Devoir : Événements rationnels et requêtes

Ce sujet s'intéresse à des logiques linéaires étendues par des expressions rationnelles.

1 Événements rationnels

La logique RELTL ajoute des événements rationnels sous la forme d'expressions rationnelles sur l'alphabet $\Sigma = 2^{\mathrm{AP}}$. On définit une formule RELTL selon la grammaire abstraite suivante :

$$\begin{aligned} \varphi &::= \top \mid \bot \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle E \rangle \varphi \mid [E]\varphi && \text{(formules)} \\ E &::= \varepsilon \mid \emptyset \mid a \mid E^* \mid E \cup E \mid EE && \text{(expressions)} \end{aligned}$$

où p est une proposition atomique de AP et a un symbole de Σ, c'est-à-dire un ensemble de propositions atomiques issues de AP. On définit pour tout a de Σ la formule propositionnelle

$$\varphi_a = \bigwedge_{p \in a} p \wedge \bigwedge_{p \notin a} \neg p \ ,$$

et pour tout p de AP l'expression rationnelle

$$\Sigma_p = \bigcup_{p \in a, a \in \Sigma} a \ .$$

Le langage défini par une expression rationnelle est noté $L(E)$. Les modalités $\langle E \rangle$ et $[E]$ ont pour sémantique, pour toute séquence $\sigma = s_0 s_1 \dots$ de Σ^ω, pour toute position i de $\mathbb{N}$, et pour toute formule RELTL φ :

$$\begin{aligned} \sigma, i \models \langle E \rangle \varphi &\text{ ssi } \exists j \geq i,\ s_i \dots s_{j-1} \in L(E) \wedge \sigma, j \models \varphi \\ \sigma, i \models [E]\varphi &\text{ ssi } \forall j \geq i,\ s_i \dots s_{j-1} \in L(E) \Rightarrow \sigma, j \models \varphi \end{aligned}$$

Pour le cas où $i = j$, on utilise la convention $s_i \dots s_{i-1} = \varepsilon$.

1.1 Identités

1.1.1 On note $\setminus$ l'opérateur de complémentation sur les expressions rationnelles. La formule $\neg\langle E \rangle \varphi$ est-elle équivalente à $\langle \Sigma^* \setminus E \rangle \varphi$?

1.1.2 Montrer que $\langle E \rangle \varphi \equiv \neg[E]\neg\varphi$.

1.1.3 Montrer que toute formule RELTL peut être mise sous la forme alternante suivante

$$\begin{aligned} \varphi &::= \bot \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \langle E \rangle \psi && \text{(formules } \Diamond\text{)} \\ \psi &::= \top \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid [E]\varphi && \text{(formules } \Box\text{)} \end{aligned}$$

Quelle est la taille des formules que vous obtenez sous cette forme ?

1.2 Expressivité & Co.

1.2.1 Définir les modalités LTL usuelles G, F et X à l'aide de formules RELTL.

1.2.2 Pour une expression rationnelle E et un mot u de Σ^*, on note $u^{-1}E$ une expression dérivée qui vérifie $L(u^{-1}E) = \{v \in \Sigma^* \mid uv \in L(E)\}$ (on sait construire une telle expression dérivée par la méthode de Brzozowski).

Peut-on donner une formule RELTL *réduite*, où toutes les modalités sont sous la portée d'une modalité X, équivalente à la formule $\langle E\rangle\varphi$? Même question pour $[E]\varphi$.

1.2.3 Une preuve du fait que tout langage rationnel sur Σ^ω peut être exprimé à l'aide d'une formule RELTL est donnée en section 1.4.2 du chapitre par Stéphane Demri et Paul Gastin disponible depuis la page web du cours `http://mpri.master.univ-paris7.fr/C-1-22.html`.

Appliquez cette preuve et donnez une formule RELTL pour le langage $(\Sigma_p\Sigma)^\omega$.

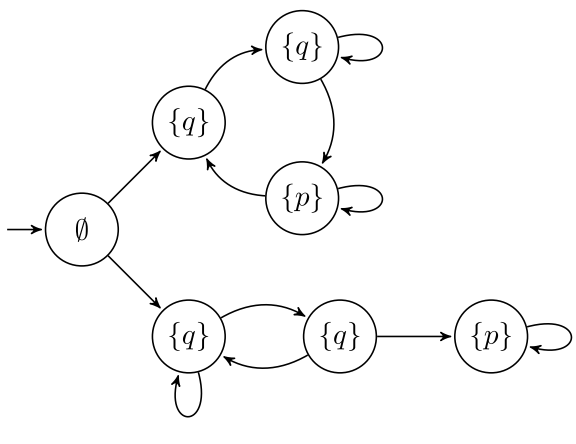
1.2.4 Montrer que le problème du model checking existentiel $\mathsf{MC}^\exists$(RELTL) d'une structure de Kripke à l'aide d'une formule RELTL est PSPACE-difficile.

On admettra pour la suite qu'il est en fait PSPACE-complet.

2 Requêtes RELTL

Des formules logiques comme les formules RELTL servent non seulement à vérifier qu'un modèle suit bien une spécification, mais peuvent aussi être utilisées pour explorer les comportements d'un modèle complexe.

Prenons pour exemple la structure de Kripke M suivante sur l'ensemble AP $= \{p, q\}$:



Au delà de la simple vérification $M \models_\exists \mathsf{F}p$, on pourrait ainsi souhaiter connaître les scénarios qui amènent à des états où p est vrai.

2.0.1 Donnez une expression rationnelle E qui vérifie $M \models_\exists [E]p$.

2.0.2 On peut observer que $M \models_\forall [\Sigma_{p\wedge q}]p$, c'est-à-dire que des comportements non permis par le modèle donnent lieu à des scénarios.

Donnez une expression rationnelle E qui vérifie $L(E) = \{w \in \Sigma^* \mid M \models_\forall [w]p\}$, c'est-à-dire une solution maximale au sens de l'inclusion des langages à l'équation $M \models_\forall [X]p$ d'inconnue X.

2.1 Requêtes

Pour la suite, on définit une structure de KRIPKE comme un tuple $\langle \mathrm{AP}, Q, I, T, l\rangle$ où Q est un ensemble fini d'états, $I \subseteq Q$ l'ensemble des états initiaux, $T \subseteq Q \times Q$ une relation de transition, et $l : Q \to 2^{\mathrm{AP}}$ une fonction d'étiquetage.

Une solution à une requête $M \models_\forall [X]\varphi$ où M est une structure de KRIPKE et φ une formule RELTL, est un langage rationnel L tel que

$$L = \{w \in \Sigma^* \mid M \models_\forall [w]\varphi\} \ .$$

Pour un ensemble d'états Q', on note $M, Q' \models \varphi$ si toutes les exécutions de M commençant dans un des états de Q' vérifient la formule φ. Dans le cas où $Q' = I$ l'ensemble des états initiaux de M, on a alors $M, Q' \models \varphi$ ssi $M \models_\forall \varphi$.

2.1.1 Pour tout mot w de Σ^*, on note $T(w)$ l'ensemble d'états de M

$$T(w) = \{q \mid \exists n \geq 0, \exists (q_0, \ldots, q_n) \in I \times Q^n, l(q_0 \ldots q_{n-1}) = w \wedge q_n = q\}$$

accessible par w. En particulier, $T(\varepsilon) = I$.

Montrer que $M \models_\forall [w]\varphi$ ssi $M, T(w) \models \varphi$.

2.1.2 On note

$$[\![\varphi]\!]_M = \{q \in Q \mid M, \{q\} \models \varphi\}$$

l'ensemble des états de M qui vérifient toujours la formule φ.

Montrer que la solution de la requête $M \models_\forall [X]\varphi$ est

$$L = \{w \in \Sigma^* \mid T(w) \subseteq [\![\varphi]\!]_M\} \ .$$

2.1.3 Construisez un automate fini $\mathcal{A}$ tel que $L(\mathcal{A})$ est le langage solution de $M \models_\forall [X]\varphi$. Quelle est la complexité de votre construction ?

2.2 Complexité

2.2.1 Montrer que construire une solution à une requête $M \models_\forall [X]\varphi$ est PSPACE-difficile.

2.2.2 Montrer par une réduction depuis l'universalité d'une expression rationnelle que construire une solution à une requête $M \models_\forall [X]\varphi$ est PSPACE-difficile *même quand* φ *est de longueur fixée.*

Ce dernier point montre que calculer une solution à une requête est sensiblement plus difficile que le problème du model checking.