



Vérification avec les outils TINA

Bernard Berthomieu Didier le Botlan
Silvano Dal Zilio François Vernadat

LAAS/CNRS/INSA

STP GdR MACS
Albi, 29 Mars 2012

Processus de vérification

3 étapes :

Modélisation

- ⇒ modèle formel M (du comportement) de l'application
- ⇒ propriétés attendues P , dans la logique L

Abstraction

- préservant les formules de L
- ⇒ graphe d'états abstrait A fini

Vérification

- des formules P sur A (Model-Checking)
- ⇒ VRAI ou un contre-exemple

Avec TINA (Time Petri Net Analyzer)

Modèles formels

Réseaux de Petri Temporels

+ Priorités, Données (C), Chronomètres (préemption)

Descriptions de haut niveau en FIACRE (compilées)

Abstractions

Graphes de couverture

Espaces d'états exacts

Réductions ordre partiel

Graphes de classes (plusieurs abstractions)

Vérification

State/Event LTL (natif)

Mu-Calcul (natif)

exportation abstraction vers outils CADP, MEC

Boite à outils Tina

nd

Editeur graphique et textuel de réseaux temporels et d'automates
Interfacé avec outils d'abstraction et de vérification

tina, sift

Génèrent abstractions de comportements
Préservant famille choisie de propriétés

selt, muse

Vérificateurs de modèles pour State/Event LTL et μ -calcul

plan

Synthétiseur de chemins temporisés

play

Simulateur pas à pas

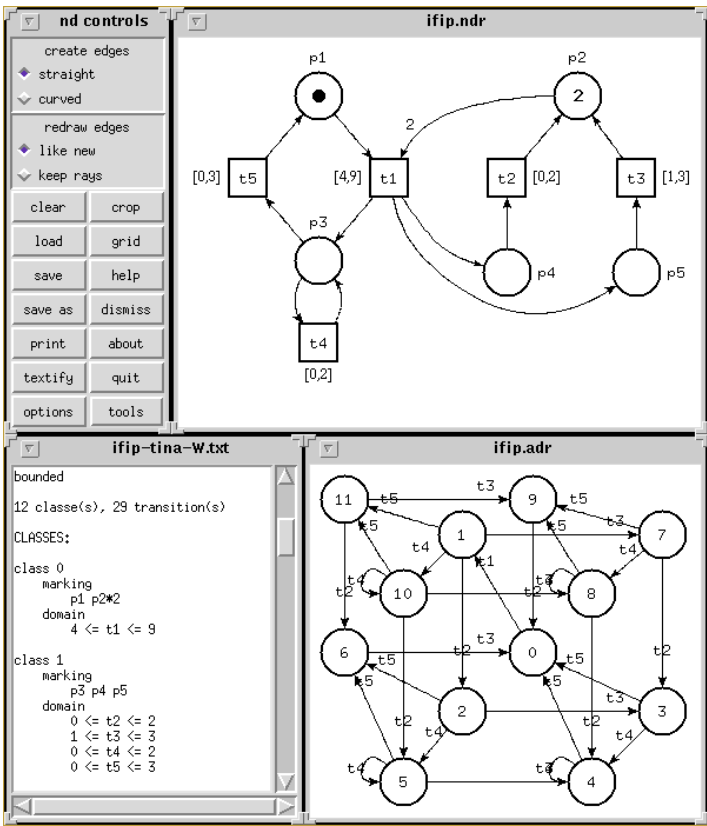
struct

Analyse structurelle

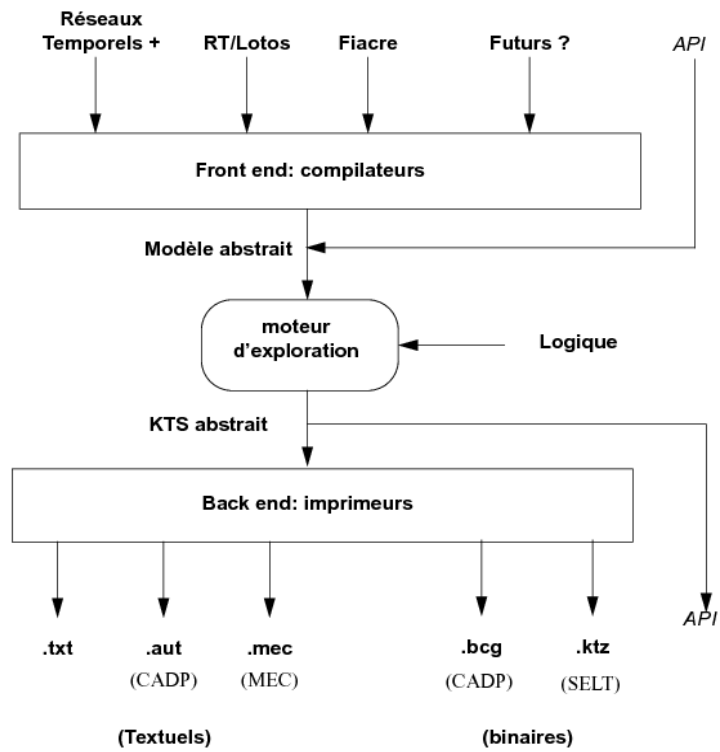
pathto, ktzio, ndrrio

utilitaires, conversions, etc

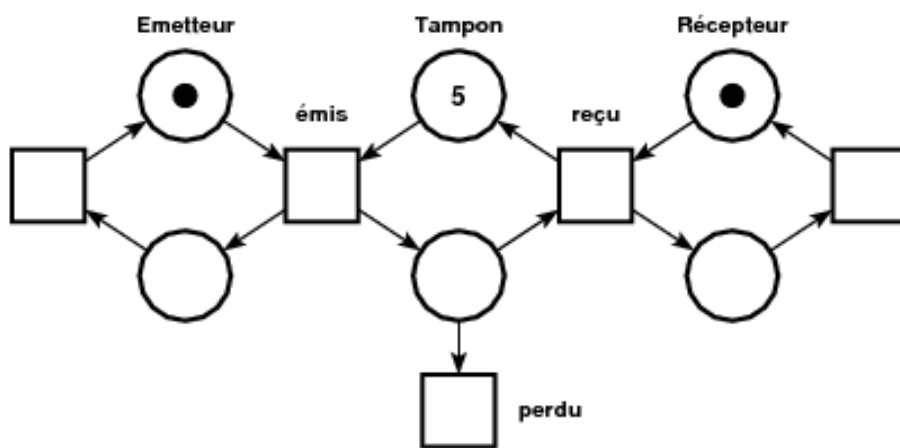
Session nd



tina – Architecture



Réseaux de Petri



Pré-conditions — Action → Post-conditions
Expriment nativement choix ET parallélisme
Caractère "borné" décidable

Réseaux de Petri – Abstractions

Abstractions

Graphes de couverture (bornes sup, détectent places bornées)

Graphes des marquages (espace d'états exact)

Réductions ordre partiel

- Ensembles persistants (blocages)

- Pas couvrants (blocages+)

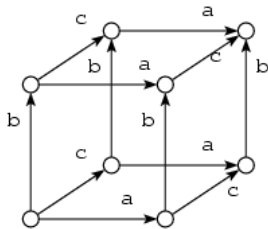
- Pas persistants (blocages+)

Analyse structurelle

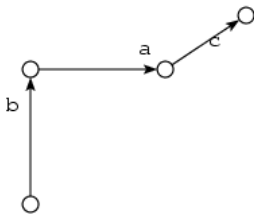
- Surapproximations par ensembles semi-linéaires

Réductions ordre partiel

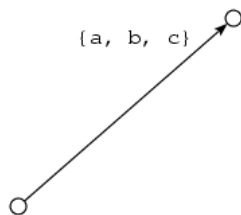
Explorations implicites



Exhaustif :
 2^n états, $n \times 2^{n-1}$ transitions

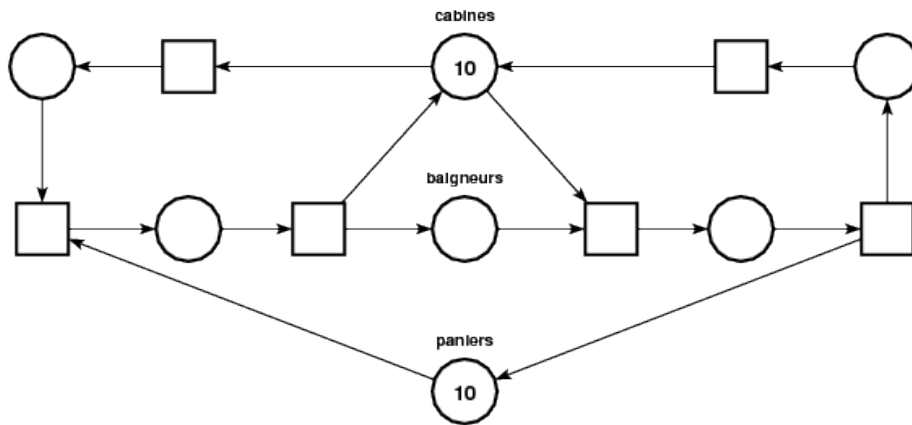


Ensembles persistants :
 $(n + 1)$ états, n transitions



Graphes de pas couvrants :
2 états, 1 transitions

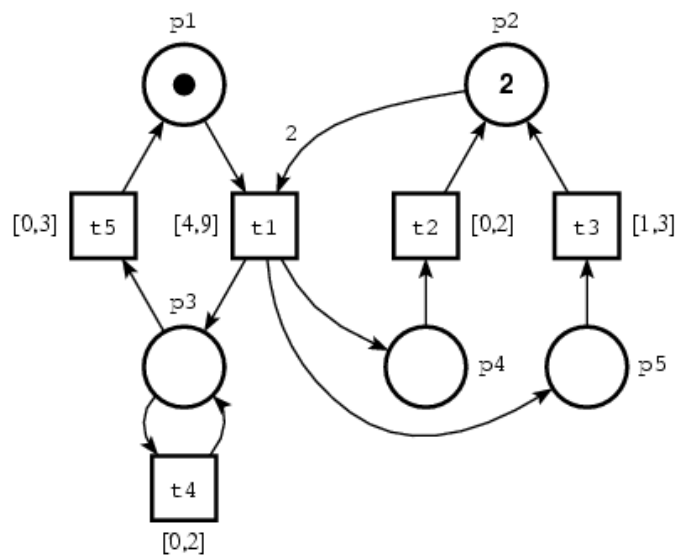
Piscine



		<i>Exact</i> tina -R	<i>Pas couvrants</i> tina -V	<i>Ens persistants</i> tina -P	<i>Pas persistants</i> tina -Q
$K = 10$	tats	7006	367	97	87
	s	0	0	0	0
$K = 100$	tats	$\sim 280M$	39517	997	897
	s	8800	0.3	0	0
$K = 1000$	tats	?	$\sim 4M$	9997	8997
	s	?	30	0	0
$K = 10000$	tats	?	$\sim 400M$	99997	89997
	s	?	4500	0.5	0.5

Réseaux Temporels (Merlin 74)

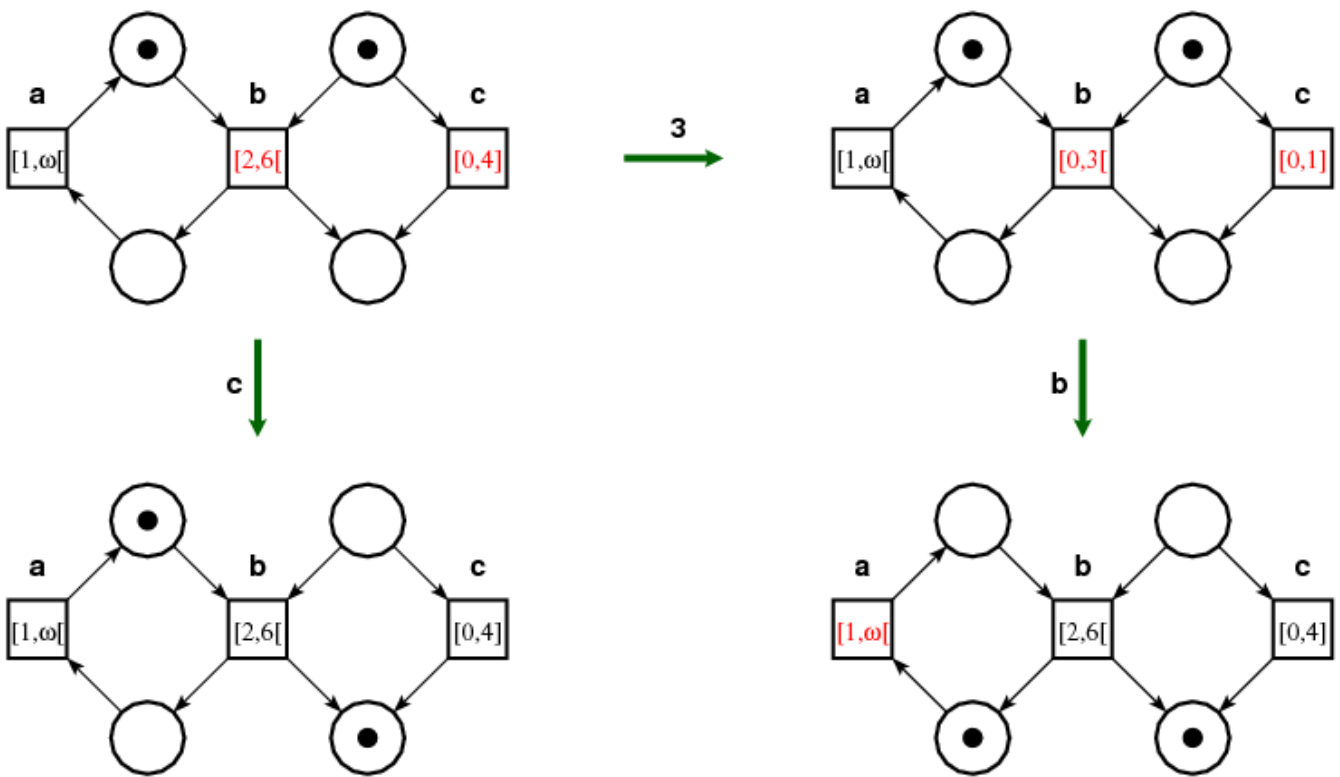
RdP + Intervalles temporels



Espaces d'états infinis (temps dense)

Caractère borné indécidable (mais conditions suffisantes)

États et transitions d'états



États et transitions d'états

$$E_0 = (m_0, l_0)$$

$$m_0 : p_1, p_2(2)$$

l_0 : solutions en t_1 of

$$4 \leq t_1 \leq 9$$

$$E_0 \xrightarrow{t_1 @ \theta_1} E_1 = (m_1, l_1) \text{ avec } (\theta_1 \in [4, 9]) :$$

$$m_1 : p_3, p_4, p_5$$

l_1 : solutions en (t_2, t_3, t_4, t_5) of

$$0 \leq t_2 \leq 2$$

$$1 \leq t_3 \leq 3$$

$$0 \leq t_4 \leq 2$$

$$0 \leq t_5 \leq 3$$

$$E_1 \xrightarrow{t_2 @ \theta_2} E_2 = (m_2, l_2) \text{ avec } (\theta_2 \in [0, 2]) :$$

$$m_2 : p_2, p_3, p_5$$

l_2 : solutions en (t_3, t_4, t_5) of

$$\max(0, 1 - \theta_2) \leq t_3 \leq 3 - \theta_2$$

$$0 \leq t_4 \leq 2 - \theta_2$$

$$0 \leq t_5 \leq 3 - \theta_2$$

TPNs – Abstractions

Abstractions : Graphes de classes d'états

Classe d'état = ensemble d'état

classe = marquage (état discret) + polyèdre (information temporelle)

Differentes constructions, préservant :

Marquages + traces (LTL) (SCG [BM83], Essential states [Popova91])

Marquages (SCG_C)

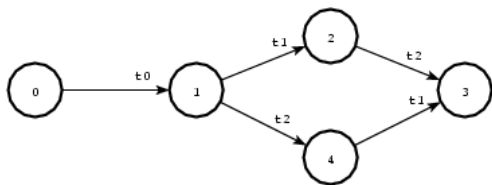
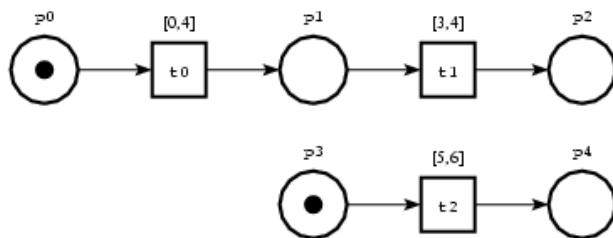
Marquages + états + traces (SSCG [BV03])

Marquages + états ($SSCG_C$)

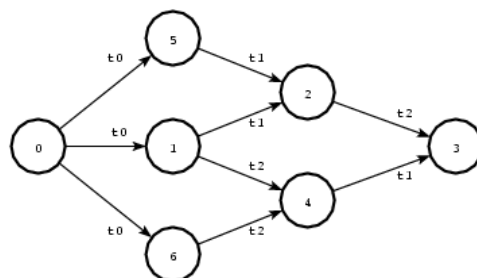
Marquages + états + traces + branchements (CTL*) (ASCG [BV03])

Théorème : Abstractions finies ssi réseau est borné

Exemples

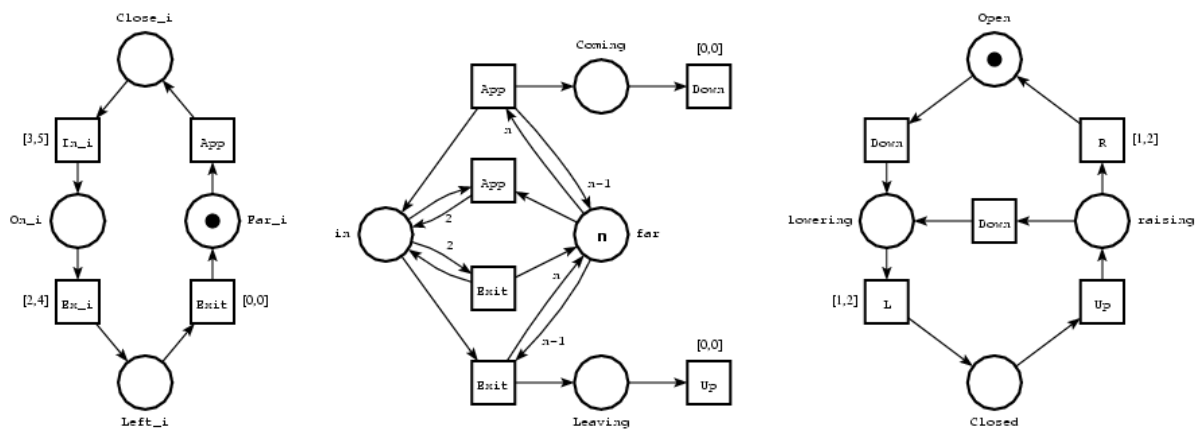


- $C_0 = (p_0 p_3, \{0 \leq t_0 \leq 4, 5 \leq t_2 \leq 6\})$
- $C_1 = (p_1 p_3, \{3 \leq t_1 \leq 4, 1 \leq t_2 \leq 6\})$
- $C_2 = (p_2 p_3, \{0 \leq t_2 \leq 3\})$
- $C_3 = (p_2 p_4, \{\})$
- $C_4 = (p_1 p_4, \{0 \leq t_1 \leq 3\})$



- $C_0 = (p_0 p_3, \{0 \leq t_0 \leq 0, 0 \leq t_2 \leq 0\})$
- $C_1 = (p_1 p_3, \{0 \leq t_1 \leq 0, 1 \leq t_2 \leq 3\})$
- $C_2 = (p_2 p_3, \{3 \leq t_2 \leq 6\})$
- $C_3 = (p_2 p_4, \{\})$
- $C_4 = (p_1 p_4, \{1 \leq t_1 \leq 4\})$
- $C_5 = (p_1 p_3, \{0 \leq t_1 \leq 0, 0 \leq t_2 \leq 1\})$
- $C_6 = (p_1 p_3, \{0 \leq t_1 \leq 0, 3 \leq t_2 \leq 4\})$

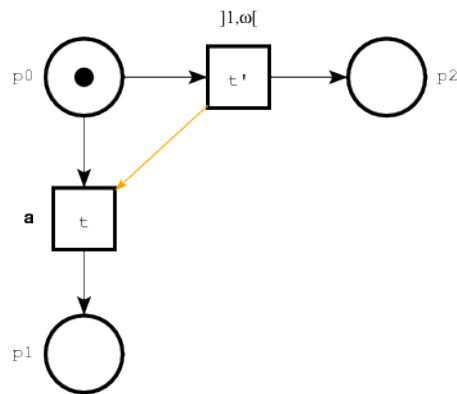
Passage à niveau



	M tina -M	$M + LTL$ tina -W	E tina -E	$E + LTL$ tina -S	$E + CTL$ tina -A	$M + LTL (discret)$ tina -D	$M + LTL (discret)$ tina -F
(1 train)	Classes 10	11	10	11	12	13	23
	Transitions 13	14	13	14	16	27	36
(2 trains)	Classes 37	123	41	141	195	116	382
	Transitions 74	218	82	254	849	198	373
(3 trains)	Classes 172	3101	232	5051	6973	1550	2280
	Transitions 492	7754	672	13019	49818	5823	5275
(4 trains)	Classes 1175	134501	1807	351271	356940	22268	28830
	Transitions 4534	436896	7062	1193376	1447835	91256	81077
(5 trains)	Classes 10972	855762	18052	35945411	23081275	313214	372264
	Transitions 53766	34337748	89166	151908273	279572133	1397517	1245355
(6 trains)	Classes 128115	697913229	217647	?	?	4299116	4830558
	Transitions 760538	3334109864	1297730	?	?	20886774	18833697

Priorités

Réseaux Temporels + Priorités (PrTPN)



Priorités augmentent l'expressivité des TPN : PrTPN bornés \approx TA

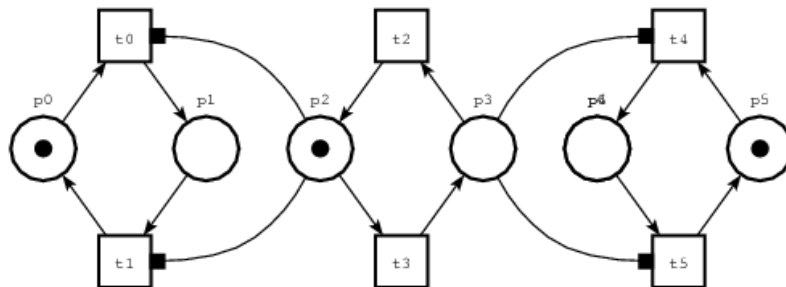
Pour PrTPN, l'écoulement du temps peut rendre une transition non tirable

Abstractions

Classes d'états restent applicables (SSCG, ASCG, mais pas SCG)

Suspension et reprise de transitions

Réseaux Temporels à Chronomètres (SwTPN)



Une transition sensibilisée peut être *Active* ou *Suspendue*

Applications : systèmes ordonnancés, préemption temporelle

Abstractions

Graphes de classes adaptables, MAIS accessibilité indécidable ...

Surapproximations fournissent des conditions suffisantes ou nécessaires

Prise en compte des données

Time Transition System =

Systemes de Keller

marquages \Rightarrow états (vecteurs d'entiers)

transitions "additives" \Rightarrow transitions arbitraires

+ contraintes temporelles à la TPN

On perd : décidabilité du caractère borné

en Tina :

format tts = TPN + traitement de données en C synchronisé

Abstractions

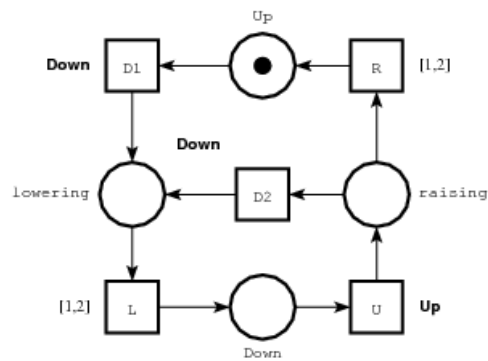
Méthode des classes d'états reste applicable

Descriptions de haut niveau — FIACRE

cf Projets COTRE, TOPCASED, OpenEmbeDD

```
process barrier_p [ Down, Lower, Up, Raise : none] is
  states up lowering down raising
  init up
  from up      Down; to lowering
  from lowering Lower; to down
  from down    Up; to raising
  from raising select Raise; to up
                []      Down; to lowering
end
```

```
component barrier [Down, Up : none] is
  port Lower, Raise : none in [1,2]
  barrier_p [Down, Lower, Up, Raise]
```



Vérification – State/Event-LTL (CMU)

Propositions atomiques

d'états

de transitions

Opérateurs logiques et temporels

Trace = suite alternée infinie d'états et de transitions

(Pour toute trace)

P	P vraie dans le premier état (transition)
$\bigcirc P$	P vraie dans le prochain état (transition)
$\square P$	P vraie dans tout état (transition)
$\diamond P$	P vraie dans un état (transition) au moins
$\square \diamond P$	P vraie infiniment souvent
$\square(P \Rightarrow \diamond Q)$	Q "répond" à P

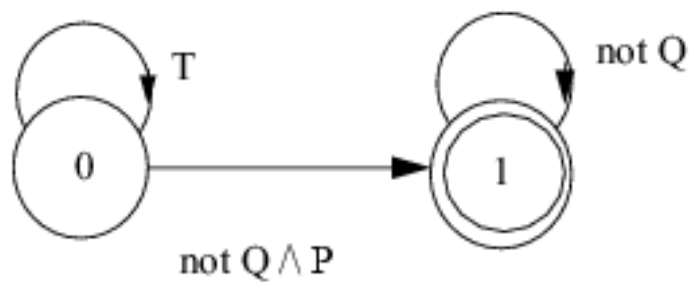
Specification patterns : <http://patterns.projects.cis.ksu.edu>

Vérifier F sur un KTS

Automates de Büchi

$$F = \Box(P \Rightarrow \Diamond Q)$$

Automate associé à $\neg F$:



Méthode

On construit synchronisation du KTS et de l'Automate de $\neg F$

Un contre exemple est un chemin contenant un circuit contenant un état acceptant

Le vérificateur SELT

Formules

S/E-LTL + arithmétique, e.g.

$\Box(t1 \Rightarrow \Diamond(p2 \geq p3 + p4 \vee p6))$

Contre exemples Abrégés

```
- [] (t1 ==> <> t4);
```

```
FALSE
```

```
state 0: p1 p2*2
```

```
-t1 ... (preserving - t4 /\ t1)->
```

```
* [accepting] state 12: p3 p4 p5
```

```
-t5 ... (preserving - t4)->
```

```
state 12: p3 p4 p5
```

Peuvent être rejoués dans le simulateur Tina

Vérification LTL “off-line”

générer l'espace d'états :

sift app.tts app.ktz

vérifier formule {et extraire contre-exemple} :

selt app.ktz -f “[] (p3 + p4 <= 1)” -S app.scn

temporiser le contre-exemple :

plan -S app.scn app.tts appt.scn

rejouer/analyser le contre-exemple temporisé :

play app.tts

? | appt.scn

? etc

Vérification à la volée avec sift

générer/vérifier espace d'états :

```
sift -f "p3 + p4 <= 1" app.tts app.ktz
```

générer contre-exemple depuis numéro d'état :

```
patho n app.ktz -t > app.scn
```

temporiser le contre-exemple :

```
plan -S app.scn app.tts appt.scn
```

rejouer/analyser le contre-exemple temporisé :

```
play app.tts
```

```
? | appt.scn
```

```
? etc
```

Prospective

Descriptions de haut niveau

Poursuite intégration Fiacre

Passage à l'échelle

Parallélisation Exploration et vérification

Démonstrateur Mercury multi-core, mémoire partagée (NUMA) :
speedup 8 à 10 sur machine 8*2 cores

Vérification probabiliste parallèle

Optimisation de modèles

Simplification de modèles (réductions, etc)

Prise en compte informations spécifiques (symetries, invariants)

Interprétation abstraite, pour systèmes infinis (Fiacre)

Liens

Outils

<http://www.laas.fr/tina>

<http://www.laas.fr/fiacre>

Méthodes

<http://www.laas.fr/tina/papers.php>

<http://www.laas.fr/fiacre/papers.php>