Examen – Concepts et Model-Checking

1er mars 2021

Durée: de 15h45 à 18h15. Envoyez vos solutions à `schwoon@lsv.fr`, **avant 18h15**, en un seul archive nommé `cmc-votrenom.zip` ou `cmc-votrenom.tgz`.

1 Spécification

On considère un circuit logique avec un signal d'entrée x, un signal de sortie y et deux signaux internes r_1, r_2. À tout moment un signal vaut 0 ou 1, du coup on note $AP = \{x, y, r_1, r_2\}$. Traduire les spécifications informelles en une formule temporelle sur AP:

(a) "Il est impossible d'avoir deux 1 consecutives dans la sortie."

(b) "Si le signal d'entrée vaut 1, au plus deux tics plus tard la sortie vaut 1."

(c) "Si le signal d'entrée vaut 1, les registres restent inchangés entre le tic actuel et le tic suivant."

(d) "Le registre r_1 vaut 1 infiniment souvent."

Vous pouvez donner des formules de LTL ou CTL, au choix. Il est possible que certaines spécifications se prêtent à plusieurs interprétations différentes (c'est tout l'intérêt des spécifications formelles !), en cas de doute justifiez votre réponse.

Solution :

Toutes les propriétés s'expriment dans les deux logiques, voici des exemples dans LTL. D'ailleurs plusieurs traductions sont possibles dans certains cas car les spéfications informelles sont justement un peu imprécis (p.ex. la première possibilité, doit elle tenir dans toujours ou juste au début ?). Toute solution "raisonnable" était acceptée.

(a) $\mathbf{G}\,(\neg y \vee \mathbf{X}\,\neg y)$

(b) $\mathbf{G}\,(x \rightarrow (\mathbf{X}\,y \vee \mathbf{X}\,\mathbf{X}\,y))$

(c) $\mathbf{G}\,(x \rightarrow (r_1 \leftrightarrow \mathbf{X}\,r_1 \wedge r_2 \leftrightarrow \mathbf{X}\,r_2))$

(d) $\mathbf{G}\,\mathbf{F}\,r_1$

2 Expressivité

Fixons un seul prédicat p, du coup $AP = \{p\}$ et $2^{AP} = \{\emptyset, \{p\}\}$. Pour chacun des langages L_i suivants, donner une formule ϕ_i telle que $[\![\phi_i]\!] = L_i$.

(a) $L_1 = \{p\}^*\emptyset^\omega$;

(b) $L_2 = \{p\}^n\emptyset^\omega$, pour n fixe ;

(c) $L_3 = (\{p\} \cdot \emptyset)^\omega$.

Solution :
Ici, on cherche des formules de LTL (qui définissent un langage, contrairement aux formules de CTL). Dans les deux premiers cas, il faut assurer que p reste vrai pendant une période puis devient faux à jamais.

(a) $p\,\mathbf{U}\,(\mathbf{G}\,\neg p)$

(b) $(p\,\mathbf{U}\,(\mathbf{G}\,\neg p)) \wedge \mathbf{X}^{\,n-1}\neg p \wedge \mathbf{X}^{\,n}\neg p$

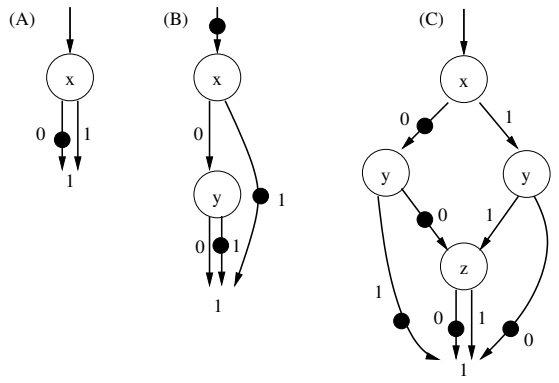
(c) $p \wedge \mathbf{G}\,(p \leftrightarrow \mathbf{X}\,\neg p)$

À noter que le troisième pas n'est pas identique à celui discuté en classe qui ne s'exprime pas en LTL ("p tient dans les positions paires") car dans ce cas-là la valeur de p est indeterminée sur les positions impaires.

3 BDD avec complément

Les *BDD avec complément* (CBDD) sont une variante des BDD où toute arête est équipée d'un bit supplémentaire dit *bit de négation*. Si ce bit est 1, alors le BDD commençant à sa destination est interprété comme le complément de ce qu'il donnerait sinon. Pour uniformité, un BDD est identifié avec un pointer sur sa racine qui est lui aussi équipé d'un tel bit de négation. Les CBDD possèdent une seule feuille 1, la feuille 0 est supprimée.

Par exemple, le CBDD A ci-dessous à gauche représente la formule x tandis que B représente $x \vee y$.



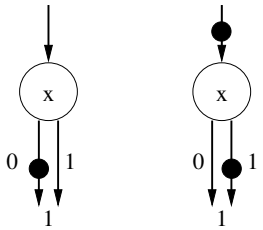
(a) Montrer que les CBDD définis ainsi ne sont pas canoniques, c'est à dire il existe des formules représentées par plusieurs CBDD non isomorphes.

(b) Prouver que pour tout CBDD on peut trouver un CBDD équivalent qui ne possède aucune arête étiquetée par 0 et avec bit de négation activé.

(c) Trouver de tels CBDD pour A and C ci-dessus.

Solution :

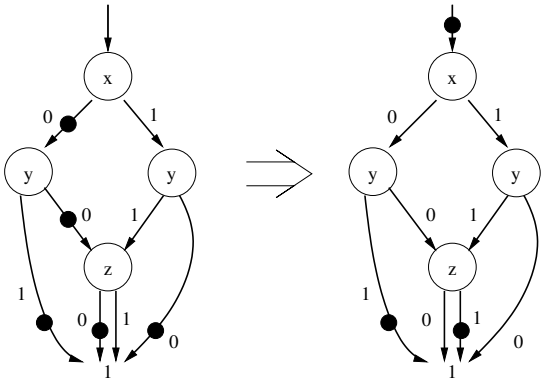
(a) Pour montrer la non-canonicité des CBDD, il suffit d'exhiber un diagramme équivalent à celui de A:



(b) Tous les chemins d'un CBDD mènent à la seule feuille 1. Du coup, pour un chemin donné, c'est la parité du nombre de négations qui détermine si le chemin vaut 0 ou 1.

Supposons qu'on possède un sommet n dont le bit de négation sur son arête sortante '0' est actif. En invertant les bits de toutes les arêtes incidentes à n (entrantes et sortantes), la parité des chemins passant par n reste inchangée (tout chemin qui passe par n contient une arête entrante et une arête sortante). Si on applique cette modification itérativement "du bas vers le haut", i.e. en commençant avec les variables maximales de l'ordre, on peut successivement éliminer toutes les bits de négations sur les arêtes '0'.

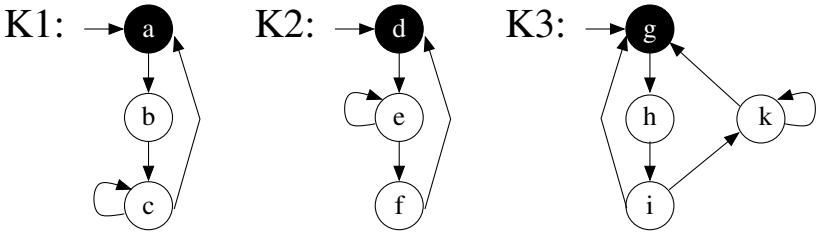
(c) Pour le résultat de cette conversion sur A, voir (a).
Pour C, le résultat est comme suite.



4 Bisimulation

On considère les structures $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3$ ci-dessous. Les états noirs satisfont p, et les états blancs ne satisfont pas p.

Pour chaque paire $\mathcal{K}_i, \mathcal{K}_j$, déterminer si $\mathcal{K}_i \equiv \mathcal{K}_j$, soit en donnant une relation de bisimulation ou en donnant une formule CTL (avec $\mathbf{EX}, \mathbf{AX}$ comme seules modalités) qui les distingue.



Solution :

$\mathcal{K}_1$ et $\mathcal{K}_3$ sont bisimilaires, avec la relation $H = \{\langle a, g\rangle, \langle b, h\rangle, \langle c, i\rangle, \langle c, k\rangle\}$. On vérifie facilement que toutes les conditions d'une bisimulation sont remplies.

Par contre, $\mathcal{K}_2$ n'est pas bisimilaire aux deux autres, p.ex. $\mathcal{K}_2$ satisfait $\mathbf{EX}\,\mathbf{EX}\,\mathbf{AX}\,p$ mais pas $\mathcal{K}_1$ ni $\mathcal{K}_3$.

5 Automates de Büchi

Dans le cours on a montré que les AB déterministes sont une sous-classe strictement moins expressifs des AB. Cette sous-classe possède néanmoins quelques propriétés intéressantes.

(a) Montrer que les langages acceptés par les AB déterministes sont clôturés sous l'intersection et l'union.

Un *automate de Muller* (AM) est un tuple $\mathcal{M} = \langle S, \Sigma, s_0, \Delta, \mathcal{F} \rangle$ dont les éléments sont comme pour les AB généralisés, avec $\mathcal{F} \subseteq 2^S$, mais une condition d'acceptance différente : Soit ρ un calcul sur AM et J l'ensemble d'états qui y apparaissent infiniment souvent. On dit que ρ est acceptant si $J \in \mathcal{F}$.

(b) Montrer que les AM acceptent les mêmes langages que les AB.

Solution :

(a) Pour l'intersection, on constate que la construction donnée dans le cours, appliquée aux AB déterministes, donne un AB déterministe.

Pour l'union, il convient de construire un automate de produit classique. Soient $\mathcal{B}_1 = \langle S, \Sigma, s_0, \delta_1, F \rangle$ et $\mathcal{B}_2 = \langle T, \Sigma, t_0, \delta_2, G \rangle$ deux AB déterministe, s.p.d.g. on suppose que δ_1, δ_2 sont complètes. On construit alors $\mathcal{B} = \langle S \times T, \Sigma, \langle s_0, t_0 \rangle, \delta, H \rangle$ avec $\delta(\langle s, t \rangle, a) = \langle \delta_1(s, a), \delta_2(t, a) \rangle$ et $H = (F \times T) \cup (S \times G)$. Un chemin est acceptant dans $\mathcal{B}$ ssi on touche infiniment souvent des états de F ou infiniment souvent des états de G. On en conclut que $\mathcal{B}$ accepte bien l'union des langages de $\mathcal{B}_1$ et $\mathcal{B}_2$.

(b) Étant donné un AB $\mathcal{B} = \langle S, \Sigma, s_0, \Delta, F \rangle$, on construit un AM $\mathcal{M} = \langle S, \Sigma, s_0, \Delta, 2^F \rangle$. Il est évident que $\mathcal{M}$ accepte le même langage que $\mathcal{B}$.

Dans l'autre sens, soit $\mathcal{M} = \langle S, \Sigma, s_0, \Delta, \mathcal{F} \rangle$ un AM. Supposons que pour un calcul donné, J est l'ensemble d'états qui apparaissent infiniment souvent. Alors tout état de $S \setminus J$ doit avoir une dernière apparance. Un AB qui simule $\mathcal{M}$ peut alors simuler un calcul de $\mathcal{M}$ pendant une partie initiale, puis à tout moment (de façon non-déterministe) deviner un $J \in \mathcal{F}$ et basculer dans une copie de $\mathcal{M}$ restreint aux états de J. S'il arrive à continuer son calcul à jamais, alors $\mathcal{M}$ aurait accepté ce calcul. Formellement, on construit l'AB suivant $\mathcal{B} = \langle S \uplus S', \Sigma, s_0, \Delta', Q' \rangle$, avec:

$$S' := \{\, s_J \mid J \in \mathcal{F},\ s \in J \,\}$$

$$\Delta' := \Delta \uplus \Delta_b \uplus \Delta_v$$

$$\Delta_b := \{\, \langle s, a, s'_J \rangle \mid \langle s, a, s' \rangle \in \Delta,\ J \in \mathcal{F},\ s' \in J \,\}$$

$$\Delta_v := \{\, \langle s_J, a, s'_J \rangle \mid \langle s, a, s' \rangle \in \Delta,\ J \in \mathcal{F},\ s, s' \in J \,\}$$

Les états avec sous-script J représentent une copie de M restreinte aux états de J. Ces états sont acceptants, du coup on doit parvenir à y rester à jamais. L'automate $\mathcal{B}$ peut basculer vers une telle copie avec les transitions dans Δ_b. Les transitions dans Δ_v continuent le calcul dans ces copies.