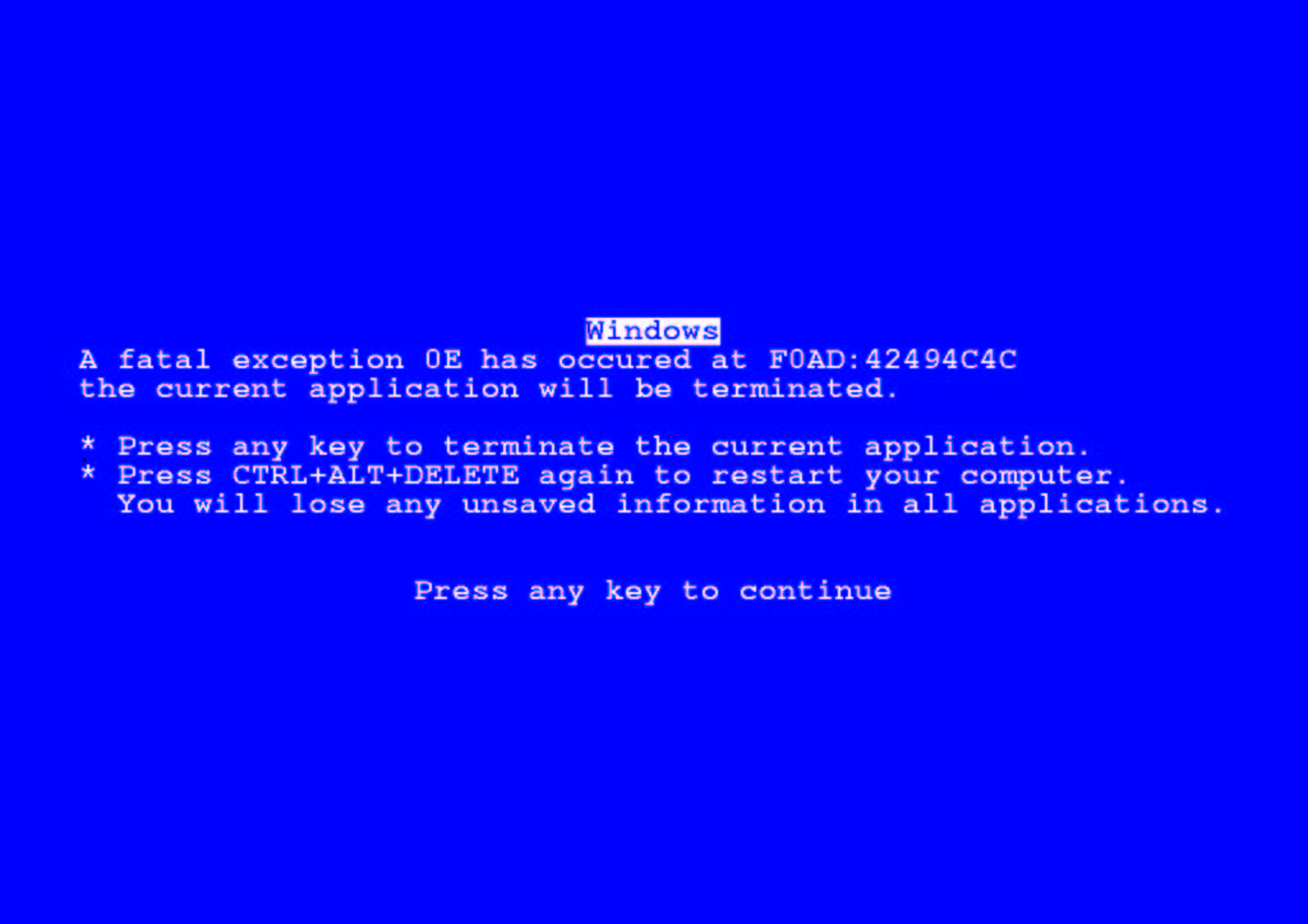
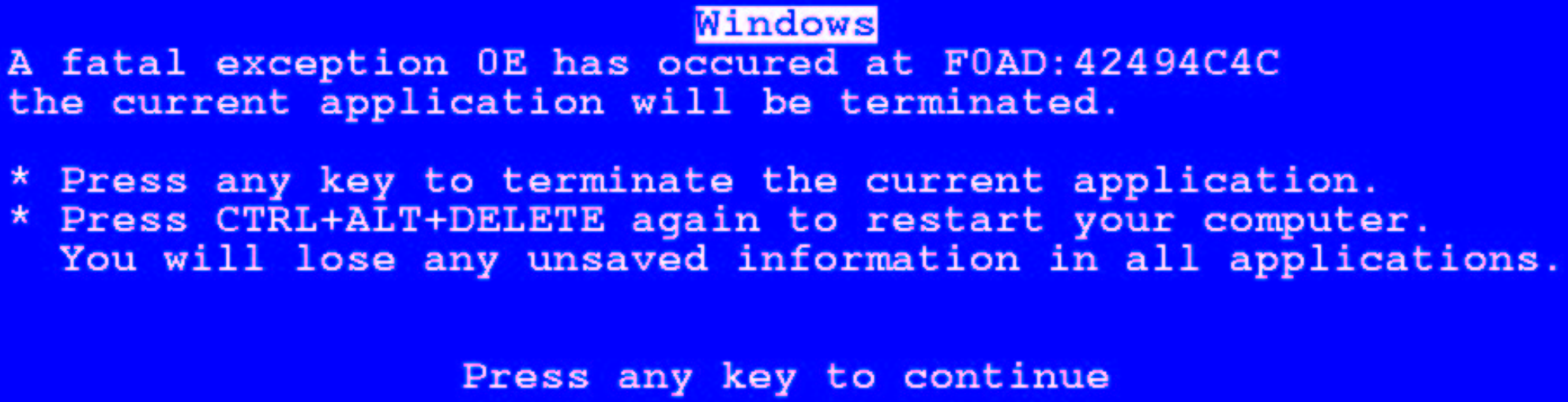
Techniques for verification of concurrent systems with invariants

Pascal Fontaine

Liège, le 17 septembre 2004


Windows
A fatal exception 0E has occured at F0AD:42494C4C
the current application will be terminated.
* Press any key to terminate the current application.
* Press CTRL+ALT+DELETE again to restart your computer.
You will lose any unsaved information in all applications.
Press any key to continue


Windows
A fatal exception 0E has occured at F0AD:42494C4C
the current application will be terminated.
* Press any key to terminate the current application.
* Press CTRL+ALT+DELETE again to restart your computer.
You will lose any unsaved information in all applications.
Press any key to continue

Introduction

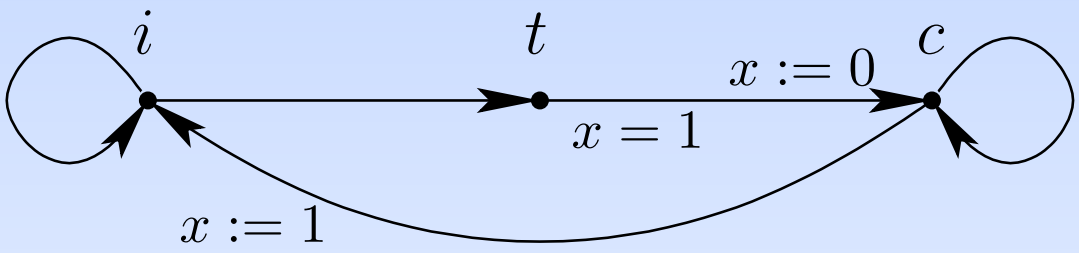
Les erreurs dans les programmes peuvent avoir des conséquences importantes: Ariane 5, Therac 25, . . .

Des tests intensifs peuvent améliorer la qualité des programmes

Il existe des techniques permettant d'atteindre un plus haut niveau de confiance

L'utilisation d'invariants est une de ces techniques: *Météor* (RATP)

Introduction: un exemple



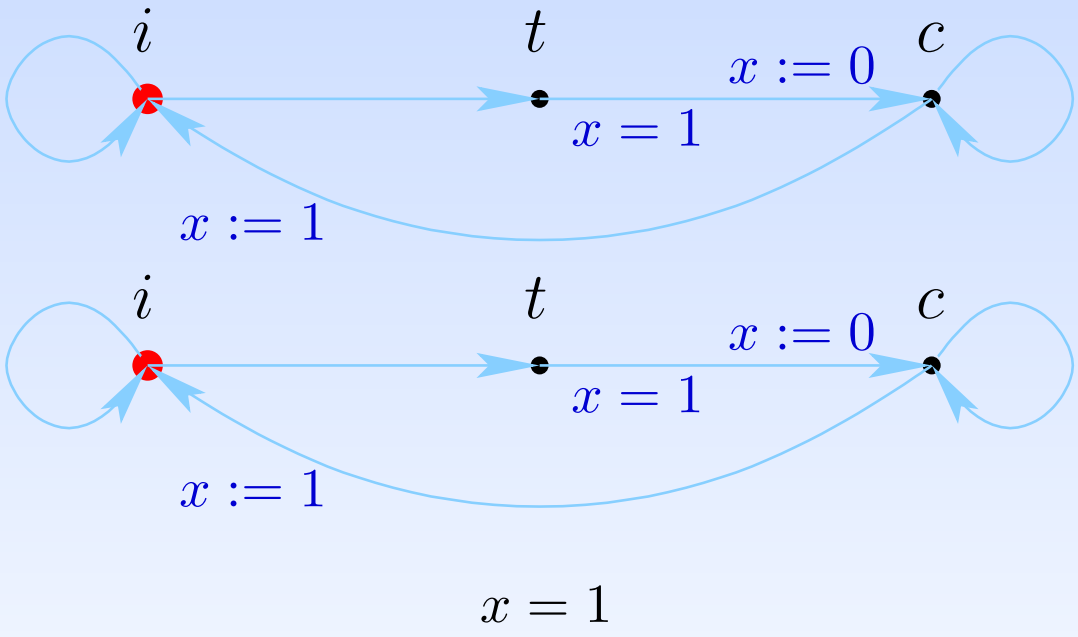
Transitions:

$$\left\{ (l[p] = i \longrightarrow \text{skip}), (l[p] = i \longrightarrow l[p] := t), (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c), \ldots \right\}$$

Propriété:

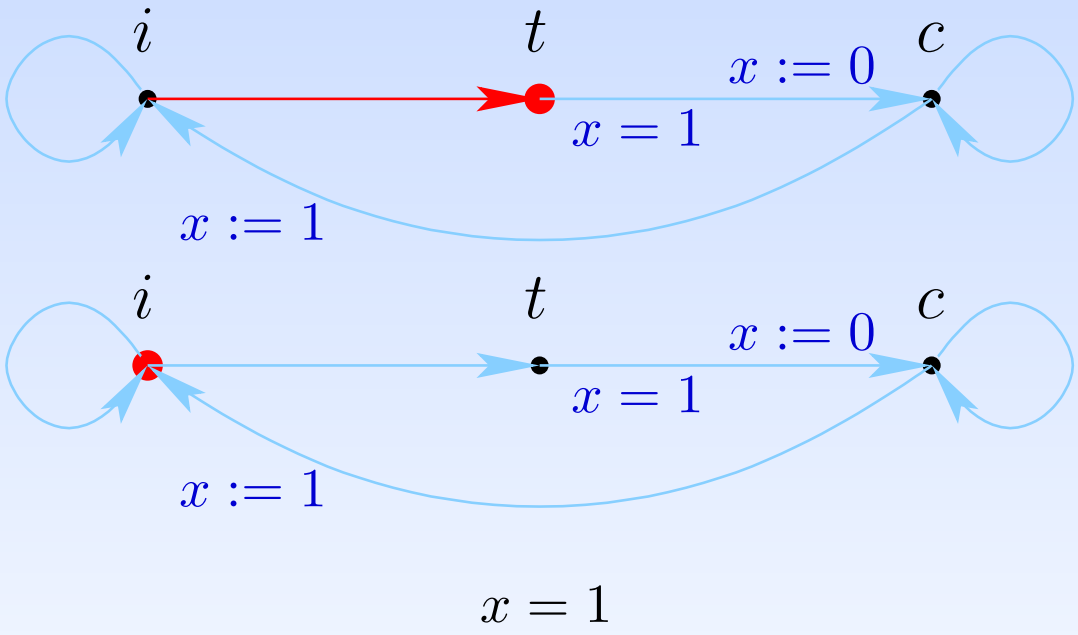
$$P = \forall q \forall r \left[q \neq r \Rightarrow \neg \left(l[q] = c \wedge l[r] = c \right) \right]$$

Introduction: un exemple (2)



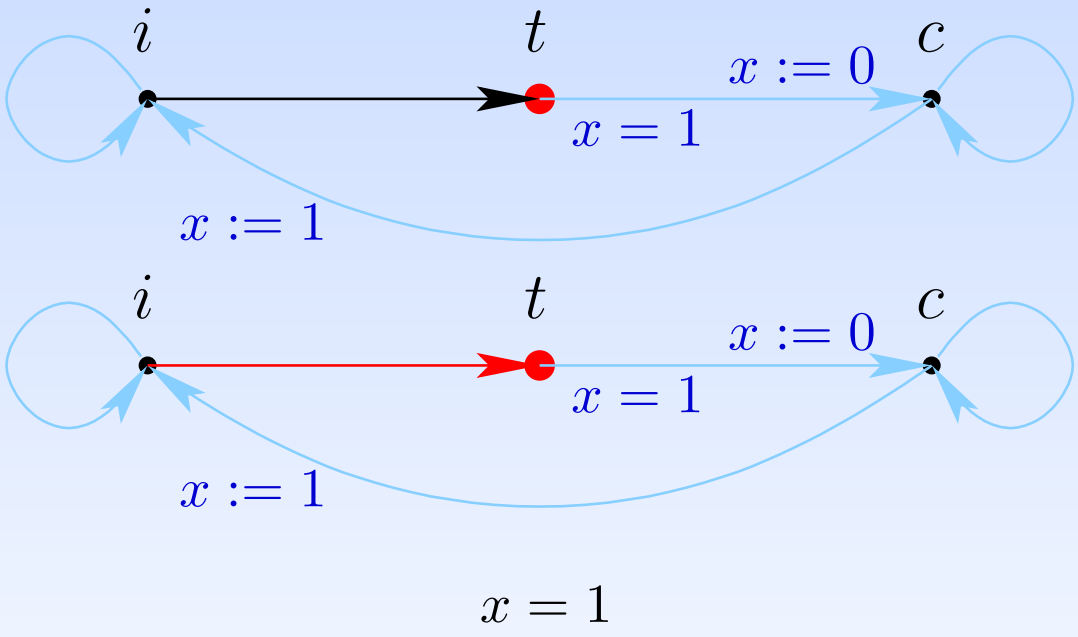
$$x = 1$$

Introduction: un exemple (2)



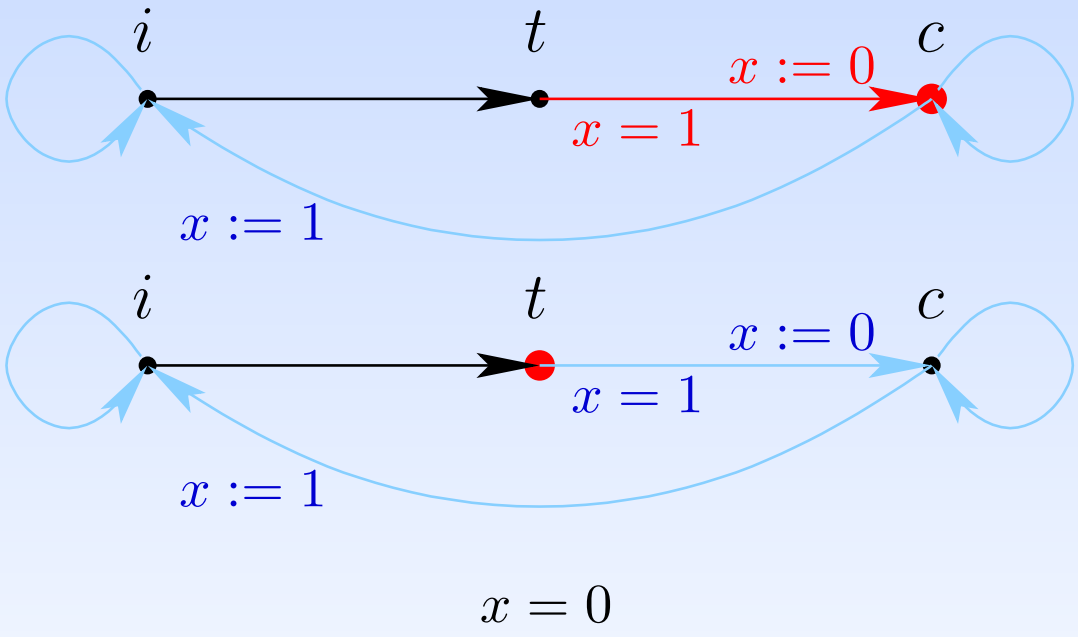
$$x = 1$$

Introduction: un exemple (2)



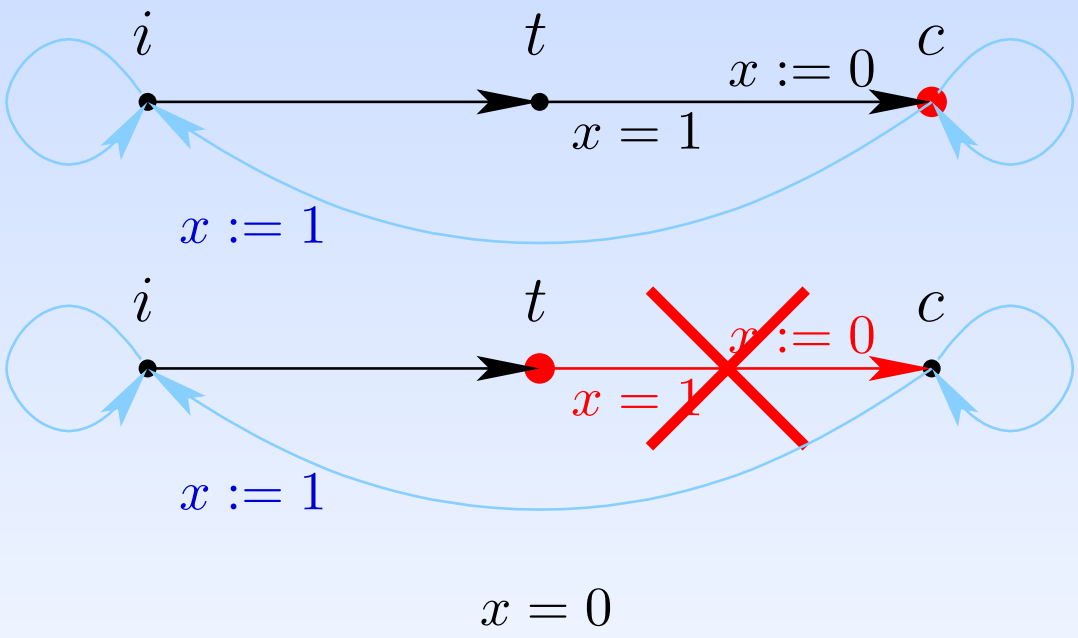
$$x = 1$$

Introduction: un exemple (2)



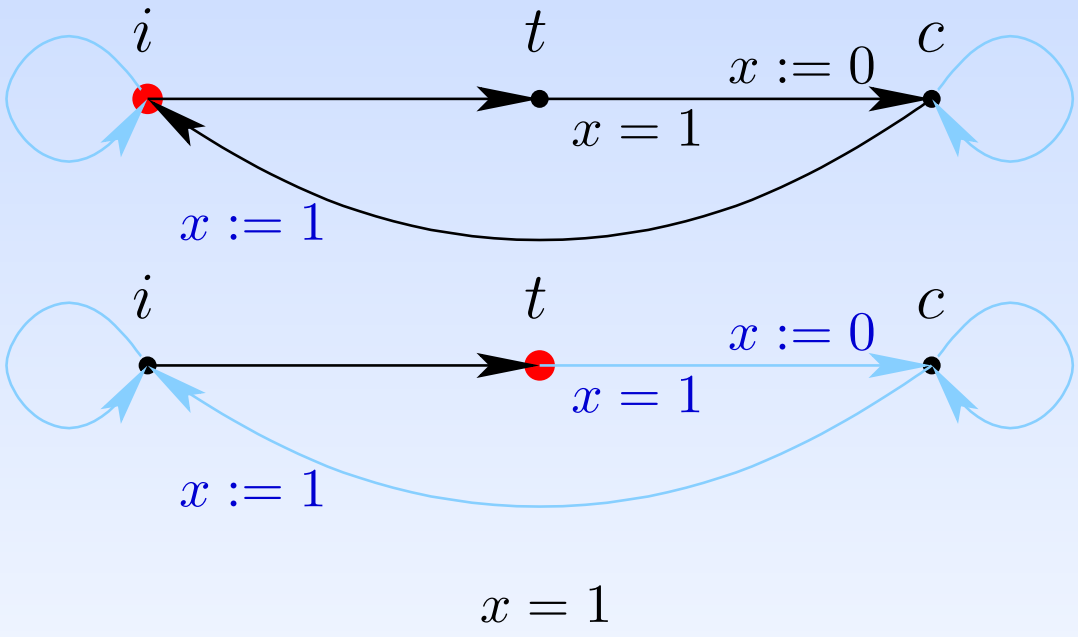
$x = 0$

Introduction: un exemple (2)



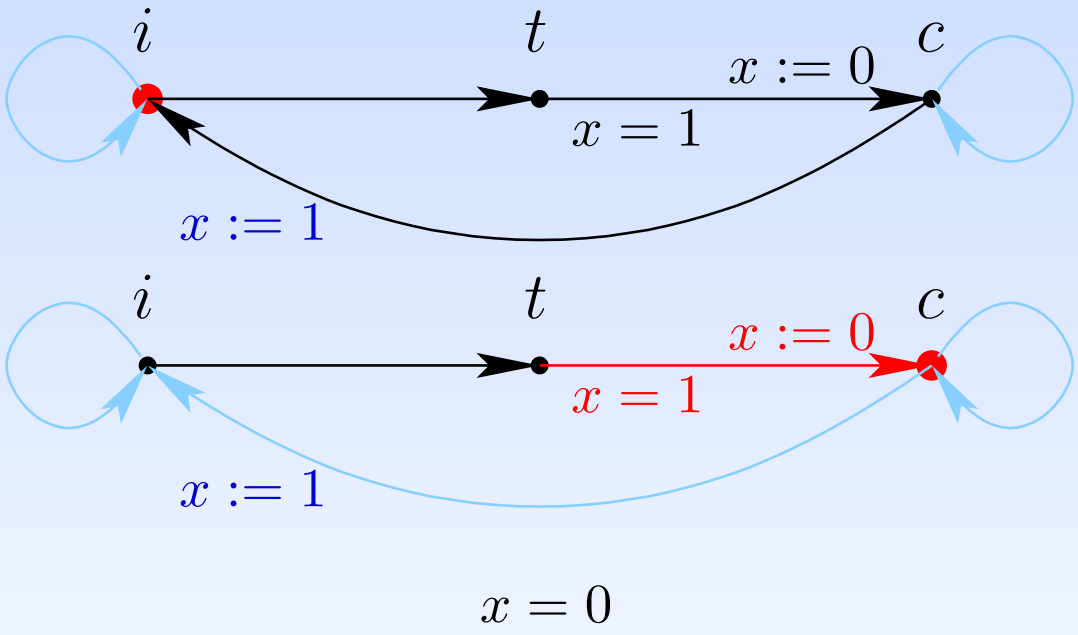
$x = 0$

Introduction: un exemple (2)



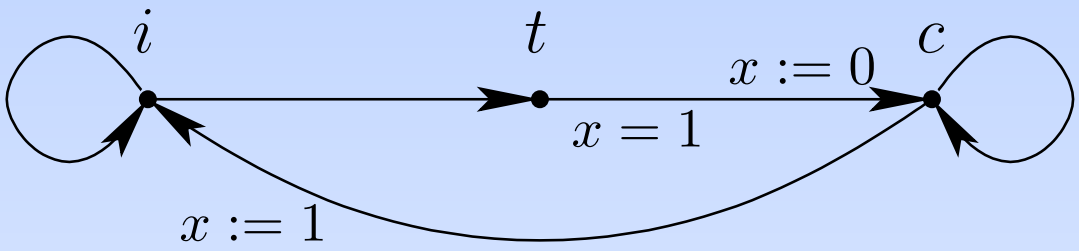
$x = 1$

Introduction: un exemple (2)



$x = 0$

Introduction: un exemple (3)



Invariant: I invariant si et seulement si $\models \{I\}\ \tau\ \{I\}$, pour toute transition τ

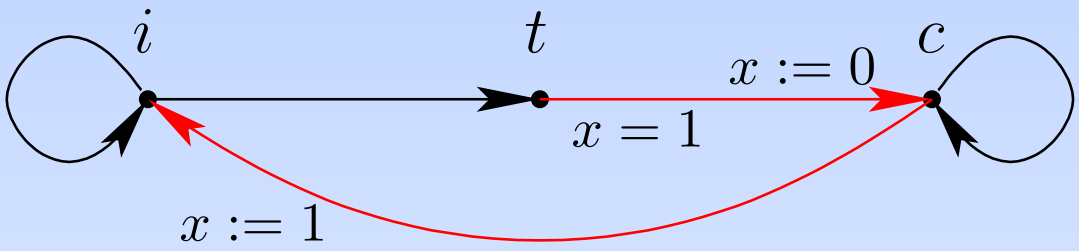
$$I \quad = \quad \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$$

$$I_1(q) \ =_{\text{def}} \ l[q] = c \Rightarrow x = 0$$

$$I_2(q) \ =_{\text{def}} \ x = 0 \Rightarrow l[q] = c$$

$$I_3(q,r) \ =_{\text{def}} \ q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

Introduction: un exemple (3)



Invariant: I invariant si et seulement si $\models \{I\}\ \tau\ \{I\}$, pour toute transition τ

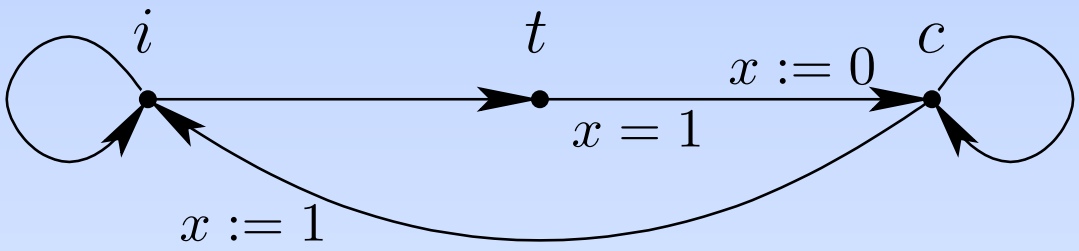
$$I \quad = \quad \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$$

$$I_1(q) \ =_{\text{def}} \ l[q] = c \Rightarrow x = 0$$

$$I_2(q) \ =_{\text{def}} \ x = 0 \Rightarrow l[q] = c$$

$$I_3(q,r) \ =_{\text{def}} \ q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

Introduction: un exemple (3)



Invariant: I invariant si et seulement si $\models \{I\}\ \tau\ \{I\}$, pour toute transition τ

$$I \quad = \quad \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$$

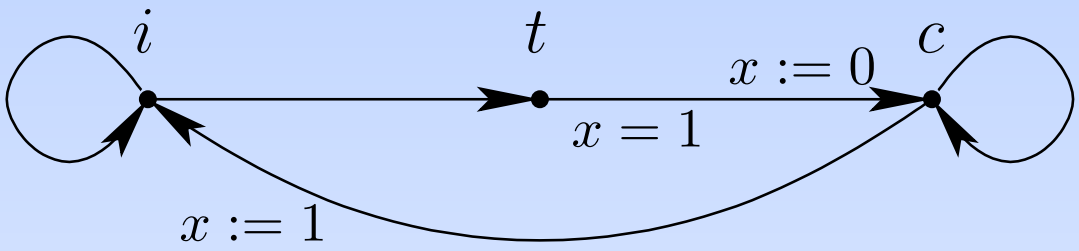
$$I_1(q) \ =_{\text{def}}\ l[q] = c \Rightarrow x = 0$$

$$I_2(q) \ =_{\text{def}}\ x = 0 \Rightarrow l[q] = c$$

$$I_3(q,r) \ =_{\text{def}}\ q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

Les conditions initiales rendent vrai l'invariant

Introduction: un exemple (3)



Invariant: I invariant si et seulement si $\models \{I\}\ \tau\ \{I\}$, pour toute transition τ

$$I \quad = \quad \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$$

$$I_1(q) \ =_{\text{def}}\ l[q] = c \Rightarrow x = 0$$

$$I_2(q) \ =_{\text{def}}\ x = 0 \Rightarrow l[q] = c$$

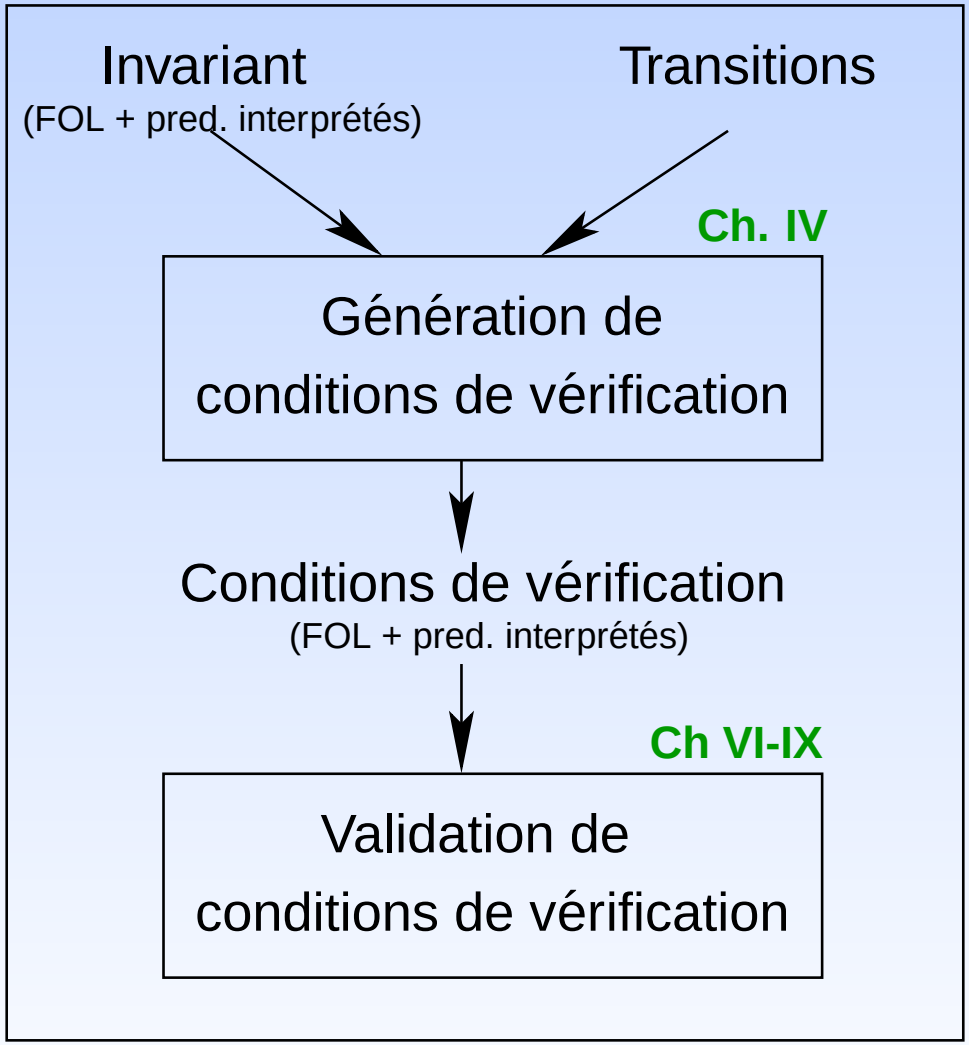
$$I_3(q,r) \ =_{\text{def}}\ q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

Les conditions initiales rendent vrai l'invariant

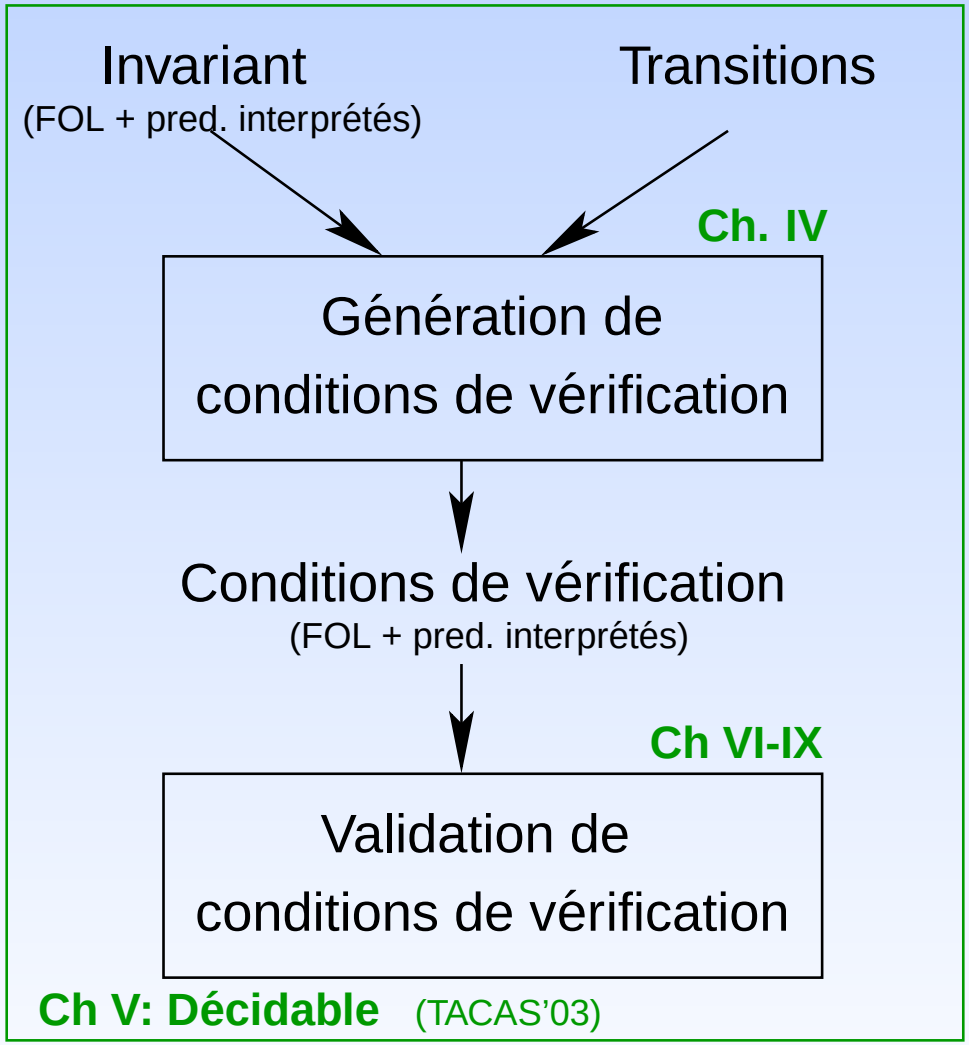
L'invariant a pour conséquence logique la propriété d'exclusion mutuelle

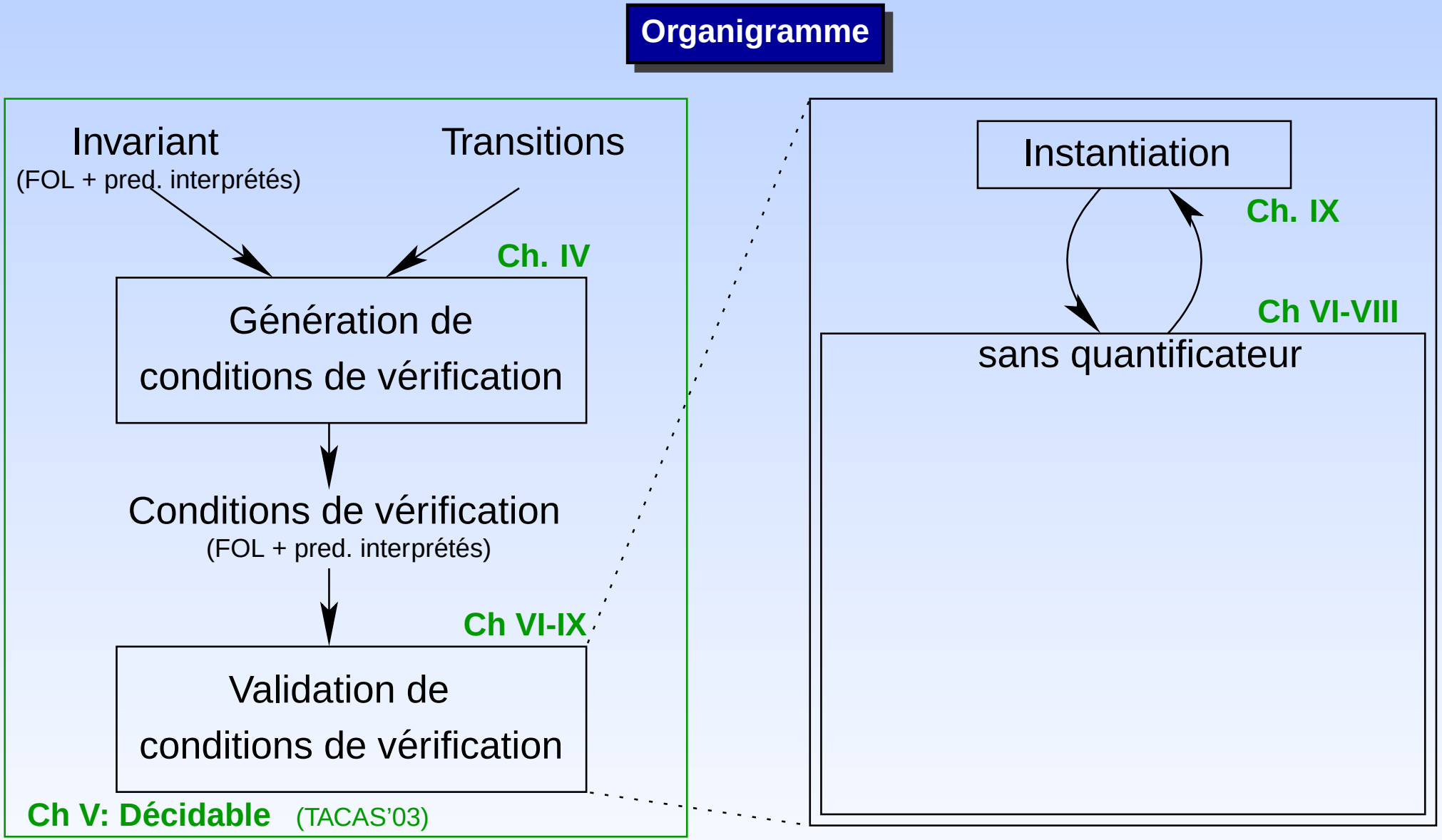
L'exclusion mutuelle est vérifiée pour toute exécution

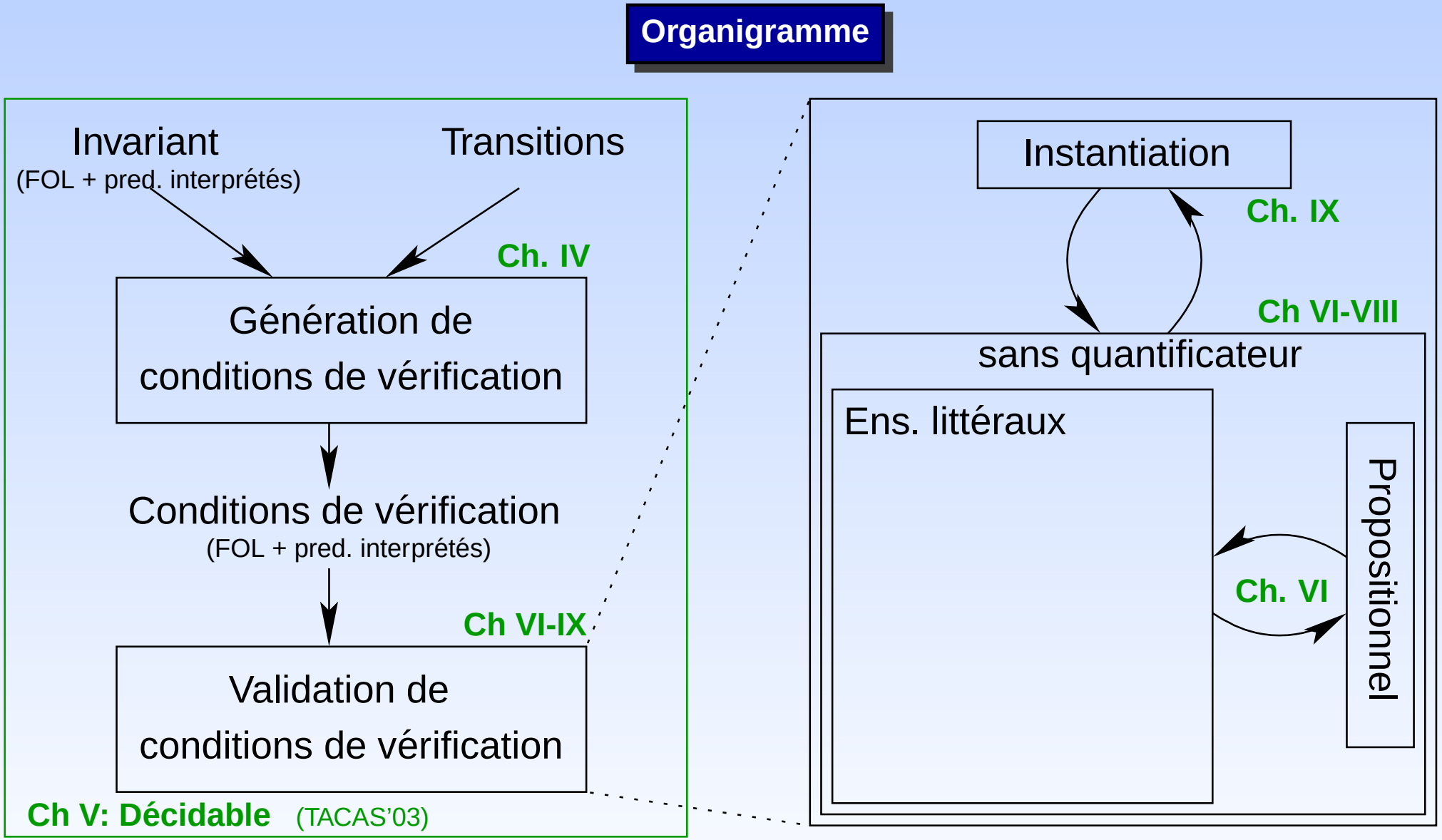
Organigramme

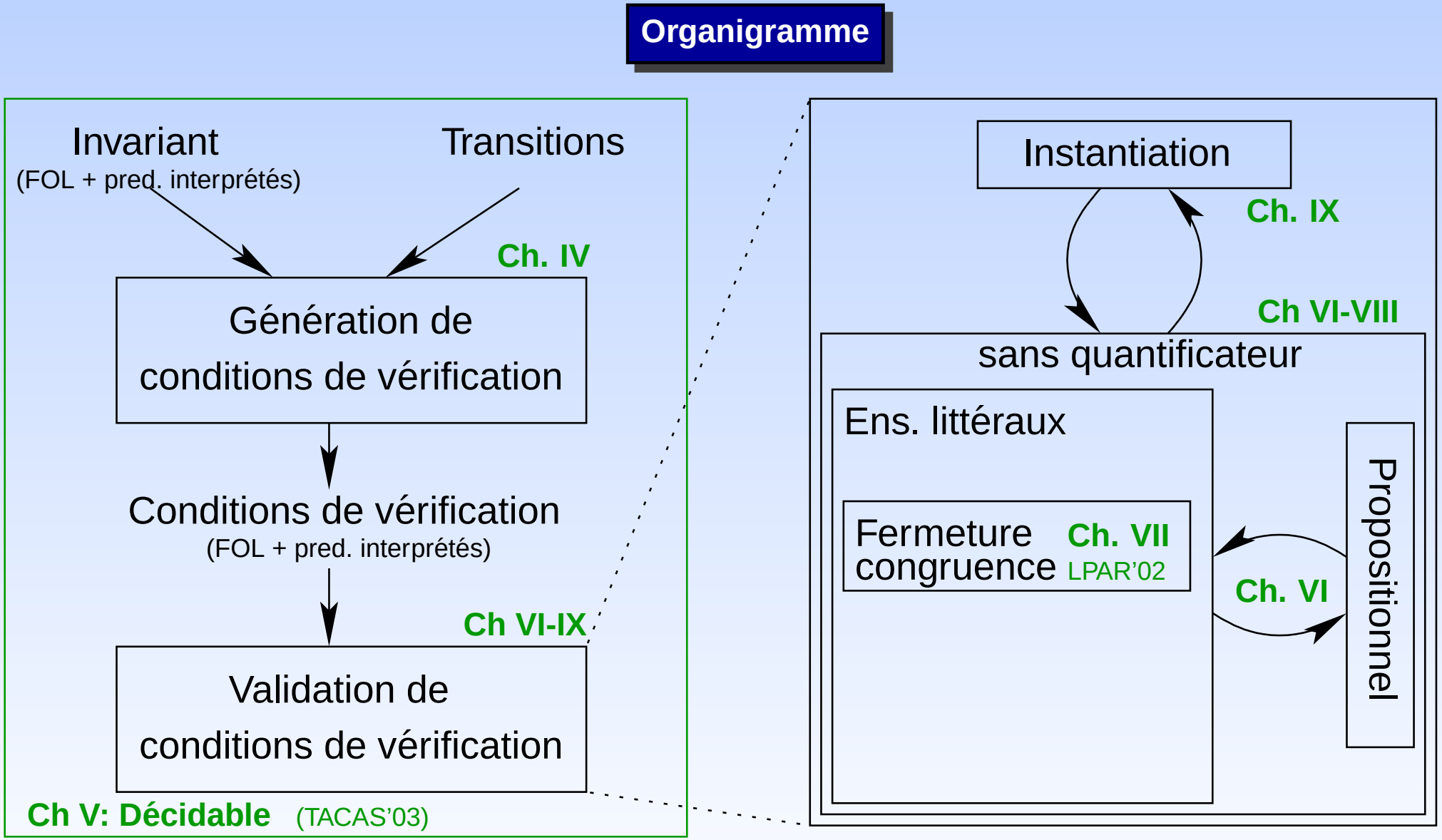


Organigramme

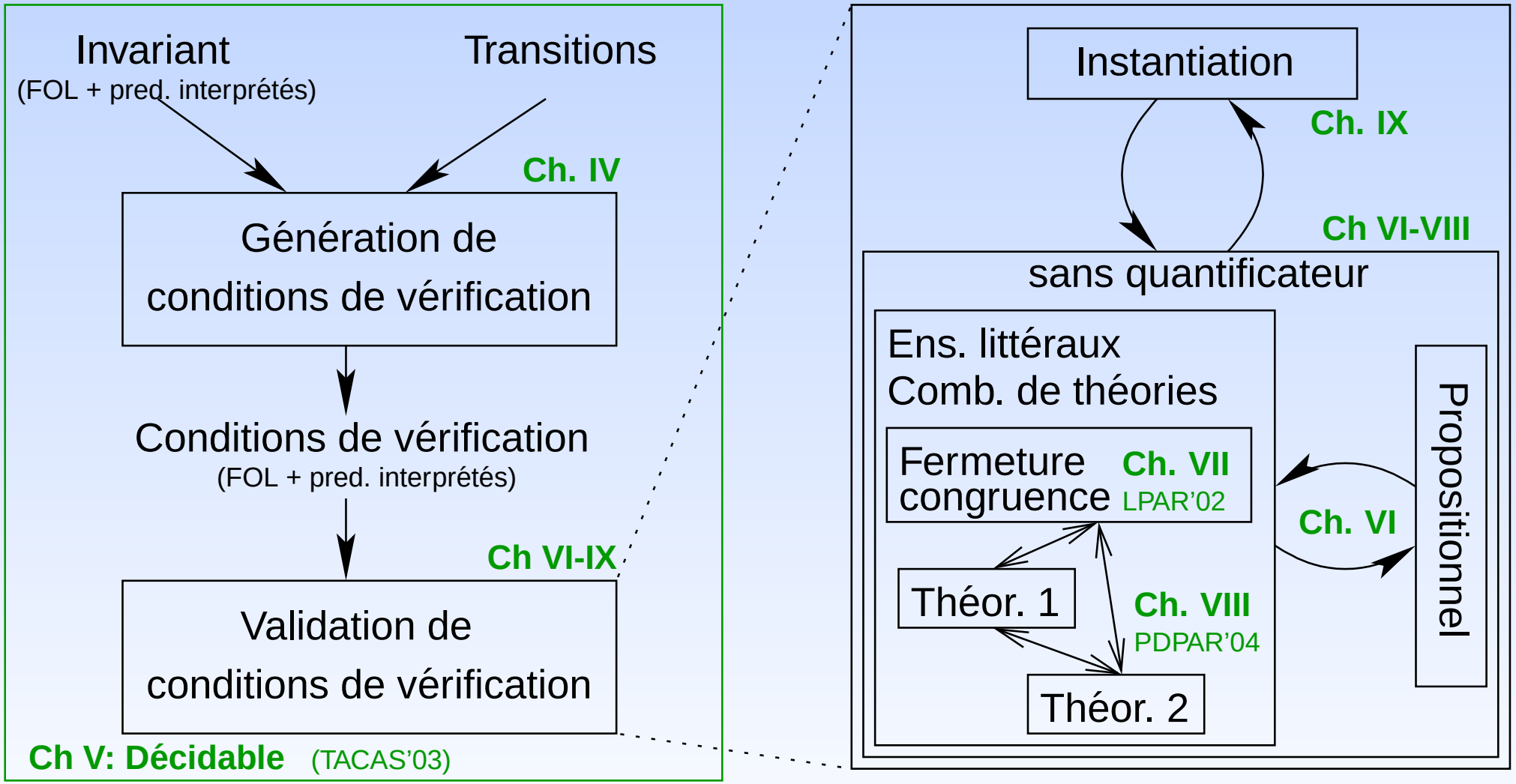



Organigramme
Invariant
(FOL + pred. interprétés)
Transitions
Ch. IV
Génération de
conditions de vérification
Conditions de vérification
(FOL + pred. interprétés)
Ch VI-IX
Validation de
conditions de vérification
Ch V: Décidable (TACAS'03)
Instantiation
Ch. IX
Ch VI-VIII
sans quantificateur


Organigramme
Invariant
(FOL + pred. interprétés)
Transitions
Ch. IV
Génération de
conditions de vérification
Conditions de vérification
(FOL + pred. interprétés)
Ch VI-IX
Validation de
conditions de vérification
Ch V: Décidable (TACAS'03)
Instantiation
Ch. IX
Ch VI-VIII
sans quantificateur
Ens. littéraux
Propositionnel
Ch. VI


Organigramme
Invariant
(FOL + pred. interprétés)
Transitions
Ch. IV
Génération de
conditions de vérification
Conditions de vérification
(FOL + pred. interprétés)
Ch VI-IX
Validation de
conditions de vérification
Ch V: Décidable (TACAS'03)
Instantiation
Ch. IX
Ch VI-VIII
sans quantificateur
Ens. littéraux
Fermeture
congruence
Ch. VII
LPAR'02
Ch. VI
Propositionnel

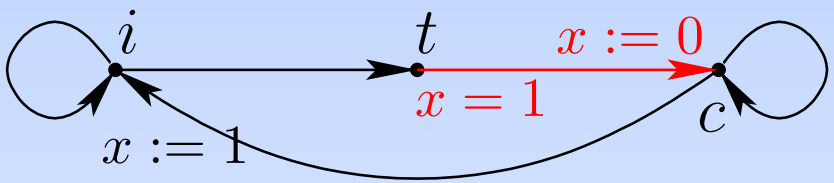
Organigramme



Organigramme

- La validation d'invariant pour systèmes paramétrés est (parfois) décidable
- Conditions de vérification sans quantificateur
  - Les méthodes propositionnelles au delà de la logique propositionnelle
  - Procédure de décision pour les symboles non interprétés
  - Combinaison de procédures de décision
- Conditions de vérification avec quantificateurs

Décidabilité de la validation d'invariant...

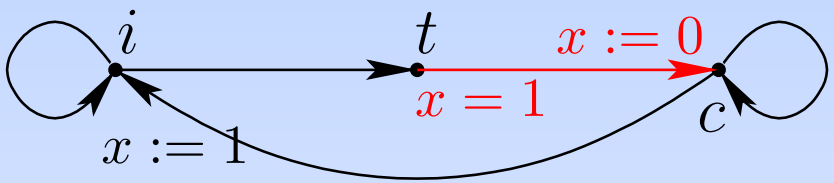


$$I_1(q) =_{\text{def}} l[q] = c \Rightarrow x = 0$$
$$I_2(q) =_{\text{def}} x = 0 \Rightarrow l[q] = c$$
$$I_3(q,r) =_{\text{def}} q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

Candidat Invariant: $I = \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$

Il faut $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I\}$.

Décidabilité de la validation d'invariant...



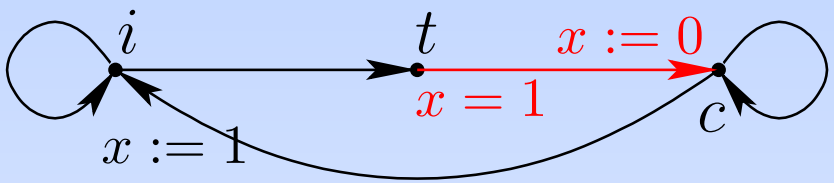
$$I_1(q) =_{\text{def}} l[q] = c \Rightarrow x = 0$$
$$I_2(q) =_{\text{def}} x = 0 \Rightarrow l[q] = c$$
$$I_3(q,r) =_{\text{def}} q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

Candidat Invariant: $I = \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$

Il faut $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I\}$.

En particulier, $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{\forall q \forall r\, I_3(q,r)\}$

Décidabilité de la validation d'invariant...



$$I_1(q) =_{\text{def}} l[q] = c \Rightarrow x = 0$$
$$I_2(q) =_{\text{def}} x = 0 \Rightarrow l[q] = c$$
$$I_3(q,r) =_{\text{def}} q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

Candidat Invariant: $I = \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$

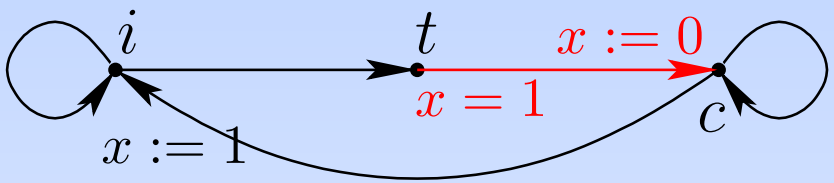
Il faut $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I\}$.

En particulier, $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{\forall q \forall r\, I_3(q,r)\}$

ou encore $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I_3(q,r)\}$

pour des processus q et r quelconques.

Décidabilité de la validation d'invariant...



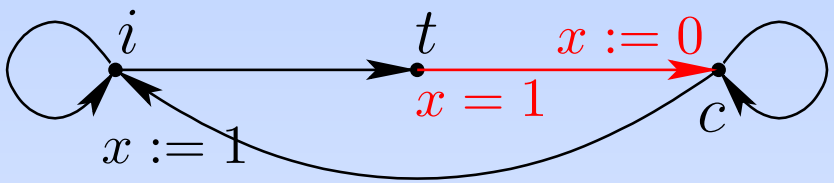
$$I_1(q) =_{\text{def}} l[q] = c \Rightarrow x = 0$$
$$I_2(q) =_{\text{def}} x = 0 \Rightarrow l[q] = c$$
$$I_3(q,r) =_{\text{def}} q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

Candidat Invariant: $I = \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$

Il faut $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I\}$.

En particulier, $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{\forall q \forall r\, I_3(q,r)\}$
ou encore (Skolemization) $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I_3(q,r)\}$
pour des processus q et r quelconques.

Décidabilité de la validation d'invariant...



$$I_1(q) =_{\text{def}} l[q] = c \Rightarrow x = 0$$
$$I_2(q) =_{\text{def}} x = 0 \Rightarrow l[q] = c$$
$$I_3(q,r) =_{\text{def}} q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

Candidat Invariant: $I = \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$

Il faut $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I\}$.

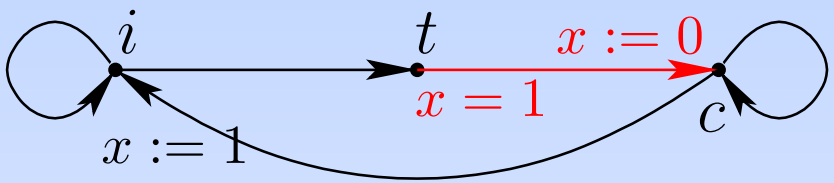
En particulier, $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{\forall q \forall r\, I_3(q,r)\}$

ou encore (Skolemization) $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I_3(q,r)\}$

pour des processus q et r quelconques.

$$\models \{\forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I_3(q,r)\}$$

Décidabilité de la validation d'invariant...



$$I_1(q) =_{\text{def}} l[q] = c \Rightarrow x = 0$$
$$I_2(q) =_{\text{def}} x = 0 \Rightarrow l[q] = c$$
$$I_3(q,r) =_{\text{def}} q \neq r \Rightarrow \neg (l[q] = c \wedge l[r] = c)$$

Candidat Invariant: $I = \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$

Il faut $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I\}$.

En particulier, $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{\forall q \forall r\, I_3(q,r)\}$

ou encore (Skolemization) $\models \{I\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I_3(q,r)\}$

pour des processus q et r quelconques.

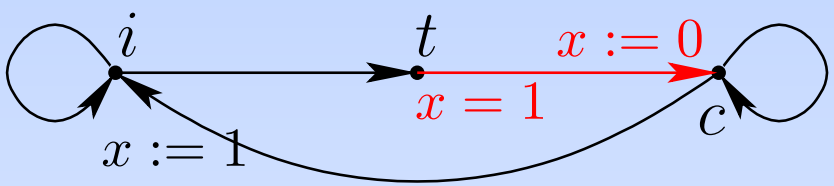
$\models \{\forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I_3(q,r)\}$

De même (Skolemization). Intuition: nommer certains processus. Ici, pas d'introduction de fonction.

$\models \{\forall q\, I_1(q) \wedge I_2(s) \wedge \forall q \forall r\, I_3(q,r)\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I_3(q,r)\}$

Skolemization des quantificateurs existentiels dans les hypothèses et universels dans la conclusion

Décidabilité de la validation d'invariant. . . (2)



$$I_1(q) =_{\text{def}} l[q] = c \Rightarrow x = 0$$
$$I_2(q) =_{\text{def}} x = 0 \Rightarrow l[q] = c$$
$$I_3(q,r) =_{\text{def}} q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

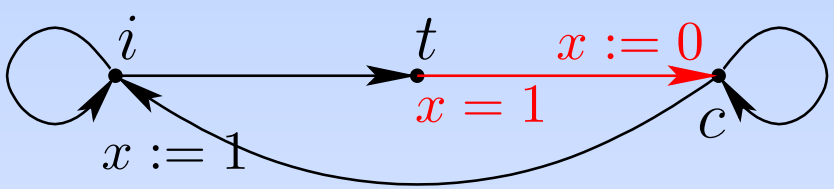
Candidat Invariant: $I = \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$

Après Skolemization:

$$\models \{\forall q\, I_1(q) \wedge I_2(s) \wedge \forall q \forall r\, I_3(q,r)\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I_3(q,r)\}$$

Les seuls processus qui interviennent explicitement sont p, q, r, s (Nombre fini).

Décidabilité de la validation d'invariant. . . (2)



$$I_1(q) =_{\text{def}} l[q] = c \Rightarrow x = 0$$
$$I_2(q) =_{\text{def}} x = 0 \Rightarrow l[q] = c$$
$$I_3(q,r) =_{\text{def}} q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

Candidat Invariant: $I = \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$

Après Skolemization:

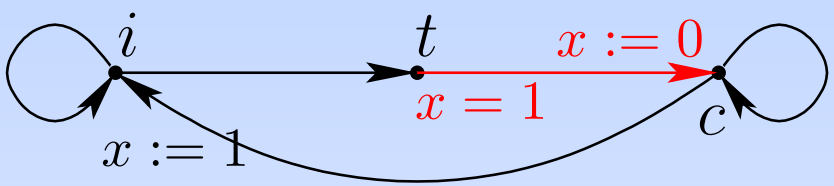
$$\models \{\forall q\, I_1(q) \wedge I_2(s) \wedge \forall q \forall r\, I_3(q,r)\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I_3(q,r)\}$$

Les seuls processus qui interviennent explicitement sont p, q, r, s (Nombre fini).

On peut remplacer (par exemple) $\forall q\, I_1(q)$ par $I_1(p) \wedge I_1(q) \wedge I_1(r) \wedge I_1(s)$.

Ceci est lié à l'absence de fonction. Sinon $f(p)$, $f(f(p))$, . . . Infinité de processus à considérer.

Décidabilité de la validation d'invariant. . . (2)



$$I_1(q) =_{\text{def}} l[q] = c \Rightarrow x = 0$$
$$I_2(q) =_{\text{def}} x = 0 \Rightarrow l[q] = c$$
$$I_3(q,r) =_{\text{def}} q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

Candidat Invariant: $I = \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$

Après Skolemization:

$$\models \{\forall q\, I_1(q) \wedge I_2(s) \wedge \forall q \forall r\, I_3(q,r)\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I_3(q,r)\}$$

Les seuls processus qui interviennent explicitement sont p, q, r, s (Nombre fini).

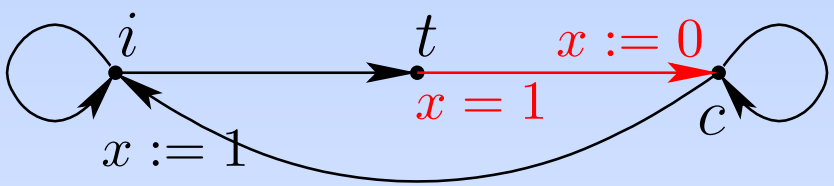
On peut remplacer (par exemple) $\forall q\, I_1(q)$ par $I_1(p) \wedge I_1(q) \wedge I_1(r) \wedge I_1(s)$.

Ceci est lié à l'absence de fonction. Sinon $f(p)$, $f(f(p))$, . . . Infinité de processus à considérer.

La condition de vérification obtenue ne contient plus de quantificateur: décidable.

Si I est un invariant pour 4 processus, il l'est aussi pour un nombre quelconque.

Décidabilité de la validation d'invariant. . . (2)



$$I_1(q) =_{\text{def}} l[q] = c \Rightarrow x = 0$$
$$I_2(q) =_{\text{def}} x = 0 \Rightarrow l[q] = c$$
$$I_3(q,r) =_{\text{def}} q \neq r \Rightarrow \neg\,(l[q] = c \wedge l[r] = c)$$

Candidat Invariant: $I = \forall q\, I_1(q) \wedge \exists q\, I_2(q) \wedge \forall q \forall r\, I_3(q,r)$

Après Skolemization:

$$\models \{\forall q\, I_1(q) \wedge I_2(s) \wedge \forall q \forall r\, I_3(q,r)\}\ (l[p] = t \wedge x = 1 \longrightarrow x := 0; l[p] := c)\ \{I_3(q,r)\}$$

Les seuls processus qui interviennent explicitement sont p, q, r, s (Nombre fini).

On peut remplacer (par exemple) $\forall q\, I_1(q)$ par $I_1(p) \wedge I_1(q) \wedge I_1(r) \wedge I_1(s)$.

Ceci est lié à l'absence de fonction. Sinon $f(p)$, $f(f(p))$, . . . Infinité de processus à considérer.

La condition de vérification obtenue ne contient plus de quantificateur: décidable.

Si I est un invariant pour 4 processus, il l'est aussi pour un nombre quelconque.

Formalisation: Herbrand, avec domaine fini.

Herbrand classique inapplicable:

- ✘ égalité
- ✘ fonctions
- ✘ symboles interprétés
- ✘ types

Herbrand classique inapplicable:

- ✘ égalité
- ✘ fonctions
- ✘ symboles interprétés
- ✘ types

Soit

- une formule S en *forme de Skolem*
- un type τ (processus) tel qu'aucun terme $f(t_1, \ldots t_n)$ n'est de type τ

H_τ est l'ensemble des τ-constantes dans S.

Pour tout modèle $\mathcal{I} = \langle D, I \rangle$ de S, il existe un modèle $\mathcal{I}' = \langle D', I' \rangle$ tel que

- D'_τ est l'ensemble quotient de H_τ par la congruence liée à l'égalité dans $\mathcal{I}$
- $\mathcal{I}'$ et $\mathcal{I}$ similaires, pour tout $\tau' \neq \tau$, tout symbole tel que τ n'est le type d'aucun argument.

Herbrand classique inapplicable:	Herbrand (version 2):
✘ égalité	✔
✘ fonctions	✔ $f(t_1, \ldots t_n)$ n'est pas un processus, $\forall f$
✘ symboles interprétés	✔ les arguments ne sont pas des processus
✘ types	✔

Soit

- une formule S en *forme de Skolem*
- un type τ (processus) tel qu'aucun terme $f(t_1, \ldots t_n)$ n'est de type τ

H_τ est l'ensemble des τ-constantes dans S.

Pour tout modèle $\mathcal{I} = \langle D, I \rangle$ de S, il existe un modèle $\mathcal{I}' = \langle D', I' \rangle$ tel que

- D'_τ est l'ensemble quotient de H_τ par la congruence liée à l'égalité dans $\mathcal{I}$
- $\mathcal{I}'$ et $\mathcal{I}$ similaires, pour tout $\tau' \neq \tau$, tout symbole tel que τ n'est le type d'aucun argument.

Décidabilité. . . : critères

Critères (invariants et systèmes paramétrés) garantissant un ensemble de Herbrand (version 2) fini.

Sur les quantificateurs:

- les quantificateurs portent sur les processus seulement (invariant et gardes des transitions)
- chaque terme de la conjonction I peut être mis en forme $\exists^\star\forall^\star$ et $\forall^\star\exists^\star$
- les gardes de transitions peuvent être mises en forme $\exists^\star\forall^\star$.

Chaque condition de vérification ($I \Rightarrow I'$) peut être mise en forme $\forall^\star\exists^\star$.

Sa négation ($I \wedge \neg I'$) peut être mise en forme $\exists^\star\forall^\star$.

La Skolemization n'introduira pas de fonction

Décidabilité. . . : critères (2)

Autres critères:

- aucun prédicat interprété sur l'ensemble des processus (excepté $=$, $<$)
- aucun terme $f(t_1, \ldots t_n)$ du type processus.

Le domaine de Herbrand est fini, le théorème est applicable

Les quantifications universelles deviennent des conjonctions finies

Les quantifications existentielles deviennent des disjonctions finies

Décidabilité. . . : nombre fini de processus

I est un invariant du système ssi I est un invariant d'un système de n_0 processus. n_0 est la somme

- du nombre de quantificateurs existentiels dans I (en forme $\exists^\star\forall^\star$)
- du nombre maximum de quantificateurs existentiels dans les gardes
- du nombre maximum de quantificateurs universels dans chacun des termes de la conjonction I (conjonction de formes $\forall^\star\exists^\star$)
- du nombre de constantes dans I
- du nombre maximum de processus qui prennent explicitement part à une transition.

Décidabilité. . . : exemple, Burns

Transitions:

$$
\begin{array}{l}
\big(\mathrm{s}_0[p],\ \mathrm{flag}[p]{:=}\mathrm{false},\ \mathrm{s}_1[p]\big)\\
\big(\mathrm{s}_1[p], \neg S[p,q] \wedge q < p \wedge \mathrm{flag}[q] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_0[p]\big)\\
\big(\mathrm{s}_1[p], \neg S[p,q] \wedge q < p \wedge \neg\mathrm{flag}[q] \ \rightarrow\ S[p,q]{:=}\mathrm{true},\ \mathrm{s}_1[p]\big)\\
\big(\mathrm{s}_1[p], \forall q\big[q < p \Rightarrow S[p,q]\big] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_2[p]\big)\\
\big(\mathrm{s}_2[p],\ \mathrm{flag}[p]{:=}\mathrm{true},\ \mathrm{s}_3[p]\big)\\
\big(\mathrm{s}_3[p], \neg S[p,q] \wedge q < p \wedge \mathrm{flag}[q] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_0[p]\big)\\
\big(\mathrm{s}_3[p], \neg S[p,q] \wedge q < p \wedge \neg\mathrm{flag}[q] \ \rightarrow\ S[p,q]{:=}\mathrm{true},\ \mathrm{s}_3[p]\big)\\
\big(\mathrm{s}_3[p], \forall q\big[q < p \Rightarrow S[p,q]\big] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_4[p]\big)\\
\big(\mathrm{s}_4[p], \neg S[p,q] \wedge p < q \wedge \mathrm{flag}[q] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_4[p]\big)\\
\big(\mathrm{s}_4[p], \neg S[p,q] \wedge p < q \wedge \neg\mathrm{flag}[q] \ \rightarrow\ S[p,q]{:=}\mathrm{true},\ \mathrm{s}_4[p]\big)\\
\big(\mathrm{s}_4[p], \forall q\big[p < q \Rightarrow S[p,q]\big] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_5[p]\big)\\
\big(\mathrm{s}_5[p],\ \mathrm{flag}[p]{:=}\mathrm{false},\ \mathrm{s}_0[p]\big)
\end{array}
$$

Décidabilité. . . : exemple, Burns

Transitions:

$$\begin{array}{l}
\big(\mathrm{s}_0[p],\ \mathrm{flag}[p]{:=}\mathrm{false},\ \mathrm{s}_1[p]\big)\\
\big(\mathrm{s}_1[p], \neg S[p,q] \wedge q < p \wedge \mathrm{flag}[q] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_0[p]\big)\\
\big(\mathrm{s}_1[p], \neg S[p,q] \wedge q < p \wedge \neg\mathrm{flag}[q] \ \rightarrow\ S[p,q]{:=}\mathrm{true},\ \mathrm{s}_1[p]\big)\\
\big(\mathrm{s}_1[p], \forall q\big[q < p \Rightarrow S[p,q]\big] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_2[p]\big)\\
\big(\mathrm{s}_2[p],\ \mathrm{flag}[p]{:=}\mathrm{true},\ \mathrm{s}_3[p]\big)\\
\big(\mathrm{s}_3[p], \neg S[p,q] \wedge q < p \wedge \mathrm{flag}[q] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_0[p]\big)\\
\big(\mathrm{s}_3[p], \neg S[p,q] \wedge q < p \wedge \neg\mathrm{flag}[q] \ \rightarrow\ S[p,q]{:=}\mathrm{true},\ \mathrm{s}_3[p]\big)\\
\big(\mathrm{s}_3[p], \forall q\big[q < p \Rightarrow S[p,q]\big] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_4[p]\big)\\
\big(\mathrm{s}_4[p], \neg S[p,q] \wedge p < q \wedge \mathrm{flag}[q] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_4[p]\big)\\
\big(\mathrm{s}_4[p], \neg S[p,q] \wedge p < q \wedge \neg\mathrm{flag}[q] \ \rightarrow\ S[p,q]{:=}\mathrm{true},\ \mathrm{s}_4[p]\big)\\
\big(\mathrm{s}_4[p], \forall q\big[p < q \Rightarrow S[p,q]\big] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_5[p]\big)\\
\big(\mathrm{s}_5[p],\ \mathrm{flag}[p]{:=}\mathrm{false},\ \mathrm{s}_0[p]\big)
\end{array}$$

Aucun quantificateur existentiel dans les gardes 0

Décidabilité. . . : exemple, Burns

Transitions:

$$\big(\mathrm{s}_0[p],\ \mathrm{flag}[p]{:=}\mathrm{false},\ \mathrm{s}_1[p]\big)$$
$$\big(\mathrm{s}_1[p], \neg S[p,q] \wedge q < p \wedge \mathrm{flag}[q] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_0[p]\big)$$
$$\big(\mathrm{s}_1[p], \neg S[p,q] \wedge q < p \wedge \neg\mathrm{flag}[q] \ \rightarrow\ S[p,q]{:=}\mathrm{true},\ \mathrm{s}_1[p]\big)$$
$$\big(\mathrm{s}_1[p], \forall q\big[q < p \Rightarrow S[p,q]\big] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_2[p]\big)$$
$$\big(\mathrm{s}_2[p],\ \mathrm{flag}[p]{:=}\mathrm{true},\ \mathrm{s}_3[p]\big)$$
$$\big(\mathrm{s}_3[p], \neg S[p,q] \wedge q < p \wedge \mathrm{flag}[q] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_0[p]\big)$$
$$\big(\mathrm{s}_3[p], \neg S[p,q] \wedge q < p \wedge \neg\mathrm{flag}[q] \ \rightarrow\ S[p,q]{:=}\mathrm{true},\ \mathrm{s}_3[p]\big)$$
$$\big(\mathrm{s}_3[p], \forall q\big[q < p \Rightarrow S[p,q]\big] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_4[p]\big)$$
$$\big(\mathrm{s}_4[p], \neg S[p,q] \wedge p < q \wedge \mathrm{flag}[q] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_4[p]\big)$$
$$\big(\mathrm{s}_4[p], \neg S[p,q] \wedge p < q \wedge \neg\mathrm{flag}[q] \ \rightarrow\ S[p,q]{:=}\mathrm{true},\ \mathrm{s}_4[p]\big)$$
$$\big(\mathrm{s}_4[p], \forall q\big[p < q \Rightarrow S[p,q]\big] \ \rightarrow\ \forall q : \big[S[p,q]{:=}\mathrm{false}\big],\ \mathrm{s}_5[p]\big)$$
$$\big(\mathrm{s}_5[p],\ \mathrm{flag}[p]{:=}\mathrm{false},\ \mathrm{s}_0[p]\big)$$

Aucun quantificateur existentiel dans les gardes	0
Au plus deux processus prenant part à la même transition	2
Total	2

Décidabilité. . . : exemple, Burns (2)

Invariant : $I = \forall p\, I_1(p) \wedge \forall p \forall q\, [I_2(p,q) \wedge I_3(p,q)]$

$I_1(p) = \neg \mathrm{flag}[p] \Rightarrow (\mathrm{s}_0[p] \vee \mathrm{s}_1[p] \vee \mathrm{s}_2[p])$
$I_2(p,q) = \mathrm{s}_2[p] \Rightarrow \neg S[p,q]$
$I_3(p,q) = \big[q < p \wedge \mathrm{flag}[q] \wedge (\mathrm{s}_5[p] \vee \mathrm{s}_4[p] \vee (\mathrm{s}_3[p] \wedge S[p,q]))\big] \Rightarrow \big[\neg \mathrm{s}_5[q] \wedge \neg(\mathrm{s}_4[q] \wedge S[q,p])\big]$

Contribution des transitions	2
Nombre de quantificateurs existentiels dans I	0
Nombre de max. $\forall$ dans termes de I	2
Nombre de constantes dans I	0
Total	4

Si I est un invariant pour 4 processus, il l'est aussi pour un nombre quelconque.

Décidabilité. . . : conclusion

De nombreux algorithmes paramétrés remplissent les critères permettant de ramener l'étude à un nombre fini de processus: Burns, Dijkstra, Ricart & Agrawala, Szymanski, General Railroad Crossing,. . .

Le nombre de processus est généralement petit (de 4 à 6).

Les conditions de vérification obtenues sont sans quantificateur mais peuvent contenir des symboles interprétés ($+$, $<$, . . .).

La validation de candidats invariants pour de tels systèmes paramétrés est donc décidable, si la validation de conditions de vérification sans quantificateur est décidable.

Propositionnel et au delà

La validation d'invariant engendre de nombreuses conditions de vérification (avec et) sans quantificateur.

Elles contiennent les connecteurs habituels ($\wedge$, $\vee$, $\Rightarrow$, $\equiv$, ...), des propositions, des fonctions et des prédicats non interprétés, mais aussi des fonctions ($+$, ...) et prédicats interprétés ($<$, ...).

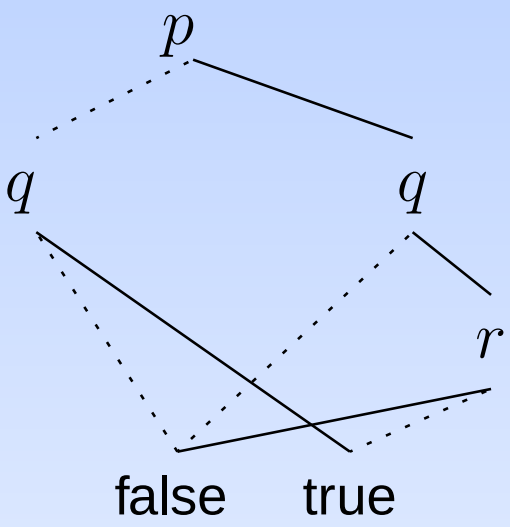
Méthodes modernes de satisfaisabilité propositionnelle: BDD (Binary Decision Diagrams), méthodes SAT (Davis, Putnam, Logemann, Loveland: DPLL).

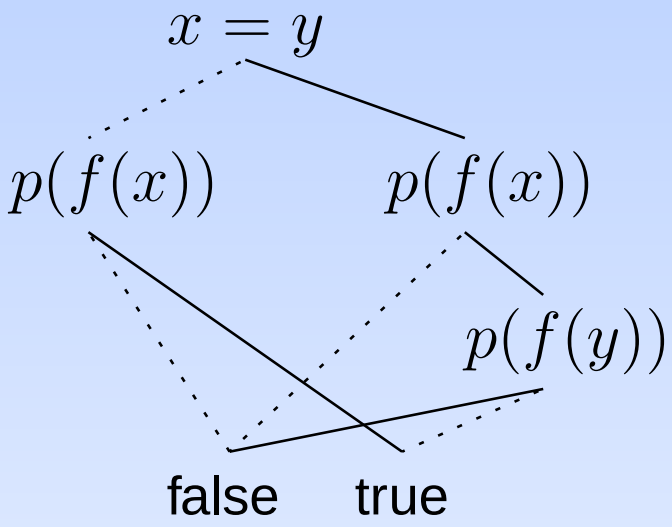
Un ensemble inconsistant de littéraux propositionnels contient une paire complémentaire de littéraux. Cette propriété est perdue avec l'introduction de l'égalité:

- $p \wedge q \wedge r$ satisfaisable
- $x = y \wedge p(f(x)) \wedge \neg p(f(y))$ inconsistant

Les méthodes propositionnelles doivent collaborer avec des procédures pour la satisfaisabilité d'ensembles de littéraux.

Propositionnel. . . : BDD

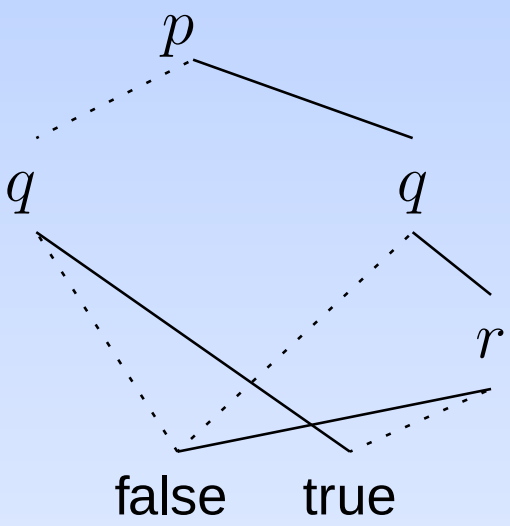


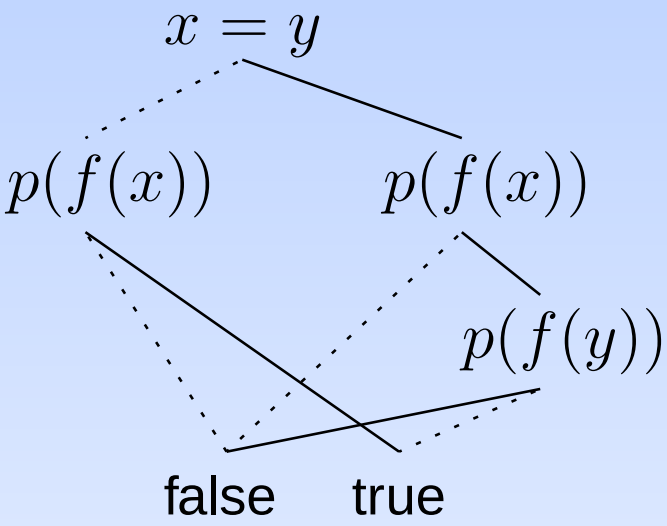


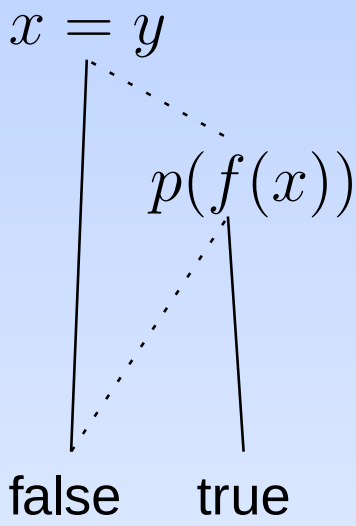
Un BDD (réduit, ordonné) est une représentation canonique d'une formule propositionnelle.

Le BDD fournit des modèles (partiels) de l'abstraction propositionnelle. Ces modèles doivent être validés par une procédure externe pour les ensembles de littéraux.

Propositionnel. . . : BDD







Un BDD (réduit, ordonné) est une représentation canonique d'une formule propositionnelle.

Le BDD fournit des modèles (partiels) de l'abstraction propositionnelle. Ces modèles doivent être validés par une procédure externe pour les ensembles de littéraux.

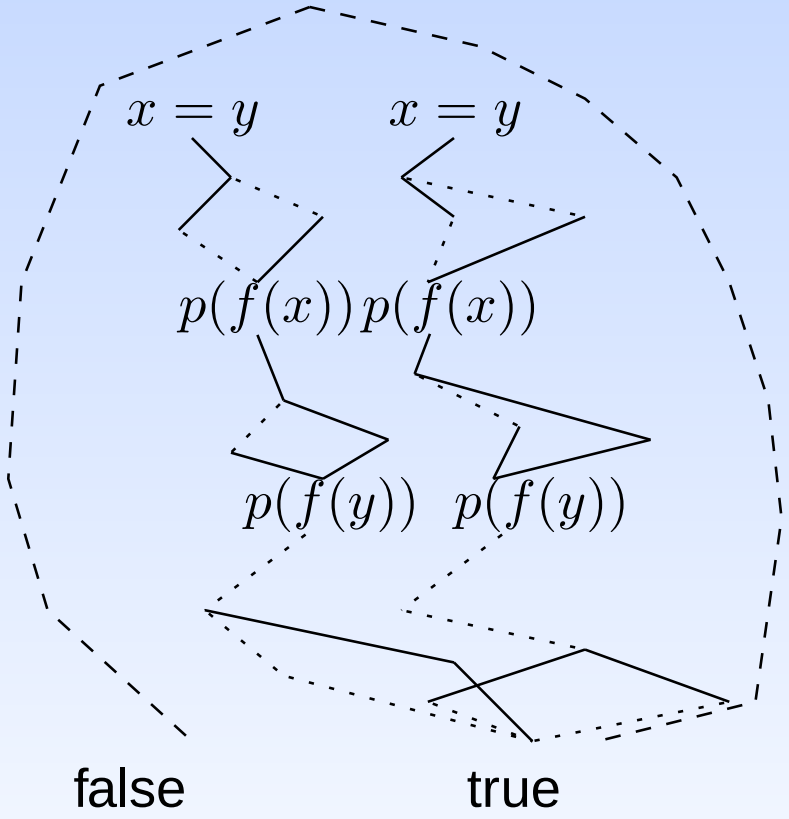
$\{x = y, p(f(x)), \neg p(f(y))\}$ inconsitant.

Une branche du BDD doit être éliminée, en ajoutant une contrainte.

La canonicité est perdue.

Problème: examiner l'ensemble des chemins vers "true".

Propositionnel. . . : BDD (2)

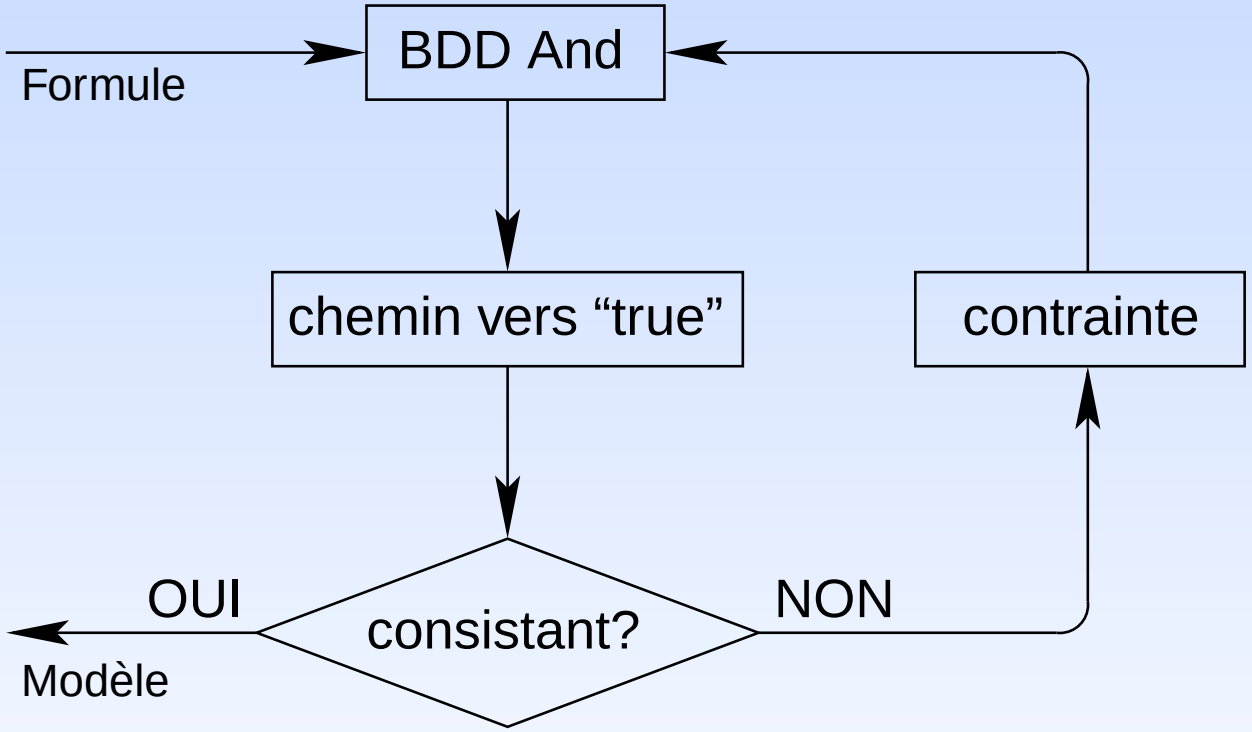


- Chemin: $\{x = y, \ldots p(f(x)), \ldots \neg p(f(y)), \ldots\}$
- Ce chemin est insatisfaisable, et doit être éliminé
- Contrainte générale: $x \neq y \vee \neg p(f(x)) \vee p(f(y))$
- Tout chemin vers “true” contenant $x = y$, $p(f(x))$ et $\neg p(f(y))$ est éliminé.

Seuls quelques chemins doivent être examinés

La procédure pour les littéraux doit fournir une information raffinée: un sous-ensemble inconsistant minimal

Propositionnel. . . : BDD (3)



Propositionnel. . . DPLL

Méthode de satisfaisabilité pour un ensemble de clauses.

Construction incrémentale d'un modèle:

- Propagation des valeurs de vérités: $p \vee q$ avec p faux, alors q vrai

$$\neg p$$
$$p \vee q$$
$$\neg q \vee r \vee t$$
$$\neg q \vee r \vee \neg t$$

Propositionnel. . . DPLL

Méthode de satisfaisabilité pour un ensemble de clauses.

Construction incrémentale d'un modèle:

- Propagation des valeurs de vérités: $p \vee q$ avec p faux, alors q vrai

$\neg p$

$p \vee q$

$\neg q \vee r \vee t$

$\neg q \vee r \vee \neg t$

Propositionnel. . . DPLL

Méthode de satisfaisabilité pour un ensemble de clauses.

Construction incrémentale d'un modèle:

- Propagation des valeurs de vérités: $p \vee q$ avec p faux, alors q vrai

$\neg p$

$p \vee q$

$\neg q \vee r \vee t$

$\neg q \vee r \vee \neg t$

Propositionnel. . . DPLL

Méthode de satisfaisabilité pour un ensemble de clauses.
Construction incrémentale d'un modèle:

- Propagation des valeurs de vérités: $p \vee q$ avec p faux, alors q vrai
- Choix d'une valeur de vérité pour une proposition non assignée.

$\neg p$
$p \vee q$
$\neg q \vee r \vee t$
$\neg q \vee r \vee \neg t$

Propositionnel. . . DPLL

Méthode de satisfaisabilité pour un ensemble de clauses.
Construction incrémentale d'un modèle:

- Propagation des valeurs de vérités: $p \vee q$ avec p faux, alors q vrai
- Choix d'une valeur de vérité pour une proposition non assignée.

$\neg p$
$p \vee q$
$\neg q \vee r \vee t$
$\neg q \vee r \vee \neg t$

Techniques évoluées de prétraitement, de backtracking, d'apprentissage, de décision.

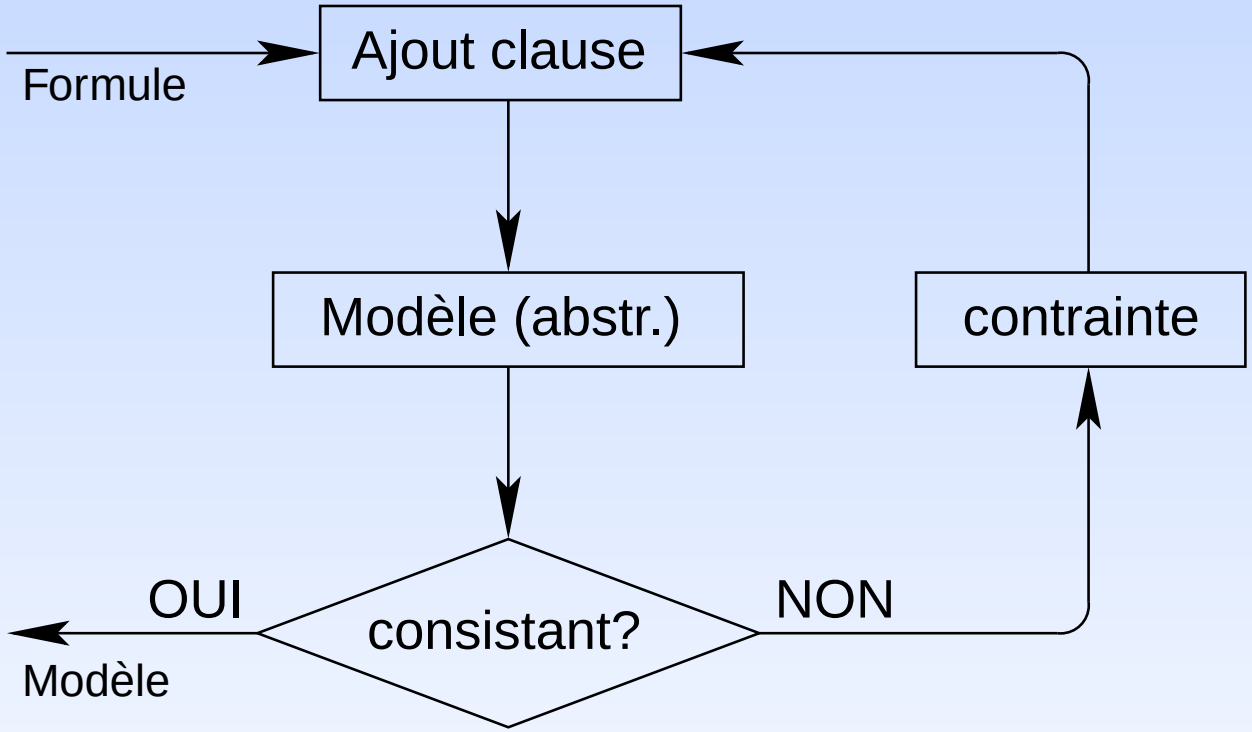
Implémentations: Berkmin, Limmat, zChaff,. . .

Un ensemble de clauses en entrée, un modèle (total) en sortie, ou insatisfaisable.

Acceptent les clauses de façon incrémentale.

Propositionnel. . . DPLL (2)

Lazy evaluation

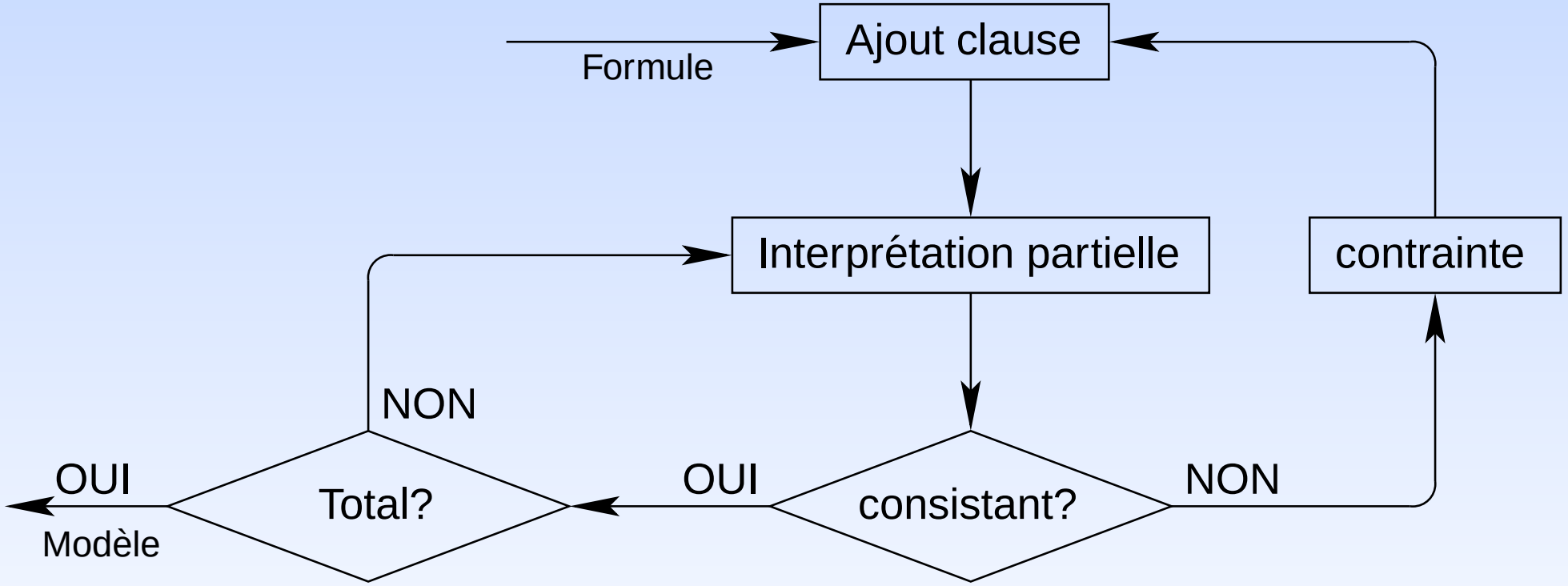


La clause de raffinement doit être la plus générale possible: ensemble minimal inconsistant

Techniques pour construire un modèle partiel de l'abstraction

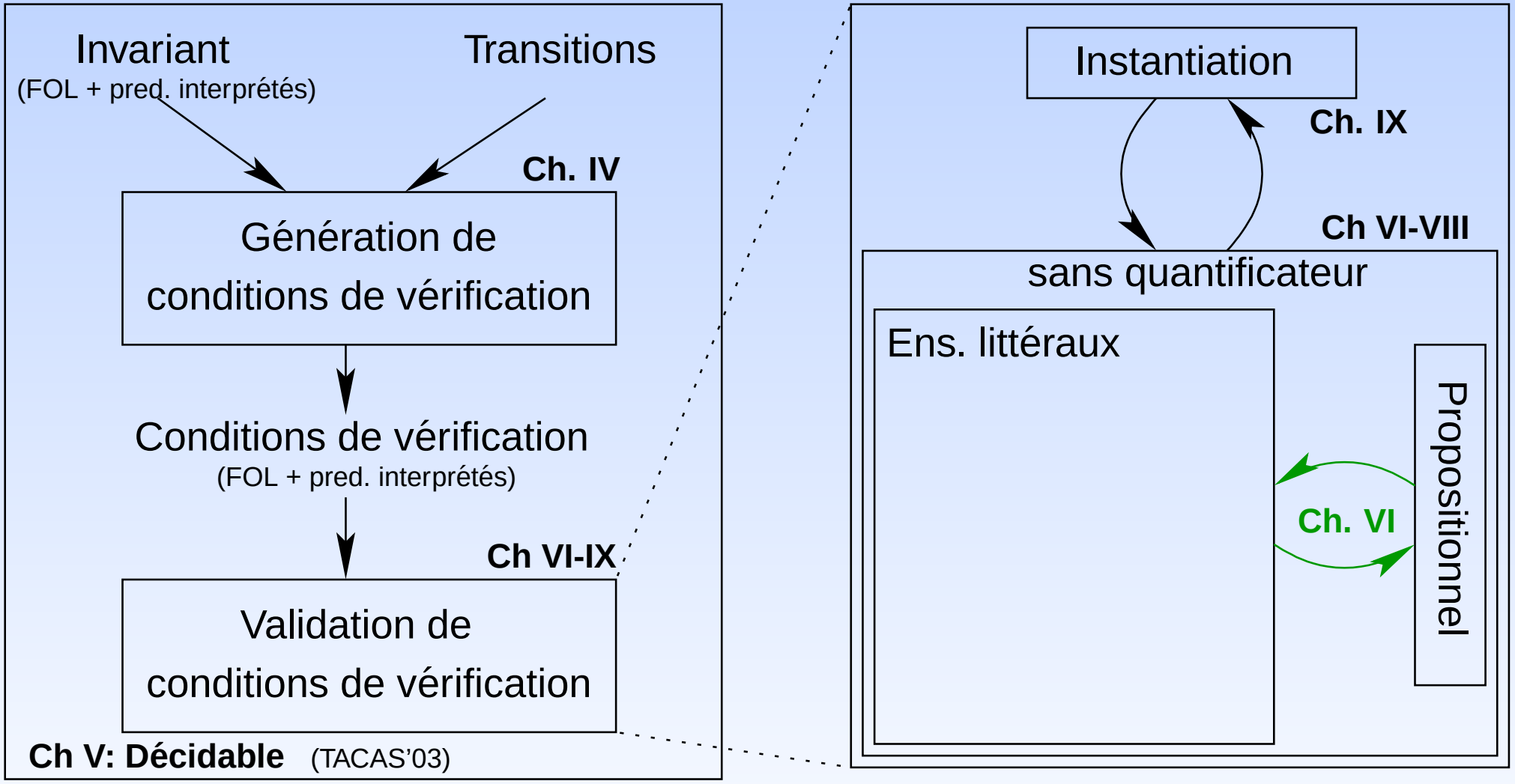
Propositionnel. . . DPLL (2)

Eager evaluation

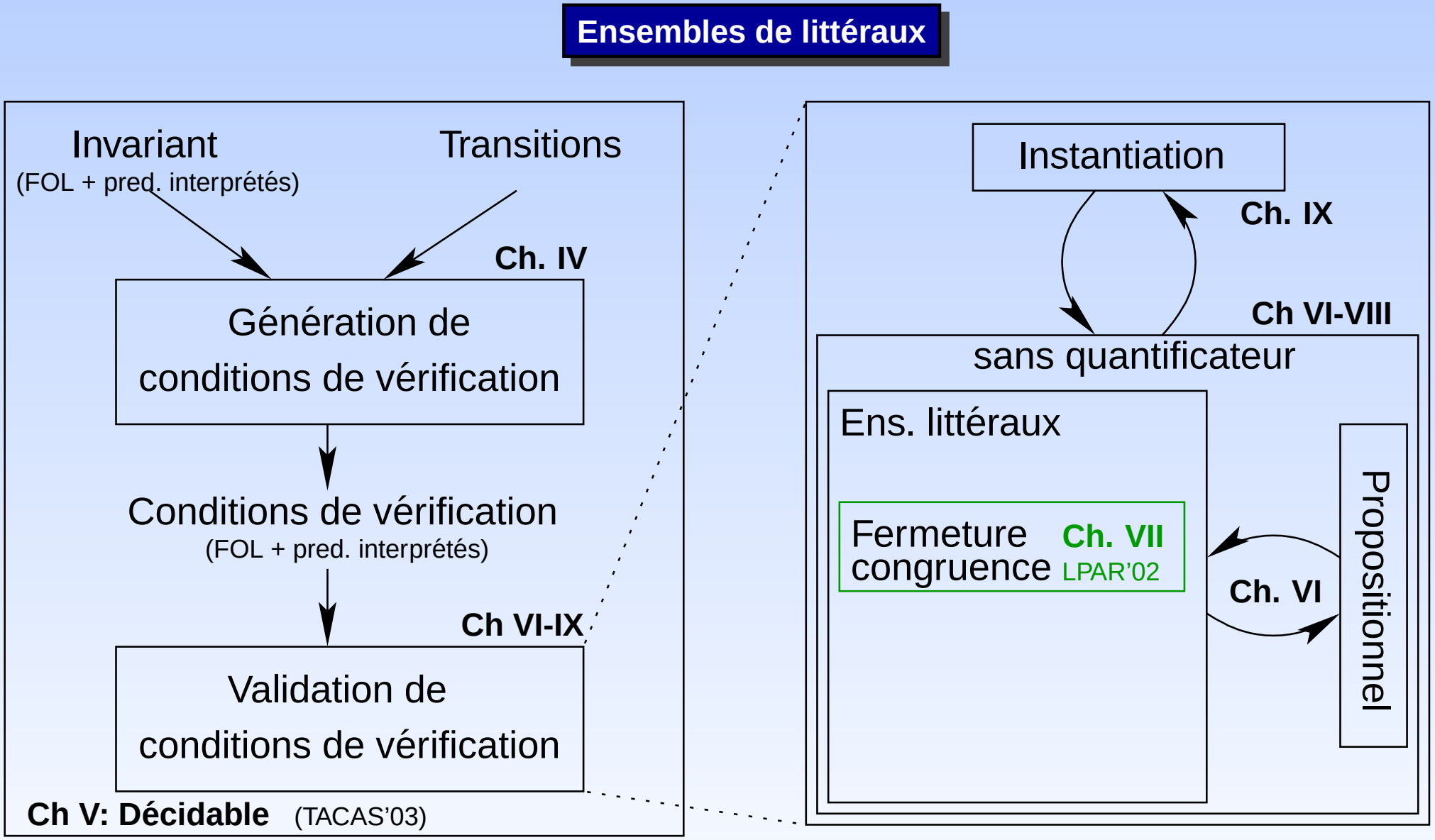


La procédure pour les littéraux doit accepter les littéraux de manière incrémentale

Propositionnel. . . : conclusion



Procédure pour les littéraux: incrémentale, produire des preuves


Ensembles de littéraux
Invariant
(FOL + pred. interprétés)
Transitions
Ch. IV
Génération de conditions de vérification
Conditions de vérification
(FOL + pred. interprétés)
Ch VI-IX
Validation de conditions de vérification
Ch V: Décidable (TACAS'03)
Instantiation
Ch. IX
Ch VI-VIII
sans quantificateur
Ens. littéraux
Fermeture congruence
Ch. VII
LPAR'02
Ch. VI
Propositionnel

Fermeture de congruence

Construction de la relation de congruence entre termes induite par des égalités.

Permet de décider la satisfaisabilité d'un ensemble de littéraux avec des fonctions et prédicats non interprétés, avec égalité.

$$p(f(x)), \neg p(f(y))$$

Fermeture de congruence

Construction de la relation de congruence entre termes induite par des égalités.

Permet de décider la satisfaisabilité d'un ensemble de littéraux avec des fonctions et prédicats non interprétés, avec égalité.

$x = y, p(f(x)), \neg p(f(y))$ inconsistant

Fermeture de congruence

Construction de la relation de congruence entre termes induite par des égalités.

Permet de décider la satisfaisabilité d'un ensemble de littéraux avec des fonctions et prédicats non interprétés, avec égalité.

$x = y, p(f(x)), \neg p(f(y))$ inconsistant

$x = y, f(x) \neq f(y)$ inconsistant

Fermeture de congruence

Construction de la relation de congruence entre termes induite par des égalités.

Permet de décider la satisfaisabilité d'un ensemble de littéraux avec des fonctions et prédicats non interprétés, avec égalité.

$x = y, p(f(x)), \neg p(f(y))$ inconsistant

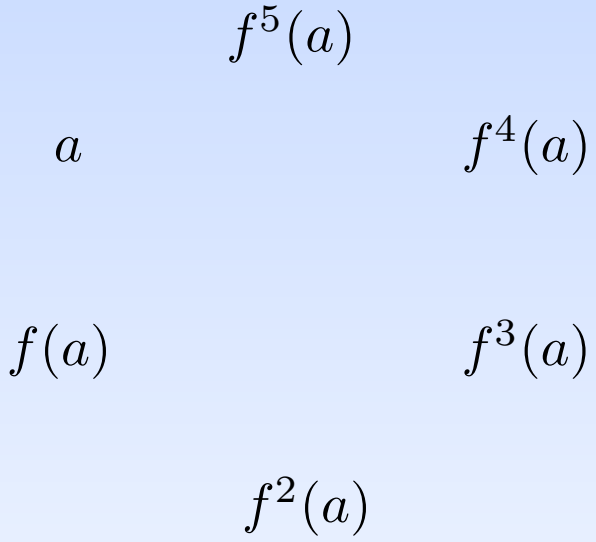
$x = y, f(x) \neq f(y)$ inconsistant

Propriétés importantes

- accepter les prédicats, et inégalités
- incrémental (accepter une à une les égalités)
- incrémental (2) (accepter de nouveaux littéraux)
- produire des preuves (cf. interaction avec méthodes propositionnelles)
- être efficace: $n \ln(n)$.

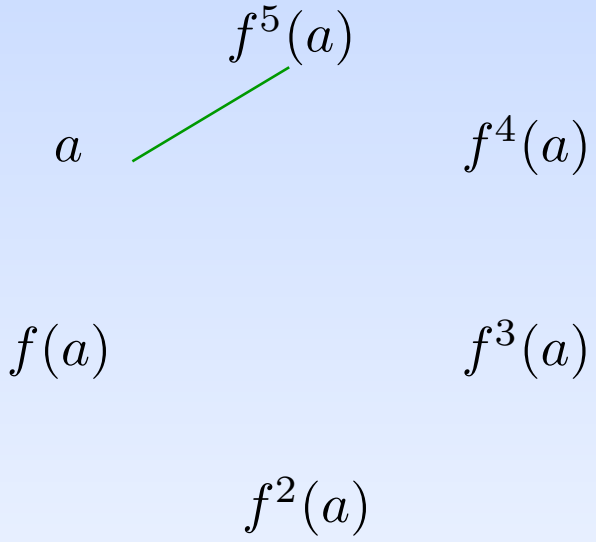
Fermeture de congruence (2)

$f(f(f(f(f(a))))) = a$



Fermeture de congruence (2)

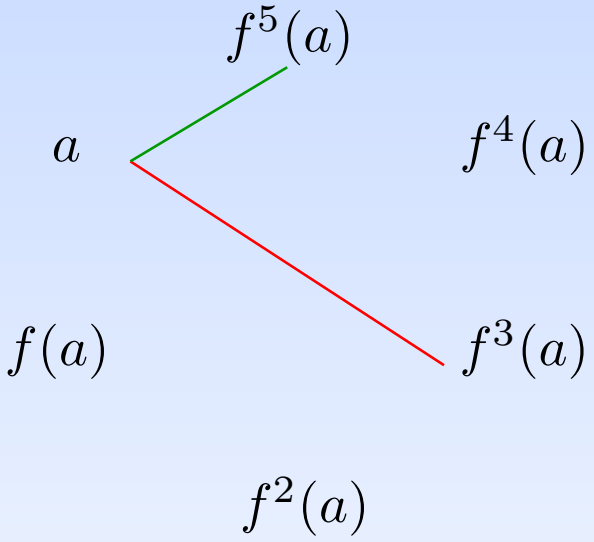
$f(f(f(f(f(a))))) = a$



Fermeture de congruence (2)

$f(f(f(f(f(a))))) = a$

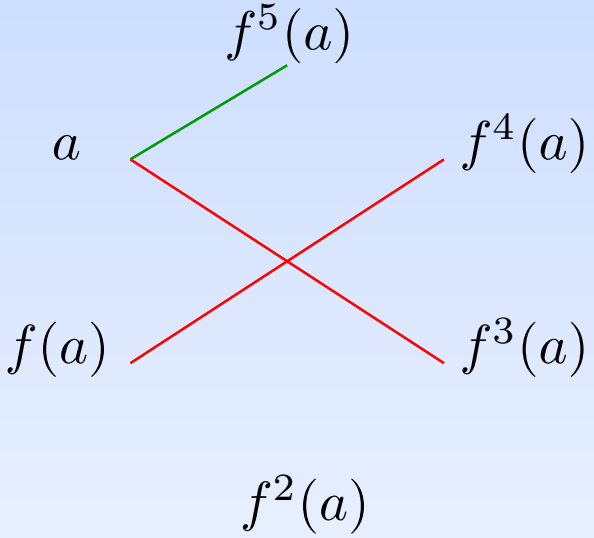
$f(f(f(a))) = a$



Fermeture de congruence (2)

$f(f(f(f(f(a))))) = a$

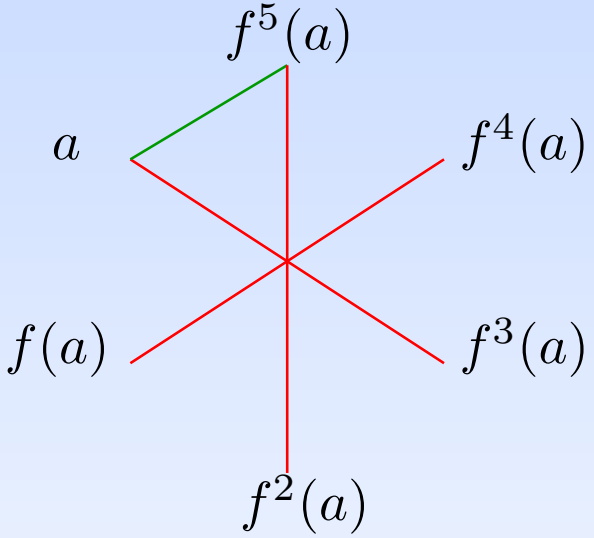
$f(f(f(a))) = a$



Fermeture de congruence (2)

$f(f(f(f(f(a))))) = a$

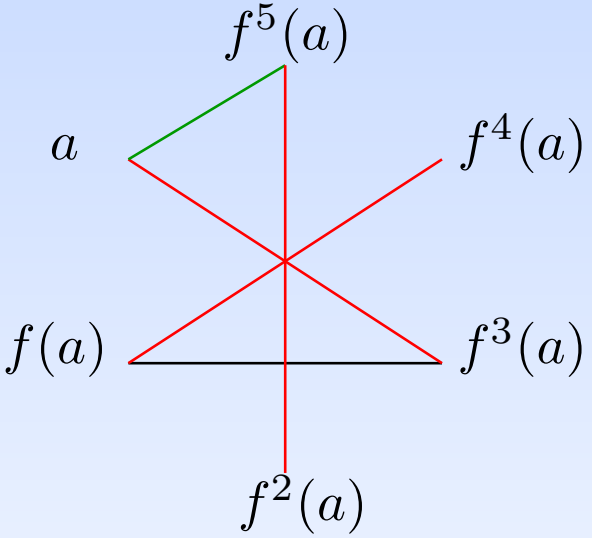
$f(f(f(a))) = a$



Fermeture de congruence (2)

$f(f(f(f(f(a))))) = a$

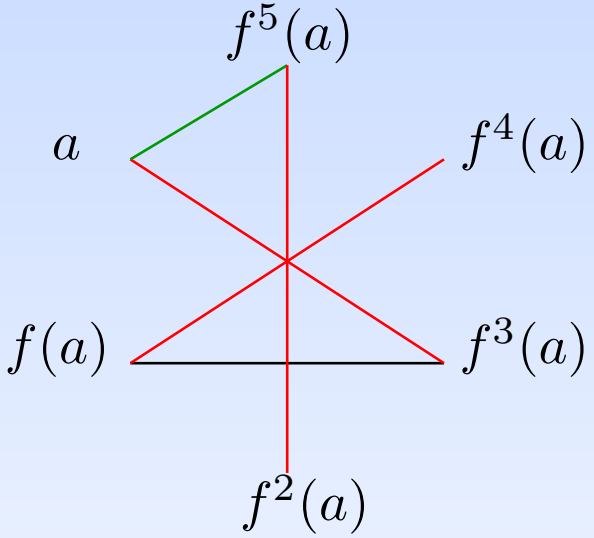
$f(f(f(a))) = a$



Fermeture de congruence (2)

$f(f(f(f(f(a))))) = a$

$f(f(f(a))) = a$

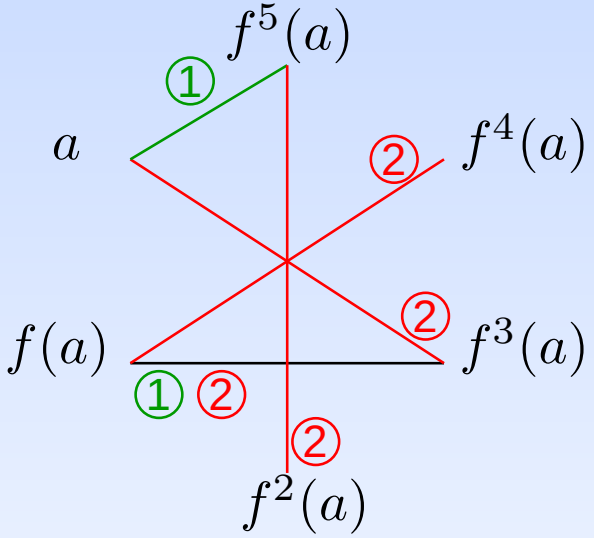


$\{f(f(f(a))) = a, f(f(f(f(f(a))))) = a, a \neq f(a)\}$:
tous les littéraux sont nécessaires à l'insatisfaisabilité.

Fermeture de congruence (2)

$f(f(f(f(f(a))))) = a$

$f(f(f(a))) = a$



$\{f(f(f(a))) = a, f(f(f(f(f(a))))) = a, a \neq f(a)\}$:
tous les littéraux sont nécessaires à l'insatisfaisabilité.

$\{f(f(f(a))) = a, f(f(f(f(f(a))))) = a, f(f(f(f(a)))) \neq f(a)\}$:
seuls les premier et dernier littéraux sont nécessaires à l'insatisfaisabilité.

Combinaison de théories

Ensemble de littéraux mixtes:

$$\{x \leq y,\ y \leq x + f(x),\ P(h(x) - h(y)),\ \neg P(0),\ f(x) = 0\}$$

contient des prédicats et fonctions non interprétés (P, f, h, c.f. fermeture de congruence), mais aussi de l'arithmétique ($+$, $-$, $\leq$, 0)

L'introduction de nouvelles variables permet de séparer les littéraux mixtes en littéraux purs:

$$\begin{aligned} L_1 &= \{x \leq y,\ y \leq x + v_1,\ v_1 = 0,\ v_2 = v_3 - v_4,\ v_5 = 0\} \\ L_2 &= \{P(v_2),\ \neg P(v_5),\ v_1 = f(x),\ v_3 = h(x),\ v_4 = h(y)\}. \end{aligned}$$

Combinaison de théories (2)

L'échange d'égalités permet aux procédures de décision de collaborer:

$$\begin{aligned} L_1 &= \{x \leq y,\ y \leq x + v_1,\ v_1 = 0, v_2 = v_3 - v_4,\ v_5 = 0\} \\ L_2 &= \{P(v_2),\ \neg P(v_5),\ v_1 = f(x),\ v_3 = h(x),\ v_4 = h(y)\} \end{aligned}$$

L_2'' est inconsistant.

Combinaison de théories (2)

L'échange d'égalités permet aux procédures de décision de collaborer:

$$\begin{aligned} L_1 &= \{x \leq y,\ y \leq x + v_1,\ v_1 = 0, v_2 = v_3 - v_4,\ v_5 = 0\} \\ L_2 &= \{P(v_2),\ \neg P(v_5),\ v_1 = f(x),\ v_3 = h(x),\ v_4 = h(y)\} \end{aligned}$$

de L_1, $x = y$:

$$\begin{aligned} L_1 &= \{x \leq y,\ y \leq x + v_1,\ v_1 = 0,\ v_2 = v_3 - v_4,\ v_5 = 0\} \\ L_2' &= \{P(v_2),\ \neg P(v_5),\ v_1 = f(x),\ v_3 = h(x),\ v_4 = h(y),\ x = y\} \end{aligned}$$

L_2'' est inconsistant.

Combinaison de théories (2)

L'échange d'égalités permet aux procédures de décision de collaborer:

$$\begin{aligned} L_1 &= \{x \leq y,\ y \leq x + v_1,\ v_1 = 0, v_2 = v_3 - v_4,\ v_5 = 0\} \\ L_2 &= \{P(v_2),\ \neg P(v_5),\ v_1 = f(x),\ v_3 = h(x),\ v_4 = h(y)\} \end{aligned}$$

de L_1, $x = y$:

$$\begin{aligned} L_1 &= \{x \leq y,\ y \leq x + v_1,\ v_1 = 0,\ v_2 = v_3 - v_4,\ v_5 = 0\} \\ L_2' &= \{P(v_2),\ \neg P(v_5),\ v_1 = f(x),\ v_3 = h(x),\ v_4 = h(y),\ x = y\} \end{aligned}$$

de L_2', $v_3 = v_4$:

$$\begin{aligned} L_1' &= \{x \leq y,\ y \leq x + v_1,\ v_1 = 0,\ v_2 = v_3 - v_4,\ v_5 = 0, v_3 = v_4\} \\ L_2' &= \{P(v_2),\ \neg P(v_5),\ v_1 = f(x),\ v_3 = h(x),\ v_4 = h(y),\ x = y\} \end{aligned}$$

L_2'' est inconsistant.

Combinaison de théories (2)

L'échange d'égalités permet aux procédures de décision de collaborer:

$$\begin{aligned} L_1 &= \{x \leq y,\ y \leq x + v_1,\ v_1 = 0, v_2 = v_3 - v_4,\ v_5 = 0\} \\ L_2 &= \{P(v_2),\ \neg P(v_5),\ v_1 = f(x),\ v_3 = h(x),\ v_4 = h(y)\} \end{aligned}$$

de L_1, $x = y$:

$$\begin{aligned} L_1 &= \{x \leq y,\ y \leq x + v_1,\ v_1 = 0,\ v_2 = v_3 - v_4,\ v_5 = 0\} \\ L_2' &= \{P(v_2),\ \neg P(v_5),\ v_1 = f(x),\ v_3 = h(x),\ v_4 = h(y),\ x = y\} \end{aligned}$$

de L_2', $v_3 = v_4$:

$$\begin{aligned} L_1' &= \{x \leq y,\ y \leq x + v_1,\ v_1 = 0,\ v_2 = v_3 - v_4,\ v_5 = 0, v_3 = v_4\} \\ L_2' &= \{P(v_2),\ \neg P(v_5),\ v_1 = f(x),\ v_3 = h(x),\ v_4 = h(y),\ x = y\} \end{aligned}$$

de L_1', $v_2 = v_5$:

$$\begin{aligned} L_1' &= \{x \leq y,\ y \leq x + v_1,\ v_1 = 0,\ v_2 = v_3 - v_4,\ v_5 = 0, v_3 = v_4\} \\ L_2'' &= \{P(v_2),\ \neg P(v_5),\ v_1 = f(x),\ v_3 = h(x),\ v_4 = h(y),\ x = y, v_2 = v_5\} \end{aligned}$$

L_2'' est inconsistant.

Combinaison de théories (3)

Techniques classiques de combinaison de théories:

- ✘ sans types
- ✘ théories du premier ordre
- ✘ théories infiniment stables.

Ces trois prérequis sont des contraintes fortes dans notre contexte.

Combinaison de théories (3)

Techniques classiques de combinaison de théories:

- ✘ sans types
- ✘ théories du premier ordre
- ✘ théories infiniment stables.

Ces trois prérequis sont des contraintes fortes dans notre contexte.

Combinaison (version 2):

- ✔ logique à types
- ✔ les “théories” sont des ensembles quelconques de structures
- ✔ les “théories” peuvent n’avoir que des modèles finis (ens. de processus).

Combinaison de théories (4)

De théories du premier ordre vers ensembles de structures:

Pour chaque type, une “théorie” au moins est *flexible*. Les éléments du domaines sont liés aux interprétations *modulo une permutation*.

Combinaison de théories (4)

De théories du premier ordre vers ensembles de structures:

Pour chaque type, une “théorie” au moins est *flexible*. Les éléments du domaines sont liés aux interprétations *modulo une permutation*.

D’infiniment stables vers quelconques:

Certaines “théories” ont d’excellentes propriétés de *compatibilité*, elles peuvent être combinées sans contrainte avec d’autres théories quelconques.

Exemple: on peut ajouter des fonctions et prédicats non interprétés à tout langage (FOL, sans quantificateur) décidable. Il n’est pas requis que cette “théorie” soit infiniment stable.

Combinaison de théories (5)

La théorie des tableaux

Deux fonctions: $\mathrm{read}(T, i)$, $\mathrm{write}(T, i, e)$.

Combinaison de théories (5)

La théorie des tableaux

Deux fonctions: $\mathrm{read}(T, i)$, $\mathrm{write}(T, i, e)$.

Compatible avec toute théorie qui n'utilise pas le type "tableau":

$$\mathrm{write}(T, i+1, \mathrm{cons}(0, \mathrm{read}(T', i))) = T''.$$

Combinaison de théories (5)

La théorie des tableaux

Deux fonctions: $\mathrm{read}(T, i)$, $\mathrm{write}(T, i, e)$.

Compatible avec toute théorie qui n'utilise pas le type "tableau":

$$\mathrm{write}(T, i+1, \mathrm{cons}(0, \mathrm{read}(T', i))) = T''.$$

Précaution avec les théories non infiniment stables:

Exemple: les éléments et indices appartiennent à un ensemble à deux éléments (Théorie combinée avec celle des tableaux).

Soit les littéraux: $\{T_1 \neq T_2 \neq T_3 \neq T_4 \neq T_5\}$.

Pas de variable de type élément-indice, pas d'équation à propager.

Combinaison de théories (5)

La théorie des tableaux

Deux fonctions: $\mathrm{read}(T, i)$, $\mathrm{write}(T, i, e)$.

Compatible avec toute théorie qui n'utilise pas le type "tableau":

$$\mathrm{write}(T, i+1, \mathrm{cons}(0, \mathrm{read}(T', i))) = T''.$$

Précaution avec les théories non infiniment stables:

Exemple: les éléments et indices appartiennent à un ensemble à deux éléments (Théorie combinée avec celle des tableaux).

Soit les littéraux: $\{T_1 \neq T_2 \neq T_3 \neq T_4 \neq T_5\}$.

Pas de variable de type élément-indice, pas d'équation à propager.

Cependant, on ne peut conclure à la satisfaisabilité.

Pour avoir la propriété de compatibilité, il faut éliminer les inégalités entre tableaux:

$$\{\mathrm{read}(T_1, i_{1,2}) \neq \mathrm{read}(T_2, i_{1,2}), \mathrm{read}(T_1, i_{1,3}) \neq \mathrm{read}(T_3, i_{1,3}), \ldots \mathrm{read}(T_4, i_{4,5}) \neq \mathrm{read}(T_5, i_{4,5})\}$$

Ici, des égalités doivent être propagées.

Conclusion

- ✔ Utilité des procédures de décision:
  Propositionnel $\longrightarrow$ Sans quantificateur $\longrightarrow$ Avec quantificateurs.
  Les conditions de vérification appartiennent à des langages décidables, dans des cas non triviaux
- ✔ Décidable, implémentable (implémentée), temps de validation acceptables (CAVEAT)

Le futur

- Des méthodes plus puissantes (application à des formules plus compliquées, plus longues)
  - collaboration propositionnel - procédure pour ens. de littéraux
  - techniques spécialisées pour l'intégration harmonieuse de procédures dans une combinaison
  - heuristiques, méthodes d'instantiation
- Un langage plus étendu
  - identification de théories avec de bonnes propriétés de compatibilité
  - constructions d'ordre supérieur
- Application à d'autres domaines des techniques de validation de formules (HARVEY).