Logique et Calculabilité 2010. Examen

19 janvier 2011. Durée 3h.

Tous les documents sont autorisés. Les exercices sont indépendants. Le barême est indicatif (noter qu'il laisse le choix des exercices à traiter). Tous les résultats et les preuves du cours peuvent être utilisés en les mentionnant. Les exercices vus en TD doivent être redémontrés s'ils sont utilisés.

Exercice 1 (4 points)

Montrer que l'ensemble de clauses suivant est insatisfaisable.

$$\forall x, y, z.\ P(x, y) \vee \neg Q(f(g(y)), x), \qquad \forall x, y.\ P(x, y) \vee Q(x, f(g(a))),$$
$$\forall x, y.\ \neg P(x, y) \vee Q(g(f(x)), y), \qquad \forall x, y.\ \neg P(x, y) \vee \neg Q(g(f(y)), x)$$

Exercice 2 (15 points)

Parmi les problèmes suivants, dire ceux qui sont (in) décidables. Justifier. (Les questions sont approximativement par ordre de difficulté croissante. Toutes les questions ont des solutions en moins de 10 lignes, sauf sans doute la dernière).

1. **Donnée :** Le code d'une machine de Turing M_1 et le code d'une machine de Turing M_2 qui s'arrête, pour tout mot d'entrée w après au plus $2 \times |w|$ transitions

   Question : Pour tout mot w, $M_1(w) = M_2(w)$
2. **Donnée :** Un ensemble fini de formules du premier ordre $\mathcal{S}$ et une formule ϕ

   Question : $\mathcal{S} \models \phi$ ou bien $\mathcal{S} \models \neg\phi$.
3. **Donnée :** Le code d'une machine de Turing M

   Question : il existe deux mots w_1, w_2 de même longueur tels que $w_1, w_2 \in L(M)$.
4. **Donnée :** les codes de deux machines de Turing M_1, M_2.

   Question : $L(M_1) \subseteq L(M_2)$.
5. **Donnée :** Une formule du premier ordre ϕ sur l'ensemble de symboles de fonction $\mathcal{F}$ et les symboles de prédicats $\mathcal{P}$, une $\mathcal{F}, \mathcal{P}$-structure *finie* $\mathcal{S}$

   Question : $\mathcal{S} \models \phi$

   Note : une structure finie sur un ensemble fini de symboles de fonction et de prédicats est donnée en extension par les graphes des fonctions $f_{\mathcal{S}}$ pour $f \in \mathcal{F}$ et la liste des tuples qui sont dans $P_{\mathcal{S}}$ pour chaque $P \in \mathcal{P}$.
6. **Donnée :** Le code d'une machine de Turing M

Question : M calcule-t-elle en temps polynomial ?

Rappel : une machine de Turing M calcule en temps polynômial s'il existe un polynôme P tel que, pour toute donnée w, M s'arrête sur la donnée w après au plus $P(|w|)$ étapes de calcul.

7. **Donnée :** Un ensemble fini de tuiles T, deux relations de compatibilité $H, V \subseteq T \times T$ et une tuile de bordure $t_0 \in T$

   Question : Existe-t-il un rectangle (non vide) que l'on peut paver avec T ?

   Un rectangle $(n, m) \in \mathbb{N}^2$ est pavable (par T, V, H, t_0) s'il existe une application $f : [0, ..., n+1] \times [0, m+1] \to T$ telle que
   - pour tout i, $f(0, i) = f(i, m+1) = f(i, 0) = f(n+1, i) = t_0$
   - pour tout $i \leq n$, pour tout j, $(f(i, j), f(i+1, j)) \in H$
   - pour tout $j \leq m$, pour tout i, $(f(i, j), f(i, j+1)) \in V$
   - pour tout $1 \leq i \leq n$ et tout $1 \leq j \leq m$, $f(i, j) \neq t_0$

   Un rectangle est non vide quand $n, m \geq 1$.

Exercice 3 (5 points)

$\mathcal{P}$ est un ensemble fini de symboles de prédicat binaires et $\mathcal{F}$ est composé de la constante 0 et d'un ensemble fini de symboles de fonction unaires.

On considère deux types d'ensembles de clauses

clauses de type (1)	clauses de type (2)	
$P(x, y) \to Q(f(x), g(y))$	$P(x, y) \to Q(f(x), y)$	$P, Q \in \mathcal{P},\ f, g \in \mathcal{F}$
	$P(x, y) \to Q(x, f(y))$	$P, Q \in \mathcal{P},\ f \in \mathcal{F}$
$P(f(x), g(y)) \to Q(x, y)$	$P(f(x), f(y)) \to Q(x, y)$	$P, Q \in \mathcal{P},\ f, g \in \mathcal{F}$
$P(0, 0)$	$P(0, 0)$	$P \in \mathcal{P}$
$\neg P(0, 0)$	$\neg P(0, 0)$	$P \in \mathcal{P}$

On considère les deux problèmes (pour $i = 1, 2$) :

Donnée : Un ensemble fini S de clauses de type i

Question : S est satisfaisable.

Sachant que, pour l'un des deux indices le problème est indécidable et pour l'autre, le problème est décidable, pour quel indice le problème est il décidable ? Justifier (soit en montrant que P_i est décidable, soit en montrant que P_{2-i} est indécidable).

Solution

Exercice 1

$$\dfrac{\dfrac{\neg P(x_3, y_3) \vee Q(g(f(x_3)), y_3) \qquad \neg P(x_4, y_4) \vee \neg Q(g(f(y_4)), x_4)}{\neg P(x_3, x_4) \vee \neg P(x_4, x_3)}\, F}{\neg P(x, x)}$$

$$\dfrac{P(x_1, y_1) \vee \neg Q(f(g(y_1)), x_1) \qquad \neg P(x, x)}{\neg Q(f(g(y_1)), y_1)}\, R$$

$$\dfrac{P(x_2, y_2) \vee Q(x_2, f(g(a))) \qquad \neg P(x, x)}{Q(x_2, f(g(a)))}\, R$$

$$\dfrac{\neg Q(f(g(y_1)), y_1) \qquad Q(x_2, f(g(a)))}{\bot}\, R$$

Exercice 2

1. C'est indécidable. On réduit le probème de l'arrêt sur le mot vide :

   Donnée : le code d'une machine de Turing M

   Question : M s'arrête sur le mot vide

   On construit une instance du problème de l'énoncé comme suit : M_1 est une machine qui, sur une entrée x, ignore son entrée et simule M. Puis, lorsque M est sur le point de s'arrêter, M_1 efface son ruban et s'arrête. On considère une machine M_2 qui, sur l'entrée x, efface son ruban et s'arrête : $M_2(x) = \epsilon$ pour tout x.
   M_2 effectue bien moins de $2 \times |w|$ étapes de calcul sur toute entrée w. De plus, $M_1(x) = \epsilon$ ssi M s'arrête sur ϵ. Donc $M_1(w) = M_2(w)$ $(= \epsilon)$ pour tout w ssi M s'arrête sur ϵ.

2. Indécidable. (Ci-après solution dûe à D. Stan). On réduit le problème d'insatisfaisabilité :

   Donnée : une formule ψ de la logique du premier ordre

   Question : ψ est insatisfaisable

   On construit une instance du problème de l'énoncé comme suit : $\mathcal{S} = \emptyset$ et $\phi = \psi \wedge P$ où P est une variable propositionnelle qui n'apparait pas dans ψ. ψ est insatisfaisable ssi $\models \neg\psi$ ssi $\models \neg\psi \vee \neg P$ (car on peut toujours choisir des modèles dans lesquels P est vrai) ssi $\models \neg\phi$ ou $\models \phi$ (car on peut toujours choisir des modèles dans lesquels P est faux : on a toujours $\not\models \psi \wedge P$).

3. C'est indécidable par le théorème de Rice. La propriété "pour tout n, $L(M)$ contient au plus un mot de longueur n" est en effet une propriété non triviale des langages récursivement énumérables.

4. C'est indécidable : la question de savoir si, étant donnée M, $L(M)$ est vide, est un problème indécidable, d'après le théorème de Rice. On réduit ensuite ce problème à $L(M_1) \subseteq L(M_2)$ en choisissant une machine M_2 qui n'accepte aucun mot.

5. C'est décidable : on énumère toutes les affectations possibles des variables de la formule. Le plus simple est de le montrer par récurrence sur la formule. Par exemple, si $\phi = \forall x.\psi$, on énumère les éléments de la structure et, pour chaque $a \in \mathcal{S}$, on teste $\mathcal{S}, x \mapsto a \models \phi$ (ce qui est possible par hypothèse de récurrence). En cas d'échec pour l'un d'eux, pon retourne non. En cas de succès pour tous, on retourne oui. Remarquons qu'on obtient ainsi un algorithme qui a pour complexité $|\mathcal{S}|^n \times |\phi|$ où n est l'alternance de quantificateurs.
6. C'est indécidable. On réduit le problème de l'arrêt : soit M, w les données du problème de l'arrêt. On construit la machine M_1, qui, sur la donnée x, ignore son entrée, puis simule M sur w. Si M s'arrête sur w, alors M_1 s'arrête sur x et renvoie un résultat r. M_1 calcule alors une fonction constante, et son temps de calcul est indépendant de x, donc polynômial. Sinon, M_1 ne s'arrête pas sur x (ni sur aucune autre donnée) : M ne calcule pas en temps polynômial. Donc M_1 calcule en temps polytomial ssi M s'arrête sur w.
7. C'est indécidable. On réduit un problème de l'arrêt légèrement modifié, dans lequel on demande à la machine de s'arrêter avec un ruban vide (i.e. ne contenant qu'un symbole spécial $\square$, puisqu'on interdit d'écrire B ; la machine n'écrit $\square$ que dans cette phase d'effacement), dans un état spécial q_a. Autrement dit, on pose la question de l'accessibilité de la configuration $q_a\$$. On suppose de plus que la machine n'écrit jamais de blanc et n'écrit \$ que quand la tête de lecture pointe sur \$. On effectue un codage de la même manière que dans le cours, avec, en supplément, la tuile droite et les relations de compatibilité avec celle-ci : $(t_0, t_0), (t_0, q\$\alpha), (t_0, \$q\alpha), (t_0, \$\alpha q), (t_0, \$\alpha'\beta), (BBB, t_0) \in H$, pour $\alpha, \beta \in \Sigma$, $q \in Q$, $\alpha' \in \Sigma \setminus \{B\}$ et $(t_0, t_0), (t_0, q_0\$B), (t_0, \$BB), (t_0, BBB)$, $(q_a\$\square, t_0), (\$\square\square, t_0), (\$\square B, t_0)$, $(\square BB, t_0), (\square\square\square, t_0)$, $(\square\square B, t_0), (BBB, t_0) \in V$.
   Montrons qu'il existe un rectangle pavable ssi la machine s'arrête sur le mot vide.

   Si un rectangle $n \times m$ est pavable Pour commencer, montrons que la première ligne du pavage ($j = 1$) contient toujours le codage de la configuration initiale. Comme $f(0,0) = f(0,1) = f(1,0) = t_0$, il faut que $(t_0, f(1,1)) \in V \cap H$ et $f(1,1) \neq t_0$. Ceci n'est possible que si $f(1,1) = q_0\$B$. Alors, comme dans le cours, la première ligne est bien un codage de la configuration initiale.
   Montrons maintenant que l'avant-dernière dernière ligne contient un codage de la configuration finale. Comme $f(n+1,0) = f(n+1,1) = f(n,0) = t_0$, $(f(n,1), t_0) \in V$ et $(t_0, f(n,1)) \in H$ et de plus $f(n,1) \neq t_0$. La seule tuile qui satisfait cette condition est $f(n,1) = q_a\$\square$. Il en résulte que l'avant dernière ligne contient une configuration finale.
   Pour les lignes intermédiaires, la preuve est la même que celle du cours.

   Si la machine s'arrête après n étapes de calcul, $q_0\$B^n = \gamma_0 \vdash_M \cdots \vdash_M \gamma_f = q_a\$\square^k B^{n-k}$, on montre qu'on peut paver un rectangle $n \times n$: $f(i,j) = u$ si $\gamma_{i-1} = \delta_{i,j} \cdot u \cdot v$ où $\delta_{i,j}$ est un préfixe de longueur j et u est de longueur 3. La preuve est essentiellement la même que celle du cours ; il suffit de verifier en plus les bordures.

Exercice 3

Le problème P_1 est décidable et le problème P_2 est indécidable.

P_1 est décidable : On montre que, par résolution avec une stratégie ordonnée, seulement un ensemble fini de clauses est engendré. On fixe un ordre arbitraire sur les symboles

de prédicat. Ensuite, on ordonne les formules atomiques comme suit : $P(u,v) > Q(s,t)$ si

1. toute variable de s,t apparait a au moins autant d'occurrence dans u,v que dans s,t
2. la taille de u,v est au moins égale à celle de s,t (taille= nombre de symboles de fonction apparaissant dans les termes, en comptant les variables et les constantes).
3. Ou bien la taille de u,v est strictement supérieure à celle de s,t, ou bien $P > Q$

Cet ordre est stable par substitution (...). La stratégie consistant à n'effectuer les résolutions que sur les littéraux maximaux est complète.

On note que, par résolution avec stratégie, on obtient des clauses qui sont ou bien du même type, ou bien d'un des types suivants :

– $P(x,y) \to Q(x,y)$
– $P(0,0) \to Q(0,0)$
– $\bot$

À renommage près, il n'y a qu'un nombre fini de clauses de ce type. On les calclule toutes : l'ensemble clauses initial si et seulement si $\bot$ est dans cet ensemble saturé.

P_2 est indécidable On code le problème de correspondance de Post modifié, supposant (sans perte de généralité) qu'il n'y a aucune paire (ϵ,ϵ). Pour chaque paire (u_i,v_i), on ajoute $|u_i|$ symboles de prédicat $P_{v,i}$, un par préfixe strict v de u_i et $|v_i|$ symboles de prédicat $Q_{v,i}$, un par préfixe strict v de v_i. On a aussi deux symbole de prédicat P_i, Q_i par indice.

1. Si av est un préfixe de u_i et $av \neq u_i$, on ajoute la clause $P_{v,i}(x,y) \to P_{av,i}(a(x),y)$
2. Si $av = u_i$, on ajoute la clause $P_{v,i}(x,y) \to Q_i(a(x),y)$
3. Si $u_i = va$, on ajoute la clause $P_i(x,y) \to P_{a,i}(a(x),y)$
4. Si $u_i = \epsilon$ et $v_i = va$, on ajoute la clause $P_i(x,y) \to Q_{a,i}(x,a(y))$
5. Si av est un préfixe de v_i et $av \neq v_i$, on ajoute la clause $Q_{v,i}(x,y) \to Q_{av,i}(x,a(y))$
6. Si $av = v_i$, on ajoute les clauses $Q_{v,i}(x,y) \to P_j(x,a(y))$ pour tous j
7. Si $v_i = va$, on ajoute la clause $Q_i(x,y) \to Q_{a,i}(x,a(y))$
8. Si $v_i = \epsilon$ et $u_j = va$, on ajoute la clause $Q_i(x,y) \to P_{a,j}(a(x),y)$
9. Si $v_i = \epsilon$ et $u_j = \epsilon$ et $v_j = va$, on ajoute la clause $Q_i(x,y) \to Q_{a,j}(x,a(y))$
10. Enfin, on ajoute $P_0(0,0)$ et $P_0(x,y) \to P_{a,1}(a(x),y)$ si $u_1 = au$ (on peut toujours supposer $u_1 \neq \epsilon$, quitte à échanger les deux suites.

On ajoute ensuite une autre famille de clauses :

1. $P_i(a(x),a(y)) \to R(x,y)$
2. $R(a(x),a(y)) \to R(x,y)$
3. $\neg R(0,0)$

En ne considérant pas la clause $\neg R(0,0)$, l'ensemble de clauses a un (unique) modèle de Herbrand minimal défini par récurrence par : $H_0 = \{P_0(0,0)\}$ et $H_{n+1} = \{Q(f(u),v) |\ \exists P(u,v) \in H_n, P(x,y) \to Q(f(x),y) \in C\} \cup \{Q(u,f(v)) \mid \exists P(u,v) \in H_n, P(x,y) \to Q(x,f(y)) \in C\} \cup ....$

Dans ce modèle, (il existe un k et une suite $i_1, \ldots i_k$ d'indices telle que $u = u_1 u_{i_1} \cdots u_{i_k}$ et $v = v_1 v_{i_1} \cdots v_{i_k}$) ssi il existe un indice i tel que $(u, v) \in P_i^{\mathcal{M}}$. Et $(u, v) \in R^{\mathcal{M}}$ ssi il existe un mot w tel que $(uw, vw) \in P_i^{\mathcal{M}}$ pour un certain i.

Par le théorème de Herbrand, il en résulte que PCP modifié a une solution ssi l'ensemble de clauses est insatisfaisable.