Sous-classes décidables de la logique du premier ordre

Documents autorisés (en particulier le poly).

Les questions sont annotées par un niveau de difficulté, variant de (0) (facile) à (3) (difficile).

En comptant 4 minutes passées par problème facile, 8 minutes par problème de difficulté (1), 12 minutes par problème de difficulté (2), et 16 minutes par problème de difficulté (3), vous aurez besoin de 1h52 pour tout faire, ce qui vous laisse 8 minutes pour lire l'énoncé. Toutes les parties sont indépendantes.

Le but de ce problème est d'étudier quelques classes de formules du premier ordre, définies par des critères syntaxiques, dont l'insatisfiabilité est décidable, contrairement à la classe de toutes les formules du premier ordre.

Partie I : classe $\exists^*\forall^*$, dite de Bernays-Schönfinkel (trivial)

On considère dans cette partie la classe des formules closes F de la forme :

$$\exists x_1, \ldots, x_m \cdot \forall y_1, \ldots, y_n \cdot G$$

où G est une formule (i) sans quantificateur, et (ii) sur un langage ne contenant pas d'autre symboles de fonction que des constantes. On note cette classe $\exists^*\forall^*$.

1. (0) Skolémiser F; montrer que l'univers de Herbrand de la skolémisée est fini.
2. (0) En déduire que le problème :

   ENTRÉE : une formule F de $\exists^*\forall^*$.

   QUESTION : F est-elle insatisfiable ?

   est décidable. (Un algorithme stupide suffira.)

Partie II : classe monadique (facile)

On considère dans cette partie la classe des formules closes F telles que (i) les symboles de prédicats P apparaissant dans F sont d'arité 1, et (ii) il n'y a aucun symbole de fonction dans F (en particulier, pas de constante).

Cette classe est notée **Mon**, et est appelée la classe des formules *monadiques*.

1. (1) Montrer sur un exemple que la skolémisée de F peut avoir un univers de Herbrand infini. En déduire que la méthode de preuve de décidabilité de la partie I ne s'applique pas en général ici.
2. (1) Fixons une formule close monadique F, soit I une interprétation de domaine D, et soient $P_1, \ldots, P_n$ les symboles de prédicats apparaissant dans F. On définit la relation d'équivalence $\equiv$ sur D par :

   $$v \equiv w \text{ si et seulement si } I(P_i)(v) = I(P_i)(w) \text{ pour tout } i, 1 \leq i \leq n.$$

   Rappelons que l'ensemble quotient $D/\equiv$ est l'ensemble des classes d'équivalence $\overline{v} = \{w \in D \mid w \equiv v\}$ d'éléments v de D. On définit l'interprétation quotient $I/\equiv$ sur le domaine quotient $D/\equiv$ par :

   $$(I/\equiv)(P_i)(c) = \top \quad \Leftrightarrow \quad \exists v \in c \cdot I(P_i)(v) = \top$$

   pour tout $c \in D/\equiv$. (Noter qu'on n'a pas besoin de définir l'interprétation des symboles de fonction, car il n'y a pas de symbole de fonction.)

   Montrer que $(I/\equiv)(P_i)(\overline{v}) = I(P_i)(v)$ pour tout $v \in D/\equiv$. (Autrement dit, $(I/\equiv)(P_i)(c)$ ne dépend pas du choix du représentant v dans c.)

3. (2) Montrer que, si I est un modèle de F, alors $I/\equiv$ est aussi un modèle de F. Indication : on montrera que, pour toute formule monadique G dont les prédicats sont parmi $P_1, \ldots, P_n$, pour toute valuation ρ de l'ensemble des variables dans D, $[G]I\rho = [G](I/\equiv)\overline{\rho}$, où $\overline{\rho}$ est une valuation de l'ensemble des variables dans $D/\equiv$ à trouver.

4. (1) Montrer que $D/\equiv$ est toujours un ensemble fini, isomorphe à un ensemble de parties de $\{1, \ldots, n\}$. (On pourra considérer la fonction f qui à toute valeur $v \in D$ associe la partie $f(v)$ de $\{1, \ldots, n\}$ des i tels que $I(P_i)(v) = \top$.)

5. (0) Déduire des questions précédentes que la satisfiabilité des formules closes monadiques est décidable.

Ce résultat est dû à Löwenheim, qui a en fait prouvé que la classe reste décidable même si on ajoute un prédicat binaire d'égalité et la quantification d'ordre 2 sur les prédicats. La preuve ci-dessus est essentiellement la version simplifiée due à Ackermann.

Partie III : classe $\forall\exists^*$, dite d'Ackermann (ça se corse un peu)

On considère maintenant la classe des formules closes F de la forme :

$$\forall x \cdot \exists y_1, \ldots, y_n \cdot G$$

où G est une formule (i) sans quantificateur, et (ii) sur un langage ne contenant pas de symbole de fonction (en particulier, pas de constante). On note cette classe $\forall\exists^*$. La particularité de cette classe est qu'elle ne permet la quantification universelle que sur une variable.

1. (0) Soit $\hat{F}$ une forme clausale (et donc préalablement skolémisée) comme dans le cours. Montrer que toute clause C de $\hat{F}$ a les propriétés suivantes :

   (i) C contient au plus une variable libre x;

   (ii) tous les atomes de C sont de la forme $P(t_1, \ldots, t_n)$, où chaque t_i est soit x, soit de la forme $f(x)$.

2. (1) On rappelle que la règle de résolution *ordonnée* est la composée des règles de *factorisation ordonnée*:

$$\frac{C \vee A \vee B}{C\sigma \vee A\sigma} \quad \frac{C \vee \neg A \vee \neg B}{C\sigma \vee \neg A\sigma}$$

   où σ est le mgu de A et B, et A et B sont *maximaux* par rapport à C, au sens où A et B sont supérieurs à ou incomparables avec chaque atome de C; et d'une règle de *résolution binaire ordonnée* :

$$\frac{C \vee A \qquad \neg B \vee C'}{C\sigma \vee C'\sigma}$$

   où σ est le mgu de A et B, les deux prémisses sont supposées renommées de sorte qu'elles n'ont aucune variable libre en commun, et A est maximal par rapport à C, B est maximal par rapport à C'. "Supérieur à" est dans cette définition, un ordre strict (irréflexif, transitif) $\succ$ *stable*, c'est-à-dire tel que $A \succ B$ implique $A\sigma \succ B\sigma$ pour toute substitution σ.

   On définit la *profondeur* $d(A)$ par $d(x) = 0$, $d(f(t)) = 1 + d(t)$, $d(P(t_1, \ldots, t_n)) = \max(d(t_1), \ldots, d(t_n))$ (le max étant pris égal à 0 si $n = 0$). On choisira comme ordre celui défini par : $A \succ B$ si et seulement si $d(A) > d(B)$. Il est facile de voir que $\succ$ est un ordre strict stable.

   Montrer que, si A est un atome maximal par rapport à C, si $C' = C \vee A$ ou $C' = C \vee \neg A$, et si C' vérifie les propriétés (i) et (ii) de la question 1, alors A est de la forme $P(t_1, \ldots, t_n)$, où :

   (iii) soit l'un des t_i est de la forme $f(x)$;

   (iv) soit $t_i = x$, pour tout i, et tous les atomes de C sont de la forme $Q(x, \ldots, x)$.

3. (3) En déduire que tout résolvant ordonné de clauses vérifiant (i) et (ii) vérifie encore (i) et (ii). On considérera séparément les étapes de factorisation et de résolution binaire ordonnée, et on examinera la forme des mgu possibles.

4. (2) Montrer que, à renommage près, la résolution ordonnée partant d'un ensemble fini S de clauses vérifiant (i) et (ii) ne produit qu'un nombre fini de clauses.

   Formellement, on pose $S_0 = S$, et pour tout n, S_{n+1} égale S_n union tous les résolvants ordonnés entre clauses de S_n, enfin $S_\infty = \bigcup_{n\geq 0} S_n$. On définit une fonction de normalisation N des clauses par : fixons une variable x une fois pour toutes; pour toute clause C de variable libre y, $N(C) = C[x/y]$. Ce qu'on demande de montrer, c'est que l'image par N de S_∞ est finie.

5. (1) En déduire que la satisfiabilité des formules de $\forall\exists^*$ est décidable.

Ce résultat est en fait encore vrai pour les formules $\exists^*\forall\exists^*$ (classe d'Ackermann étendue initialement), et c'est juste un tout petit peu plus dur à montrer. La classe $\exists^*\forall\forall\exists^*$, dite de Gödel, est aussi décidable. Pour une taxonomie des classes décidables et indécidables, consulter B. Dreben et W. D. Goldfarb, *The decision problem: solvable classes of quantificational formulas*, Addison-Wesley, Reading, 1979 (illisible mais exhaustif). Pour la décidabilité par résolution, consulter W. H. Joyner, *Resolution strategies as decision procedures*, Journal of the ACM 23(3), 396–417, 1976 ou T. Tammet, *Resolution methods for decision problems and finite model-building*, Ph.D. Thesis, Chalmers University, Göteborg, Suède, `ftp://ftp.cs.chalmers.se/pub/users/tammet/ds.ps.gz` et les pointeurs qui s'y trouvent.

Partie IV : contraintes ensemblistes (facile)

On considère un langage, dit d'*expressions ensemblistes*, défini comme suit :

$$e ::= X \mid 0 \mid 1 \mid e \cap e \mid e \cup e \mid \overline{e} \mid f(e_1, \ldots, e_n)$$

où f parcourt l'ensemble des symboles de fonction (d'arité n), et X parcourt un ensemble dit de *variables d'ensembles*. Les expressions e s'interprètent comme des ensembles de termes clos, modulo une valuation χ, qui à chaque variable d'ensemble X associe un ensemble de termes clos (dans le langage du premier ordre habituel). La définition est la suivante, où T est l'ensemble de tous les termes clos (sur le langage du premier ordre que l'on considère) :

$$\begin{array}{rcl}
[X]\chi & = & \chi(X) \\
[0]\chi & = & \emptyset \\
[1]\chi & = & T \\
[e_1 \cap e_2]\chi & = & [e_1]\chi \cap [e_2]\chi \\
[e_1 \cup e_2]\chi & = & [e_1]\chi \cup [e_2]\chi \\
[\overline{e}]\chi & = & T \setminus [e]\chi \\
[f(e_1, \ldots, e_n)]\chi & = & \{f(t_1, \ldots, t_n) \mid t_1 \in [e_1]\chi, \ldots, t_n \in [e_n]\chi\}
\end{array}$$

Par exemple, l'expression ensembliste $f(g(\overline{h(X)}) \cup h(X))$ dénote l'ensemble des termes de la forme $f(t)$, où t est soit de la forme $g(u)$ avec $u \neq h(v)$ pour tout v dans X, soit de la forme $h(u)$ avec u dans X.

Une *contrainte ensembliste élémentaire* est une expression de la forme $e_1 \subseteq e_2$.

On définit une relation de satisfaction par :

$$\chi \models e_1 \subseteq e_2 \text{ ssi } [e_1]\chi \subseteq [e_2]\chi$$

On cherche à montrer que le problème suivant est décidable :

ENTRÉE : un ensemble fini de contraintes ensemblistes élémentaires $K = \{E_i \mid 1 \leq i \leq n\}$.

QUESTION : K est-il satisfiable, autrement dit existe-t-il une valuation χ telle que $\chi \models E_i$ pour tout i, $1 \leq i \leq n$?

1. (2) Pour chaque sous-expression e dans K, on définit un prédicat unaire P_e, et les formules suivantes (dites formules *structurelles*) :

$$\begin{array}{ll}
\forall x \cdot \neg P_e(x) & \text{si } e = 0 \\
\forall x \cdot P_e(x) & \text{si } e = 1 \\
\forall x \cdot P_e(x) \Leftrightarrow P_{e_1}(x) \wedge P_{e_2}(x) & \text{si } e = e_1 \cap e_2 \\
\forall x \cdot P_e(x) \Leftrightarrow P_{e_1}(x) \vee P_{e_2}(x) & \text{si } e = e_1 \cup e_2 \\
\forall x \cdot P_e(x) \Leftrightarrow \neg P_{e_1}(x) & \text{si } e = \overline{e_1} \\
\forall x_1, \ldots, x_n \cdot P_e(f(x_1, \ldots, x_n)) \Leftrightarrow P_{e_1}(x_1) \wedge \ldots \wedge P_{e_n}(x_n) & \text{si } e = f(e_1, \ldots, e_n) \\
\forall x_1, \ldots, x_m \cdot \neg P_e(g(x_1, \ldots, x_m)) & \text{si } e = f(e_1, \ldots, e_n), \text{ pour tout } g \neq f
\end{array}$$

ainsi que les formules (dites *non structurelles*) :

$$\forall x \cdot P_{e_1}(x) \Rightarrow P_{e_2}(x)$$

pour toute contrainte élémentaire $e_1 \subseteq e_2$ dans K.

Soit S l'ensemble de toutes les formules ainsi obtenues. Montrer que K est satisfiable si et seulement si S est Herbrand-satisfiable.

2. (0) En déduire, ainsi que de la partie II, que le problème de satisfiabilité de contraintes ensemblistes est décidable.

Les résultats de cette partie sont tirés de L. Bachmair, H. Ganzinger, U. Waldmann, *Set constraints are the monadic class*, rapport MPI-I-92-240, Max-Planck-Institut für Informatik, Saarbrücken, décembre 1992, `ftp://ftp.mpi-sb.mpg.de/pub/papers/conferences/BGW-LICS93.dvi.Z`. Les contraintes ensemblistes sont un formalisme très pratique d'analyse de programmes. Notamment, on peut prédire le résultat d'un programme Prolog en approximant l'ensemble des termes acceptés par chaque clause Prolog par un sur-ensemble défini par des contraintes ensemblistes. On consultera par exemple la page d'Andreas Podelski en `http://www.mpi-sb.mpg.de/~podelski/papers.html`.