Devoir de logique. À rendre avant le 3 mai 2017.

L'objet du problème est de donner une procédure de décision par résolution pour les logiques de description, utilisées en intelligence artificielle.

Partie 1: Résolution étiquetée

L'objet de cette partie est de montrer un résultat général de complétude pour une famille de stratégies de résolution.

$\mathcal{L}$ est un ensemble fini d'entiers, appelés étiquettes. Un *littéral étiqueté* est une paire d'un littéral et d'un élément de $\mathcal{L}$, noté $L\!:\!e$. Une *clause étiquetée* est une disjonction de littéraux étiquetés. Comme d'habitude, la disjonction vide est notée $\bot$. La sémantique d'une clause étiquetée est la même que celle de la clause à laquelle on a retiré les étiquettes. L'application d'une substitution σ à $L\!:\!e$ est définie par $(L\!:\!e)\sigma = L\sigma\!:\!e$. Cette définition est étendue aux clauses.

Une *fonction de sélection* s est une application qui associe à toute clause étiquetée un sous-ensemble des littéraux de c.

Dans cette partie, on considère les deux règles d'inférence suivantes:

$$R \qquad \frac{L\!:\!e \vee C \quad \overline{L'}\!:\!e' \vee C'}{(C \vee C')\sigma} \quad \texttt{Si} \begin{cases} \sigma = \mathsf{mgu}(L, L') \\ L\!:\!e \in s(L\!:\!e \vee C) \\ \overline{L'}\!:\!e' \in s(\overline{L'}\!:\!e' \vee C') \end{cases}$$

$$F \qquad \frac{L\!:\!e \vee L'\!:\!e' \vee C}{(L\!:\!e \vee C)\sigma} \quad \texttt{Si} \begin{cases} \sigma = \mathsf{mgu}(L, L') \\ L'\!:\!e' \in s(L\!:\!e \vee L'\!:\!e' \vee C) \end{cases}$$

Soit C un ensemble de clauses. On affecte à chaque formule littéral de chaque clause de C une étiquette dans un ensemble fini (on peut affecter des étiquettes différentes à un même littéral apparaissant dans deux clauses différentes). C est ainsi considéré comme un ensemble de clauses étiquetées.

Soit $\geq$ un ordre sur les littéraux clos (sans variable) étiquetés. *Dans tout le probème*, on considère la fonction de sélection suivante: $s(c)$ est l'ensemble des littéraux $L : e$ tels qu'il existe une substitution σ telle que $(L\!:\!e)\sigma$ est maximal dans $c\sigma$.

On note S_e cette stratégie (paramétrée par $\geq$ et l'étiquetage).

Question 1.1

Dans cette question, on ne considère que des clauses closes (sans variable) et étiquetées et on suppose que $\geq$ est un ordre total.

À une clause $C = L_1\!:\!a_1 \vee \cdots \vee L_n\!:\!a_n$ on associe le multi-ensemble des littéraux étiquetés $m(C) = \{L_1\!:\!a_1, \ldots, L_n\!:\!a_n\}$. Les clauses sont ainsi ordonnées par l'extension multi-ensemble de $\geq$.

Si $\mathcal{S}$ est un ensemble de clauses et $L\!:\!a$ est un littéral étiqueté, on note $\mathcal{S}(L)$ l'ensemble des clauses C ne contenant aucun littéral $L\!:\!b$ et telles que $C \in \mathcal{S}$ ou bien $C \vee \overline{L}\!:\!b \in \mathcal{S}$ pour au moins un b. (Autrement dit, on remplace L par $\top$ dans les clauses de $\mathcal{S}$ et on simplifie).

Si $\mathcal{E}$ est un ensemble de clauses, on note de plus $\mathcal{E}^*$ l'ensemble des clauses déductibles de $\mathcal{E}$ par la stratégie S_e.

Montrer que, si $\perp \notin \mathcal{E}^*$, C est une clause minimale de $\mathcal{E}^*$ et L est un littéral maximal de C, alors $\perp \notin (\mathcal{E}^*(L))^*$.

Montrer la complétude réfutationnelle de la stratégie S_e sur les clauses closes étiquetées.

Question 1.2

Montrer que, pour tout ordre $\geq$ et tout étiquetage initial de l'ensemble de clauses, la stratégie S_e est réfutationnellement complète pour le calcul des prédicats en forme clausale.

Question 1.3

Une clause (étiquetée) C *subsume* une clause (étiquetée) C', s'il existe une substitution σ et une clause C'' telles que $C' = C\sigma \vee C''$.

une clause $C \in \mathcal{E} \setminus \{C'\}$ qui subsume C'.

Montrer que, si $\mathcal{E}$ est un ensemble de clauses étiquetées et $\mathcal{E}'$ un ensemble de clauses étiquetées telle que

- pour toute clause $C \in \mathcal{E}$, il existe une clause $C' \in \mathcal{E}'$ qui subsume C
- pour toute clause C déductible de $\mathcal{E}'$ par résolution et factorisaton, suivant la stratégie de sélection de la question 1.1, il existe une clause $C' \in \mathcal{E}'$ qui subsume C

Alors, si $\mathcal{E}$ est insatisfaisable, $\mathcal{E}'$ contient la clause vide.

Partie 2: FO_2

Cette partie a pour objectif de montrer que la stratégie de résolution de la première partie donne une procédure de décision pour le fragment FO_2 de la logique du premier ordre.

Le fragment FO_2 de la logique du premier ordre est l'ensemble des formules sans variable libre, sur un ensemble fini de symboles de fonction d'arité 0, un ensemble fini de symboles de prédicats unaires, un ensemble fini de symboles de prédicats binaires, qui n'utilisent que deux symboles de variables: x, y. Par exemple:

$$\forall x.(\exists y.R(x,y) \wedge (R(y,a) \vee \exists x.R(y,x) \wedge R(x,a)))$$

est dans FO_2. En revanche, la formule suivante n'est pas dans FO_2:

$$\forall x, \forall y, \forall z. \quad (R(x,y) \wedge R(y,z)) \Rightarrow R(x,z)$$

(En fait, on peut montrer, ce qui n'est pas demandé ici, qu'il n'existe aucune formule de FO_2 qui est logiquement équivalente à la formule de transitivité ci-dessus.)

Question 2.1

Montrer qu'il existe un algorithme polynomial qui, étant donnée une formule $\phi \in \mathsf{FO}_2$, calcule une formule $\tilde{\phi} = \phi_1 \wedge \ldots \wedge \phi_n$ telle que

1. pour tout i, ϕ_i est sans variable libre et de l'une des formes $\forall x \forall y.\psi_i$, $\forall x \exists y.\psi_i$, $\forall x.\psi_i$, $\exists x.\psi_i$, ψ_i, où ψ_i est sans quantificateur.
2. $\widetilde{\phi}$ ne contient pas de symbole de constante et tous ses symboles de prédicat sont d'arité au plus deux.
3. ϕ est satisfaisable si et seulement si $\tilde{\phi}$ est satisfaisable

Ind: On pourra, pour certaines sous-formule $Qx.\theta$ de ϕ où $Q \in \{\exists, \forall\}$, introduire un nouveau symbole de prédicat unaire $R_{Qx.\theta}$, remplacer $Qx.\theta$ par $R_{Qx.\theta}(y)$ dans ϕ et ajouter $\forall y.(R_{Qx.\theta}(y) \leftrightarrow Qx.\theta)$.

Question 2.2

On note $\mathcal{C}_2$ l'ensemble des clauses c telles que:

1. c contient au plus deux variables et, si elle contient exactement deux variables, alors elle ne contient aucun symbole de fonction
2. Si f est un symbole de fonction qui apparait dans c, alors toutes ses occurrences dans c sont dans un même terme $f(x)$
3. Il existe un littéral de c qui contient toutes les variables de c
4. Si c contient un littéral clos, alors c ne contient pas de variable et pas de symbole de fonction.

Montrer qu'il existe un algorithme qui, étant donnée une formule $\phi \in \mathsf{FO}_2$, calcule des ensembles de clauses $C_1^\phi, \dots C_n^\phi \subseteq \mathcal{C}_2$ tels que ϕ est satisfaisable ssi l'un des ensembles de clauses C_i^ϕ est satisfaisable.
Remarque: on pourra utiliser un "splitting": remarquer qu'un ensemble de clauses $C = \{c_1 \vee c_2\} \cup C'$ où c_1 et c_2 n'ont pas de variable en commun est satisfaisable ssi l'un des deux ensembles $\{c_1\} \cup C'$, $\{c_2\} \cup C'$ est satisfaisable. Cette règle de Splitting est aussi utilisée dans la question suivante.

Question 2.3

On utilise un ensemble d'étiquettes $\mathcal{L} = \{0, 1\}$. On considère l'ordre strict suivant sur les littéraux clos étiquetés: $L : a < L' : b$ ssi

- L est strictement moins profond que L'
- ou bien L et L' ont même profondeur et $a < b$

Si $\mathcal{E}$ est un ensemble de clauses de $\mathcal{C}_2$, on choisit d'étiqueter par 1 tous les littéraux contenant 2 variables et par 0 tous les littéraux contenant au plus une variable.

1. Montrer que si $C, C' \in \mathcal{C}_2$ et C'' est obtenue par résolution ou factorisation suivant la stratégie S_e, alors, les clauses obtenues par splitting de C'' sont aussi dans $\mathcal{C}_2$.
2. Montrer que la stratégie S_e (avec splitting et subsomption, cf. question 1.3) permet d'obtenir une procédure de décision de la satisfaisabilité d'un ensemble fini de clauses de $\mathcal{C}_2$, et donc une procédure de décision pour FO_2.

Partie 3: Logiques de description et logique du premier ordre

Syntaxe de $\mathcal{DL}$

N_c est un ensemble de *noms de concepts* et N_R un ensemble de *noms de rôles.*

Les *concepts* $\mathcal{C}_{\mathcal{DL}}$ de $\mathcal{DL}$ sont le plus petit ensemble tel que:

- $\top, \bot \in \mathcal{C}_{\mathcal{DL}}$ et $N_C \subseteq \mathcal{C}_{\mathcal{DL}}$
- Si $C_1, C_2 \in \mathcal{C}_{\mathcal{DL}}$ et $r \in N_R$, alors, $C \sqcap D, C \sqcup D, \neg C, \forall r.C, \exists r.C \in \mathcal{C}_{\mathcal{DL}}$

Une *TBox* $\mathcal{T}$ est un ensemble fini d'inclusion de concepts $C \sqsubseteq D$ où $C, D \in \mathcal{C}_{\mathcal{DL}}$. Une *ABox* $\mathcal{A}$ est un ensemble fini d'*assertions* d'une des formes $a : C$ ou bien $(a, b) : r$ où $a, b \in N_C$, $C \in \mathcal{C}_{\mathcal{DL}}$ et $r \in N_R$. Une *Base de données* BD est une paire constituée d'une TBox et d'une ABox.

Sémantique de $\mathcal{DL}$

Une *interprétation* $\mathcal{I}$ est donnée par un *domaine* $D_\mathcal{I}$ et une application $[\![\ ]\!]_\mathcal{I}$ de N_C dans $2^{D_\mathcal{I}}$ et de N_R dans $2^{D_\mathcal{I} \times D_\mathcal{I}}$ (autrement dit, un concept est un sous-ensemble du domaine et un rôle est une relation binaire sur le domaine). $[\![\_]\!]_\mathcal{I}$ est étendue aux concepts de la manière suivante:

- $[\![\top]\!]_\mathcal{I} = D_\mathcal{I}$, $[\![\bot]\!]_\mathcal{I} = \emptyset$
- $[\![C \sqcap D]\!]_\mathcal{I} = [\![C]\!]_\mathcal{I} \cap [\![D]\!]_\mathcal{I}$, $[\![C \sqcup D]\!]_\mathcal{I} = [\![C]\!]_\mathcal{I} \cup [\![D]\!]_\mathcal{I}$, $[\![\neg C]\!]_\mathcal{I} = D_\mathcal{I} \setminus [\![C]\!]_\mathcal{I}$.
- $[\![\exists r.C]\!]_\mathcal{I} = \{a \in D_\mathcal{I} \mid \text{il existe } (a, b) \in [\![r]\!]_\mathcal{I}, b \in [\![C]\!]_\mathcal{I}\}$
- $[\![\forall r.C]\!]_\mathcal{I} = \{a \in D_\mathcal{I} \mid \text{ pour tout } (a, b) \in [\![r]\!]_\mathcal{I}, b \in [\![C]\!]_\mathcal{I}\}$

La relation de satisfaction est alors définie par:

- $\mathcal{I} \models C \sqsubseteq D$ ssi $[\![C]\!]_\mathcal{I} \subseteq [\![D]\!]_\mathcal{I}$
- $\mathcal{I} \models a : C$ ssi $[\![a]\!]_\mathcal{I} \in [\![C]\!]_\mathcal{I}$
- $\mathcal{I} \models (a, b) : r$ ssi $([\![a]\!]_\mathcal{I}, [\![b]\!]_\mathcal{I} \in [\![r]\!]_\mathcal{I}$

$\mathcal{I}$ est un modèle d'une base de données $(\mathcal{T}, \mathcal{A})$ si elle satisfait toutes les inclusions de concepts de $\mathcal{T}$ et toutes les assertions de $\mathcal{A}$.

Les problèmes des logiques de description

On se pose le problème de la conséquence: étant donnés une base de données $\mathcal{K}$ et une inclusion de concepts $C \sqsubseteq D$ (resp. une assertion $a : C$ ou $(a, b) : r$), est ce que $C \sqsubseteq D$ (resp. $a : C$ ou $(a, b) : r$) est conséquence logique de $\mathcal{K}$?

Comme en logique du premier ordre, cet *Entscheidungsproblem* se ramène au problème de satisfaisabilité, qui est le problème que nous considérerons dans les parties suivantes.

Traduction en logique du premier ordre de $\mathcal{DL}$

À chaque nom de concept on associe un prédicat unaire et à chaque nom de rôle on associe un prédicat binaire. On considère les deux fonctions π_x et π_y qui associe à un concept de $\mathcal{DL}$ une formule de FO_2 comme suit:

- $\pi_x(A) = A(x)$ et $\pi_y(A) = A(y)$ si $A \in N_C$
- $\pi_x(C \sqcap D) = \pi_x(C) \wedge \pi_x(D)$, $\pi_y(C \sqcap D) = \pi_y(C) \wedge \pi_y(D)$
- $\pi_x(C \sqcup D) = \pi_x(C) \vee \pi_x(D)$, $\pi_y(C \sqcup D) = \pi_y(C) \vee \pi_y(D)$
- $\pi_x(\exists r.C) = \exists y.r(x, y) \wedge \pi_y(C)$ et $\pi_y(\exists r.C) = \exists x.r(y, x) \wedge \pi_x(C)$
- $\pi_x(\forall r.C) = \forall y.r(x, y) \Rightarrow \pi_y(C)$ et $\pi_y(\forall r.C) = \forall x.r(y, x) \Rightarrow \pi_x(C)$

On définit alors la traduction d'une base de données $(\mathcal{T}, \mathcal{A})$ par:

$$\pi(\mathcal{T}) = \bigwedge_{C \sqsubseteq D \in \mathcal{T}} \forall x.(\pi_x(C) \Rightarrow \pi_x(D))$$

$$\pi(\mathcal{A}) = \bigwedge_{a:C \in \mathcal{A}} \pi_x(C)\{x \mapsto a\} \quad \wedge \quad \bigwedge_{(a,b):r \in \mathcal{A}} r(a, b)$$

Question 3

Montrer que $(\mathcal{T}, \mathcal{A})$ est satisfaisable ssi $\pi(\mathcal{T}) \wedge \pi(\mathcal{A})$ est satisfaisable

En déduire que le problème de satisfaisabilité de $(\mathcal{T}, \mathcal{A})$ est décidable.