École polytechnique — X2016 — INF412 Fondements de l'informatique

PC5 notée

Sujet proposé par Stéphane Graham-Lengrand et Samuel Mimram et Bruno Salvy

Cet énoncé comporte trois parties indépendantes et qui pourront être résolues dans n'importe quel ordre. Dans chaque partie, on pourra, pour répondre à une question, admettre les résultats dont on demande la démonstration aux questions *précédentes*. Il n'est pas nécessaire de traiter toutes les questions pour avoir la note maximale. Les correcteurs vous remercient d'avance d'écrire lisiblement.

1 Complétude équationnelle des algèbres de Boole

Dans cette exercice, on fixe un entier n et on considère les formules propositionnelles sur l'ensemble de variables $\mathcal{V} = \{x_1, \ldots, x_n\}$ et les connecteurs $\wedge$, $\vee$, $\neg$, $\top$ et $\bot$ (ces deux derniers étant respectivement interprétés par vrai et faux). On note $\approx$, la plus petite relation d'équivalence sur les formules telle que

$$\begin{array}{rcll}
(a \wedge b) \wedge c & \approx & a \wedge (b \wedge c) & (A) \\
a \wedge \top & \approx & a & (B) \\
a \wedge \bot & \approx & \bot & (C) \\
a \wedge b & \approx & b \wedge a & (D) \\
a \wedge a & \approx & a & (E) \\
a \wedge (b \vee c) & \approx & (a \wedge b) \vee (a \wedge c) & (F) \\
\neg(a \wedge b) & \approx & \neg a \vee \neg b & (G) \\
a \wedge \neg a & \approx & \bot & (H) \\
\neg\neg a & \approx & a & (I)
\end{array}
\qquad
\begin{array}{rcll}
(a \vee b) \vee c & \approx & a \vee (b \vee c) & (J) \\
a \vee \bot & \approx & a & (K) \\
a \vee \top & \approx & \top & (L) \\
a \vee b & \approx & b \vee a & (M) \\
a \vee a & \approx & a & (N) \\
a \vee (b \wedge c) & \approx & (a \vee b) \wedge (a \vee c) & (O) \\
\neg(a \vee b) & \approx & \neg a \wedge \neg b & (P) \\
a \vee \neg a & \approx & \top & (Q)
\end{array}$$

et qui soit une congruence, c'est-à-dire si $F \approx F'$ et $G \approx G'$ alors $F \wedge G \approx F' \wedge G'$, $F \vee G \approx F' \vee G'$ et $\neg F \approx \neg F'$.

Deux formules F et G telles que $F \approx G$ sont dites *congruentes*. La *valeur de vérité* d'une formule est la fonction $\{0,1\}^n \to \{0,1\}$ qui lui est associée, de la façon décrite dans le cours. On admettra que deux formules congruentes ont toujours la même valeur de vérité. L'objectif de cet exercice est de montrer la réciproque : si deux formules admettent la même valeur de vérité alors elles sont congruentes.

Question 1.1. *Montrez que la formule*

$$\neg x_1 \wedge \neg(x_3 \wedge \neg x_1) \tag{1}$$

est congruente à une formule sous forme normale disjonctive.

Dans la suite on admettra que toute formule est congruente à une formule sous forme normale disjonctive. Une formule F est *canonique* si elle est sous forme normale disjonctive

$$F \quad = \quad \bigvee_{i=1}^{k} C_i \tag{2}$$

et chaque formule C_i est une conjonction de littéraux, appelée *cube*, de la forme

$$C_i \quad = \quad L_{i,1} \wedge L_{i,2} \wedge \ldots \wedge L_{i,n} \tag{3}$$

où $L_{i,j}$ est soit x_j, soit $\neg x_j$, pour $1 \leq j \leq n$.

Autrement dit dans ces cubes, toutes les variables dans $\mathcal{V} = \{x_1, \ldots, x_n\}$ apparaissent exactement une fois, et dans l'ordre.

Question 1.2. *Montrez que la formule* (1) *est congruente à une formule canonique (pour $n = 3$).*

Question 1.3. *Montrez que toute formule est congruente à une formule canonique.*

Question 1.4. *Donnez la valeur de vérité de la formule* (1)*.*

Question 1.5. *Étant donnée une formule sous forme canonique, notée comme dans* (2) *et* (3)*, donnez sa valeur de vérité en fonction des littéraux apparaissant dans les différents cubes.*

Question 1.6. *Donnez trois formules canoniques distinctes qui ont la même valeur de vérité que* (1)*.*

Question 1.7. *Montrez que deux formules F et G quelconques qui ont la même valeur de vérité sont nécessairement congruentes.*

2 Théorie des différences

Terminologie : étant donné une signature Σ et une théorie $\mathcal{T}$ sur Σ, une formule F sur Σ est *$\mathcal{T}$-satisfiable* s'il existe un modèle $\mathfrak{M}$ de $\mathcal{T}$ et une valuation v qui satisfont F.

La théorie des différences [1] s'exprime avec la signature $\Sigma = (\emptyset, \emptyset, \mathcal{R})$ qui n'a aucune constante ni symbole de fonction, et où $\mathcal{R}$ est un ensemble dénombrable de symboles de relation $(R_c)_{c \in \mathbb{Z}}$, tous d'arité 2, et indexés par les entiers relatifs.

On considère la structure $\mathfrak{M}_{\mathbb{Z}}$ de signature Σ, dont le domaine de base est précisément l'ensemble $\mathbb{Z}$ des entiers relatifs, et qui interprète tout symbole R_c comme l'ensemble des paires (r, r') telles que $r - r' \leq c$.

La théorie des différences $\mathcal{D}$iff désigne l'ensemble des formules closes sur la signature Σ qui sont satisfaites dans la structure $\mathfrak{M}_{\mathbb{Z}}$.

Question 2.1. *Montrez que pour tous entiers relatifs c et c', la formule suivante est un axiome de la théorie $\mathcal{D}iff$.*

$$\forall x \forall y \forall z ((R_c(x, y) \wedge R_{c'}(y, z)) \Rightarrow R_{c+c'}(x, z)$$

On appelle "problème de différences" toute conjonction de formules atomiques sur Σ. On cherche à savoir si un tel problème est $\mathcal{D}$iff-satisfiable.

Pour un tel problème P, on définit le graphe $\mathcal{G}_P$ avec poids suivant : les sommets sont les variables libres de P, et pour toute formule atomique $R_c(x, y)$ dans la conjonction P, on place une arête orientée de poids c du sommet x vers le sommet y, notée $x \xrightarrow{c} y$.

On rappelle qu'un *chemin* d'un sommet x à un sommet y est une suite d'arêtes de la forme $x_0 \xrightarrow{c_1} x_1 \xrightarrow{c_2} \cdots \xrightarrow{c_n} x_n$ $(n \geq 0)$, avec $x = x_0$ et $y = x_n$, dont on définit le poids comme $c_1 + \cdots + c_n$.

On appelle *cycle négatif* un chemin qui va d'un sommet à lui-même et qui a un poids strictement négatif.

Question 2.2. *Montrez que s'il existe dans $\mathcal{G}_P$ un cycle négatif, alors le problème P n'est pas $\mathcal{D}iff$-satisfiable.*

On cherche maintenant à montrer la réciproque. On définit le graphe $\mathcal{G}_P^*$ comme l'extension du graphe $\mathcal{G}_P$ avec un sommet de plus, y, et avec, pour tout sommet x de $\mathcal{G}_P$, une arête de plus $x \xrightarrow{0} y$. Pour tout sommet x de $\mathcal{G}_P$, on considère l'ensemble $\mathcal{C}(x)$ des chemins de x à y dans $\mathcal{G}_P^*$.

1. *difference logic* en anglais

Question 2.3. *Montrez que si $\mathcal{G}_P$ n'a pas de cycle négatif, alors pour tout sommet x de $\mathcal{G}_P$, il existe dans $\mathcal{C}(x)$ un chemin de poids minimal. On notera $\delta(x)$ son poids.*

Question 2.4. *Montrez que si $\mathcal{G}_P$ n'a pas de cycle négatif, alors P est $\mathcal{D}$iff-satisfiable.*

On cherche maintenant à généraliser la notion de "problème de différences". Faisons d'abord une première remarque :

Question 2.5. *Montrez que si $\mathfrak{M}_\mathbb{Z}$ avec la valuation v satisfait un problème de différences P, alors pour tout entier relatif c, $\mathfrak{M}_\mathbb{Z}$ avec la valuation v' satisfait P, où $v'(x) = v(x) + c$ pour toute variable libre x de P.*

On étend alors la signature Σ en une signature Σ^+ qui rajoute à Σ une infinité de symboles de relation $(U_c)_{c\in\mathbb{Z}}$ et $(L_c)_{c\in\mathbb{Z}}$, tous d'arité 1, et indexés par les entiers relatifs. La structure $\mathfrak{M}_\mathbb{Z}$ de signature Σ est étendue en une structure $\mathfrak{M}^+_\mathbb{Z}$ de signature Σ^+ qui interprète tout symbole U_c comme l'ensemble des entiers r tels que $r \leq c$ et interprète tout symbole L_c comme l'ensemble des entiers r tels que $-r \leq c$. La théorie $\mathcal{D}\text{iff}^+$ désigne l'ensemble des formules closes sur Σ^+ qui sont satisfaites dans $\mathfrak{M}^+_\mathbb{Z}$. Enfin, on appelle "problème de différences borné" toute toute conjonction de formules atomiques sur Σ^+. On cherche à savoir si un tel problème est $\mathcal{D}\text{iff}^+$-satisfiable.

Question 2.6. *Pour tout problème de différences borné P, proposez un problème de différences P' tel que P est $\mathcal{D}\text{iff}^+$-satisfiable si et seulement si P' est $\mathcal{D}$iff-satisfiable.*
Indication : on pourra utiliser la question précédente.

On continue d'étendre la notion de "problème de différences".

Question 2.7. *Soit F une formule sur Σ^+, sans quantificateur et en forme normale disjonctive. Proposez un ensemble fini de problèmes de différences $\{P_1, \ldots, P_n\}$, tel que F est $\mathcal{D}\text{iff}^+$-satisfiable si et seulement si l'un des problèmes $\{P_1, \ldots, P_n\}$ est $\mathcal{D}$iff-satisfiable.*

On appelle "problème de différences généralisé" toute formule sans quantificateurs sur Σ^+.

Question 2.8. *Etant donné un algorithme pour décider si un graphe avec poids possède un cycle négatif, proposez un algorithme (à décrire en Français) pour décider si un problème de différences généralisé est $\mathcal{D}\text{iff}^+$-satisfiable.*

3 Machines à 2 compteurs

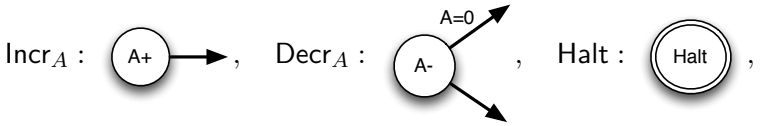
Une machine à 2 compteurs (M2C) possède deux compteurs A et B à valeurs dans $\mathbb{N}$ et un nombre fini d'états. Leur définition est similaire à la présentation des machines à k compteurs présentée dans les transparents du cours 5. On utilise une paire (a, b) pour représenter la valeur des compteurs à un instant donné, a (resp. b) étant la valeur de A (resp. B). Les instructions possibles d'une M2C sont :

- $\mathsf{Incr}_A(j)$: (a, b) devient $(a + 1, b)$ et on saute à l'état j ;
- $\mathsf{Decr}_A(j, k)$: si $a = 0$, on saute à l'état j ; sinon, (a, b) devient $(a - 1, b)$ et on saute à l'état k ;
- Incr_B et Decr_B définies de façon analogue ;
- Halt : le calcul s'arrête, le résultat est dans le compteur A.

Un état particulier est appelé état initial, l'entrée est alors dans le compteur A, le compteur B contenant généralement la valeur initiale 0.

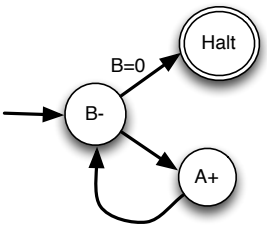
Une M2C *calcule* une fonction totale F de $\mathbb{N}$ dans $\mathbb{N}$ si pour tout $n \in \mathbb{N}$, l'exécution de la machine sur les compteurs $(n, 0)$ atteint un état Halt avec $(F(n), 0)$ dans ses compteurs.

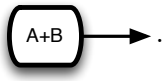
On utilisera la représentation graphique suivante :



ainsi que les graphes analogues pour Incr_B et Decr_B ; et on indiquera par une flèche sans origine l'état initial.

Par exemple, la machine suivante effectue l'opération $(a, b) \mapsto (a + b, 0)$:



On l'appellera A+B et on pourra la réutiliser comme sous-routine sous la forme 

.

Dans une telle utilisation, l'état Halt correspond à une *branche de sortie* qui pointe sur l'état suivant du calcul.

Le cours montre que les M2C peuvent simuler toute machine de Turing. L'objet de cet exercice est d'approfondir la question de la puissance de calcul de ces machines, et de montrer le résultat, paradoxal au premier abord, que ces machines ne peuvent pas calculer la fonction $N \mapsto 2^N$.

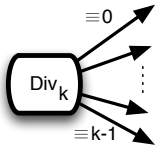
3.1 Premières machines

Question 3.1. *Écrire une M2C* Mult_3 *effectuant l'opération* $(a, 0) \mapsto (3a, 0)$.

On admet pour la suite que pour tout $k \in \mathbb{N}$, on peut ainsi écrire une M2C Mult_k.

Question 3.2. *Écrire une M2C* Div_2*, qui prend en entrée* $(a, 0)$*, possède deux états* Halt *et donc deux branches de sortie selon que* a *est pair ou impair, et termine avec* $(\lfloor a/2 \rfloor, 0)$ *dans ses compteurs.*

On admet pour la suite que pour tout $k \in \mathbb{N}^\star$, on peut ainsi écrire une M2C Div_k, avec k branches de sortie selon la valeur de a mod k. On notera cette machine



3.2 Machines à 1 compteur puissant (M1CP)

On considère maintenant des machines à un seul compteur, A, à valeur dans $\mathbb{N}$, avec un nombre fini d'états, et dont les instructions sont

1. Add_k, qui ajoute k au compteur, k étant un entier positif fixé ;
2. Mul_k, Div_k, comme ci-dessus ;
3. Decr, Halt comme pour les machines à 2 compteurs.

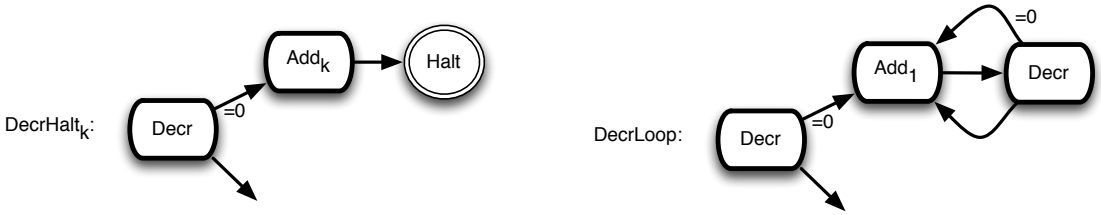
Une telle machine *calcule* une fonction totale F de $\mathbb{N}$ dans $\mathbb{N}$ si pour tout $n \in \mathbb{N}$, l'exécution de la machine sur le compteur initialisé à n atteint un état Halt avec $F(n)$ dans le compteur.

Les questions de la section précédente montrent que ces machines peuvent être simulées par des machines à 2 compteurs. La limitation à un seul compteur facilite l'analyse du fonctionnement de ces machines et en particulier des branches $A = 0$ de leurs instructions Decr. Pour voir cela, on peut également ajouter aux machines à 1 compteur :

— pour tout entier k, une instruction $\mathsf{DecrHalt}_k$, qui décrémente le compteur si sa valeur est non-nulle, et s'arrête avec k dans le compteur sinon.

— une instruction DecrLoop, qui décrémente le compteur si sa valeur est non-nulle, et déclenche une boucle infinie sinon.

Ces instructions peuvent s'encoder avec Add_k, Decr, et Halt de la manière suivante :



Question 3.3. *Montrez que, réciproquement, toute machine à 1 compteur peut être simulée par une machine à 1 compteur qui utilise* $\mathsf{DecrHalt}_k$ *et* DecrLoop *mais qui n'utilise pas d'instructions* Decr*.*

Pour une telle machine $\mathcal{M}$, on définit $S_{\mathcal{M}}$ comme l'ensemble des entiers k de ses instructions $\mathsf{DecrHalt}_k$.

Question 3.4. *Soit $\mathcal{M}$ une telle machine qui calcule une fonction totale $N \mapsto F(N)$. En suivant le chemin effectué lors d'un calcul, montrer que pour tout N vérifiant $F(N) \notin S_{\mathcal{M}}$, il existe deux entiers M et D strictement positifs tels que*

$$F(N + pD) = F(N) + pM, \qquad \forall p \in \mathbb{N}.$$

Question 3.5. *En déduire que les M1CPs ne peuvent pas calculer $N \mapsto 2^N$.*

La suite de l'exercice consiste à montrer que les M1CPs peuvent néanmoins simuler les M2Cs.

3.3 Boucles fortes et normalisation des machines à 2 compteurs

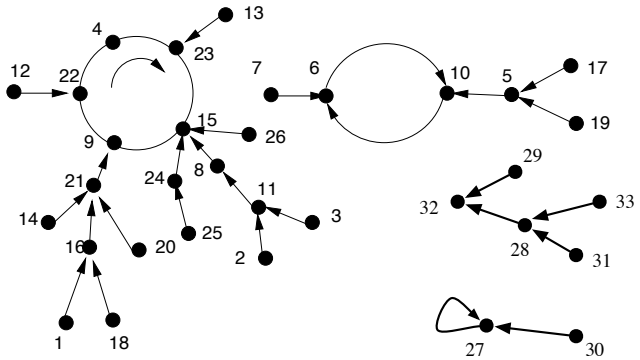


FIGURE 1 – Un graphe orienté à 33 sommets ayant chacun au plus une arête sortante

Si on efface temporairement les branches $= 0$ des instructions Decr_A et Decr_B d'une M2C, il reste un graphe orienté où chaque instruction a au plus une arête sortante. Dans un tel graphe, suivre les flèches à partir d'un sommet arbitraire mène soit à un sommet sans arête sortante (Halt), soit à un sommet déjà visité. Un exemple de tel graphe est présenté en Figure 1. Tout graphe de ce type se décompose donc en un ensemble de cycles orientés (éventuellement réduits à un sommet), sur lesquels viennent pointer des arbres. Les arêtes formant ces cycles sont appelées *boucles fortes* de la machine, et les sommets de ces cycles sur lesquels pointent des arêtes autres que celles du cycle sont appelés *sommets d'entrée* de ces boucles fortes. Dans la suite, on raisonne sur l'ensemble de la machine, cette construction ayant permis d'identifier ses boucles fortes et leurs sommets d'entrée.

Question 3.6. *Montrer qu'on peut simuler toute M2C par une M2C où les boucles fortes n'ont qu'un sommet d'entrée.*

Question 3.7. *Montrer qu'on peut récrire toute M2C préparée comme à la question précédente de sorte que dans le parcours des boucles fortes à partir de leur sommet d'entrée, les instructions* Decr *précèdent les instructions* Incr*.*

Question 3.8. *Soit une machine qui contient une boucle forte préparée comme à la question précédente. Soit a^+ le nombre de Incr_A de la boucle, b^+ le nombre de Incr_B, a^- le nombre de Decr_A et b^- le nombre de Decr_B. Écrire une machine qui simule cette boucle et telle que, à tout moment d'une exécution sur une entrée (a, b),*

- *soit la valeur du compteur B est inférieure à $\max(b, b^+ - b^-)$;*
- *soit la valeur du compteur A est 0.*

Dans cette machine, on pourra utiliser des sous-routines Div_k, Mul_k, A+B, et à l'intérieur de ces sous-routines, la valeur du compteur B pourra devenir arbitrairement grande.

On dira qu'une M2C est *normalisée* si elle vérifie les trois propriétés spécifiées aux questions 3.6, 3.7 et 3.8.

3.4 Simulations

On introduit enfin une troisième sorte de machines : des machines à 1 compteur et 1 compteur borné (M1.5C), toujours avec un nombre fini d'états. Ces machines disposent d'un compteur C contenant un entier naturel arbitraire, et d'un compteur D contenant un entier limité à un intervalle $\{0, \ldots, K\}$. On utilise une paire (c, d) pour représenter la valeur des compteurs à un instant donné, c (resp. d) étant la valeur de C (resp. D). Ces machines disposent en outre d'un drapeau F à valeur dans $\{0, 1\}$. Lorsqu'on les utilisera pour simuler une M2C, F indiquera si C contient la valeur de A (F est à 1) ou celle de B (F est à 0). L'ensemble d'instructions est le suivant :

- les instructions de la M1CP, n'agissant que sur C ;
- les instructions Decr_D et Incr_D, cette dernière transformant (c, d) en $(c, d + 1)$ si $d < K$, et échouant si $d = K$;
- une instruction Test_F avec deux sorties selon que F est à 0 ou à 1 ;
- une instruction Switch_F qui change la valeur du drapeau F (de 0 à 1 ou de 1 à 0).

La notion de calcul d'une fonction totale est la même que pour les machines à deux compteurs, le drapeau F étant initialement à 1.

Il est facile de voir qu'une M2C peut simuler une telle machine. L'inverse est plus intéressant.

Question 3.9. *Montrer qu'on peut simuler une M2C normalisée (et donc une M2C quelconque) par une M1.5C. Précisez une borne sur la valeur de K pour cette machine.*

Question 3.10. *Montrer qu'on peut simuler une M1.5C dont le compteur D est borné par K et ayant M états par une M1CP ayant au plus $2M(K + 1)$ états.*

Question 3.11. *Conclure que les machines à 2 compteurs ne peuvent pas calculer la fonction $N \mapsto 2^N$. Expliquer pourquoi ce résultat n'est pas contradictoire avec le fait que ces machines peuvent simuler des machines de Turing.*