Le projet du GdT “intégration”

M. Sighireanu (LIAFA), S. Bardin, A. Finkel et
D. Nowak (LSV), G. Sutre, A. Vincent et
F. Herbreteau (LaBRI)

Journée PERSÉE - 26 mai 2005
LIAFA, Paris

Objectifs

- **Cadre** : calcul d'accessibilité accéléré d'automates étendus infinis et hétérogènes

```
R:=R0; R':=⊥
while (not R ⊑ R') {
R' := R
R := R ⊔ post(R) ⊔ acc(R)
}
```

- Intégration de model-checkers :
  - FAST : accélération exacte pour automates à compteurs + boucles imbriquées
  - TREX : interpolation pour automates avec lossy FIFO + compteurs + horloges

Intégrer l'expertise de l'utilisateur

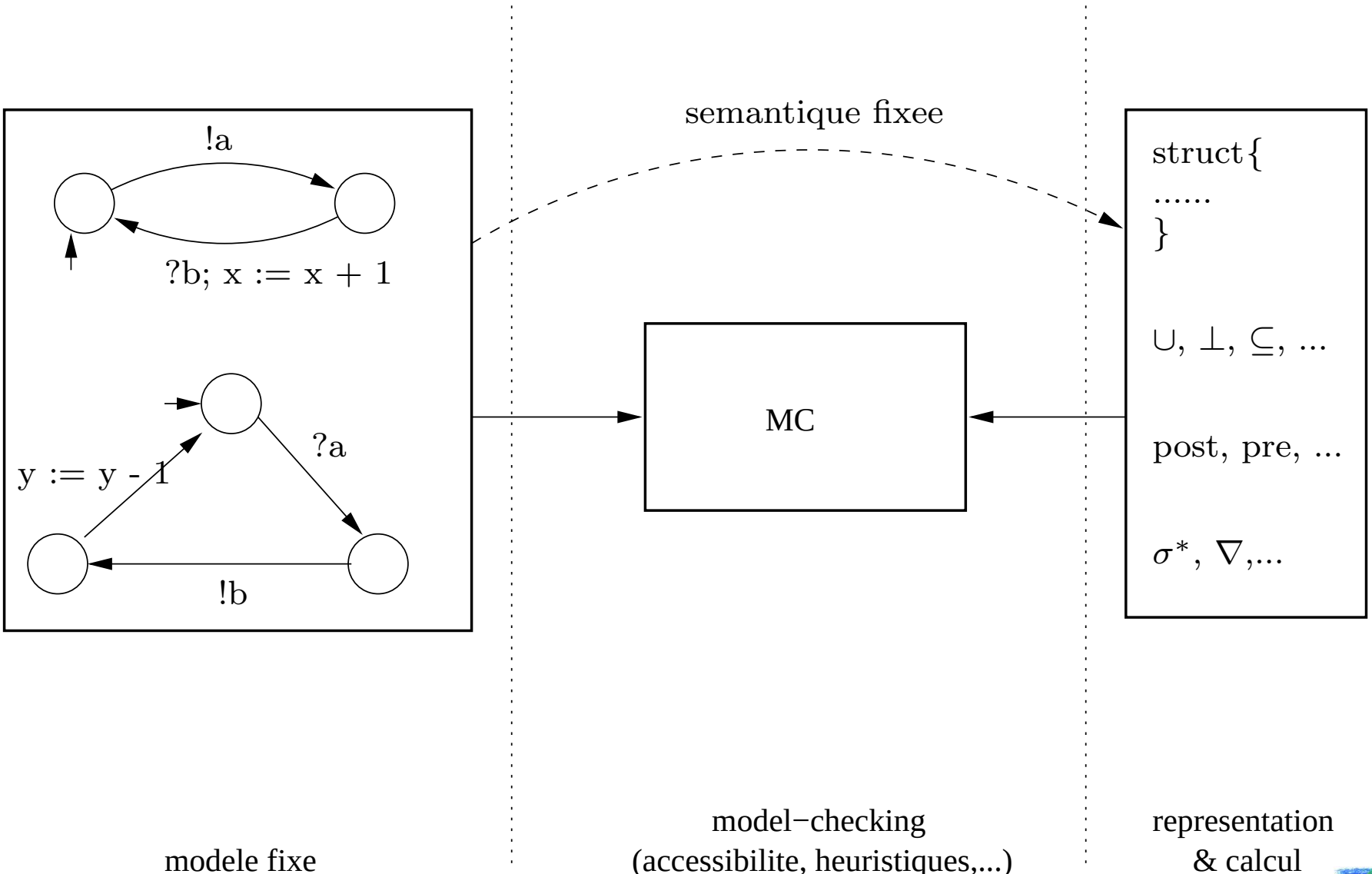
- **Expérience** : l'approche "presse-bouton" n'est actuellement pas adaptée, la vérification doit être guidée :
  - diagnostique : assistance à l'utilisateur (ex : proposition de représentation adaptée dans TReX)
  - stratégie : programmation de la vérification "à la HyTech" (ex : quelle heuristique utiliser ? Ordre des calculs ? ...)

- Plateforme d'expérimentation générique et modulaire pour les techniques d'accélération et d'abstraction (projet VeriFast)

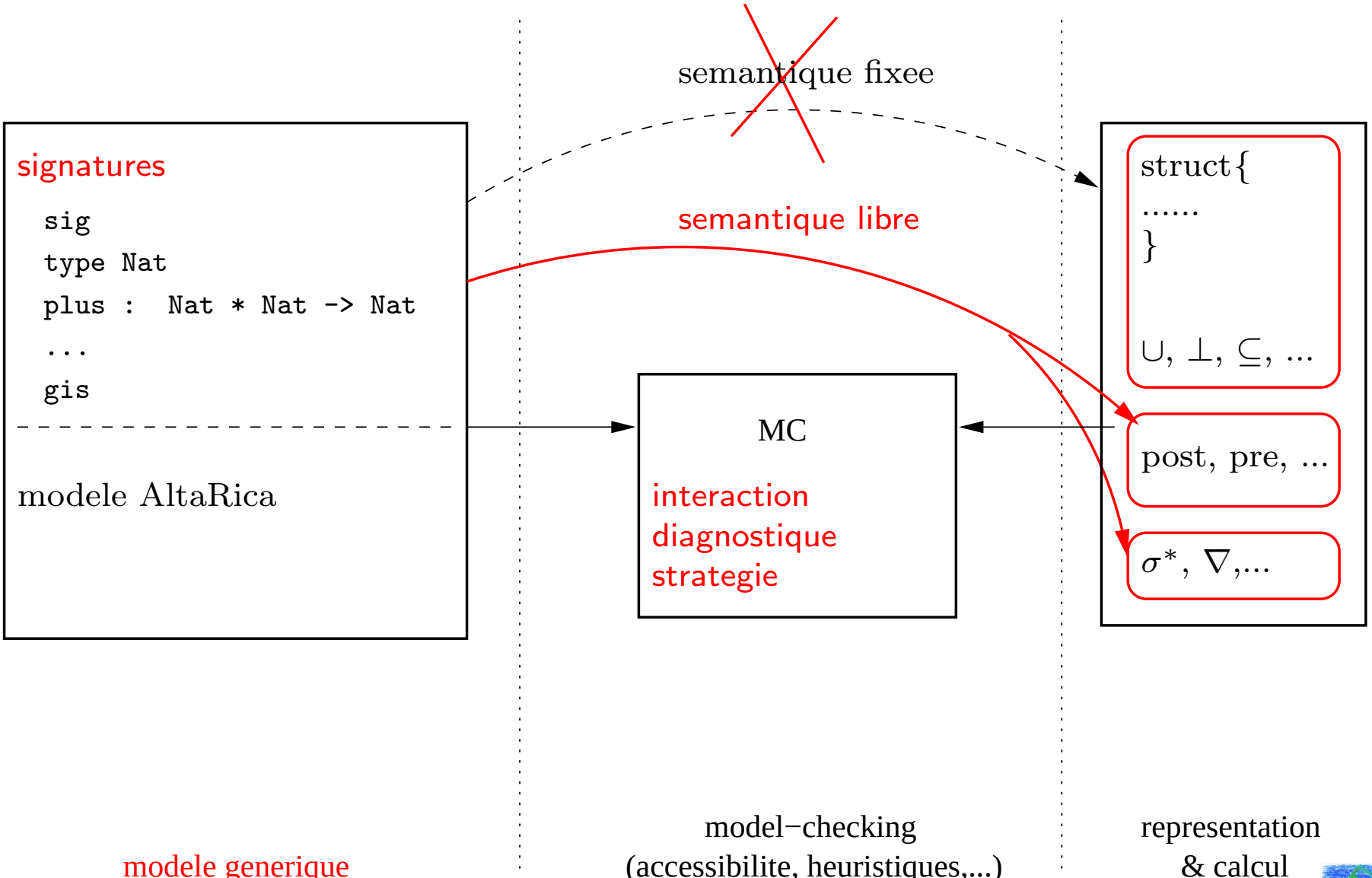
- **Solution** : langage de script et boucle interactive

```
...
> let Rinit = region_of_string(``x<=2 and y=7'');;
> cap (post*(Rinit)) Rbad;;
empty : int
...
```

Architecture classique d'un model-checker



Architecture générique et modulaire



Notre modèle générique : AltaRica + TAD

- Système à contraintes $(\overrightarrow{x}, E, A, \mathtt{Op}, T)$
  - $\overrightarrow{x}$: variables, E : événements, $A(\overrightarrow{x})$: assertion, $\mathtt{Op}$: opérations
  - T : transitions gardées $G(\overrightarrow{x}) \xrightarrow{e} \overrightarrow{x} := \rho(\overrightarrow{x}, \mathtt{Op})$

- **Généricité** : domaine $\mathbb{D}$ et opérations Op (signatures)

```
sig
type Nat
0, 1 : Nat
plus : Nat * Nat -> Nat
geq : Nat * Nat -> bool
gis
```

sémantique des opérations $\delta : \mathbb{D}^{\overrightarrow{x}} \times \mathtt{Op} \rightarrow \mathbb{D}^{\overrightarrow{x}}$

Calcul symbolique : les régions

- **Objectif** : représentation symbolique d'ensembles infinis + opérations ensemblistes

$$(\overrightarrow{x}, R, \bot, \top, \sqcup, \sqcap, \sqsubseteq, [\![\cdot]\!])$$

  - $\overrightarrow{x}$ variables représentées
  - R ens. de régions interprétées par $[\![\cdot]\!]$: $R \longrightarrow 2^{\mathbb{D}^{\overrightarrow{x}}}$
  - $\bot$ est la région vide : $[\![\bot]\!] = \emptyset$ et $\top$ est la région univers : $[\![\top]\!] = \mathbb{D}^{\overrightarrow{x}}$
  - $\sqcup$ est l'union de régions : $[\![r \sqcup r']\!] \supseteq [\![r]\!] \cup [\![r']\!]$ (idéalement $=$), de même pour l'intersection de régions $\sqcap$
  - $\sqsubseteq$ est le test d'inclusion : $r \sqsubseteq r' \Rightarrow [\![r]\!] \subseteq [\![r']\!]$ (idéalement $\Leftrightarrow$)

Calcul symbolique : évaluation des opérations

- **Objectif** : évaluation des opérations sur les régions
- Sémantique symbolique des opérations $\hat{\delta}$: $R \times \mathtt{Op} \longrightarrow R$ telle que :

$$[\![\ \hat{\delta}(r, \mathrm{op})\ ]\!] = \left\{ v' \in \mathbb{D}^{\overrightarrow{x}} \mid \exists v \in [\![\ r\ ]\!]\ \ v' = \delta(v, \mathrm{op}) \wedge v' \in A \right\}$$

- Pour une transition $G(\overrightarrow{x}) \xrightarrow{e} \overrightarrow{x} := \rho(\overrightarrow{x}, \mathtt{Op})$:

$$\mathtt{post}(r, e) = \hat{\delta}(r \sqcap r_G, \rho)$$

où r_G est la région correspondant à G

- C'est ici qu'est définie la sémantique des opérations !

Calcul symbolique : les accélérations (1/2)

- **Objectif** : convergence du calcul d'accessibilité
- Accélération exacte d'une séquence d'opérations σ :

$$[\![\hat{\sigma}_A(r,\sigma)]\!] = \left\{ v' \in \mathbb{D}^{\overrightarrow{x}} \mid \exists v \in [\![r]\!], \exists i \; v' = \delta(v, \sigma^i) \wedge v' \in A \right\}$$

- Interpolation d'une séquence d'opérations σ par Δ :

$$[\![\hat{\sigma}_I(r,\sigma,\Delta)]\!] = \left\{ v' \in \mathbb{D}^{\overrightarrow{x}} \mid \exists v \in [\![r]\!], \exists i \; v' = v + i.\Delta \wedge v' \in A \right\}$$
$$\supseteq [\![\hat{\sigma}_A(r,\sigma)]\!]$$

$\Delta = r_{i+1} - r_i$ pour tout $i \in [0; n]$ où $r_i = \delta(r, \sigma^i)$ et $n \in \mathbb{N}$

Calcul symbolique : les accélérations (2/2)

- Widening appliqué à une séquence d'opérations σ :

$$[\![\hat{\sigma}_W(r,\sigma)]\!] = \{v' \in \mathbb{D}^{\vec{x}} \mid \exists v \in [\![r]\!], \exists i \; v' \in r_i \wedge v' \in A\}$$
$$\supseteq [\![\hat{\sigma}_A(r,\sigma)]\!]$$

  où la séquence $(r_i)_{i\geq 0}$ est définie par $r_{i+1} = r_i \nabla \hat{\delta}(r, \sigma^{i+1})$ et $r_0 = r$, et l'opérateur ∇ doit satisfaire :
  - $r \nabla r' \supseteq r \cup r'$ pour toutes régions r, r'
  - la séquence croissante $(r_i)_{i\geq 0}$ converge

- D'une façon générale, une fonction d'accélération $\hat{\sigma}$ satisfait :

$$[\![\hat{\sigma}(r,\sigma)]\!] \supseteq [\![\hat{\sigma}_A(r,\sigma)]\!]$$

- Extension à des langages d'opérations plutôt que des séquences (ex : $(\sigma_1 + \sigma_2)^*$ dans FAST)

Modularité du calcul symbolique

La séparation des 3 modules permet des stratégies de model-checking ainsi que l'expérimentation

- **Régions** : représentation et opérations ensemblistes

  ex : NDD, Automates partagés, PDBM, Polyèdres convexes

- **Représentations symboliques** : évaluation des opérations `post`/`pre` (sémantique)

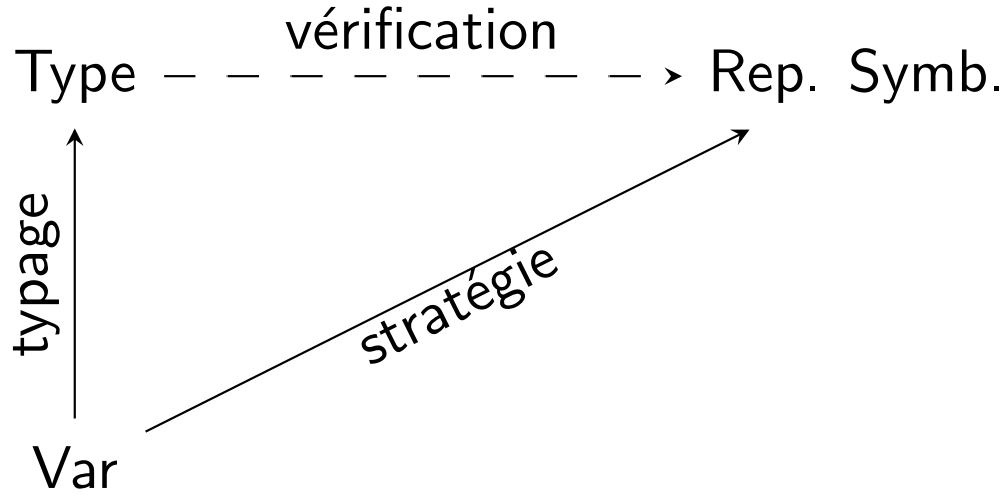
  ex : $P.\overrightarrow{x} \leq q \rightarrow \overrightarrow{x} := A.\overrightarrow{x} + b$ ou $\phi_{\text{pres.}} \rightarrow \overrightarrow{x} := A.\overrightarrow{x} + b$

- **Accélérations** : clôture transitive de séquences d'opérations

  ex : [BW] (exact), [FL] (exact), [BLW] (interpolation), [AAB] (interpolation), [H] (widening)

Définition de la sémantique des opérations

- **Sémantique des opérations** : définie par `post`/`pre`, donc par les représentations symboliques
- Sémantique fixée par l'utilisateur par association "variable-représentation symbolique"



- Stratégie de vérification basée sur le choix de la représentation la mieux adaptée

Évaluation des opérations

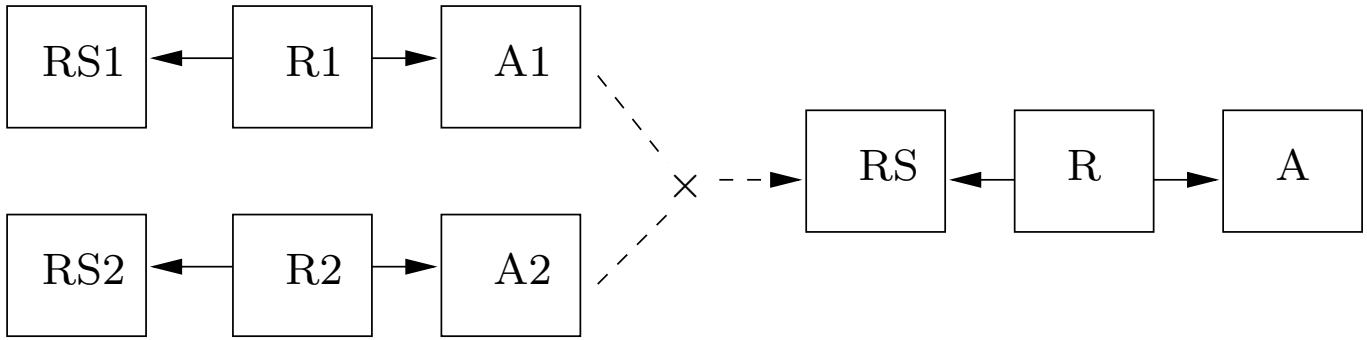
$$\texttt{geq(x,0)} \rightarrow \texttt{x := plus(x,1)}$$

Le module de représentation symbolique doit interpréter les opérateurs

- **Garde** :
  - traduction de `geq(x,0)` dans le format de la région sous-jacente (ex : inéquation linéaire $x \geq 0$)
  - obtention la région r_G correspondante, et test $r \sqcap r_G \neq \bot$
- **Affectation** :
  - traduction du terme `plus(x,1)` dans une représentation permettant son évaluation (ex : fonction affine pour les NDD)
  - évaluation de la fonction obtenue sur la région $r \sqcap r_G$
- Utilisation d'un cache pour ne traduire qu'une seule fois

Composition des représentations symboliques

- **Systèmes hétérogènes** : il est nécessaire de composer les régions, les représentations symboliques et les accélérations



- Le produit cartésien (orthogonal) peut être défini (presque) génériquement

- S'il y a de l'information partagée : définir un produit ad hoc (ex : les PDBM de TReX)

Produit cartésien

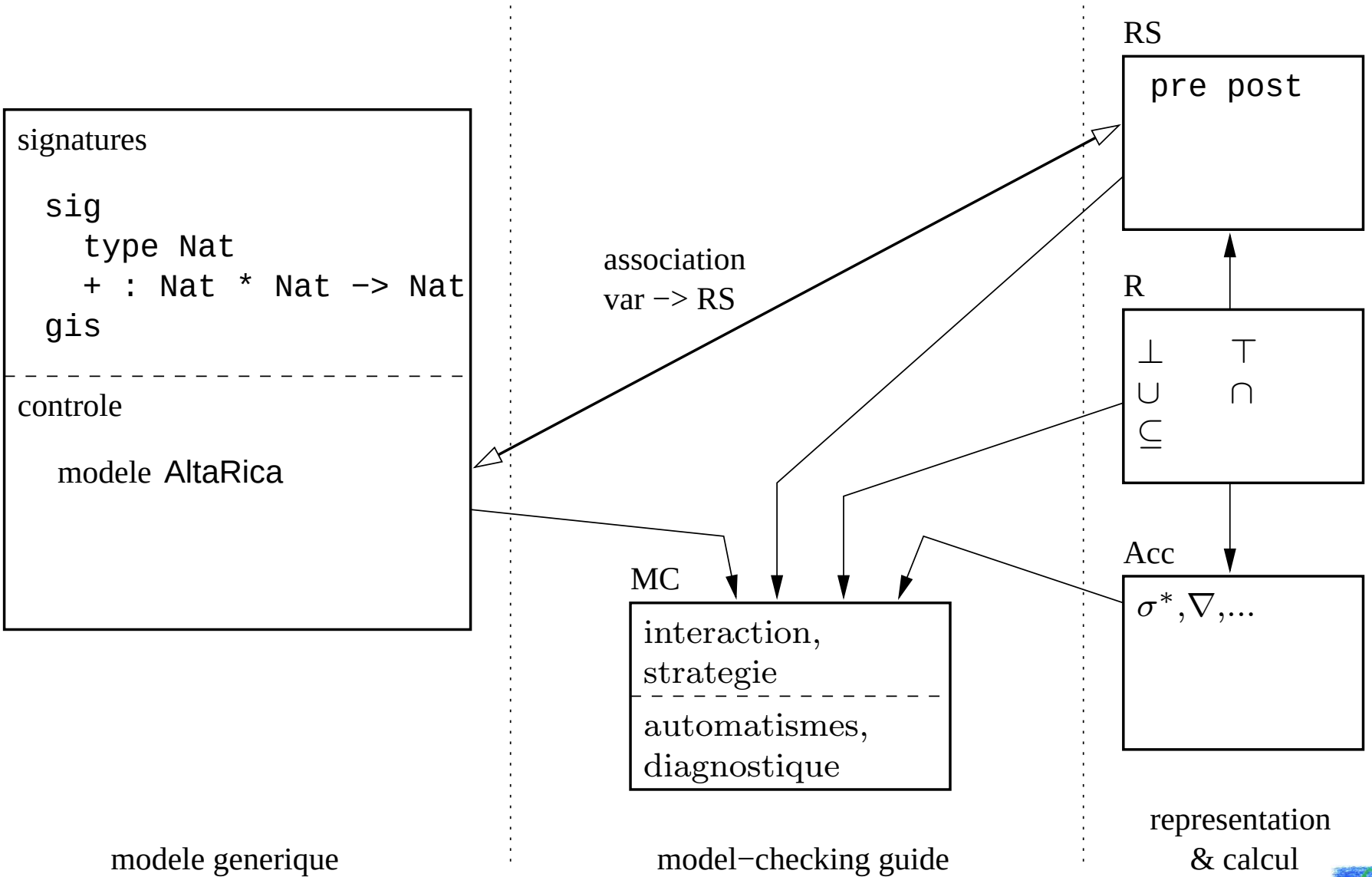
- **Régions** : $\bot = (\bot_1, \bot_2)$, $\sqsubseteq = \sqsubseteq_1 \times \sqsubseteq_2$,...
- **Représentations symboliques** :
  - il faut séparer les transitions suivant ce qui concerne chaque sous-membre :

    recv(q,a) $\rightarrow$ q:=send(q,b); x:=x+1

    donne :

    recv(q,a) $\rightarrow$ q:=send(q,b) et true $\rightarrow$ x:=x+1
  - attention aux gardes qu'il faut évaluer globalement avant les affectations
- **Accélérations** : prévoir une procédure globale pour gérer correctement les gardes

Notre architecture générique



Conclusion

- **Objectif** : développer un assistant de vérification accélérée modulaire et générique
  - validation de l'architecture
  - intégration de TReX et Fast dans l'architecture (régions, représentations symboliques, accélérations)
  - mise en œuvre de la boucle d'interaction
- Gérer l'abstraction de modèles
- À plus long terme, intégration d'autres outils (automates partagés, Lash,...)