

Étude du langage de programmation prologue

Julie Parreaux

2018-2019

Référence du développement : Stern [1]

Leçons où on présente le développement : 916 (Logique propositionnelle) ; 918 (Preuve en logique du premier ordre).

1 Introduction

Prolog est un langage logique qui nous permet de résoudre un but à l'aide d'une série de faits et d'une série d'implication. Il a été introduit pour l'intelligence artificielle : il permet de simuler une forme d'intelligence. Pour cela, on se restreint à des formules du type de Horn : le langage Prolog peut alors être vu comme un ensemble de clause de Horn. L'exécution est alors une réfutation (en logique du premier ordre ou de la logique propositionnelle) d'une clause de Horn.

Une clause de Horn est de la forme $x_1 \wedge \dots \wedge x_n \Rightarrow y$ soit $\neg x_1 \vee \dots \vee \neg x_n \vee y$. En Prolog, une clause de Horn s'écrit sous la forme $L \rightarrow P_1 \dots P_k$ où L est la tête et $P_1 \dots P_k$ la queue de la règle représente la connaissance à l'aide de $L \rightarrow$ fait et $> Q_1 \dots Q_r$ est le but. Le langage Prolog repose sur la réfutation de type LD de $\{Q_1, \dots, Q_r\}$ à l'aide de la connaissance. De plus, si la résolution est possible, on est capable de donner toutes les solutions. Donnons les étapes de la résolution de la réfutation :

- Effacement des buts : on cherche la règle $L \rightarrow P_1 \dots P_k$ telle que L s'identifie à Q_1 . On cherche donc un unificateur σ ce qui nous donne $\{Q_2\sigma, \dots, Q_r\sigma\}$. La correction est assurée par le théorème que nous devons dénombrer, on parle alors de résolution SLR.
- Arbre d'effacement et backtracking. On énumère l'ensemble des unificateurs pour Q_1 et L et on les explore les uns à la suite des autres (on a une infinité à gérer).
- L'introduction du cut (qui permet de réduire le backtracking : les choix sont fixes) donne un langage à part entière.

Remarques sur le développement

Le développement donne le fonctionnement de Prolog selon la logique de base. Pour les deux logiques le développement s'articule de la même façon.

1. Définition du cadre.
2. Étude de la résolution dans le cadre des clauses de Horn.
3. Définition d'une preuve.
4. Existence d'une preuve LD pour toute clause de départ.

2 Prolog dans le cadre de la logique propositionnelle

Définition. Soit C un ensemble de clauses. On appelle preuve par résolution linéaire une suite c_0, \dots, c_n de clauses telles que c_0 soit élément de C et telle que $c_i, \forall 1 \leq i \leq n$, soit obtenue comme résolvante de c_{i-1} et d'une clause de $c \cup \{c_0, \dots, c_{i-2}\}$ que l'on notera l_{i-1} .

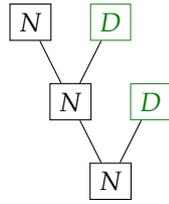


FIGURE 1 – Arbre d'une LD-réfutation.

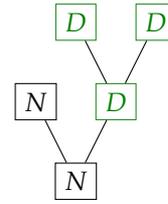


FIGURE 2 – Configuration élémentaire.

Définition. On appelle clause de Horn une clause dont au plus un littéral est une variable propositionnelle. Une clause de Horn possédant une variable propositionnelle est appelée une clause définie. Sinon une clause de Horn sans aucune variable propositionnelle est appelée une clause négative ou but.

Proposition. Soit C un ensemble de clauses de Horn et soit C_0 le sous-ensemble formé des clauses définies, alors

1. C_0 est non contradictoire ;
2. Si C est contradictoire, il existe une clause g négative dans C telle que $C_0 \cup \{g\}$ soit contradictoire.

Démonstration. 1. La résolvente de deux clauses définies est aussi définie (car on supprime uniquement un littéral positif donc il en reste un littéral positif dans la résolvente).

2. C est contradictoire. Étudions les cas de la résolution :
 - La résolution de deux clauses négatives est impossible.
 - La résolution d'une clause négative et d'une clause positive donne une clause négative (on supprime un littéraux positifs : il n'en reste plus).
 - La résolvente de deux clauses définies est aussi définie (car on supprime uniquement un littéral positif donc il en reste un littéral positif dans la résolvente).

Donc l'arbre de résolution pour C ne contient que des clauses de Horn (étude précédente : on ne pas obtenir plus d'un littéral positif).

Si l'arbre de C contient une racine négative :

- il existe une unique branche négative ;
- tous les autres sommets sont des clauses définies.

En particulier, c'est vrai pour les arbres de réfutation. On pose alors g l'unique feuille contenant une clause négative.

□

Définition. Une preuve par LD-résolution (L pour linéaire et D pour définie) est une preuve par réfutation dont la prémisse est négative et les entrées sont positives. Les arbres sont donc de la forme (Figure 1).

Théorème. Soit C un ensemble de clause de Horn définies, g une clause négative telle que $C \cup \{g\}$ soit contradictoire, alors il existe une réfutation de $C \cup \{g\}$ par LD-résolution.

Démonstration. On se donne un arbre de réfutation de $C \cup \{g\}$ qui admet une unique branche négative (existe par la proposition).

Si cet arbre est un arbre de réfutation LD-résolution alors on a le résultat.

Sinon il existe un sommet non terminal tel qu'il est étiqueté par des clauses définies (D) et que ces fils également. Prenons le sommet de profondeur minimal vérifiant ces propriétés, on obtient ainsi une configuration élémentaire (Figure 2). Par construction, on remarque qu'il est le fils d'un noeud étiqueté par une clause négative (N).

On prend deux variables p et q autorisant les deux réfutations. On peut trouver des clauses négatives g_1, g_2, g_3 telles que $\neg q \notin g_1$ et $\neg p \notin g_3$ ce qui nous permet d'étiqueté cette configuration élémentaire (Figure 3).

On remarque qu'on obtient également $g_1 \cup g_2 \cup g_3$ en résolvant $\{\neq q\} \cup g_1$ avec $\{q\} \cup \{\neq p\} \cup g_3$ et leur résolvente avec $\{p\} \cup g_2$ ce qui nous donne l'arbre (Figure 4). On a ainsi éliminer un sommet non terminal étiqueté par une clause définie.

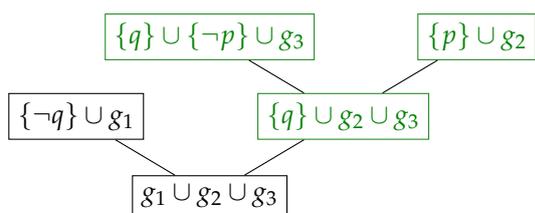


FIGURE 3 – La configuration élémentaire de notre preuve en logique propositionnelle.

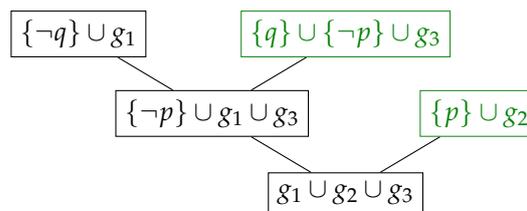


FIGURE 4 – La transformation en réfutation LD de notre configuration élémentaire en logique propositionnelle.

En itérant ce procédé sur l'arbre, comme le nombre des défaut de linéarité (c'est le nombre de sommet non terminal étiqueté par des clauses définies) diminue à chaque étape, on conclut la preuve. \square

3 Prolog dans le cadre de la logique du premier ordre

Dans le cadre de la logique du premier ordre la preuve repose sur les même argument. En effet, la règle de résolution au premier ordre demande une unification qui ne touche pas à la positivité ou à la négativité d'un littéral puisque c'est uniquement une substitution d'une variable (on ne touche pas aux différentes opérations autour de celle-ci). La preuve est rigoureusement la même en ajoutant cette notion d'unification.

Définition. Soit C un ensemble de clauses. On appelle preuve par résolution linéaire une suite c_0, \dots, c_n de clauses telles que c_0 soit élément de C et telle que $c_i, \forall 1 \leq i \leq n$, soit obtenue comme résolvente de c_{i-1} et d'une clause de $c \cup \{c_0, \dots, c_{i-2}\}$ que l'on notera l_{i-1} .

Définition. On appelle clause de Horn une clause dont au plus un littéral est une variable propositionnelle. Une clause de Horn possédant une variable propositionnelle est appelée une clause définie. Sinon une clause de Horn sans aucune variable propositionnelle est appelée une clause négative ou but.

Proposition. Soit C un ensemble de clauses de Horn et soit C_0 le sous-ensemble formé des clauses définies, alors

1. C_0 est non contradictoire ;
2. Si C est contradictoire, il existe une clause g négative dans C telle que $C_0 \cup \{g\}$ soit contradictoire.

Démonstration. Comme l'unification est une substitution syntaxique des variables (sans toucher à leur signe), on peut en déduire que la réfutation n'enlève qu'une seule variable positive dans l'union des deux clauses unifiées.

1. La résolvente de deux clauses définies est aussi définie (car on supprime uniquement un littéral positif donc il en reste un littéral positif dans la résolvente).
2. C est contradictoire. Étudions les cas de la résolution :
 - La résolution de deux clauses négatives est impossible.
 - La résolution d'une clause négative et d'une clause positive donne une clause négative (on supprime un littéraux positifs : il n'en reste plus).
 - La résolvente de deux clauses définies est aussi définie (car on supprime uniquement un littéral positif donc il en reste un littéral positif dans la résolvente).

Donc l'arbre de résolution pour C ne contient que des clauses de Horn (étude précédente : on ne pas obtenir plus d'un littéral positif).

Si l'arbre de C contient une racine négative :

- il existe une unique branche négative ;
- tous les autres sommets sont des clauses définies.

En particulier, c'est vrai pour les arbres de réfutation. On pose alors g l'unique feuille contenant une clause négative.

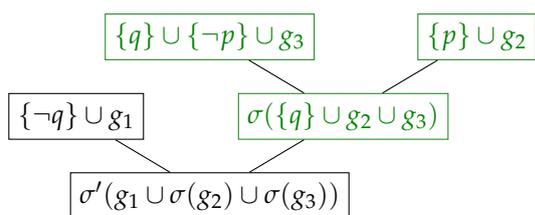


FIGURE 5 – La configuration élémentaire de notre preuve en logique du premier ordre.

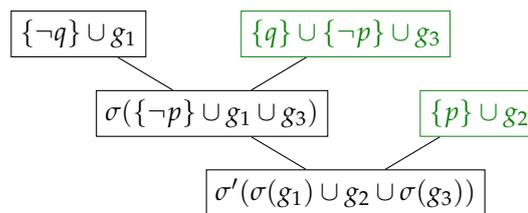


FIGURE 6 – La transformation en réfutation LD de notre configuration élémentaire en logique du premier ordre.

□

Définition. Une preuve par LD-résolution (L pour linéaire et D pour définie) est une preuve par réfutation dont la prémisse est négative et les entrées sont positives. Les arbres sont donc de la forme (Figure 1).

Théorème. Soit C un ensemble de clause de Horn définies, g une clause négative telle que $C \cup \{g\}$ soit contradictoire, alors il existe une réfutation de $C \cup \{g\}$ par LD-résolution.

Démonstration. On se donne un arbre de réfutation de $C \cup \{g\}$ qui admet une unique branche négative (existe par la proposition).

Si cet arbre est un arbre de réfutation LD-résolution alors on a le résultat.

Sinon il existe un sommet non terminal tel qu'il est étiqueté par des clauses définies (D) et que ces fils également. Prenons le sommet de profondeur minimal vérifiant ces propriétés, on obtient ainsi une configuration élémentaire (Figure 2). Par construction, on remarque qu'il est le fils d'un noeud étiqueté par une clause négative (N).

On prend deux variables p et q autorisant les deux réfutations via leur unifications respectives. On peut trouver des clauses négatives g_1, g_2, g_3 telles que $\neg q \notin g_1$ (après unification des termes) et $\neg p \notin g_3$ (après unification des termes) ce qui nous permet d'étiqueté cette configuration élémentaire (Figure 5).

On remarque qu'on obtient également $g_1 \cup g_2 \cup g_3$ en résolvant $\{ \neq q \} \cup g_1$ avec $\{ q \} \cup \{ \neq p \} \cup g_3$ et leur résolvante avec $\{ p \} \cup g_2$ ce qui nous donne l'arbre (Figure 6). On a ainsi éliminer un sommet non terminal étiqueté par une clause définie.

En itérant ce procédé sur l'arbre, comme le nombre des défaut de linéarité (c'est le nombre de sommet non terminal étiqueté par des clauses définies) diminue à chaque étape, on conclut la preuve. □

Références

- [1] J. Stern. *Fondements mathématiques de l'informatique*. Ediscience international, 1990.