Réseaux de Pétri colorés: **CPNTools**

Alain Freddy Kiraga

Automne 2009

Plan

Principale parties

- **Index**
  - **Tool Box**: Ensemble d'outils pour éditer et analyser le CPN
  - **Help**: Liens vers un ensemble de pages Web utiles
  - **Options**: Variation d'options pour la simulation,$\cdots$
  - **CPNs ouverts dans l'outil**
- **Espace de travail pour éditer le CP-Net (Marking Menu, Tool Box)**
  - **Binder**: Conteneur de Pages
  - **Page**: Contient des élements CP-Net

Outils

- **Créer le CP-Net**
  - Éditer le Net(Places, Transitions, Arcs)
  - Ajouter/Éditer les déclaration(color sets, Fonctions,...)
  - Ajouter/Editer les inscriptions (sur les places, transitions et arcs)
- **Déboguer et analyser le Net**
  - Analyse syntaxique
  - Simulation
  - Analyse de l'espace d'états(Fairness,liveness,deadlock,...)

Élements de l'entrée "nameNet.cpn"

- **History**: liste des commandes effectuées sur le CP-Net
- **Steps**: nombre d'étapes effectués dans la simulation
- **Time**: le modèle de temps courant
- **Declaration**: déclarations des ensembles de couleurs, fonctions, constantes . . . écrites dans le langage CPN ML

Palettes d'outils

- **Auxilliary**: Créer des élements auxilliaire du CP-Net(aucune sémantique)
- **Create**: Créer les structures basiques du CP-Net(places, transitions, arcs,...)
- **Hierarchy**: Créer des structures hiérarchique du CP-Net(modularité)
- **Net**: nouveau, enregistrer, charger,...
- **Monitoring**: créer les moniteurs
- **Simulation**: Simuler le CP-Net, Evaluer le code ML
- **State Space**: Caculer l'espace d'états, graphe connexe,...
- **Style**: Mise en forme
- **View**: pour la visualisation

Plan

Places, Transitions, Arcs

- **Place**
  - **Ensemble de couleurs**: Type de tous les jetons que peut contenir une place
  - **Marquage initial**: Expression multi-set exprimant le nombre de jetons initial(optionel)
  - **Nom de la place**(optionel)

Places, Transitions, Arcs

- **Transition**
  - **Nom de la transition**(optionel)
  - **Garde**: Expression booléenne CPN ML

$$\textbf{Syntaxe} : [Exp]\ ou\ [Exp1, Exp2, \ldots, Expn]$$

  - **Temps**: Temps d'attente (Expression entière positive) ajoutée au temps courant du modèle
  - **Segment de code**
    - ⋆ **input(optionel)**: Tuple des variables CPN utilisées dans le code
    - ⋆ **output(optionel)**: Tuple des variables CPN produites comme résultat de l'exécution du code
    - ⋆ **action(optionel)**: Expression ML (pas de déclaration d'ensembles de couleurs, variables ou références)

Place, Transitions, Arcs

- **Arcs**
  - **Inscription**: Expression CPN ML pouvant être évaluée à un multi-set ou à un seul élément

Plan

- **Unit colour set**

syntaxe

colset name = *unit*[*with new_unit*]

Exemple

colset U = *unit*;
colset E = *unit with e*;

- **Boolean colour set**

syntaxe

colset name = *bool*[*with new_false*; *new_true*]

Exemple

colset B = *bool*;
colset B = *bool with*(*no*, *yes*);

opérations

not, andalso, orelse

- **Integer colour set**

syntaxe

colset name = *int*[*with int_expr*1 · ·*int_expr*2]

Exemple

colset ENT = *int*;
colset ENT_1 = *int with* 1 · ·10;

opérations

*+, -, div, *, mod,...*

- **String colour set**

syntaxe

colset name = *string*[*with string_exp*1 ··*string_exp*2[*and int_exp*1 ··*int_exp*2]]

Exemple

colset s = *string*;
colset s_1 = *string with* "*a*" ··"*z*";
colset s_2 = *string with* "*a*" ··"*d*" *and* 2 ··5;

opérations

size, substring, explode, implode

- **Enumerated colour set**

syntaxe

$colset\ name = with\ id_0 | id_1 | \cdots | id_n;$

Exemple

$colset\ day = with\ Mon | Tue | Wed | Thu | Fri | Sat | Sun;$

opérations

lt,legal,...

- **Index colour set**

syntaxe

colset name = *index id with int_exp*1 · ·*int_exp*2;

Exemple

colset PH = *index ph with* 1 · ·2;

opérations

lt,legal,...

- **Product colour set**

syntaxe

$colset\ name = product\ name_1 * \cdots * name_n;$

Exemple

$colset\ INT = int;$
$colset\ BOOL = bool;$
$colset\ P = INT * BOOL;$
Les éléments de P sont (I_1, B_1) avec $I_1 \in INT$ et $B_1 \in BOOL$

opérations

$\#, -, \cdots$

- **Record colour set**

syntaxe

$colset\ name = record\ id_1 : name_1 * \cdots * id_n : name_n;$

Exemple

$val\ NbrSite = 4;$
$colset\ INT = int;$
$colset\ SITES = index\ S\ with\ 1 \cdot\cdot NbrSite;$
$colset\ PACK = record\ se : SITES * re : SITES * no : INT$

opérations

$\#, \cdots$

- **List colour set**

syntaxe

colset name = *list name*$_0$ [*with int_exp*1 · · *int_exp*2];

Exemple

colset INTList = *list int*;
colset BOOLList = *list bool with* 2 · · 4;

opérations

length, tl, hd,...

- **Union colour set**

syntaxe

$colset\ name = union\ id_1[: name_1] + id_2[: name_2] + \cdots id_n[: name_n];$

Exemple

$colset\ Data = string\ with\ "a" \cdot\cdot "z"\ and\ 0 \cdot\cdot 5;$
$colset\ Paquet = union\ DATA : Data + Ack;$

opérations

legal, mkstr,...

- **Alias colour set**

syntaxe

colset name = *newname*;

Exemple

colset AllNumber = *INT*;
colset DayOff = *WeekEnd*;

- **Timed colour set**

syntaxe

colset name $=\cdots$ *timed*;

Exemple

colset clock $=$ *int timed*;
colset P $=$ *product bool* $*$ *clock timed*;

opérations

legal, lt,...

- **Valeurs**
  Lie une valeur à un identificateur

syntaxe

val id $=$ *exp*;
id $\longrightarrow$ *identificateur*
exp $\longrightarrow$ expression CP-ML incluant les expressions *muli-set*

- **Variables**
  Identificateur dont la valeur peut changer durant l'exécution

syntaxe

var id $=$ *cs_name*;
id $\longrightarrow$ *identificateur*
cs_name $\longrightarrow$ ensemble de couleurs déclaré

- **Références globales**
  Similaire aux pointeurs en C.

syntaxe

globref id $=$ *exp*;
id $\longrightarrow$ *identificateur*
exp $\longrightarrow$ valeur assignée

Exemple

globref i $=$ 10;
globref date $=$ (*Mon*, 10);

opérations

inc, dec,...

- **Fonctions**

Syntaxe

$$\begin{aligned} fun \quad & id\ pat_1 = exp_1 \\ | \quad & id\ pat_2 = exp_2 \\ | \quad & \cdots \\ | \quad & id\ pat_n = exp_n; \end{aligned}$$

$exp_1, exp_2, \cdots, exp_n$ sont de même type

Exemple

$$fun\ Comp(x, y) = (x <> y);$$

- **Déclarations locales avec *let-in***
  Declaration des variables locales à une fonction
  Peut être utilisé dans un segment de code

Syntaxe

let
  val $pat_1 = exp_1$
  val $pat_2 = exp_2$
  ...
  val $pat_n = exp_n$
in
  exp
end;

Exemple

fun member (e, l)=
if ...else
let
  val $head = List.hd\ l$
  val $tail = List.tl\ l$
in
  if $e = head$ then $true$
  else member $(e, tail)$
end;

Syntaxe 1

if bool-exp
then exp_1
else exp_2;

Syntaxe 2

case exp of
$pat_1 \Rightarrow exp_1$
$|pat_2 \Rightarrow exp_2$
$|\ldots$
$|pat_n \Rightarrow exp_n$;

Exemple

if $x = p$ then 1‘e else 2‘e

Multi-sets

Syntaxe

i‘c

où i: entier non négatif
c: un élément d’un ensemble de couleurs

Exemple

3‘*true* $+ +$ 2‘*false*
multi-set de 3 instances *true* et 2 instances *false*

Référence:

http://wiki.daimi.au.dk/cpntools-help/_home.wiki