Sous-classes décidables de la logique du premier ordre

Correction.

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.

   La skolémisée $\hat{F}$ de F est :
   $$\forall y_1, \ldots, y_n \cdot G[c_1/x_1, \ldots, c_m/x_m]$$
   où c_1, …, c_m sont m constantes nouvelles, distinctes deux à deux.

   Si c_{m+1}, …, c_p sont les symboles de fonction qui apparaissent dans F (ils sont en nombre fini, puisque F est de taille finie), ce sont par (ii) tous des constantes. L'univers de Herbrand de $\hat{F}$ est donc réduit à $\{c_1, \ldots, c_p\}$ (complété éventuellement par une constante si $p = 0$), qui est donc 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.)

   Le théorème 14 du chapitre sur la logique du premier ordre du cours exprime que toute formule existentielle est valide si et seulement si elle est vraie dans toute interprétation de Herbrand sur $\{c_1, \ldots, c_p\}$ (sans perte de généralité, on supposera $p \geq 1$). De façon duale, soit $\hat{F} \equiv \forall y_1, \ldots, y_n \cdot G_0$ la skolémisée de F. $\hat{F}$ est universelle, et est donc insatisfiable si et seulement si elle est fausse dans toute interprétation de Herbrand sur $\{c_1, \ldots, c_p\}$.

   Mais ces interprétations de Herbrand sont des ensembles d'atomes clos. Si P_1, …, P_k sont les symboles de prédicats dans F, d'arités respectives m_1, …, m_k, les atomes clos sont tous de la forme $P_i(c_{j_1}, \ldots, c_{j_{m_i}})$, $1 \leq i \leq k$, et $1 \leq c_{j_\ell} \leq p$, pour tout $1 \leq i \leq m_i$. Les atomes clos sont donc *en nombre fini : si* $m = \max(m_\ell, 1 \leq \ell \leq k)$*, il y en a au plus* $k.p^m$*. Il n'y a donc qu'un nombre fini d'interprétations de Herbrand (au plus* $2^{k.p^m}$*), on peut ainsi toutes les tester : pour chaque interprétation de Herbrand* I*, la valeur de* $[\hat{F}]I\rho$ *(pour n'importe quel* ρ*, vu que* $\hat{F}$ *est close) est :*

$$\bigwedge_{i_1=1}^{p} \cdots \bigwedge_{i_n=1}^{p} [G_0]I(\rho[c_{i_1}/y_1, \ldots, c_{i_n}/y_n])$$

qui est calculable en temps p^n *multiplié par un polynôme en la taille de* F*.*

De façon plus pratique, on pourra construire la formule :

$$\bigwedge_{i_1=1}^{p} \cdots \bigwedge_{i_n=1}^{p} G_0[c_{i_1}/y_1, \ldots, c_{i_n}/y_n]$$

et en tester l'insatisfiabilité par n'importe quelle méthode d'insatisfiabilité propositionnelle (BDD, Davis-Putnam, tableaux, résolution, etc.)

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.

   Prenons par exemple $\forall x \cdot \exists y \cdot P(x) \Rightarrow P(y)$*. Sa skolémisée est* $\forall x \cdot P(x) \Rightarrow P(f(x))$*. Son univers de Herbrand contient au moins une constante* c*, ainsi que* $f(c)$*,* $f(f(c))$*, …,* $\underbrace{f(\ldots f(}_{n}c)\ldots)$*, pour tout* n*.*

2. (1) Fixons une formule close monadique F, soit I une interprétation de domaine D, et soient P_1, …, 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.)

   Pour tout $v \in D$*,* $(I/\equiv)(P_i)(\overline{v}) = \top$ *si et seulement s'il existe* $w \in \overline{v}$*, autrement dit* $w \in D$ *tel que* $w \equiv v$*, et tel que* $I(P_i)(w) = \top$*. Donc si* $(I/\equiv)(P_i)(\overline{v}) = \top$*, alors* $I(P_i)(w) = \top$ *pour un* w *tel que* $w \equiv v$*, donc par la définition de* $\equiv$*,* $I(P_i)(v) = \top$*. Réciproquement, si* $I(P_i)(v) = \top$*, alors comme* $v \in \overline{v}$*, par la définition de* $I/\equiv$ *on a* $(I/\equiv)(P_i)(\overline{v}) = \top$*. Pour résumer,* $(I/\equiv)(P_i)(\overline{v}) = \top$ *si et seulement si* $I(P_i)(v) = \top$*. Autrement dit,* $(I/\equiv)(P_i)(\overline{v}) = I(P_i)(v)$*.*

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, …, 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.

Choisissons $\overline{\rho}$ la valuation qui envoie chaque variable x vers $\overline{\rho(x)}$. On montre le résultat indiqué par récurrence structurelle sur G. Si G est un atome, alors il est de la forme $P_i(x)$ pour un certain i, $1 \leq i \leq n$, et pour une certaine variable x : alors $[G]I\rho = I(P_i)(\rho(x)) = (I/\equiv)(P_i)(\overline{\rho(x)})$ (par la question précédente) $= (I/\equiv)(P_i)(\overline{\rho}(x)) = [G](I/\equiv)\overline{\rho}$. Les cas des connecteurs booléens sont des appels triviaux à l'hypothèse de récurrence. Si G est de la forme $\forall x \cdot H$, alors $[G]I\rho = \bigwedge_{v\in D}[H]I(\rho[v/x]) = \bigwedge_{v\in D}[H](I/\equiv)(\overline{\rho}[\overline{v}/x])$. Nous devons montrer que ceci est égal à $[G](I/\equiv)\overline{\rho} = \bigwedge_{c\in D/\equiv}[H](I/\equiv)(\overline{\rho}[c/x])$. Or si $[G]I\rho$ est vrai, alors pour tout $v \in D$, $[H](I/\equiv)(\overline{\rho}[\overline{v}/x])$ est vrai, en particulier $[H](I/\equiv)(\overline{\rho}[c/x])$ est vrai pour toute classe $c \in D/\equiv$, puisque toute classe c s'écrit comme un $\overline{v}$ pour au moins un $v \in D$. Réciproquement, si $[G](I/\equiv)\overline{\rho}$ est vrai, alors pour toute classe c, en particulier pour toute classe de la forme $\overline{v}$, où $v \in D$, $[H](I/\equiv)(\overline{\rho}[c/x])$ est vrai : mais ceci signifie que $[G]I\rho$ est vrai. Le raisonnement est similaire pour les quantifications existentielles.

Maintenant, lorsque $G = F$, pour tout ρ, on en déduit que $I, \rho \models F$ si et seulement si $(I/\equiv), \overline{\rho} \models F$. Mais F étant close, sa valeur de vérité ne dépend ni de ρ ni de $\overline{\rho}$, donc $I \models F$ si et seulement si $(I/\equiv) \models F$. Comme $I \models F$ par hypothèse, on en conclut que $(I/\equiv) \models F$.

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 $I(P_i)(v) = \top$.)

   Montrons que f passe au quotient : $i \in f(v)$ si et seulement si $I(P_i)(v) = \top$, si et seulement si $I(P_i)(\overline{v}) = \top$, par la question 2. Donc la propriété pour i d'appartenir à $f(v)$ ne dépend que de $\overline{v}$. La fonction $\overline{f}$ qui à tout $\overline{v}$ de $D/\equiv$ associe l'ensemble des i tels que $i \in f(v)$ est donc bien définie.

   Montrons que $\overline{f}$ est injective, autrement dit que $f(v) = f(w)$ implique $v \equiv w$. En effet, $f(v) = f(w)$ implique que pour tout i dans $\{1, \ldots, n\}$, $I(P_i)(v) = I(P_i)(w)$, autrement dit que $v \equiv w$ par définition.

   Comme $\overline{f}$ est injective, le cardinal de $D/\equiv$ est au plus celui de $\mathbb{P}\{1, \ldots, n\}$, soit 2^n, et est donc fini.

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

   Si F est satisfiable, alors F a un modèle quotient fini comme dans les questions précédentes, dont les valeurs sont des parties de $\{1, \ldots, n\}$ (à isomorphisme près). Comme en partie I, il n'y a qu'un nombre fini d'interprétations possibles I sur ce domaine fini, on peut alors toutes les énumérer et tester si F est vraie sur chaque I, en temps fini.

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 $f(x)$.

   Skolémisons $\hat{F}$: on obtient $G[f_1(x)/y_1, \ldots, f_n(x)/y_n]$, dans laquelle la seule variable libre est x. $\hat{F}$ a donc encore la même propriété, donc (i) est vraie. De plus, tous les atomes de G sont de la forme $P(s_1, \ldots, s_n)$, où les s_i sont soit x, soit des y_j. Donc les atomes de $G[f_1(x)/y_1, \ldots, f_n(x)/y_n]$ sont de la forme $P(t_1, \ldots, t_n)$ où chaque t_i est soit x, soit de la forme $f(x)$. Ceci est préservé par la mise en forme clausale, donc (ii) est vérifiée.

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 $Q(x, \ldots, x)$.

Comme C' vérifie (ii), $d(A)$ vaut soit 0 soit 1. Si $d(A) = 1$, alors par définition l'un des t_i est un terme de la forme $f(t)$, avec $d(t) = 0$, donc $t = x$ par (i), et donc (iii) est vérifiée. Sinon, tous les t_i valent x. Comme A est maximal par rapport à C, tous les atomes B de C sont tels que $d(B) = 0$, donc de la forme $Q(s_1, \ldots, s_m)$ avec s_j une variable, donc $s_j = x$, pour tout j, par (i). Donc (iv) est vérifiée.

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.

On le montre d'abord pour les facteurs ordonnés. Considérons une clause $C \vee A \vee B$ (le cas $C \vee \neg A \vee \neg B$ est similaire), avec A et B maximaux par rapport à C, et σ le mgu de A et B. Comme A et B sont unifiables, on peut écrire $A = P(s_1, \ldots, s_n)$ et $B = P(t_1, \ldots, t_n)$. Si pour tout i, $s_i = t_i$, alors σ est la substitution vide, et le facteur $C \vee A$ obéit encore clairement à (i) et (ii). Sinon, soit i tel que $s_i \neq t_i$. Alors le couple (s_i, t_i) ne peut être que de la forme $(x, f(x))$ ou $(f(x), x)$ puisque (iii) ou (iv) doit être vrai. Mais aucun de ces cas n'est unifiable. En clair, on n'a jamais besoin d'effectuer que des factorisations triviales.

Montrons-le ensuite pour les résolvants binaires ordonnés. Soit donc $C \vee A$ et $C' \vee \neg B$ deux clauses vérifiant (i) et (ii), σ le mgu de A et B, et $C\sigma \vee C'\sigma$ le résolvant. Comme A et B sont maximaux, ils vérifient (iii) ou (iv), donc en particulier $A = P(s_1, \ldots, s_n)$ avec $s_i \in \{x, f(x)\}$ pour tout i, et $B = P(t_1, \ldots, t_n)$ avec $t_i \in \{y, f(y)\}$ pour tout i, où x est l'unique variable de $C \vee A$, et y est l'unique variable de $C' \vee \neg B$, et $x \neq y$ puisque les clauses sont supposées renommées de sorte à porter sur des variables distinctes. L'unificateur σ de A et B est donc soit $[y/x]$ (ou de façon équivalente $[x/y]$), soit $[f(x)/y]$, soit $[f(y)/x]$. Dans le premier cas, le résolvant est $C[y/x] \vee C'$, qui vérifie clairement (i) et (ii).

Dans le second cas, où $\sigma = [f(x)/y]$, (i) est clairement vérifiée. Montrons (ii). Nécessairement, pour que σ soit de cette forme, il existe i tel que $s_i = f(x)$ et $t_i = y$. Montrons que pour tout $j \neq i$, t_j doit valoir y aussi. En effet, sinon t_j vaudrait $g(y)$ pour un certain symbole de fonction g, mais que s_j vaille x ou soit de la forme $h(x)$, σ ne peut pas unifier s_j avec t_j. Donc $t_j = y$ pour tout j, et la clause $C' \vee \neg B$ est donc dans le cas (iv). Le résolvant binaire est $C \vee C'[f(x)/y]$: tous les atomes de ce dernier sont soit des atomes de C, auquel cas ils sont bien de la forme $P(u_1, \ldots, u_k)$ avec u_i valant

x ou $f(x)$ pour tout i, puisque $C \vee A$ vérifie (ii); soit des atomes de $C'[f(x)/y]$, qui sont donc de la forme $Q(f(x), \ldots, f(x))$. Donc le résolvant binaire vérifie (ii).

Le troisième cas est entièrement similaire.

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.

   Par récurrence sur n, toutes les clauses de S_∞ vérifient (i) et (ii). Soit m l'arité maximale d'un symbole de prédicat dans S, p le nombre de symboles de prédicat dans S, k le nombre de symboles de Skolem. Alors le nombre d'atomes de la forme $P(t_1, \ldots, t_n)$, où chaque t_i est soit x soit de la forme $f(x)$, est majoré par $p.(k+1)^m$. C'est donc aussi un majorant du nombre d'atomes apparaissant dans $N(S_\infty)$. Le nombre de littéraux est alors majoré par $2p.(k+1)^m$. Comme les clauses de $N(S_\infty)$ sont des ensembles de littéraux parmi ces derniers, il ne peut y en avoir plus de $2^{2p.(k+1)^m}$.

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

   Soit F une formule de $\forall\exists^*$, et soit S l'ensemble de clauses correspondant. Saturons S par la règle de résolution de résolution ordonnée, avec l'ordre de la question 2, couplée avec une élimination des clauses subsumées nouvelles ("subsomption en avant"). Ceci est une méthode complète de preuve. En particulier, si F est insatisfiable, alors cette procédure finira par produire la clause vide.

   Maintenant, en général, par la question 1 toute clause de S vérifie (i) et (ii). Mais par la question 5 la procédure ci-dessus ne produira qu'un nombre fini N de clauses différentes même à renommage près. Donc une fois que l'on aura engendré (des représentants à renommage près) de ces N clauses, ce qui arrive après un temps fini pourvu que notre stratégie soit équitable (par exemple par saturation par ensembles de niveaux S_n), toutes les clauses supplémentaires C que l'on pourra fabriquer ensuite seront des renommages de clauses D précédemment fabriquées; en particulier, toutes les nouvelles clauses C seront subsumées par des clauses anciennes D. Elles sont donc toutes éliminées, ce qui force la procédure à terminer.

   Si la clause vide a été engendrée à un moment quelconque, alors par la correction de la résolution F est insatisfiable. Sinon, par complétude F doit être satisfiable. Ceci fournit un algorithme (une procédure qui termine) pour décider de la satisfiabilité de toute formule monadique F.

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.

Supposons que K soit satisfiable, et soit χ une valuation telle que $\chi \models K$. On définit l'interprétation de Herbrand suivante I : l'ensemble des atomes clos de la forme $P_e(t)$ validés par I est exactement ceux tels que $t \in [e]\chi$. Par définition de la sémantique des expressions ensemblistes, toutes les formules structurelles sont validées par I. De plus, les formules non-structurelles sont validées par I, parce que pour toute contrainte $e_1 \subseteq e_2$ dans K, pour tout terme clos t qui est dans $[e_1]\chi$, t est aussi dans $[e_2]\chi$.

Réciproquement, supposons que S soit Herbrand-satisfiable, et soit I une interprétation de Herbrand validant S. Posons χ la valuation qui à toute variable ensembliste X associe l'ensemble des termes clos t tels que $P_X(t) \in I$. On montre par récurrence structurelle sur la sous-expression e de K que : (a) $t \in [e]\chi$ si et seulement si $P_e(t) \in I$. C'est par définition si e est une variable ensembliste X. Sinon, c'est une utilisation facile des formules structurelles. Le cas le moins trivial est celui où e est de la forme $f(e_1, \ldots, e_n)$.

Dans un sens, si $t \in [f(e_1, \ldots, e_n)]\chi$, alors t est de la forme $f(t_1, \ldots, t_n)$ avec $t_i \in [e_i]\chi$ pour tout i, $1 \leq i \leq n$, donc par récurrence $P_{e_i}(t_i) \in I$ pour tout I : comme la formule structurelle $\forall x_1, \ldots, x_n \cdot$

$P_e(f(x_1,\ldots,x_n)) \Leftrightarrow P_{e_1}(x_1) \wedge \ldots \wedge P_{e_n}(x_n)$ est satisfaite par I, il s'ensuit que $P_e(f(t_1,\ldots,t_n))$ est satisfaite par I, autrement dit $P_e(t)$ est satisfaite par I.

Dans l'autre sens, si $P_e(t)$ est satisfaite par I, à cause de la formule structurelle $\forall x_1,\ldots,x_m \cdot \neg P_e(g(x_1,\ldots,x_m))$ pour tout $g \neq f$, il est nécessaire que t soit de la forme $f(t_1,\ldots,t_n)$. Par la clause $\forall x_1,\ldots,x_n \cdot P_e(f(x_1,\ldots,x_n)) \Leftrightarrow P_{e_1}(x_1) \wedge \ldots \wedge P_{e_n}(x_n)$, on a donc en plus $P_{e_i}(t_i)$ satisfait par I pour tout i, $1 \leq i \leq n$. Donc par récurrence $t_i \in [e_i]\chi$ pour tout i, et donc $t \in [e]\chi$.

Maintenant, comme I satisfait toutes les clauses $\forall x \cdot P_{e_1}(x) \Rightarrow P_{e_2}(x)$, pour toute contrainte $e_1 \subseteq e_2$ dans K, en particulier pour tout terme clos t tel que $P_{e_1}(t)$ soit satisfaite par I, $P_{e_2}(t)$ est satisfaite par I. Par (a), ceci signifie que pour tout terme clos t tel que $t \in [e_1]\chi$, alors $t \in [e_2]\chi$. Donc $\chi \models e_1 \subseteq e_2$.

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

   K est satisfiable si et seulement si S est Herbrand-satisfiable, si et seulement si S est satisfiable (puisque S est une conjonction de formules universelles). Or S est une formule monadique, et la satisfiabilité de S est donc satisfiable par les résultats de la partie II.

   En fait, c'est l'idée, mais l'argument ci-dessus est faux: S n'est pas une formule monadique, puisqu'elle fait intervenir des symboles de fonction. Mais on peut effectuer une skolémisation inverse, c'est-à-dire trouver une formule F dont la skolémisée soit S, comme suit. Fixons-nous une énumération d'une infinité de variables une fois pour toutes : x_1, x_2, ..., x_n, ..., et pour chaque symbole de fonction f, soit y_f une nouvelle variable, distincte de toutes les x_i, $i \geq 0$; pour chaque expression e dans K, produisons les formules structurelles :

$\forall x \cdot \neg P_e(x)$	*si* $e = 0$
$\forall x \cdot P_e(x)$	*si* $e = 1$
$\forall x \cdot P_e(x) \Leftrightarrow P_{e_1}(x) \wedge P_{e_2}(x)$	*si* $e = e_1 \cap e_2$
$\forall x \cdot P_e(x) \Leftrightarrow P_{e_1}(x) \vee P_{e_2}(x)$	*si* $e = e_1 \cup e_2$
$\forall x \cdot P_e(x) \Leftrightarrow \neg P_{e_1}(x)$	*si* $e = \overline{e_1}$
$P_e(y_f) \Leftrightarrow P_{e_1}(x_1) \wedge \ldots \wedge P_{e_n}(x_n)$	*si* $e = f(e_1,\ldots,e_n)$
$\neg P_e(y_g)$	*si* $e = f(e_1,\ldots,e_n)$, *pour tout* $g \neq f$

   Ce qui change par rapport à la question 1, c'est qu'on ne quantifie plus universellement sur les x_i dans les deux dernières formules, et qu'on a remplacé $f(x_1,\ldots,x_n)$ par y_f et $g(x_1,\ldots,x_m)$ par y_g.

   Soit S' l'ensemble des toutes ces formules, plus les formules non structurelles de K. On définit la formule F souhaitée par :

$$F = \exists y_{f_1^0},\ldots,y_{f_{n_0}^0} \cdot \forall x_1 \cdot \exists y_{f_1^1},\ldots,y_{f_{n_1}^1} \cdot \forall x_2 \cdot \ldots \forall x_k \cdot \exists y_{f_1^k},\ldots,y_{f_{n_k}^k} \cdot \bigwedge S'$$

   où $\bigwedge S'$ est la conjonction des formules de S', k est l'arité maximale d'un symbole de fonction f dans la signature, f_1^0, ..., $f_{n_0}^0$ sont tous les symboles de fonction d'arité 0, f_1^1, ..., $f_{n_1}^1$ sont tous les symboles de fonction d'arité 1, ..., et f_1^k, ..., $f_{n_k}^k$ sont tous les symboles de fonction d'arité k. Il est clair (à permutation de quantifications universelles et de conjonctions près que la skolémisée de F est S. De plus, F est maintenant réellement une formule monadique, dont la satisfiabilité est équivalente à celle de S, donc à celle de K.

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`.