`ltl2ba`
Un compilateur de formules LTL en automate de Büchi généralisés

Emile ROLLEY Thomas MORIN

Université de Bordeaux

12 mai 2022

Automates de Büchi sur *les transitions*

Automates de Büchi sur *les transitions*

Même définition que pour un automate de Büchi généralisé :

$$\mathcal{A} = (S, \rightarrow, S_0, F_1, ..., F_l) \quad \text{avec} \quad \forall i \in \{1, .., l\},\ F_i \subseteq \rightarrow$$

Automates de Büchi sur *les transitions*

Même définition que pour un automate de Büchi généralisé :

$$\mathcal{A} = (S, \rightarrow, S_0, F_1, ..., F_l) \quad \text{avec} \quad \forall i \in \{1, .., l\},\ F_i \subseteq \rightarrow$$

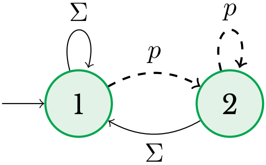

Figure 1: Exemple d'automate reconnaissant la formule LTL $\mathsf{GF}p$, avec en pointillé, les transitions appartenant à l'unique condition d'acceptation.

L'algorithme de traduction

L'algorithme de traduction

Intuition

Diviser la formule de départ φ en sous-formules plus simple (dites *réduites*) et ajouter une condition d'acceptation pour chaque sous-formule de la forme $\alpha \mathsf{U} \beta$.

L'algorithme de traduction

Intuition

Diviser la formule de départ φ en sous-formules plus simple (dites *réduites*) et ajouter une condition d'acceptation pour chaque sous-formule de la forme $\alpha \mathsf{U} \beta$.

Étapes

1. Mise en forme normale négative de φ.
2. $S_0 = \{\varphi\}$.
3. Pour chaque état Y dans S :
   - Calculer un graphe orienté temporaire $\mathcal{G}_Y$.
   - Ajouter dans $\mathcal{A}$ les transitions et les nouveaux états correspondants grâce à $\mathcal{G}_Y$.

L’algorithme de traduction

L'algorithme de traduction

Définition (*NNF*)

Une formule est en **forme normale négative** (*NNF*) si elle est constituée uniquement des sous-formules suivantes :

- $\bot, p$ et $\neg p$ avec $p \in \mathrm{AP}$
- $\mathsf{X}\alpha$ et $\alpha \circledast \beta$ avec $\circledast \in \{\mathsf{U}, \mathsf{R}, \vee, \wedge\}$

L'algorithme de traduction

Définition (*NNF*)

Une formule est en **forme normale négative** (*NNF*) si elle est constituée uniquement des sous-formules suivantes :

- $\bot, p$ et $\neg p$ avec $p \in \mathrm{AP}$
- $\mathsf{X}\alpha$ et $\alpha \circledast \beta$ avec $\circledast \in \{\mathsf{U}, \mathsf{R}, \vee, \wedge\}$

Définition (*ensemble réduit*)

Un ensemble de formules Z est **réduit** si :

- toutes les formules de Z sont **réduites**, c'est-à-dire, de la forme p, $\neg p$ ou $\mathsf{X}\alpha$ avec $p \in \mathrm{AP}$
- $\bot \notin \mathrm{Z}$, et $\{p, \neg p\} \not\subseteq \mathrm{Z}$ pour tout $p \in \mathrm{AP}$.

L'algorithme de traduction

L'algorithme de traduction

Calcul de $\mathcal{G}_Y$

Soit Y = Z $\cup\{\alpha\}$ où α n'est pas réduite et si possible maximale (càd. n'est sous-formule d'aucune autre formule non réduite de Y). Les arêtes à partir de Y sont :

- Si $\alpha = \alpha_1 \vee \alpha_2$, $Y \to Z \cup \{\alpha_1\}$ et $Y \to Z \cup \{\alpha_2\}$.
- Si $\alpha = \alpha_1 \wedge \alpha_2$, $Y \to Z \cup \{\alpha_1, \alpha_2\}$
- Si $\alpha = \alpha_1 \mathsf{R}\, \alpha_2$, $Y \to Z \cup \{\alpha_1, \alpha_2\}$ et $Y \to Z \cup \{\mathsf{X}\alpha, \alpha_2\}$.
- Si $\alpha = \alpha_1 \mathsf{U}\, \alpha_2$, $Y \to Z \cup \{\alpha_2\}$ et $Y \to^{\alpha} Z \cup \{\mathsf{X}\alpha, \alpha_1\}$.

L'algorithme de traduction

Calcul de $\mathcal{G}_Y$

Soit Y = Z $\cup \{\alpha\}$ où α n'est pas réduite et si possible maximale (càd. n'est sous-formule d'aucune autre formule non réduite de Y). Les arêtes à partir de Y sont :

- Si $\alpha = \alpha_1 \vee \alpha_2$, $Y \to Z \cup \{\alpha_1\}$ et $Y \to Z \cup \{\alpha_2\}$.
- Si $\alpha = \alpha_1 \wedge \alpha_2$, $Y \to Z \cup \{\alpha_1, \alpha_2\}$
- Si $\alpha = \alpha_1 \mathsf{R}\ \alpha_2$, $Y \to Z \cup \{\alpha_1, \alpha_2\}$ et $Y \to Z \cup \{\mathsf{X}\alpha, \alpha_2\}$.
- Si $\alpha = \alpha_1 \mathsf{U}\ \alpha_2$, $Y \to Z \cup \{\alpha_2\}$ et $Y \to^{\alpha} Z \cup \{\mathsf{X}\alpha, \alpha_1\}$.

Cette construction est appliquée récursivement jusqu'à ce que toutes les feuilles du graphe soient réduites.

L'algorithme de traduction

L'algorithme de traduction

Calcul des transitions à partir de Y

Finalement, une fois $\mathcal{G}_Y$ calculé, sont ajoutées dans $\mathcal{A}$:

- les transitions suivantes $\{Y \rightarrow^{\Sigma_Z} \mathrm{next}(Z) \mid Z \in \mathrm{Red}(Y)\}$
- pour chaque sous-formule $\alpha = \alpha_1 \mathsf{U}\, \alpha_2$, les conditions d'acceptations $F_\alpha = \{Y \rightarrow^{\Sigma_Z} \mathrm{next}(Z) \mid Y \in S,\ Z \in \mathrm{Red}_\alpha(Y)\}$

L'algorithme de traduction

Calcul des transitions à partir de Y

Finalement, une fois $\mathcal{G}_Y$ calculé, sont ajoutées dans $\mathcal{A}$:

- les transitions suivantes $\{Y \to^{\Sigma_Z} \text{next}(Z) \mid Z \in \text{Red}(Y)\}$
- pour chaque sous-formule $\alpha = \alpha_1 \mathsf{U}\, \alpha_2$, les conditions d'acceptations $F_\alpha = \{Y \to^{\Sigma_Z} \text{next}(Z) \mid Y \in S,\ Z \in \text{Red}_\alpha(Y)\}$

Avec,

$$\text{Red}(Y) = \{Z \text{ réduit} \mid Y \to^{*} Z\}$$

$$\text{Red}_\alpha(Y) = \{Z \text{ réduit} \mid Y \to^{*\setminus\alpha} Z\}$$

$$\text{next}(Z) = \{\alpha \mid \mathsf{X}\alpha \in Z\}$$

$$\Sigma_Z = \bigcap_{p \in Z} \Sigma_p \cap \bigcap_{\neg p \in Z} \Sigma_{\neg p}$$

Un exemple pour $\varphi = p \mathsf{U} \mathsf{X} q$

(Un autre exemple pour $\varphi = p \; \mathsf{U} \; \mathsf{FX}q$)

L'implémentation d'Emile

L'implémentation de Thomas