle de résolution).

Avant d'analyser les bons choix d'application de la règle et pour être sûrs que la méthode marchera dans tous les cas, on décide d'appliquer la « méthode de la force brute », c'est-à dire qu'on applique *tous les choix* pour une certaine énumération et l'on regarde si l'on a (dans l'ensemble des clauses du problème auquel on ajoute les clauses déduites) deux clauses unitaires complémentaires (la seule contradiction que l'on puisse toujours détecter de façon mécanique).

La notation à droite des formules :

$$(i,j)-(k,l) \quad (1 \leq i \leq 4, 1 \leq j \leq 3, 2 \leq K \leq 12, 1 \leq l \leq 2)$$

signifie : on applique la règle de résolution en choisissant le littéral à la position j (de gauche à droite) de la clause numérotée i et son complémentaire à la position l de la clause numérotée k.

1	$\neg P \vee \neg Q \vee R$	
2	$P \vee R$	
3	$Q \vee R$	
4	$\neg R$	
5	$\neg Q \vee R$	$(1,1)-(2,1)$
6	$\neg P \vee R$	$(1,2)-(3,1)$
7	$\neg P \vee \neg Q$	$(1,3)-(4,1)$
8	P	$(2,2)-(4,1)$
9	Q	$(3,2)-(4,1)$
10	$\neg Q \vee R$	$(1,1)-(8,1)$
11	$\neg P \vee R$	$(1,2)-(9,1)$
12	R	$(2,1)-(6,1)$
13	$\square$	$(4,1)-(12,1)$

On remarque que l'on peut déduire la même clause plus d'une fois (par exemple 5 et 10 ; 6 et 11).

En analysant la réfutation après l'avoir obtenue, on constate que seules 6 et 12 étaient nécessaires pour détecter la contradiction. Vous semble-t-il possible de savoir qu'il y a contradiction avant de trouver la réfutation ? ■

ANNEXE 2 : PROBLEMES, STRATEGIES ET ENONCES

Nous avons vu différentes procédures de preuve appelées aussi calculs (systèmes formels ou « à la Hilbert », tableaux, résolution, etc.). La question se pose alors naturellement : « existe-t-il une procédure de preuve qui soit uniformément (c'est-à dire pour tout problème) meilleure (par exemple en nombre de pas) que les autres ? » . La réponse (à laquelle on peut s'attendre) est non.

Une notion naturellement associée au problème du non-déterminisme est celle de *stratégie*. Ce mot a, au sens technique, une signification très similaire à celle qu'il a dans le langage courant.

Une stratégie est une règle pour faire les choix d'application d'une(des) règle(s) d'inférence non-déterministe(s) dans le but d'obtenir un certain objectif, en général, en diminuant le nombre de choix , donc l'espace de recherche (c'est-à-dire l'ensemble de toutes les applications avant d'arriver au résultat cherché ou d'arrêter). Parfois on cherche à diminuer le nombre des pas menant à la solution.

Dès le début on s'est rendu compte de l'impossibilité d'attaquer des problèmes intéressants de preuves automatisées sans associer des stratégies aux calculs (les calculs étant des ensembles de règles d'inférence, règles non-déterministes).

Bien entendu, on s'est posé la question concernant l'existence de la meilleure façon de gérer le non-déterminisme, via une procédure (stratégie) parfaite, c'est-à dire n'engendrant jamais de formule redondante et ce pour la résolution de n'importe quel problème.

On peut montrer (en utilisant des résultats connus de la théorie de la calculabilité) qu'il n'existe pas de procédure complète pour la réfutation (par exemple la résolution)

parfaite, c'est-à-dire n'engendrant pas de formule (clause) non nécessaire à la preuve (réfutation).

On peut définir de façon abstraite une *procédure de preuve* comme un couple (T, Σ) où T est un système formel et Σ une stratégie pour T.

Il est intéressant de remarquer qu'en général dans les ouvrages de logique on parle de *systèmes de preuve (proof systems)* en les identifiant à T et sans faire mention de la stratégie.

Dans le but de définir la notion abstraite de *preuve automatisée*, on définit la notion de *graphe de preuve*, qui suit naturellement la définition de l'opérateur $\mathcal{R}$: Les formules ont un niveau, qui n'est pas unique (d'où l'utilisation de graphe au lieu d'arbre) et qui est défini de façon standard comme 1 de plus que celui des formules dont elle est la conséquence directe. Autrement dit, si l'on utilise la résolution, le niveau de la résolvante est 1 de plus que celui de ses parents (les clauses d'entrée ont le niveau 0).

Le *problème du démonstrateur* pour un triplet :

$$(S_0, \Gamma, F)$$

est défini comme celui d'engendrer en utilisant une stratégie Σ un ensemble de formules F, avec :
S_0 : ensemble de formules de départ ;
Γ : ensemble des règles d'inférence et

$$\Gamma^*(S_j) = \bigcup_{i \geq j} \Gamma(S_i)$$

F : ensemble de formules sous-ensemble des conséquences de S_0, c'est-à-dire que $F \subseteq \Gamma^*(S_0)$;

et :

$$\Sigma : 2^G \to 2^G$$

où G est le graphe de preuve.

En dépliant le graphe sous forme d'arbre on associe à chaque nœud une dérivation, ce qui permet d'associer à une feuille une mesure de la dérivation avec la stratégie Σ.

Nous pouvons modifier légèrement la définition pour permettre de caractériser aussi les assistants de preuve, en particulier les vérificateurs de preuve : Un *démo-vérificateur abstrait* est un 5-uplet :

$$(S_0, \Gamma, F, P, \Sigma)$$

où P est l'ensemble des formules de la prétendue preuve (si $P = \emptyset$, on a un démonstrateur totalement automatisé, si P contient tous les pas d'une preuve nous avons un vérificateur, si l'on en donne, par exemple, certains lemmes on a un assistant de preuve ou démonstrateur interactif).

Nous avons inclus la stratégie qui n'est pas nécessairement uniforme : on peut penser Σ comme un ensemble de stratégies. La théorie dans laquelle la preuve est faite est incluse dans S_0.

Définition 1. *Une stratégie st pour la résolution est dite* **complète** *si et seulement si pour tout ensemble de clauses S :*

$$\textit{Si } S \vdash_{\mathcal{R}} \square \textit{ alors } S \vdash_{\mathcal{R}+st} \square$$

Exemple (stratégie d'entrée). Un exemple de stratégie pour la résolution est la *stratégie d'entrée* : étant donné un ensemble de clauses S, la résolution est appliquée toujours avec au moins une clause dans S (l'ensemble de clauses d'entrée).

Définition 2 (complexité d'une preuve, d'une méthode). *La complexité d'une preuve (réfutation) par résolution d'un ensemble de clauses S, notée $Comp_{\mathcal{R}}(S)$ est le nombre de clauses différentes de la preuve (réfutation) de S.*
La complexité de la méthode de résolution sur des ensembles de clauses de cardinalité n, notée $Comp_{\mathcal{R}}(n)$ est définie comme :

$$Comp_{\mathcal{R}}(n) = \max_{card(S)=n} \min Comp_{\mathcal{R}}(S).$$

Le problème de la complexité des preuves de la LP0 a été particulièrement étudié depuis la fin des années 1960.