Fondements de l'informatique. Examen
Durée: 3h

Sujet proposé par Olivier Bournez
Version 5
(corrigé)

L'énoncé comporte 5 parties (sections), certaines avec des sous-parties (sous-sections), chacune indépendante, qui pourront être traitées dans un ordre quelconque. En revanche, dans chaque partie, il peut être utile, dans la réponse à une question, d'utiliser les questions précédentes ! On pourra librement admettre le résultat d'une question pour passer aux questions suivantes. La difficulté des questions n'est pas une fonction linéaire ni croissante de leur numérotation.

La qualité de votre argumentation et de votre rédaction est une partie importante de votre évaluation.

On pourra utiliser les résultats et les théorèmes démontrés en cours sans chercher à les redémontrer.

Dans tout l'énoncé, on demande des algorithmes et des solutions à un haut niveau : dans aucune des questions il n'est demandé de décrire complètement une machine de Turing, ni même d'en donner une description graphique ; on pourra se contenter pour décrire un algorithme de le décrire par exemple en français ou dans un langage de programmation classique comme PYTHON, JAVA, C ou OCAML.

1 Calculabilité

Dans cette section, on considère que l'alphabet est $\{A, B, \ldots, Z, 0, \ldots, 9\}$.

Question 1. *Peut-on décider si une machine de Turing accepte le mot* INF412 *? Justifier.*

Solution : Non, c'est indécidable. Application directe du théorème de Rice. □

Question 2. *Peut-on décider si une machine de Turing accepte le mot* INF412 *en moins de 412 étapes ? Justifier.*

Solution : Oui, il suffit de simuler la machine pendant 412 étapes. □

Question 3. *Soit A un langage récursivement énumérable. On suppose que A contient exactement 412 mots de taille n pour tout n. A est-il décidable ? Justifier.*

Solution : A est décidable. Pour décider si un mot w est dans A, il suffit de calculer sa longueur n, d'énumérer les mots de A (ce qui est possible car il est récursivement énumérable) jusqu'à produire les 412 mots w_1, w_2, …, w_{412} de longueur n (on compte les mots produits et on s'arrête dès qu'il y en a 412). Si w figure parmi ces 412 mots, on accepte, sinon on refuse. □

Question 4. *Soit A un langage indécidable. Le langage $B = \{n | \exists x \in A, |x| \geq n\}$, où n est un entier codé en binaire, et $|x|$ désigne la longueur du mot x, est-il décidable ? Justifier.*

Solution : Oui il est décidable. En effet :

Si A est fini, alors B l'est aussi, et il suffit de construire un algorithme qui accepte exactement les mots de B pour décider B.

Sinon, c'est que pour tout entier n_0, il y a un mot de A de longueur plus grande que n_0. B est alors l'ensemble de tous les entiers. Il est clairement décidable. □

2 Ordres

On considère une signature Σ qui contient un symbole de relation binaire $\leq$, et un symbole binaire d'égalité $=$.

Question 5. *Ecrire des formules sur la signature Σ qui expriment que la relation $\leq$ est une relation d'ordre (i.e. est réflexive, transitive et antisymétrique).*

Solution : $\forall x\ x \leq x$, $\forall x \forall y\ (x \leq y \Rightarrow y \leq x)$, $\forall x \forall y \forall z (x \leq y \wedge y \leq z \Rightarrow x \leq z)$. □

Un ordre est total si pour tout x et y, on a $x \leq y$ ou $y \leq x$.

Tout ensemble *fini* peut être muni d'une relation d'ordre total : chaque énumération d'un ensemble fini détermine un ordre total sur cet ensemble. Il y a donc $n!$ ordres total pour un ensemble fini de n éléments.

Question 6. *Utiliser le théorème de compacité pour démontrer que tout ensemble peut-être muni d'une relation d'ordre total.*

Solution : Soit E un ensemble. Soit Σ la signature comportant un symbole de relation binaire $\leq$, et un symbole d'égalité $=$, et un symbole de constante pour chaque élément $c \in E$. On considère l'ensemble $\mathcal{T}$ des énoncés $a \leq a$, $(a \leq b \wedge b \leq a) \Rightarrow a = b$, $(a \leq b \wedge b \leq c) \Rightarrow (a \leq c)$, $\neg(a = b)$ pour tout $a, b, c \in E$, ainsi que $\forall x \forall y (x \leq y \vee y \leq x)$, et les axiomes de l'égalité. Tout sous-ensemble fini de $\mathcal{T}$ ne fait intervenir qu'un ensemble fini de constantes $c \in E$. Soit C cet ensemble fini de constantes. Comme tout ensemble fini de E peut être totalement ordonné, C peut être totalement ordonnée. Cela donne un modèle du sous-ensemble fini de $\mathcal{T}$.

Par conséquent, par le théorème de compacité, $\mathcal{T}$ est consistant. On en déduit que E peut être totalement ordonné. □

3 Principe de Robinson

On rappelle qu'un *corps commutatif* est un modèle des axiomes (6.8) à (6.17) du polycopié. On rappelle qu'un corps de *caractéristique* p est un corps où $\mathbf{1} + \mathbf{1} + \cdots + \mathbf{1} = \mathbf{0}$ où le $\mathbf{1}$ est répété p fois, et p est le plus petit entier avec cette propriété. Un corps est dit de *caractéristique* 0 s'il n'est pas de caractéristique p pour tout entier p. Par exemple, on connaît $\mathbb{Q}$ qui est de caractéristique 0, et $\mathbb{Z}/p\mathbb{Z}$ de caractéristique p.

Question 7. *Démontrer que si une formule ϕ du premier ordre*[1] *est vraie dans tous les corps de caractéristique 0, alors il existe un entier k tel que la formule ϕ est vraie dans tous les corps de caractéristique supérieure ou égale à k.*

1. C'est-à-dire une formule du calcul des prédicats vu en cours.

Solution : On considère l'ensemble Σ de formules constituée de la formule $\neg\phi$, des axiomes de la théorie des corps et $\mathbf{1}+\mathbf{1}\neq\mathbf{0}$, $\mathbf{1}+\mathbf{1}+\mathbf{1}\neq\mathbf{0}$, $\mathbf{1}+\mathbf{1}+\mathbf{1}+\mathbf{1}\neq\mathbf{0}$, ...

Par le théorème de compacité, Σ possède un modèle si et seulement si toute partie finie de Σ possède un modèle. Or Σ ne possède pas de modèle, donc il doit y avoir une partie finie de Σ qui ne possède pas de modèle.

Quitte à ajouter des formules de Σ dans cette partie, on peut toujours supposer que cette partie est de la forme ϕ, les axiomes de la théorie des corps, et $\mathbf{1}+\mathbf{1}\neq\mathbf{0}$, $\mathbf{1}+\mathbf{1}+\mathbf{1}\neq\mathbf{0}$, $\mathbf{1}+\mathbf{1}+\cdots+\mathbf{1}\neq\mathbf{0}$, avec $k-1$ occurences de $\mathbf{1}$ dans la dernière formule. Mais dire que cette partie finie ne possède pas de modèle, c'est exactement dire que ϕ est vraie dans tous les corps de caractéristique supérieure ou égale à k. □

4 NP-complétude

L'objectif de cet exercice est de prouver que le problème de décision SC suivant est NP-complet :

- **Donnée**: Une formule F' du calcul propositionnel sous la forme d'une conjonction de clauses de 2 ou 3 littéraux, avec **chaque variable qui apparaît au total au plus trois fois** (en comptant ses occurences positives et négatives) .
- **Réponse**: Décider si F' est satisfiable.

Question 8. *Soit x_1, $x_2 \dots x_k$ des variables propositionnelles. Considérons la formule ϕ définie par la conjonction des k clauses $(x_i \vee \neg x_{i+1})$ pour $i = 1, \dots, k-1$, et de la clause $(x_k \vee \neg x_1)$. Quelles sont les assignations des variables qui satisfont ϕ ?*

Solution : Supposons que la clause $(x_i \vee \neg x_{i+1})$ soit satisfaite et que x_i soit faux. Ceci implique que x_{i+1} est aussi égale à faux. La structure cyclique de la formule implique alors que toutes les variables doivent être à faux. Le raisonnement est symétrique si x_i est vraie. Donc ϕ est satisfaite si et seulement si $x_1 = x_2 = \cdots = x_k$. □

Question 9. *Prouver que le problème SC est NP-complet.*

Solution : Le problème est clairement dans NP : la donnée de la valeur des variables constitue un certificat vérifiable en temps polynomial : il suffit de vérifier que la formule est satisfatite.

Nous allons construire une instance du problème SC à partir d'une instance du problème 3SAT de la façon suivante. Pour chaque variable x qui apparaît dans plus de trois clauses dans F, en supposant que x apparaît dans k clauses, on considère la procédure suivante :

1. Créer k nouvelles variables $x_1, \dots, x_k$.
2. Remplacer la $i$ème occurrence de x par x_i (dans la clause correspondante).
3. Ajouter les k clauses $(x_i \vee \neg x_{i+1})$ pour $i = 1, \dots k-1$, et de la clause $(x_k \vee \neg x_1)$.

Nous obtenons ainsi une nouvelle formule F'.

Supposons que F est satisfiable. Soit t une assignation telle que son affectation rend F satisfiable. Nous allons construire une assignation t' à partir de t telle que

1. $t'(x) = t(x)$ pour toute variable x apparaissant moins de 3 fois.
2. $t'(x_i) = t(x)$ pour $i = 1, \dots k$, pour toute variable x apparaissant plus de $k > 3$ fois.

Il est facile de remarquer que l'assignation t' satisfait F'.

Maintenant, supposons que F' est satisfiable. Soit x une variable apparaissant k fois dans F. D'après la question précédente, pour que l'ensemble de k clauses $(x_i \vee \neg x_{i+1})$ pour $i = 1, \dots k-1$, et de la clause $(x_k \vee \neg x_1)$ soient satisfaite, il faut que pour $i = 1, \dots k-1$ on ait $x_k = x_1$.

Soit t' une assignation telle que son affectation rend F' satisfiable. Nous allons construire une assignation t à partir de t' telle que

1. $t(x) = t'(x)$ pour toute variable x apparaissant moins de 3 fois.
2. $t(x) = t(x_1)$ pour toute variable x apparaissant plus de $k > 3$ fois.

Il est facile de remarquer que l'assignation t satisfait F'.

F est satisfiable si et seulement si F' est satisfiable.

La transformation de F et F' se réalise bien en temps polynomial.

□

5 Théorème de Fagin

L'objectif de cette section est de prouver qu'il est possible de définir NP sans aucune notion de machine, par la logique du second ordre existentiel. Ce résultat, dû à Fagin [1] est fondateur de la complexité descriptive.

5.1 Quelques concepts vus par un logicien

La logique du premier ordre (le calcul des prédicats) vu en cours permet d'exprimer certains concepts facilement.

Par exemple, pour le logicien, un **graphe (non orienté et sans boucle)** se définit comme un modèle sur une signature Σ_G contenant un symbole E de relation d'arité 2, qui satisfait la formule

$$\forall x \forall y((\neg E(x,x)) \wedge (E(x,y) \Rightarrow E(y,x))). \tag{1}$$

Autre exemple, puisqu'une forme normale conjonctive est au final une conjonction de clauses, chaque clause étant une disjonction de littéraux, pour le logicien, une **forme normale conjonctive** se définit comme un modèle sur une signature Σ_{FNC} contenant les symboles de relation P et N d'arité 2 : son domaine est vu comme un ensemble de clauses et de variables et la relation $P(c,v)$ signifie que la variable v apparaît positivement dans la clause c, et $N(c,v)$ signifie que la variable v apparaît négativement dans la clause c.

Question 10. *Décrire complétement la (une) structure qui correspond à la forme normale conjonctive*

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (x_2 \vee \neg x_3 \vee x_5).$$

Solution : On peut prendre comme ensemble de base $\{1,2,3,4,5\}$ et

$$P = \{(1,1),(1,3),(2,4),(3,2),(3,5)\}.$$

$$N = \{(1,2),(2,1),(2,2),(3,3)\}.$$

□

5.2 Logique du second ordre

La logique du second ordre consiste à ajouter à la logique du premier ordre des variables de relations sur lesquelles on peut quantifier : Le principe est que la formule $\exists A^r\ \phi$ signifie qu'il y a un choix de relation A d'arité r, tel que la formule ϕ est satisfaite.

On va se limiter dans cet énoncé aux formules $SO\exists$: c'est-à-dire avec des quantifications existentielles sur des symboles de relations, placées au début. [2]

2. Voir note bibliographique pour le cadre général de la logique du second ordre.

Détails formels si nécessaire : Fixons une signature[3] Σ.

— Une formule de $SO\exists$ est de la forme

$$\exists R_1^{n_1}\ \exists R_2^{n_2}\ \ldots \exists R_k^{n_k}\ \psi$$

où ψ est une formule du premier ordre sur la signature $\Sigma \cup \{R_1^{n_1}, \ldots, R_k^{n_k}\}$, où $\Sigma \cup \{R_1^{n_1}, \ldots, R_k^{n_k}\}$ est la signature Σ à laquelle on a ajouté les symboles de relations R_1, R_2, …, R_k d'arités respectives n_1, n_2, …, n_k.

— Une telle formule est satisfaite en une structure sur la signature Σ s'il existe un choix d'interprétation pour les relations R_1, R_2, …, R_k avec les arités correspondantes telle que la formule ψ est satisfaite.

Prenons l'exemple suivant (comme ci-dessus, les exposants dans les quantifications sur les symboles de relation indiquent l'arité du symbole).

$$\exists B^1 \exists N^1 \exists R^1 \forall x((B(x) \vee N(x) \vee R(x)) \wedge (\forall y(E(x,y) \Rightarrow$$
$$(\neg(B(x) \wedge B(y)) \wedge \neg(N(x) \wedge N(y)) \wedge \neg(R(x) \wedge R(y))))))$$

Question 11. *Cette formule $\phi_{3-COLORABILITE}$ de $SO\exists$ exprime le problème $3-COLORABILITE$: Expliquer pourquoi un graphe est coloriable avec 3-couleurs si et seulement, lorsqu'on le voit comme une structure sur Σ_G,*[4] *cette formule $\phi_{3-COLORABILITE}$ est satisfaite en cette structure ? à quoi correspondent B, N et R ?*

Solution : $B(x)$ indique si le sommet est de la couleur 1, $N(x)$ s'il est de la couleur 2, et $R(x)$ de la couleur 3, et la formule signifie qu'aucune arrête n'a ses extrémités de la même couleur. □

Question 12. *Donner une formule ϕ_{SAT} de $SO\exists$ qui exprime le problème SAT : c'est-à-dire telle qu'une forme normale conjonctive est satisfiable si et seulement si, lorsqu'elle est vue comme une structure sur Σ_{FNC},*[5] *cette formule ϕ_{SAT} est satisfaite en cette structure.*

Solution :

$$\exists S \forall x \exists y((P(x,y) \wedge S(y)) \vee ((N(x,y) \wedge \neg S(y))).$$

S correspond aux variables qui prennent la valeur vraie. □

On se focalise dans toute la suite sur les modèles égalitaires finis et ordonnés, c'est-à-dire dont l'ensemble de base[6] est fini, et avec un symbole $=$ d'arité 2, interprété par l'égalité, et un symbole $\leq$ d'arité 2 avec une interprétation satisfaisant les axiomes habituels d'ordre total. Pour simplifier, on ne considérera que des signatures sans symboles de fonctions.[7]

Un graphe *ordonné* (non orienté et sans boucle) est un graphe au sens précédent dont les sommets sont ordonnés : c'est-à-dire un modèle sur une signature $\Sigma_G \cup \{=, \leq\}$, où $\Sigma_G \cup \{=, \leq\}$ désigne la signature Σ_G à laquelle sont ajoutés des symboles de relations $=$ et $\leq$ interprétés par l'égalité et une interprétation satisfaisant les axiomes habituels d'ordre total, et qui satisfait la formule (1).

Question 13. *Soit $\Sigma_G \cup \{=, \leq, s\}$ la signature $\Sigma_G \cup \{=, \leq\}$ avec une nouvelle constante s. Une structure finie sur cette signature comme dans le paragraphe précédent est donc la donnée (G, a) d'un graphe G fini ordonné et de l'un de ses sommets a interprétant s.*

Donnner une formule du premier ordre sur cette signature[8] *telle que : pour tout graphe G et tout sommet a de G, les sommets plus petits que a forment une clique*[9] *si et seulement si la*

3. k, n_1, …, n_k sont bien entendu des entiers.
4. Voir la section 5.1.
5. Voir la section 5.1.
6. synonyme domaine.
7. Les signatures ne possèdent donc que des symboles de constantes et des symboles de relations.
8. C'est-à-dire sur la signature $\Sigma_G \cup \{=, \leq, s\}$.
9. On rappelle qu'une clique est un sous-ensemble de sommets reliés deux-à-deux.

formule est satisfaite en (G, a).

Solution : $\forall x \forall y((\neg(x = y) \wedge x \leq s \wedge y \leq s) \Rightarrow E(x, y))$ □

Question 14. *Donner une formule ϕ_{CLIQUE} de $SO\exists$ qui exprime le problème $CLIQUE$: c'est-à-dire une formule sur la signature $\Sigma_G \cup \{=, \leq, s\}$ telle que pour tout graphe ordonné fini G et pour tout entier k, G possède une clique de taille k si et seulement si ϕ_{CLIQUE} est satisfaite en la structure (G, a), où a est l'interprétation de s comme le $k$ième sommet*[10] *de G.*

(Indication : on pourra pour cela identifier et utiliser une notion de relation bijective)

Solution :

En notant (pour la lisibilité) $f(x) = y$ pour $f(x, y)$, la formule suivante (écrite pour la lisibilité sur plusieurs ligne) convient :

$$\exists f^2$$
$$(\forall x \forall y \forall y'(((y = f(x)) \wedge (y' = f(x))) \Rightarrow (y = y')))$$
$$\wedge(\forall x \forall x' \forall y(((y = f(x)) \wedge (y = f(x'))) \Rightarrow (x = x')))$$
$$\wedge(\forall x \forall y \forall x' \forall y'(((y = f(x)) \wedge (y' = f(x')) \wedge (\neg(y = y')) \wedge (y \leq s) \wedge (y' \leq s)) \Rightarrow E(x, x')))$$

Cette formule exprime qu'il existe une fonction (codée comme une relation (première ligne) fonctionnelle (seconde ligne)), injective (troisième ligne), telle que les sommets en bijection avec $1, 2, \ldots, k$ forment une clique (dernière ligne).

□

Tous ces problèmes sont des problèmes de NP.

Le théorème de Fagin consiste à observer que les problèmes de décision qui s'expriment par une formule de $SO\exists$ de cette façon sont exactement les problèmes de NP.

5.3 Démonstration du théorème de Fagin

Pour formaliser cela, il faut arriver à parler de problèmes de décision sur des structures, et donc fixer une représentation.

Fixons une signature Σ.

Soit $\mathcal{M}$ une structure finie ordonnée sur cette signature. Puisque les éléments du domaine sont en nombre fini et ordonnés, on peut les appeler $1, 2, \ldots, n$ pour un certain n. Chaque symbole de relation $R \in \mathcal{R}$ d'arité k s'interprète comme un sous-ensemble de $\{1, 2, \ldots, n\}^k$. On peut le coder par un mot sur l'alphabet $\{0, 1\}$ de longueur n^k, où le 1 à la $i$ème position indique que le k-uplet correspondant est dans le sous-ensemble. De même chaque symbole de constante peut se coder par le codage en binaire de son interprétation. Le codage de $\mathcal{M}$, noté $< \mathcal{M} >$, est alors la concaténation des codages de ses constantes et relations. [11]

Question 15. *Soit ϕ une formule du premier ordre sur une signature Σ. Démontrer qu'il y a un algorithme qui prend en entrée le codage $< \mathcal{M} >$ d'une structure finie ordonnée sur Σ, et qui décide en temps polynomial si $\mathcal{M}$ satisfait ϕ.*

Solution : Cela se prouve par induction sur ϕ.

On va donner la preuve du cas d'induction où ϕ est de la forme $\forall x\ \psi$. Les autres cas sont similaires ou plus simples.

10. Puisque les sommets de G sont ordonnés, et en nombre fini, on peut les numéroter de 1 à n. Le $k$ième est donc celui de numéro k dans l'ordre total donné par $\leq$.

11. Rappel : on considère des signatures sans symboles de fonctions pour simplifier.

On considère la signature Σ avec un nouveau symbole de constante c, qui vise à coder la valeur substituée pour la variable x. On parcours toutes les éléments du domaine de M, et pour chacun on construit le codage de la structure où c vaut cet élément, et on appelle l'algorithme pour vérifier si ψ est bien vrai (qui existe par hypothèse d'induction). Si c'est bien le cas pour toutes les valeurs possibles de c, on retourne vrai, sinon on retourne faux. Cela se fait en un temps qui est de l'ordre du nombre d'éléments multiplié par le temps pour l'algorithme ψ, ce qui reste donc polynomial par hypothèse d'induction. □

On va tout d'abord démontrer le sens le plus facile du théorème de Fagin : toute problème de décision qui s'exprime par une formule de $SO\exists$ est dans NP.

Question 16. *Formellement : soit Σ une signature, et soit ϕ une formule de $SO\exists$. Démontrer que le problème suivant est dans NP :*

— ***Donnée**: le codage $< \mathcal{M} >$ d'une structure finie ordonnée sur Σ.*

— ***Réponse**: décider si $\mathcal{M}$ satisfait la formule ϕ.*

Solution : ϕ est de la forme $\exists R_1^{r_1} \dots \exists R_k^{r_k} \psi$, avec ψ du premier ordre sur la signature Σ à laquelle sont ajoutés les symboles de relations $R_1, R_2, \dots, R_k$. Pour un i, donnée, il y a $2^{n^{r_i}}$ relation R_i d'arité r_i sur le domaine à n éléments. Cela se code selon le codage plus haut comme un mot de longueur n^{r_i}, c'est-à-dire par un mot de longueur polynomiale sur l'alphabet $\{0, 1\}$. La donnée d'une interprétation pour R_1, …, R_k (c'est-à-dire des mots de longueur polynomiale codant R_1, …, R_k) constitue un certificat vérifiable en temps polynomial en raison de la question précédente : il suffit étant donné ces k-mots, de tester si la structure avec ces interprétations pour les R_i vérifie la formule ψ. □

Question 17. *Que peut-on dire de plus sur la difficulté du problème précédent ?*

Solution : Il est NP-complet. En effet, on a montré qu'il était dans NP, et le problème 3-COLORABILITE se réduit à ce problème : on fixe ϕ à la formule $\phi_{3-COLORABILITE}$. Etant donné un graphe, on peut produire en temps polynomial son codage en tant que structure finie ordonnée, et cela constitue bien une réduction. □

On va maintenant montrer l'autre direction : tout problème A de NP s'exprime par une formule de $SO\exists$.

Formellement : soit Σ une signature. Soit A un problème de décision sur les structures finies et ordonnées sur Σ qui est dans NP. Par définition, cela veut dire que l'on a $< \mathcal{M} > \in A$ si et seulement si $\exists u$, V accepte (w, u), pour un certain vérificateur polynomial V, où $w =< \mathcal{M} >$.

Question 18. *Construire une formule ϕ_A de $SO\exists$ qui exprime que V possède un calcul accepteur sur $w\#u$ pour un certain u, où $w =< \mathcal{M} >$.*

(on pourra admettre dans un premier temps le fait qu'il est possible de définir un ordre[12] *sur les k-uplets à partir de l'ordre $\leq$ sur les éléments, et on pourra (alors) noter $t + 1$ pour le successeur de t).*

Solution : (Schéma de la preuve) : On utilise une preuve similaire à celle du théorème de Cook-Levin donnée en cours : on écrit la conjonction de 4 formules $CELL$, $START$, $MOVE$ et $HALT$.

On code le tableau $T[i, j]$ de cette preuve cette fois par $T(\bar{i}, \bar{j})$, où $\bar{i}, \bar{j}$ sont des k-uplets.

On adapte chacune des formules en fonction dans ce nouveau contexte. Utiliser des k-uplets permet de coder un entier entre 1 et n^k (car il y a n^k k-uplets), et on a « simplement » besoin de parler de $i + 1, i + 2$, $j + 1$, $j + 2$ dans la formule $MOVE$ pour exprimer que c'est une fenêtre légale, ce qui peut bien s'écrire à l'aide de successeur de i, et j.

12. Par exemple l'ordre lexicographique

□

La direction manquante du théorème de Fagin en découle : Soit A un problème de décision sur les structures finies et ordonnées sur Σ qui est dans NP. La formule ϕ_A exprime le problème $A : \mathcal{M}$ est une instance positive de A si et seulement s'il satisfait la formule ϕ_A.

Notes bibliographiques

Sur la logique du second ordre

On s'est restreint dans cette énoncé aux quantifications du second ordre existentielles. Mais en logique du second ordre, on autorise aussi en général les quantifications universelles. Par exemple, la formule $\forall A^r\ \phi$ signifie que pour tout choix de relation A d'arité r, la formule ϕ est satisfaite.

Toute formule du second ordre peut se transformer en une formule équivalente où toutes les quantifications du second ordre sont au début.

Sur le théorème de Fagin

Le théorème de Fagin est un résultat remarquable, car on voit qu'il permet de définir NP sans avoir à définir la moindre notion de machine (par exemple de machine de Turing), ou d'algorithme ! C'est assez inattendu ! Il a donné naisance à ce que l'on appelle la complexité descriptive et complexité implicite, où des définitions sans notion de machine des principales classes de complexité ont été obtenues (P, $PSPACE$, ...).

Sur la partie 5

La partie 5 est inspirée de [2].

Références

[1] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp, editor, *Complexity in Computer Computations*, pages 43–73. American Mathematics Society, Providence R.I., 1974.

[2] N. Immerman. *Descriptive Complexity*. Springer, 1999.