Correction d'algorithmes de Karp et Miller en Coq

Thibault Hilaire, David Ilcinkas et Jérôme Leroux

LaBRI

11 Avril 2022

Index

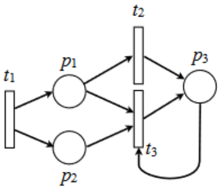
1. Introduction
   - Réseau de Petri
   - Beaux préordres
   - Couverture
   - Histoire et Contribution
2. Modélisation en Coq de Karp Miller
3. L'algorithme de Finkel, Haddad et Khmelnitsky
4. Conclusion

Definition

Un réseau de Petri est un tuple $\mathcal{N} = (P, T, Pre, Post)$ où

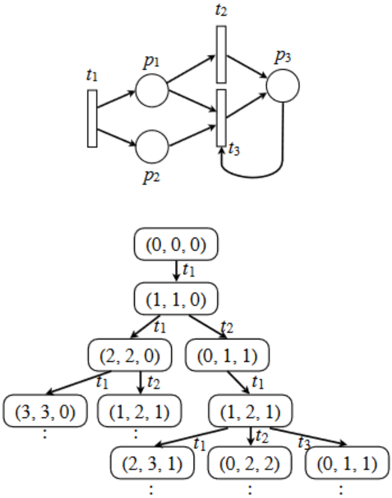
- P l'ensemble de places.
- T l'ensemble de transitions.
- $Pre \subseteq P \times T$
- $Post \subseteq T \times P$

On appelle marquage de $\mathcal{N}$ un élément m de $\mathbb{N}^{|P|}$.



Definition

m est **atteignable** depuis m_0 si il existe $t_1, \ldots, t_n \in T$ tels que $m_0 \xrightarrow{t_1} \ldots \xrightarrow{t_n} m$.



Definition

Un préordre $\preceq$ est un **beau préordre** (wqo en anglais) si et seulement si pour toute séquence infinie $x_1, x_2, \ldots$ il existe deux indices $i, j \in \mathbb{N}$ tels que $i < j$ et $x_i \preceq x_j$.

Exemples :

- $(\mathbb{N}, \leq)$ est un beau préordre.
- $(\mathbb{Z}, \leq)$ n'est pas un beau préordre.
- $(\mathbb{N}^k, \leq)$ où $(x_1, .., x_k) \leq (y_1, .., y_k)$ si $\forall i$ on a $x_i \leq y_i$, est un beau préordre.

Definition

m est **couvrable** depuis m_0 si il existe $t_1, ..., t_n \in T$ et m' tels que $m \leq m'$ et $m_0 \xrightarrow{t_1} ... \xrightarrow{t_n} m'$.

Definition

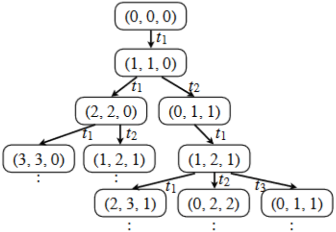
Un **ensemble de couverture** $Cover(m_0)$ désigne l'ensemble des marquages couvrables depuis m_0.

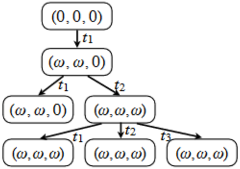
Exemples d'applications : Problèmes de sûreté, d'exclusion mutuelle

Definition

Un ideal I pour un préordre $\leq$ est un ensemble :

- clos par le bas
- dirigé vers le haut





Un petit historique :

- Karp, R.M., Miller, R.E. (1969)
- Finkel, A. (1991)
- Finkel, A., Geeraerts, G., Raskin, J.F., Van Begin, L. (2005)
- Geeraerts, G., Raskin, J.F., Van Begin, L. (2010)
- Reynier, P.A., Servais, F. (2013)
- Piipponen, A., Valmari, A. (2016)
- Reynier, P.A., Servais, F. (2019)
- Finkel, A., Haddad, S., Khmelnitsky, I. (2020)

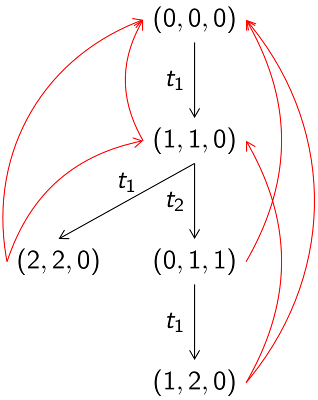
Mitsuharu Yamamoto, Shogo Sekine, and Saki Matsumoto. 2017. Formalization of Karp-Miller tree construction on petri nets. (CPP 2017)

Notre contribution : Nous essayons d'apporter une **certification en Coq** à l'algorithme de Finkel, Haddad et Khmelnitsky en **étendant** la modélisation en Coq de l'algorithme de Karp Miller par Mitsuharu et al.

```
Section PetriNetDef.

Record petri_net :=
  PetriNet {
      place : finType;
      transition : finType;
      Pre: transition -> {ffun place -> nat};
      Post : transition -> {ffun place -> nat};
    }.

Definition marking (pn : petri_net) := {ffun place pn -> nat}.
```



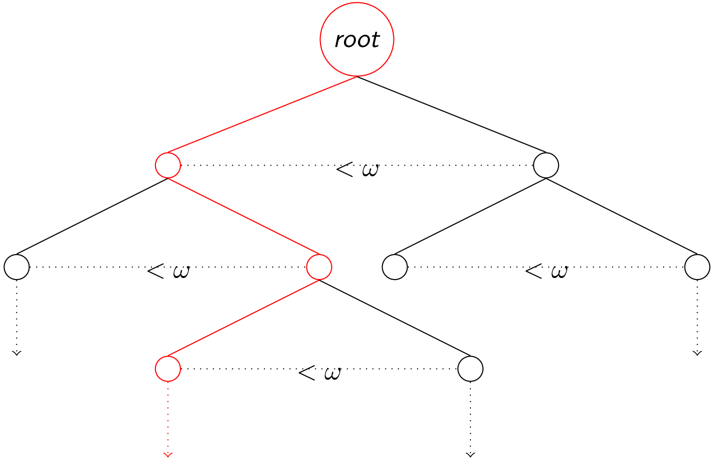
Fonction d'accélération Ω(marquage,arbre) :
$\forall p \; \Omega(m, T)(p) =$

- Si il existe un ancêtre m_a à m dans T tel que $m_a \leq m$ et $m_a(p) < m(p)$ alors ω
- Sinon $m(p)$

L'algorithme classique de Karp Miller :

INPUT : $\mathcal{N}$ un réseau de Petri, m_0.
OUTPUT : un arbre couvrant T
$T \leftarrow N(m_0)$
$Front = \{m_0\}$
Tant que $Front \neq \emptyset$ on sort un élément $m \in Front$.

- Si m admet un ancêtre dans T ayant la même étiquette alors on ne fait rien
- Sinon pour chaque transition t :
  si $m \xrightarrow{t} m_t$ alors on rajoute dans l'arbre T un fils $\Omega(m_t, T)$ à m.
  $Front \leftarrow Front \cup \Omega(m_t)$


root
$< \omega$
$< \omega$
$< \omega$
$< \omega$

Definition

Soit $(m_i)_{i \leq n}$ une séquence de marquages, $(m_i)_{i \leq n}$ est une **bonne séquence** si $\forall p \in P, \forall i < n$ on a $m_i(p) = \omega \implies m_{i+1}(p) = \omega$.

Definition

Soit s_1, s_2 deux bonnes séquences de marquages alors on dit que $s_2 \leq s_1$ si et seulement si il existe une $t \in T$ telle que $s_1 \xrightarrow{t} s_2$ avec accélération et que le dernier élément de s_2 n'apparaît pas dans s_1.

Definition

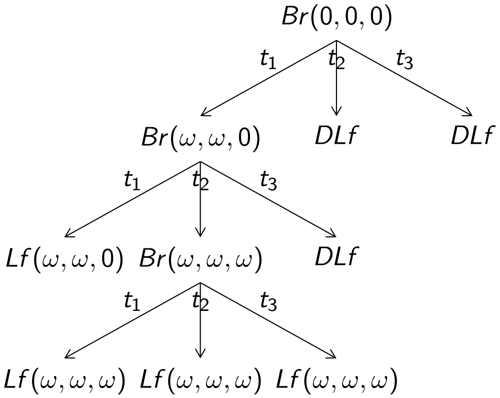
Une relation $\preceq$ est **bien fondée** (wf en anglais) si et seulement si il n'existe pas de séquence infinie décroissante.

```
Variable A : Type.
Variable R : A -> A -> Prop.

Inductive Acc (x: A) : Prop :=
    Acc_intro : (forall y:A, R y x -> Acc y) -> Acc x.


Definition well_founded := forall a:A, Acc a.
```

```
Inductive kmtree :=
  DLf | Lf of markingc | Br of markingc & {ffun transition -> kmtree}.
```



```
Definition mkkmtreeF (mcp : {mcp | mkkmt_inv mcp.1 mcp.2})
           (F' : forall y, mkkmtreeT y mcp -> kmtree) :=
  Br (val mcp).2 [ffun t : transition =>
                  match dec_op (nextma (val mcp) t) with
                  | inleft (exist mcp' Hnext) =>
                    match dec (mcp'.2 \in mcp'.1) with
                    | left _ => Lf mcp'.2
                    | right Hnotin => F' mcp' _
                    end
                  | inright _ => DLf
                  end].
```

$$\begin{tikzcd} m \arrow[r, "a"] \arrow[rr, bend left, "compo(a, b)"] & m' \arrow[r, "b"] & m'' \end{tikzcd}$$

On peut créer des **méta-transitions** qui se comportent comme des transitions mais :

- peuvent simuler des séquences de transitions.
- peuvent avoir des ω dans leur *Pre* et *Post*.
- peuvent simuler des accélérations.

$$(x, y) \xrightarrow{t} (x-1, y+1)$$

On peut créer **l'accélération** a :

- $Pre(a) = (\omega, 0)$
- $Post(a) = (\omega, \omega)$

$$m \xrightarrow{a} m' \quad \Longleftrightarrow \quad \underbrace{m \xrightarrow{t} \cdots \xrightarrow{t} m'}_{\omega}$$

L'algorithme de Finkel, Haddad et Khmelnitsky :

INPUT : $\mathcal{N}$ un réseau de Petri, m_0.

OUTPUT : un arbre couvrant T tel que T est la **plus petite couverture** de $\mathcal{N}$,

Les noeuds de T forment une **antichaîne**.

$$(0, 0, 0) \qquad\qquad \emptyset$$

L'algorithme de Finkel, Haddad et Khmelnitsky :
INPUT : $\mathcal{N}$ un réseau de Petri, m_0.
OUTPUT : un arbre couvrant T tel que T est la **plus petite couverture** de $\mathcal{N}$,
Les noeuds de T forment une **antichaîne**.

$$\begin{tikzcd} (0,0,0) \arrow[d, "t_1"'] & \emptyset \\ (1,1,0) \arrow[u, bend left, "a_1"] & \end{tikzcd}$$

L'algorithme de Finkel, Haddad et Khmelnitsky :
INPUT : $\mathcal{N}$ un réseau de Petri, m_0.
OUTPUT : un arbre couvrant T tel que T est la **plus petite couverture** de $\mathcal{N}$,
Les noeuds de T forment une **antichaîne**.

$$(0,0,0) \qquad \{a_1\}$$

L'algorithme de Finkel, Haddad et Khmelnitsky :

INPUT : $\mathcal{N}$ un réseau de Petri, m_0.

OUTPUT : un arbre couvrant T tel que T est la **plus petite couverture** de $\mathcal{N}$,

Les noeuds de T forment une **antichaîne**.

$$
\begin{tikzcd}
& (\omega,\omega,0) \arrow[dl, "t_1"'] \arrow[dr, "t_2"] & & \{a_1\} \\
(\omega,\omega,0) \arrow[ur, bend left, "="] & & (\omega,\omega,1) \arrow[ul, bend right, "a_2"'] &
\end{tikzcd}
$$

L'algorithme de Finkel, Haddad et Khmelnitsky :

INPUT : $\mathcal{N}$ un réseau de Petri, m_0.
OUTPUT : un arbre couvrant T tel que T est la **plus petite couverture** de $\mathcal{N}$,
Les noeuds de T forment une **antichaîne**.

$(\omega, \omega, 0)$ $\{a_1, a_2\}$

L'algorithme de Finkel, Haddad et Khmelnitsky :
INPUT : $\mathcal{N}$ un réseau de Petri, m_0.
OUTPUT : un arbre couvrant T tel que T est la **plus petite couverture** de $\mathcal{N}$,
Les noeuds de T forment une **antichaîne**.

$$(\omega, \omega, \omega) \qquad\qquad \{a_1, a_2\}$$

Lors de notre modélisation en Coq nous avons rencontré des difficultés sur les points suivants :

- La modélisation du non déterminisme de l'algorithme
- La terminaison

En conclusion :

- On a compris la modélisation et certification en Coq par Mitsuharu Yamamoto, Shogo Sekine, and Saki Matsumoto de l'algorithme de Karp Miller.
- On a étudié l'amélioration de l'algorithme de Karp Miller par Alain Finkel, Serge Haddad, Igor Khmelnitsky.
- On a commencé à modéliser en Coq l'algorithme d'Alain Finkel, Serge Haddad, Igor Khmelnitsky en étendant la modélisation de Mitsuharu Yamamoto, Shogo Sekine, and Saki Matsumoto.

[FGRVB05] Alain Finkel, Gilles Geeraerts, Jean-François Raskin, and Laurent Van Begin. A counter-example to the minimal coverability tree algorithm. *Université Libre de Bruxelles, Tech. Rep*, 535, 2005.

[FHK20] Alain Finkel, Serge Haddad, and Igor Khmelnitsky. Minimal coverability tree construction made complete and efficient. In Jean Goubault-Larrecq and Barbara König, editors, *Foundations of Software Science and Computation Structures - 23rd International Conference, FOSSACS 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25-30, 2020, Proceedings*, volume 12077 of *Lecture Notes in Computer Science*, pages 237–256. Springer, 2020.

[Fin91] Alain Finkel. The minimal coverability graph for petri nets. In *International Conference on Application and Theory of Petri Nets*, pages 210–243. Springer, 1991.

[GRVB10] Gilles Geeraerts, Jean-François Raskin, and Laurent Van Begin. On the efficient computation of the minimal coverability set of petri nets. *International Journal of Foundations of Computer Science*, 21(02) :135–165, 2010.

[KM69] Richard M Karp and Raymond E Miller. Parallel program schemata. *Journal of Computer and system Sciences*, 3(2) :147–195, 1969.

[PV16] Artturi Piipponen and Antti Valmari. Constructing minimal coverability sets. *Fundamenta Informaticae*, 143(3-4) :393–414, 2016.

[RS13] Pierre-Alain Reynier and Frédéric Servais. Minimal coverability set for petri nets : Karp and miller algorithm with pruning. *Fundamenta Informaticae*, 122(1-2) :1–30, 2013.

[RS19] Pierre-Alain Reynier and Frédéric Servais. On the computation of the minimal coverability set of petri nets. In *International Conference on Reachability Problems*, pages 164–177. Springer, 2019.

[YSM17] Mitsuharu Yamamoto, Shogo Sekine, and Saki Matsumoto. Formalization of karp-miller tree construction on petri nets. In Yves Bertot and Viktor Vafeiadis, editors, *Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs, CPP 2017, Paris, France, January 16-17, 2017*, pages 66–78. ACM, 2017.