Systèmes de transitions symboliques et hiérarchiques pour la conception et la validation de modèles B raffinés

Nicolas Stouls

Équipe VASCO du Laboratoire d'Informatique de Grenoble (LIG)

Travail supporté par le CNRS, ST-Microelectronics

Le 14 Décembre 2007

Travail réalisé sous la direction de :

Marie-Laure Potet *(Pr INPG)* *Directeur de thèse*
Sylvain Boulmé *(MdC INPG)* *Co-encadrant*

Plan de l'exposé

Introduction

- Omniprésence de l'informatique
  Automobiles, machines à laver, carte de paiement, etc.

Exemple : *Problème logiciel sur la Prius*

Un problème du logiciel embarqué sur la Toyota Prius fait que celle-ci peut caler ou s'arrêter lorsqu'elle roule sur l'autoroute.

Source : CNN/Money – 16/05/2005

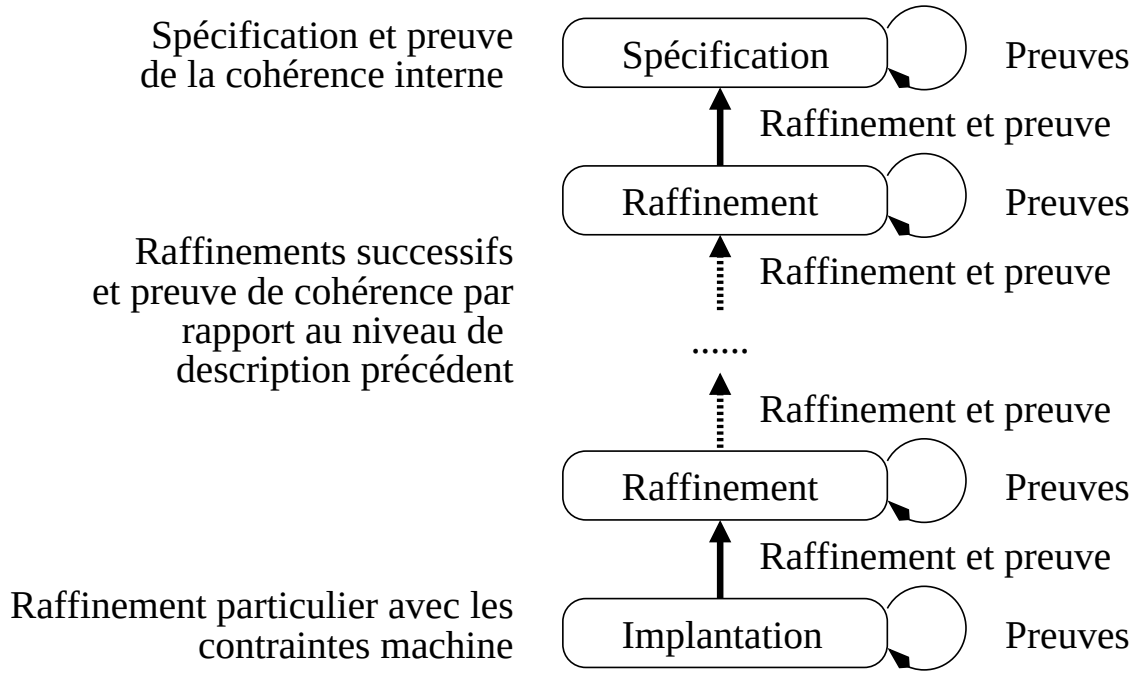
- Besoin de garanties sur la qualité des logiciels
- Méthodes de génie logiciel

Génie logiciel

- Techniques de modélisation et de développement
- Structuration du raisonnement
- Prévision au plus tôt des difficultés à venir
- Maîtrise des coûts et des délais
- Méthodes formelles :
  - Haut niveau de confiance
  - Langages expressifs
  - Sémantique des langages mathématiquement bien définie
  - Techniques de vérification décidables ou non
  - Nécessite souvent une grande expertise

Première introduction à la méthode B

- Méthode formelle de spécification et de développement
  - Langage de spécification basé sur l'affectation
  - Développement par raffinement
  - Vérifications par technique de preuve



Exemple fil rouge de spécification abstraite B événementiel

SYSTEM *Canal_De_Communication*
VARIABLES *TailleEnvoi*
INVARIANT $TailleEnvoi \in \mathbb{N}$
INITIALISATION $TailleEnvoi := 0$
EVENTS
$Envoyer \mathrel{\hat{=}}$ **SELECT** $TailleEnvoi = 0$ **THEN** $TailleEnvoi :\in \mathbb{N}_1$ **END ;**
$Traiter \mathrel{\hat{=}}$ **SELECT** $TailleEnvoi > 0$
THEN $TailleEnvoi := TailleEnvoi - 1$ **END ;**
$Reset \mathrel{\hat{=}}$ **SELECT** $TailleEnvoi > 0$ **THEN** $TailleEnvoi := 0$ **END**
END

Exemple de raffinement B événementiel

REFINEMENT *Canal_De_Communication_R*
REFINES *Canal_De_Communication*

CONSTANTS *TailleBuff*
PROPERTIES $TailleBuff \in \mathbb{N}_1$

VARIABLES *DansBuffer* , *AEnvoyer*
INVARIANT $AEnvoyer \in \mathbb{N} \land DansBuffer \in 0..TailleBuff$
$\land\ DansBuffer + AEnvoyer = TailleEnvoi$

INITIALISATION $DansBuffer := 0 \ || \ AEnvoyer := 0$
EVENTS
$Envoyer \mathrel{\widehat{=}}$ **SELECT** $AEnvoyer = 0 \land DansBuffer = 0$ **THEN** $AEnvoyer :\in \mathbb{N}_1$ **END** ;

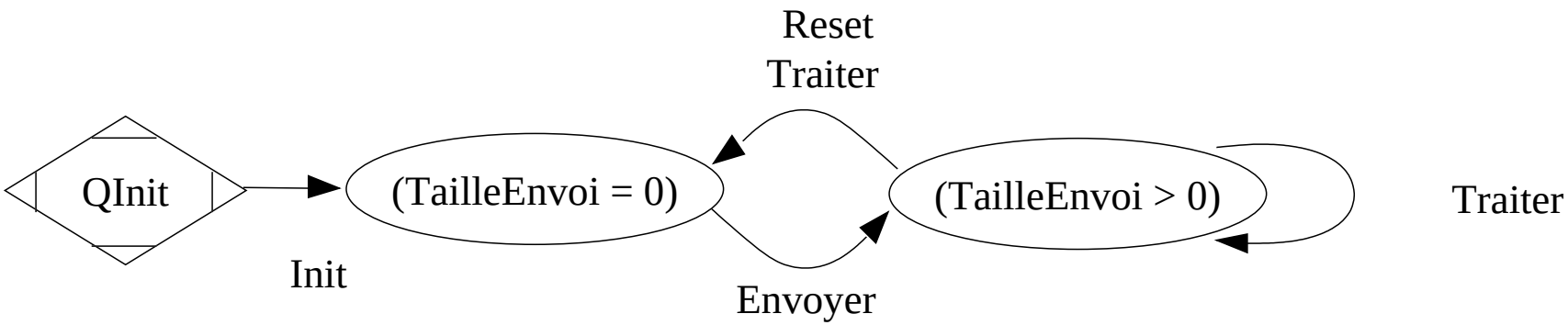
$EnvoyerSuite \mathrel{\widehat{=}}$ **SELECT** $AEnvoyer > 0 \land DansBuffer < TailleBuff$
THEN $AEnvoyer := AEnvoyer - 1 \ || \ DansBuffer := DansBuffer + 1$ **END** ;

$Traiter \mathrel{\widehat{=}}$ **SELECT** $DansBuffer > 0$ **THEN** $DansBuffer := DansBuffer - 1$ **END** ;

$Reset \mathrel{\widehat{=}}$ **SELECT** $AEnvoyer > 0 \lor DansBuffer > 0$
THEN $AEnvoyer := 0 \ || \ DansBuffer := 0$ **END**
END

Contributions de cette thèse

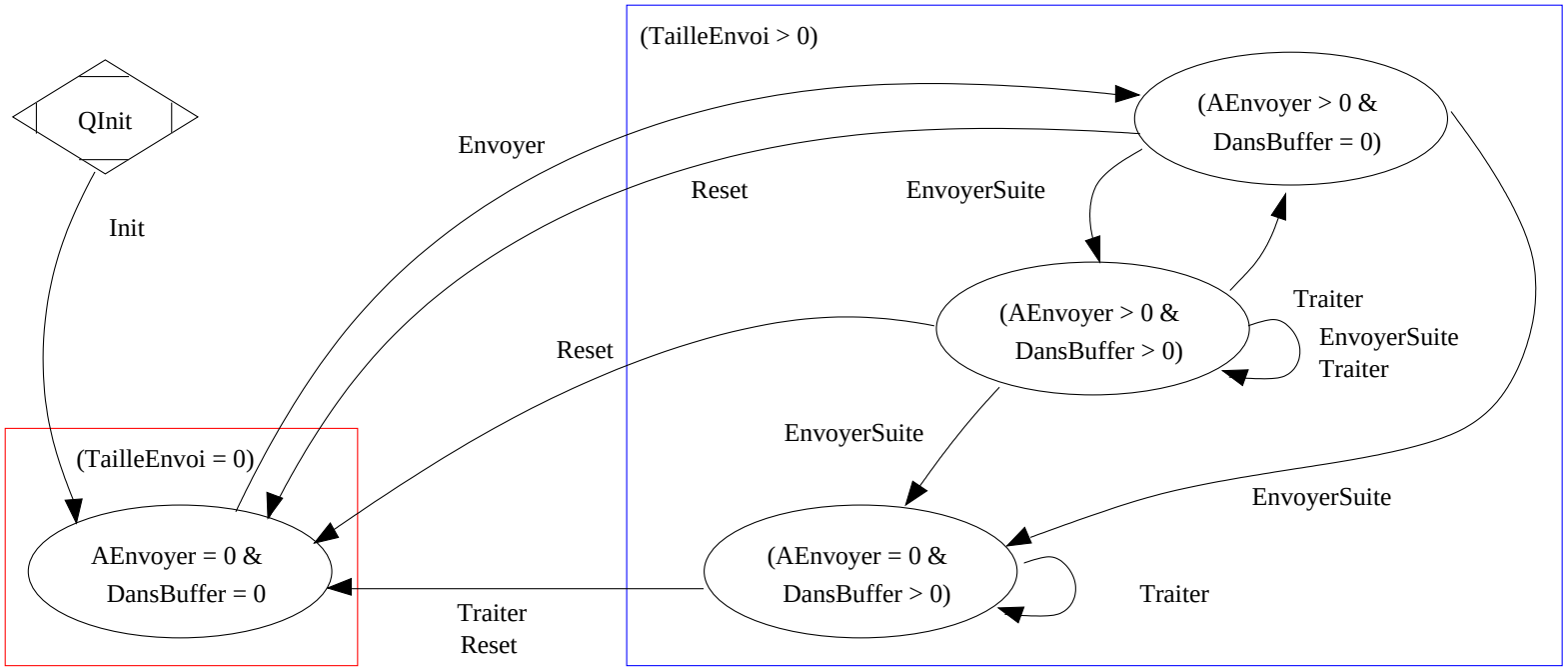
- Mise en évidence des **comportements** d'un modèle



- 2 aspects mis en avant :
  - Complémentarité des vues *(données vs comportements)*
  - Visualisation des étapes de conception

Contributions de cette thèse

- Mise en évidence des étapes de conception



- Objectifs :
  - Validation de la spécification
  - Compréhension
  - Documentation
  - Application à la vérification de propriétés

Langage B événementiel

Forme générale d'un composant de spécification abstraite

SYSTEM *M* /* *Nom du composant* */
VARIABLES *v* /* *Spécification des variables* */
INVARIANT *I*
INITIALISATION *Init* /* *Initialisation des variables* */
EVENTS /* *Liste d'événements* */
Ev $\widehat{=}$ **SELECT** *G* **THEN** *S* **END**
END

Spécification abstraite

Sémantique des substitutions généralisées ($\mathcal{WP}$)

$\mathcal{WP}$ (Weakest Precondition – $P \mathrel{\hat{=}} [S]R$) :
Calcul de la plus faible pré-condition P qui, si elle est vérifiée, garantit que l'exécution de S termine et établit R.

Exemples :

$[x := exp]R \quad \mathrel{\hat{=}}$ Remplacement par *exp* des occurrences libres de x dans R.

$[x :\in E]R \quad \mathrel{\hat{=}} \quad \forall x' \cdot (x' \in E \Rightarrow [x := x']R)$

Conditions de correction d'un système

Initialisation	$[Init]I$
Pour chaque événement	$I \wedge G \Rightarrow [S]I$

Raffinement B

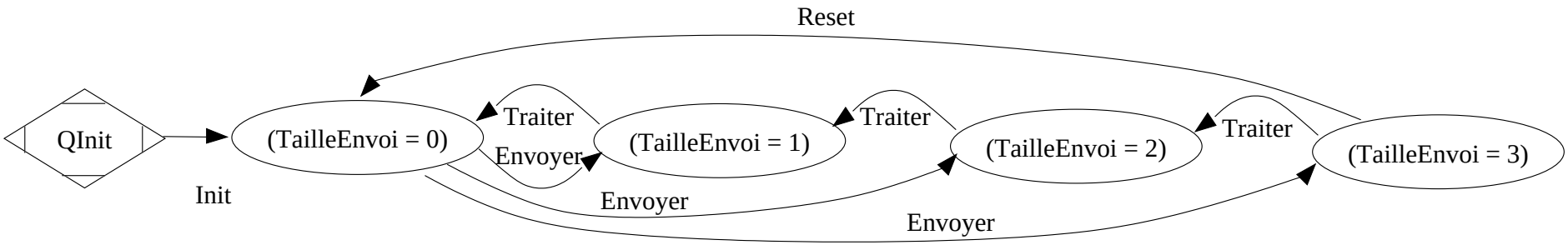
- Raffinement B événementiel :
  - Changement de représentation des données
    Invariant de liaison *L* ***:*** *prédicat mettant en relation les données de l'abstraction et celles du raffinement.*
  - Restriction des gardes
  - Diminution du non-déterminisme
  - Introduction de nouveaux événements
- Raffinement effectué par parties
  - $Init_A \sqsubseteq_L Init_R$
  - $Ev_A \sqsubseteq_L Ev_R$ *(Pour chaque événement Ev)*
  - **skip** $\sqsubseteq_L Ev_N$ *(Pour chaque nouvel événement* Ev_N*)*
- Traces(Raf) $\subseteq$ Traces(Abst)
  (Modulo les nouveaux événements)

Représentation des comportements d'un modèle B événementiel

- État de l'art : 2 aspects orthogonaux
  - *Formalisme de description des comportements*
    Cas particulier des systèmes de transitions
  - *Méthode de construction des **STES***

Systèmes de transitions étiquetées

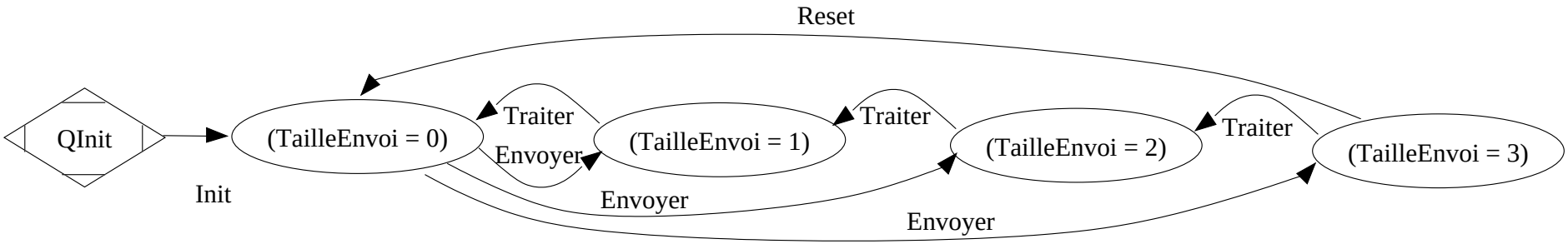
- Concrets *(ProB [Leuschel et al.])*
  - *1 état par valuation du modèle*
  - *Nombre fini de valuations représentées*



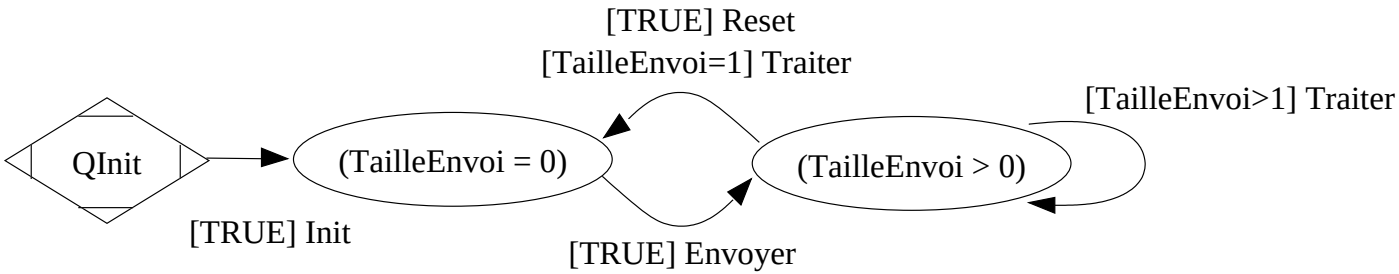
- Symboliques *(ASM [Gurevich])*
  - *États pouvant contenir une infinité de valuations*
  - *Systèmes paramétrés*
  - *Conditions de franchissement*

Systèmes de transitions étiquetées

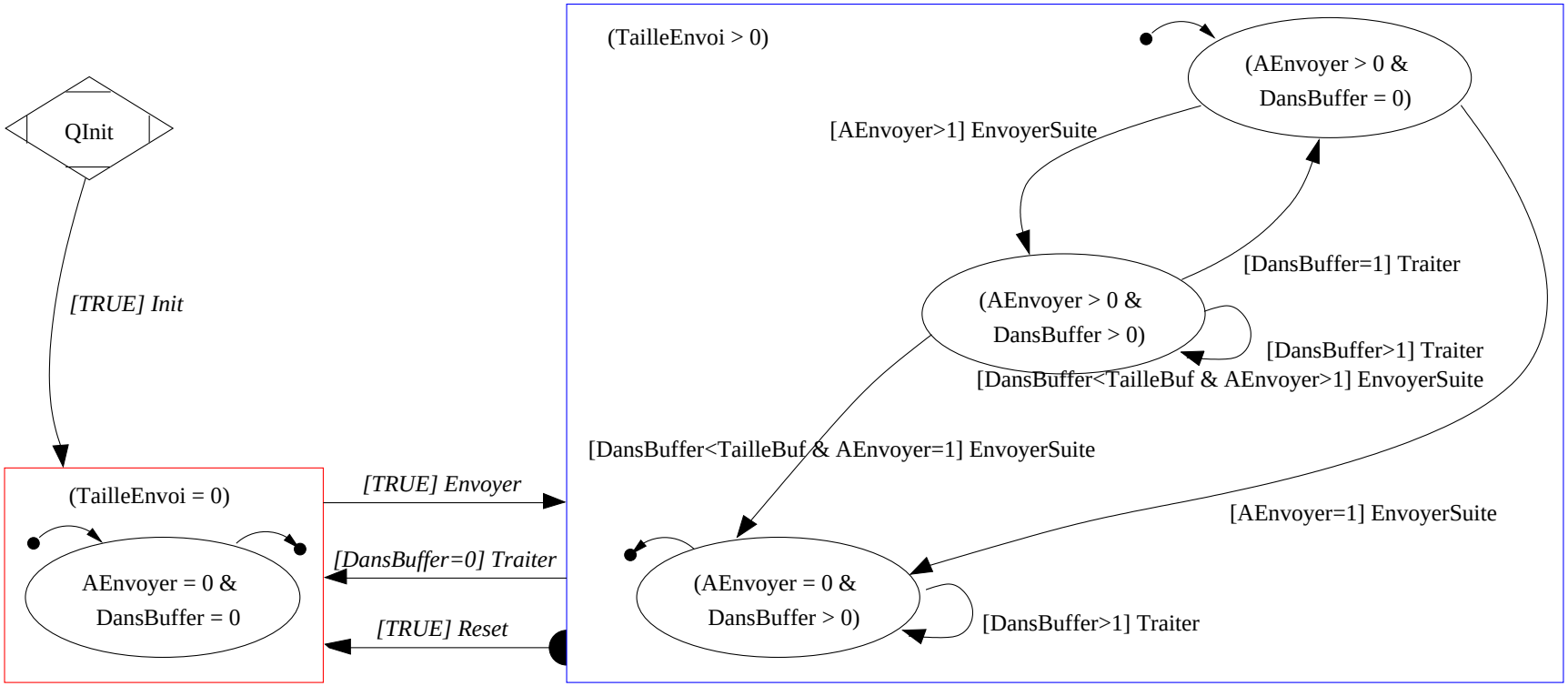
- Concrets *(ProB [Leuschel et al.])*
  - *1 état par valuation du modèle*
  - *Nombre fini de valuations représentées*



- Symboliques *(ASM [Gurevich])*
  - *États pouvant contenir une infinité de valuations*
  - *Systèmes paramétrés*
  - *Conditions de franchissement*



Systèmes de transitions étiquetées



- Hiérarchiques *(StateCharts [Harel], états-transitions [OMG])*
  - *Symbolique*
  - *Structuration des diagrammes*
    Transitions factorisées, vers sous-état initial, depuis sous-état final

Construction des comportements d'un modèle B

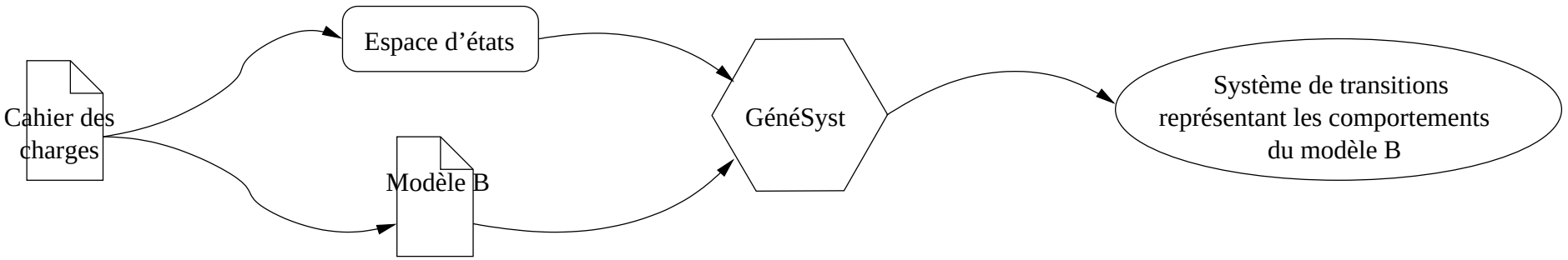
- Construction syntaxique par règles de réécriture *[Voisinnet et al.]*
  Coût faible, création de transitions infranchissables, sous-ensemble de B
- Animation par énumération *(ProB [Leuschel et al.])*
  Nombre de valuations fini, traces de longueur finie
- Animation par résolution de contraintes *(BZ-TT [Ambert et al.])*
  Nombre de valuations possiblement infini, traces de longueur finie
- Calcul par la preuve *[Graf et al., Bert et al.]*
  Nombre de valuations possiblement infini, traces de longueur infinie, choix des états par l'utilisateur

Plan de l'exposé

Présentation de la méthode

- **Modèle B** et **espace d'états** sont fournis par l'utilisateur
  - État : *prédicat portant sur les variables du système*
  - Espace d'états : *Invariant* $\Leftrightarrow (\bigvee_i Etat_i)$
- Calcul de l'ensemble des transitions entre les états

  Par la preuve (Outil GénéSyst)
- Le **STES** produit représente les comportements du modèle B

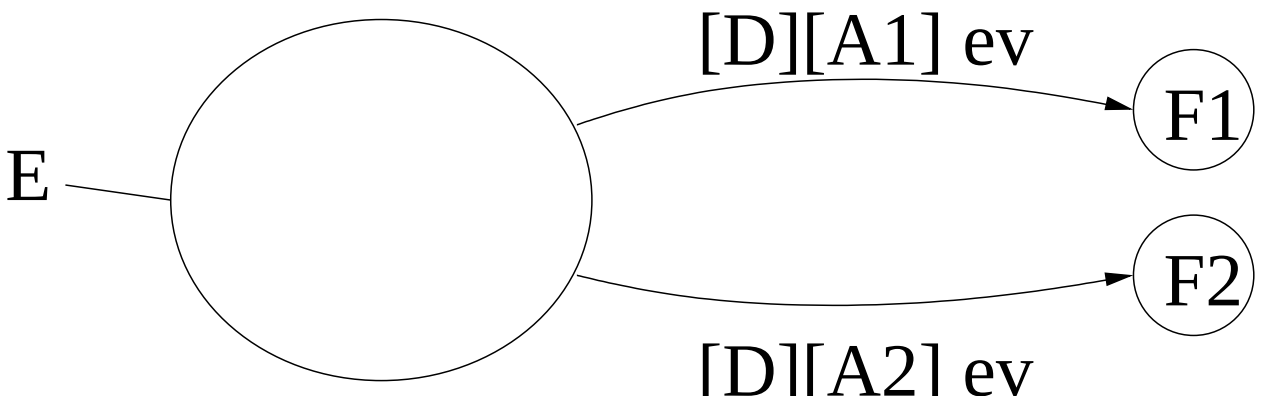


Principe de construction par la preuve

- Principe de construction par la preuve d'une transition :
  - Construction syntaxique de sa condition de franchissement
  - Condition toujours fausse ? *(Transition non franchissable)*
  - Condition toujours vraie ? *(Toujours franchissable)*
- Limitation : échec possible de la preuve automatique
  (Défauts de preuve : transitions infranchissables représentées)
- Proposition : scinder la condition en deux moitiés
- Avantages :
  - Simplification des formules à vérifier
    (Diminution du risque de défaut de preuve)
  - Aide à la lecture
    (Mise en évidence de l'origine de la condition)

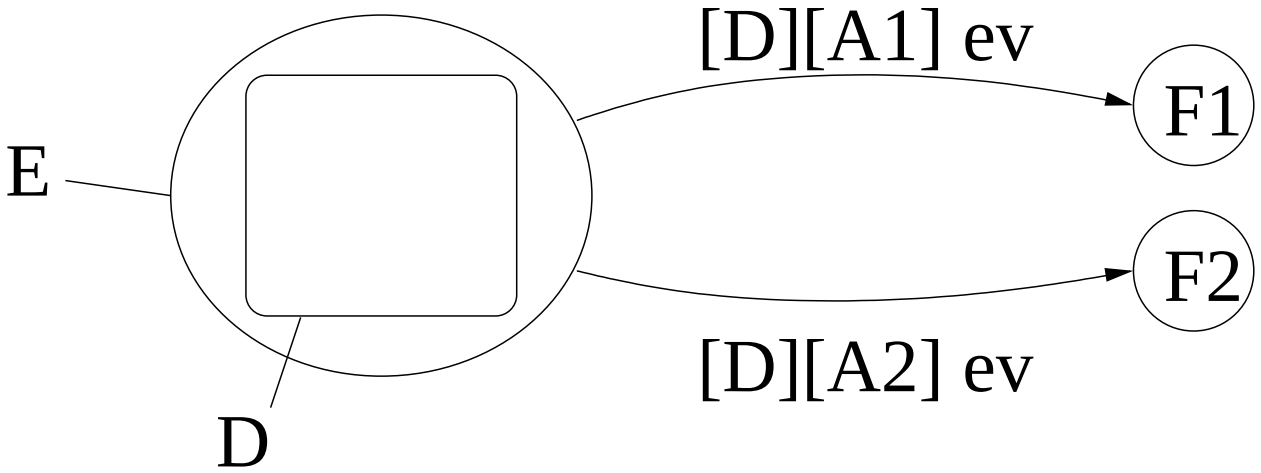
Transitions avec 2 conditions de franchissement

- Décomposition des gardes en 2 conditions *D* et *A* :
  - *ev* : **nom d'un événement** du système B ;
  - *D* : **condition de déclenchabilité**
    (Valuations de E permettant de déclencher ev)
  - *A* : **condition d'atteignabilité**
    (Valuations de E et D permettant à ev d'atteindre F.)



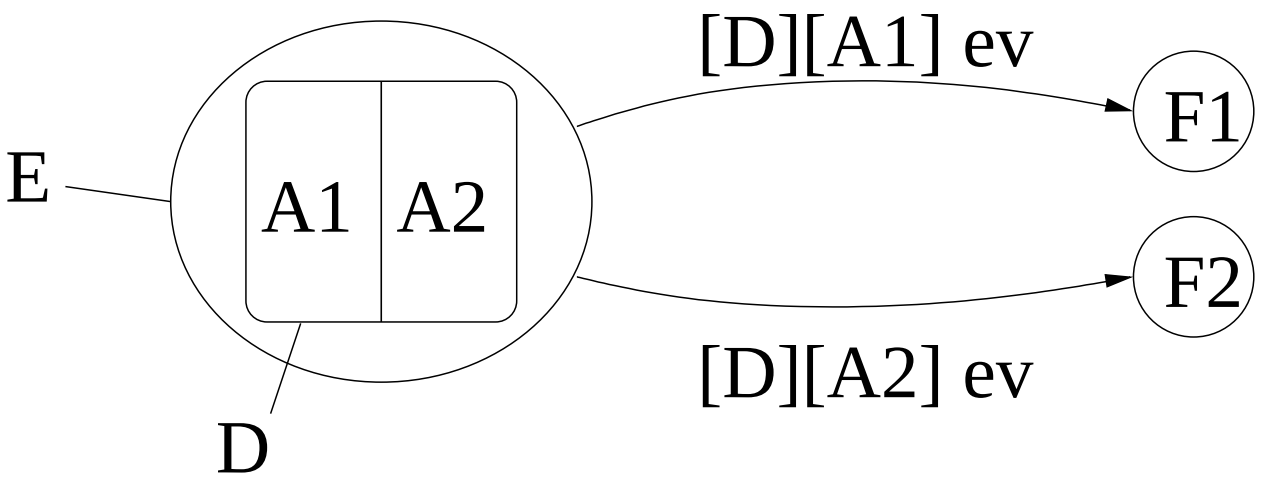
Transitions avec 2 conditions de franchissement

- Décomposition des gardes en 2 conditions *D* et *A* :
  - *ev* : **nom d'un événement** du système B ;
  - *D* : **condition de déclenchabilité**
    (Valuations de E permettant de déclencher ev)
  - *A* : **condition d'atteignabilité**
    (Valuations de E et D permettant à ev d'atteindre F.)



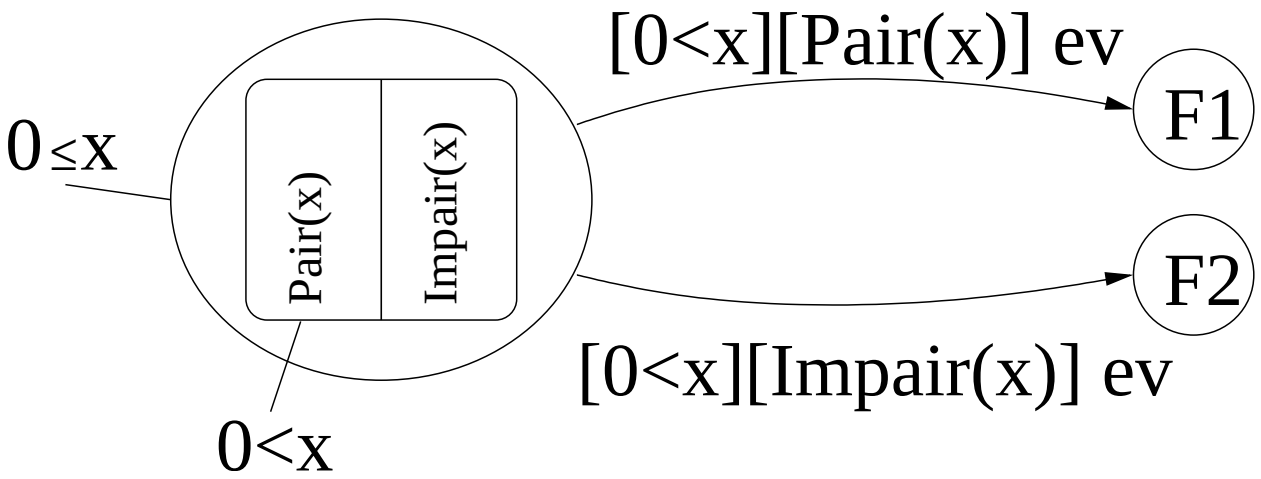
Transitions avec 2 conditions de franchissement

- Décomposition des gardes en 2 conditions *D* et *A* :
  - *ev* : **nom d'un événement** du système B ;
  - *D* : **condition de déclenchabilité**
    (Valuations de E permettant de déclencher ev)
  - *A* : **condition d'atteignabilité**
    (Valuations de E et D permettant à ev d'atteindre F.)

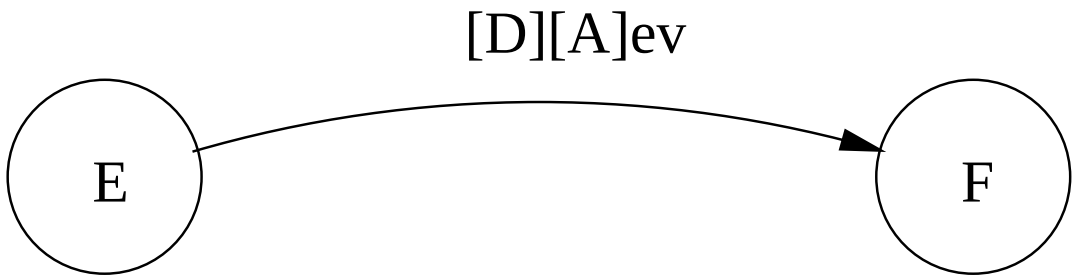


Transitions avec 2 conditions de franchissement

- Décomposition des gardes en 2 conditions *D* et *A* :
  - *ev* : **nom d'un événement** du système B ;
  - *D* : **condition de déclenchabilité**
    (Valuations de E permettant de déclencher ev)
  - *A* : **condition d'atteignabilité**
    (Valuations de E et D permettant à ev d'atteindre F.)



Construction d'une transition



- Construction syntaxique des conditions :

$D \mathrel{\hat{=}}$ Garde(ev) $\qquad A \mathrel{\hat{=}} \neg[\text{Action}(ev)]\neg\mathcal{D}ef(F)$

- Heuristique à 3 cas :
  - Condition toujours fausse ? *(Non franchissable)*

$\forall x : (\mathcal{D}ef(E) \Rightarrow \neg D)$ $\qquad \forall x : (\mathcal{D}ef(E) \wedge D \Rightarrow \neg A)$

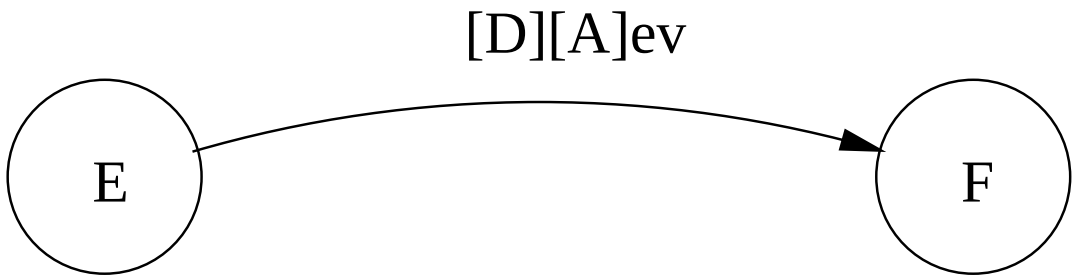
  - Condition toujours vraie ? *(Franchissable sans condition)*

$\forall x : (\mathcal{D}ef(E) \Rightarrow D)$ $\qquad \forall x : (\mathcal{D}ef(E) \wedge D \Rightarrow A)$

  - Sinon le franchissement est conditionné

Notation : $\mathcal{D}ef(E) \mathrel{\hat{=}}$ *prédicat de définition de l'état (E)*

Construction d'une transition



- Construction syntaxique des conditions :

$D \mathrel{\hat{=}} \text{Garde}(ev)$ $\qquad$ $A \mathrel{\hat{=}} \neg[\text{Action}(ev)]\neg\mathcal{D}ef(F)$

- Heuristique à 3 cas :
  - Condition toujours fausse ? *(Non franchissable)*

$\forall x \cdot (\mathcal{D}ef(E) \Rightarrow \neg D)$ $\qquad$ $\forall x \cdot (\mathcal{D}ef(E) \wedge D \Rightarrow \neg A)$

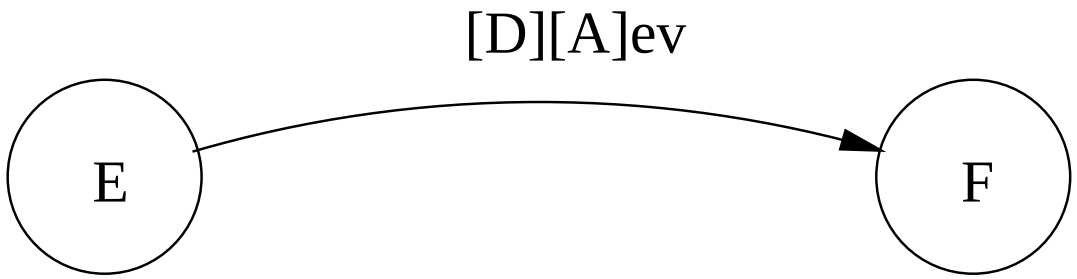
  - Condition toujours vraie ? *(Franchissable sans condition)*

$\forall x \cdot (\mathcal{D}ef(E) \Rightarrow D)$ $\qquad$ $\forall x \cdot (\mathcal{D}ef(E) \wedge D \Rightarrow A)$

  - Sinon le franchissement est conditionné

Notation : $\mathcal{D}ef(E) \mathrel{\hat{=}}$ *prédicat de définition de l'état (E)*

Construction d'une transition



- Construction syntaxique des conditions :

$D \mathrel{\hat{=}} \text{Garde}(ev)$ $\qquad A \mathrel{\hat{=}} \neg[\text{Action}(ev)]\neg\mathcal{D}ef(F)$

- Heuristique à 3 cas :
  - Condition toujours fausse ? *(Non franchissable)*

$\forall x \cdot (\mathcal{D}ef(E) \Rightarrow \neg D)$ $\qquad \forall x \cdot (\mathcal{D}ef(E) \wedge D \Rightarrow \neg A)$

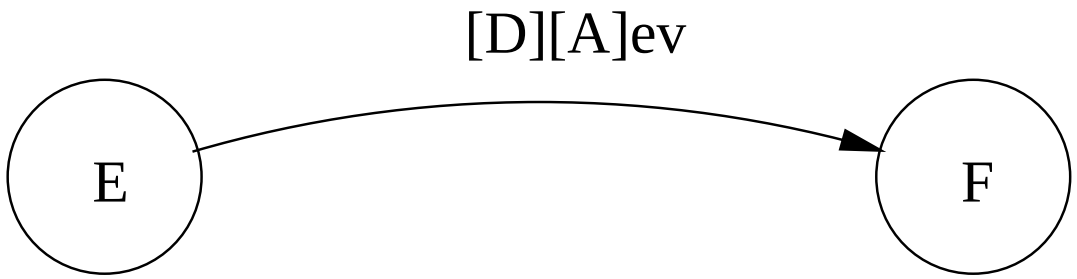
  - Condition toujours vraie ? *(Franchissable sans condition)*

$\forall x \cdot (\mathcal{D}ef(E) \Rightarrow \quad D)$ $\qquad \forall x \cdot (\mathcal{D}ef(E) \wedge D \Rightarrow \quad A)$

  - Sinon le franchissement est conditionné

Notation : $\mathcal{D}ef(E) \mathrel{\hat{=}}$ *prédicat de définition de l'état (E)*

Construction d'une transition



- Construction syntaxique des conditions :

$D \mathrel{\hat{=}} \text{Garde}(ev)$ $\qquad A \mathrel{\hat{=}} \neg[\text{Action}(ev)]\neg\mathcal{D}ef(F)$

- Heuristique à 3 cas :
  - Condition toujours fausse ? *(Non franchissable)*

$\forall x \cdot (\mathcal{D}ef(E) \Rightarrow \neg D)$ $\qquad \forall x \cdot (\mathcal{D}ef(E) \wedge D \Rightarrow \neg A)$

  - Condition toujours vraie ? *(Franchissable sans condition)*

$\forall x \cdot (\mathcal{D}ef(E) \Rightarrow \quad D)$ $\qquad \forall x \cdot (\mathcal{D}ef(E) \wedge D \Rightarrow \quad A)$

  - Sinon le franchissement est conditionné

Notation : $\mathcal{D}ef(E) \mathrel{\hat{=}}$ *prédicat de définition de l'état (E)*

Relation de transition

- Construite par induction sur les états atteignables depuis l'initialisation

Theorem (Égalité des traces)

Si S est un système B événementiel pour lequel l'invariant a été établi et si T est un ***STES*** *généré à partir de S, alors :*

$$\text{Traces}(S) = \text{Chemins}(T)$$

Relation de transition

- Construite par induction sur les états atteignables depuis l'initialisation

Theorem (Égalité des traces)

Si S est un système B événementiel pour lequel l'invariant a été établi et si T est un ***STES*** *généré à partir de S, alors :*

$$\text{Traces}(S) = \text{Chemins}(T)$$

Modélisation B événementiel réalisée

SYSTEM *Canal_De_Communication*
VARIABLES *TailleEnvoi*
INVARIANT $TailleEnvoi \in \mathbb{N}$
INITIALISATION $TailleEnvoi := 0$
EVENTS
Envoyer $\hat{=}$ **SELECT** $TailleEnvoi = 0$
THEN $TailleEnvoi :\in \mathbb{N}_1$ **END ;**

Traiter $\hat{=}$ **SELECT** $TailleEnvoi > 0$
THEN $TailleEnvoi := TailleEnvoi - 1$ **END ;**

Reset $\hat{=}$ **SELECT** $TailleEnvoi > 0$
THEN $TailleEnvoi := 0$ **END**
END

Construction d'un STES associé au modèle

- L'utilisateur décrit les états dans la clause **ASSERTIONS**

  Complétude de l'espace d'états garantie par preuve de la correction du modèle

ASSERTIONS $TailleEnvoi = 0 \vee TailleEnvoi > 0$

- **STES** construit par *GénéSyst*

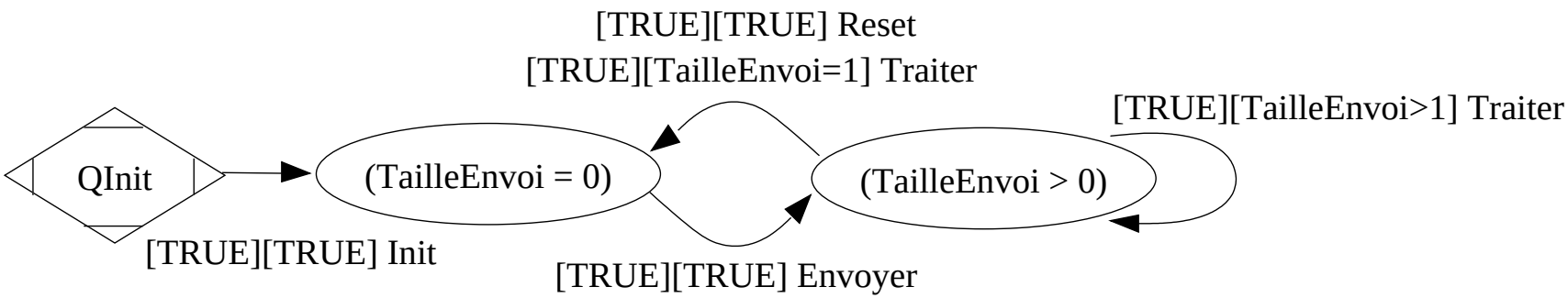
Construction d'un STES associé au modèle

- L'utilisateur décrit les états dans la clause **ASSERTIONS**

  Complétude de l'espace d'états garantie par preuve de la correction du modèle

ASSERTIONS $TailleEnvoi = 0 \vee TailleEnvoi > 0$

- **STES** construit par *GénéSyst*



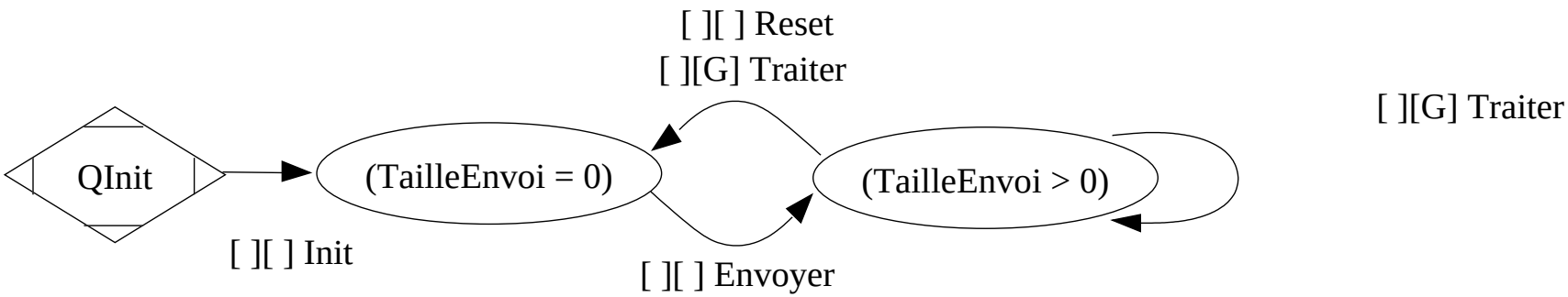
Construction d'un STES associé au modèle

- L'utilisateur décrit les états dans la clause **ASSERTIONS**

  Complétude de l'espace d'états garantie par preuve de la correction du modèle

ASSERTIONS $TailleEnvoi = 0 \vee TailleEnvoi > 0$

- **STES** construit par *GénéSyst*



Notation graphique : [ ] *si condition vraie et* [G] *sinon.*

Prise en compte du processus de raffinement

- Raffinement : vue plus précise de la spécification abstraite
  (Même méthode de construction applicable)
- Objectifs :
  - Exhiber les liens entre les données abstraites et raffinées
  - Préservation de la structure du **STES** abstrait
  - Zoomer sur les comportements internes d'un états
  - Simplifier la construction grâce aux propriétés du raffinement
- Principes :
  - Modification d'un **STES** abstrait
  - Ajout d'un niveau de hiérarchie
  - Construction des comportements

Définition des STES Hiérarchiques

- Propriétés voulues :
  - Réutilisation des méthodes de construction
  - Favoriser la lisibilité des représentations
- Solutions proposées :
  - Système de Transitions Étiquetées Symbolique
    (Contenant les états de tous les niveaux)
  - Fonction de hiérarchie des états
    (Associant un état à son état père)
  - Identification des sous-états initiaux et finaux
    (Proposition d'heuristiques de choix)
  - Factorisation des transitions
    (Proposition d'une définition favorisant la lisibilité)
- Sémantique d'un **STEH** : le **STES** en lequel il s'aplatit

Modèle B événementiel raffiné

REFINEMENT *Canal_De_Communication_R*
REFINES *Canal_De_Communication*

CONSTANTS *TailleBuff*
PROPERTIES $TailleBuff \in \mathbb{N}_1$

VARIABLES *DansBuffer* , *AEnvoyer*
INVARIANT $AEnvoyer \in \mathbb{N} \land DansBuffer \in 0..TailleBuff$
$\land\ DansBuffer + AEnvoyer = TailleEnvoi$

INITIALISATION $DansBuffer := 0 \ || \ AEnvoyer := 0$
EVENTS
$Envoyer \mathrel{\hat{=}}$ **SELECT** $AEnvoyer = 0 \land DansBuffer = 0$ **THEN** $AEnvoyer :\in \mathbb{N}_1$ **END** ;

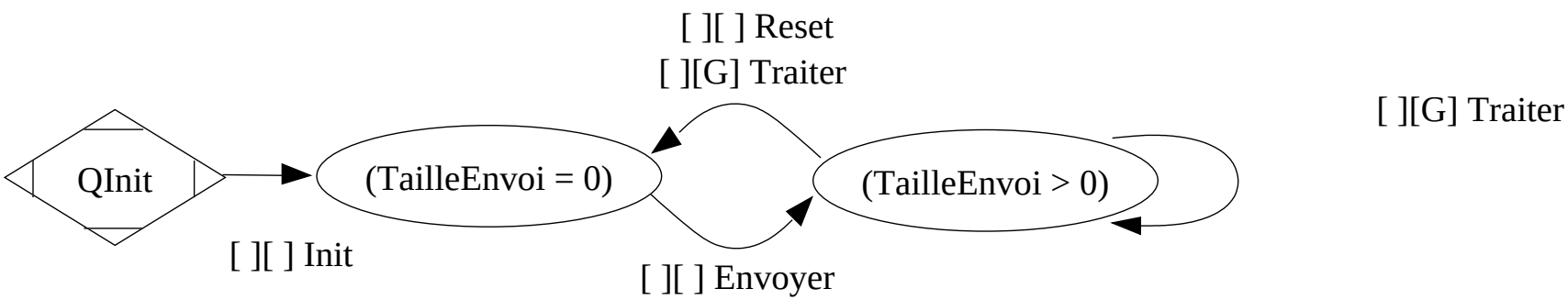
$EnvoyerSuite \mathrel{\hat{=}}$ **SELECT** $AEnvoyer > 0 \land DansBuffer < TailleBuff$
THEN $AEnvoyer := AEnvoyer - 1 \ || \ DansBuffer := DansBuffer + 1$ **END** ;

$Traiter \mathrel{\hat{=}}$ **SELECT** $DansBuffer > 0$ **THEN** $DansBuffer := DansBuffer - 1$ **END** ;

$Reset \mathrel{\hat{=}}$ **SELECT** $AEnvoyer > 0 \lor DansBuffer > 0$
THEN $AEnvoyer := 0 \ || \ DansBuffer := 0$ **END**
END

Construction du modèle raffiné

- Construire le **STES** associé au système abstrait



Construction du modèle raffiné

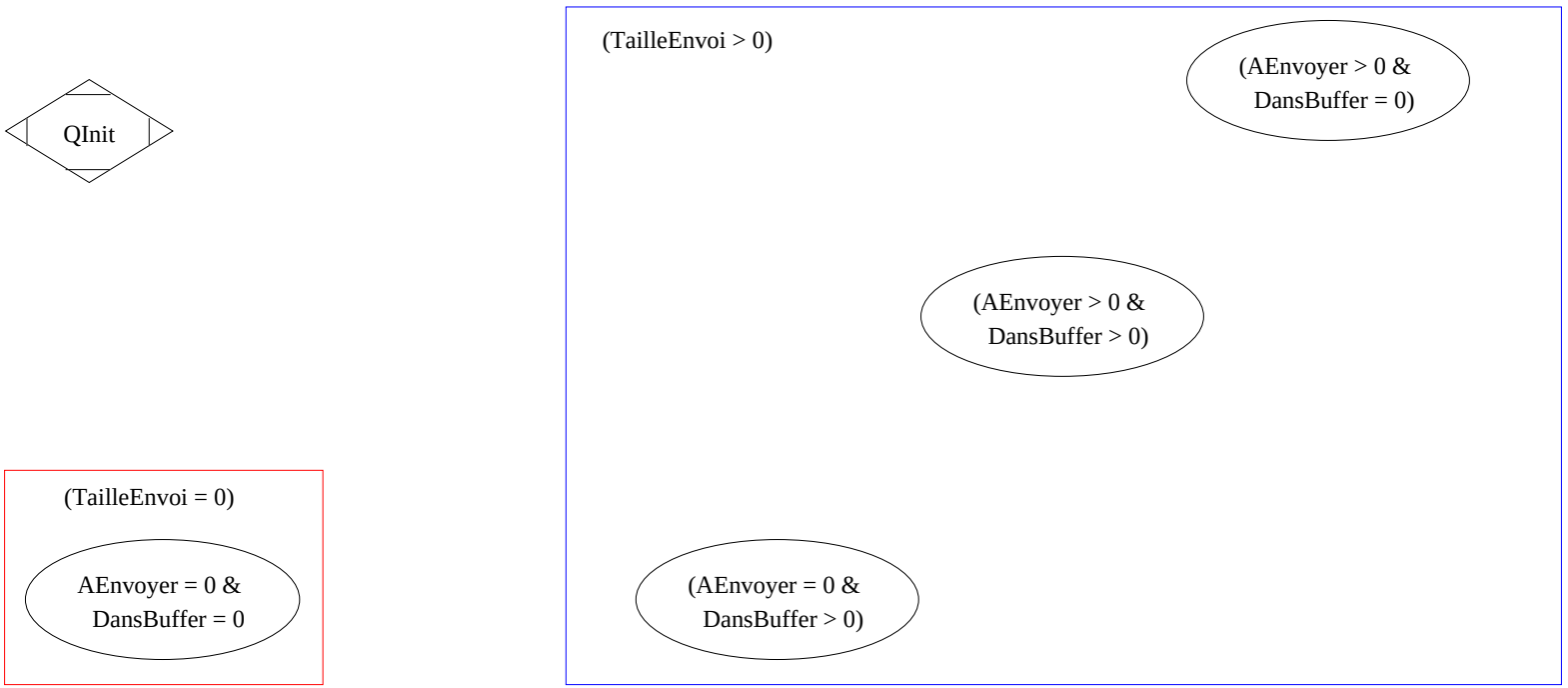
- Décomposition des états donnée par l'utilisateur

ASSERTIONS

$$((\mathit{TailleEnvoi} = 0) \Leftrightarrow (\mathit{DansBuffer} = 0 \ \wedge \ \mathit{AEnvoyer} = 0)) \wedge \left((\mathit{TailleEnvoi} > 0) \Leftrightarrow \left(\begin{array}{l} (\mathit{DansBuffer} = 0 \ \wedge \ \mathit{AEnvoyer} > 0) \vee \\ (\mathit{DansBuffer} > 0 \ \wedge \ \mathit{AEnvoyer} > 0) \vee \\ (\mathit{DansBuffer} > 0 \ \wedge \ \mathit{AEnvoyer} = 0) \end{array} \right) \right)$$

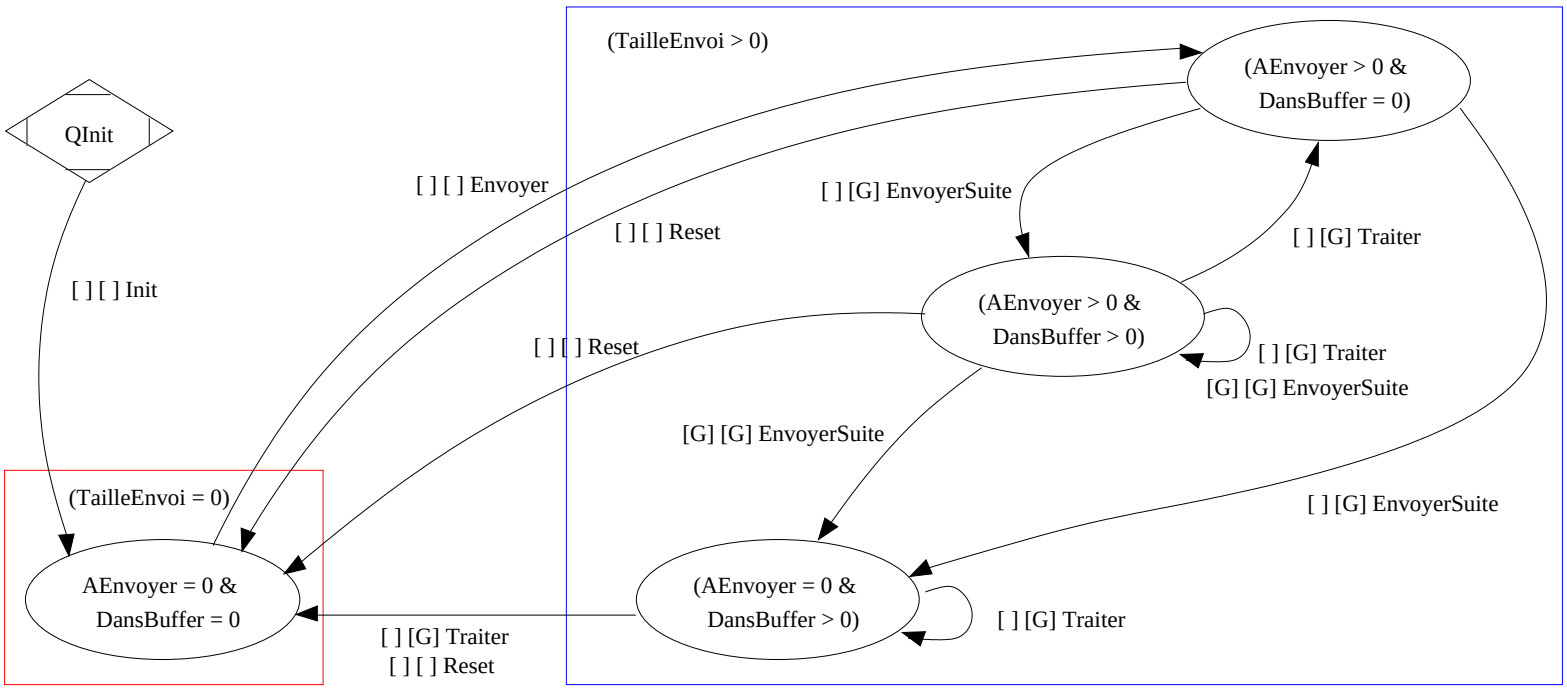
Construction du modèle raffiné

- Décomposition des états donnée par l'utilisateur



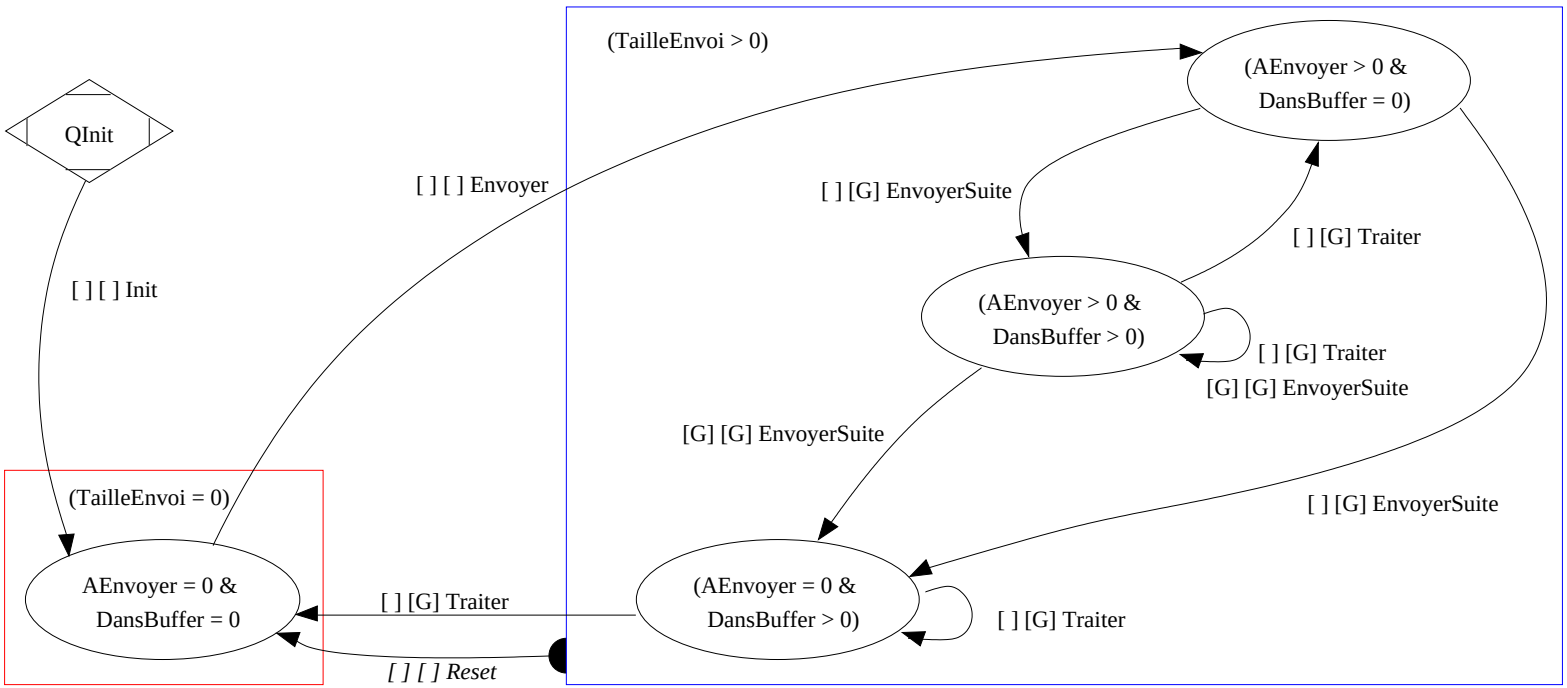
Construction du modèle raffiné

- Calcul de la relation de transition entre états feuille



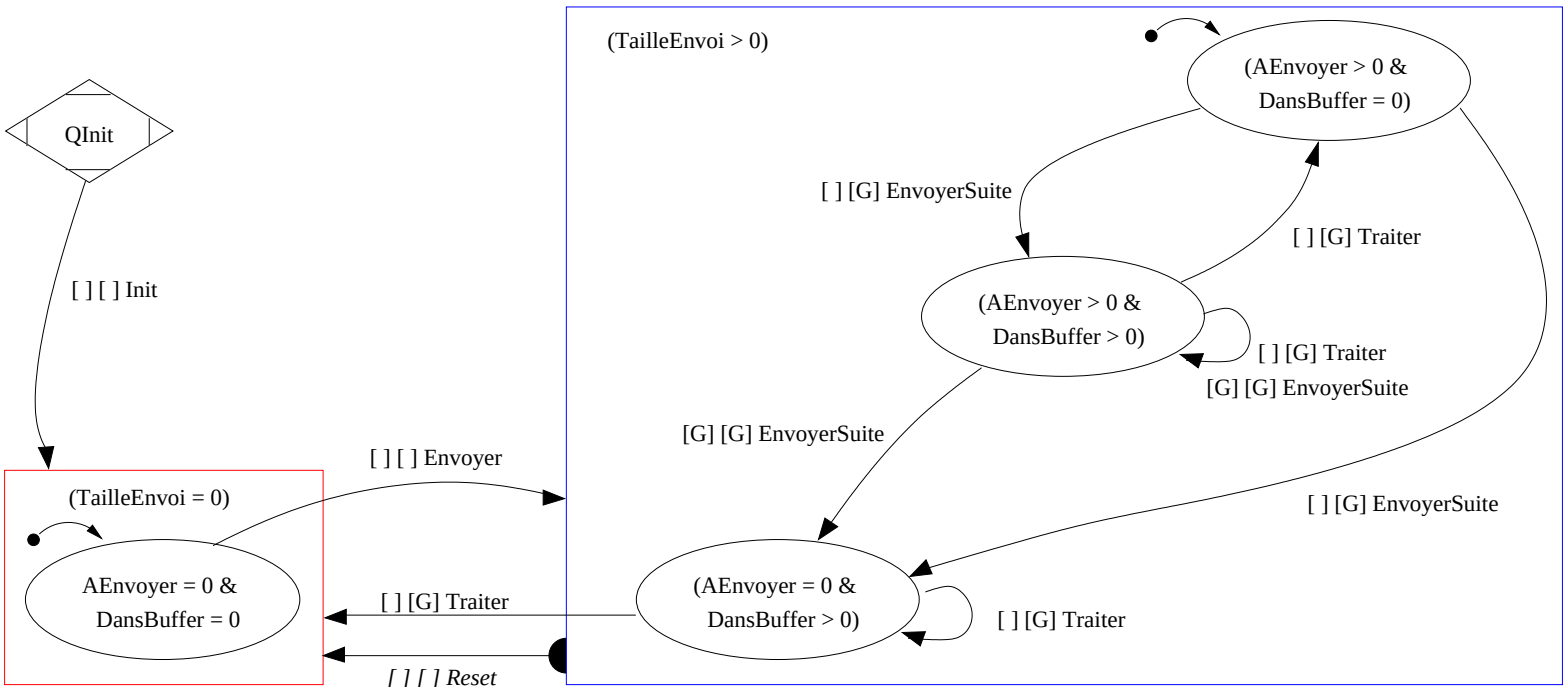
Construction du modèle raffiné

- Réduire le nombre de transitions externes
  - Factorisation des transitions
  - Choix des sous-états initiaux
  - Choix des sous-états finaux



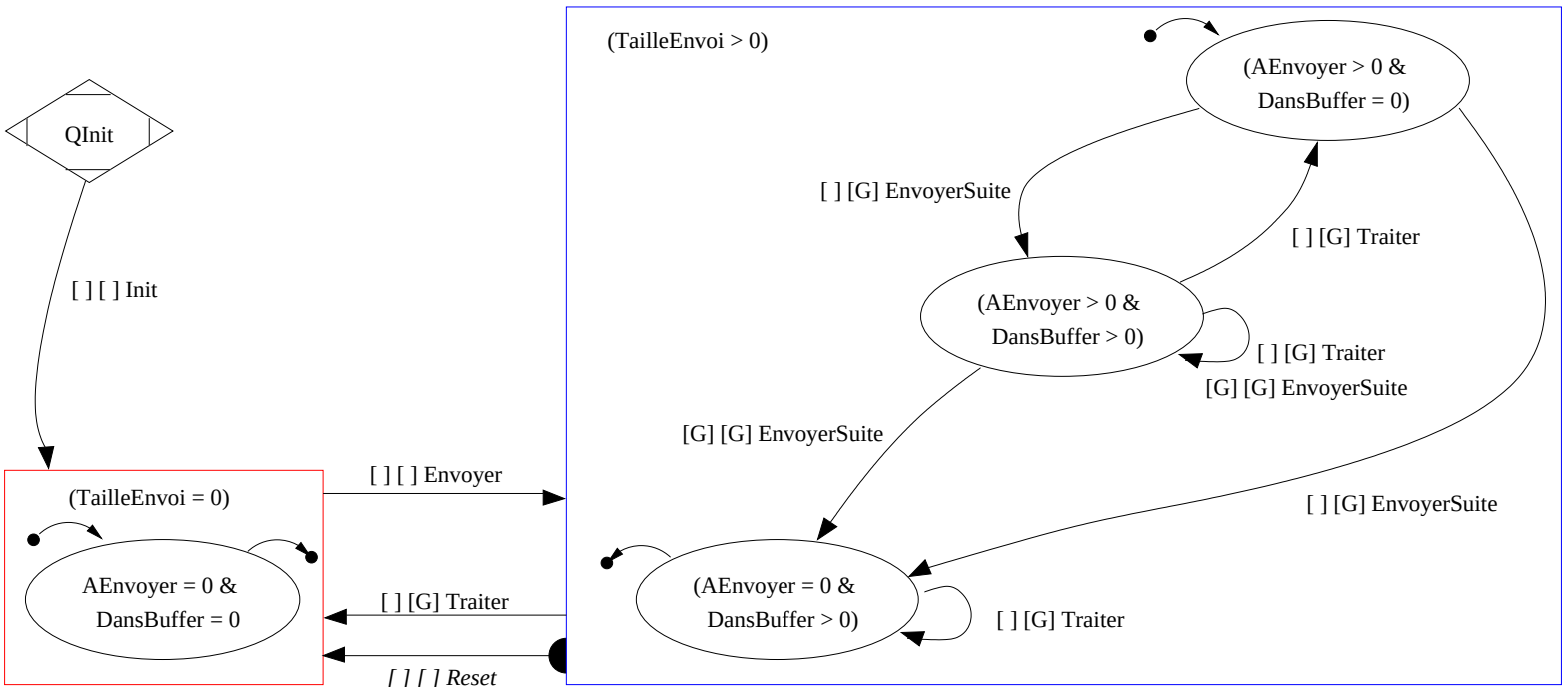
Construction du modèle raffiné

- Réduire le nombre de transitions externes
  - Factorisation des transitions
  - Choix des sous-états initiaux
  - Choix des sous-états finaux



Construction du modèle raffiné

- Réduire le nombre de transitions externes
  - Factorisation des transitions
  - Choix des sous-états initiaux
  - Choix des sous-états finaux



Plan de l'exposé

Heuristiques de choix d'un espace d'états

- Énumération *(Intéressant pour les variables de contrôle)*

$$\{x = 0 \quad , \quad x = 1 \quad , \quad x = 2 \quad , \quad \dots \quad , \quad x = n\}$$

- Exhibition d'un témoin *(Exhibition du cycle de vie)*

$$\{T \in e_1 \quad , \quad T \in e_2 \quad , \quad \dots \quad , \quad T \in e_n\}$$

Avec T une nouvelle constante non valuée (Exhibition du cycle de vie de T).

- Utilisation des gardes *(Focus sur les modifications de données)*

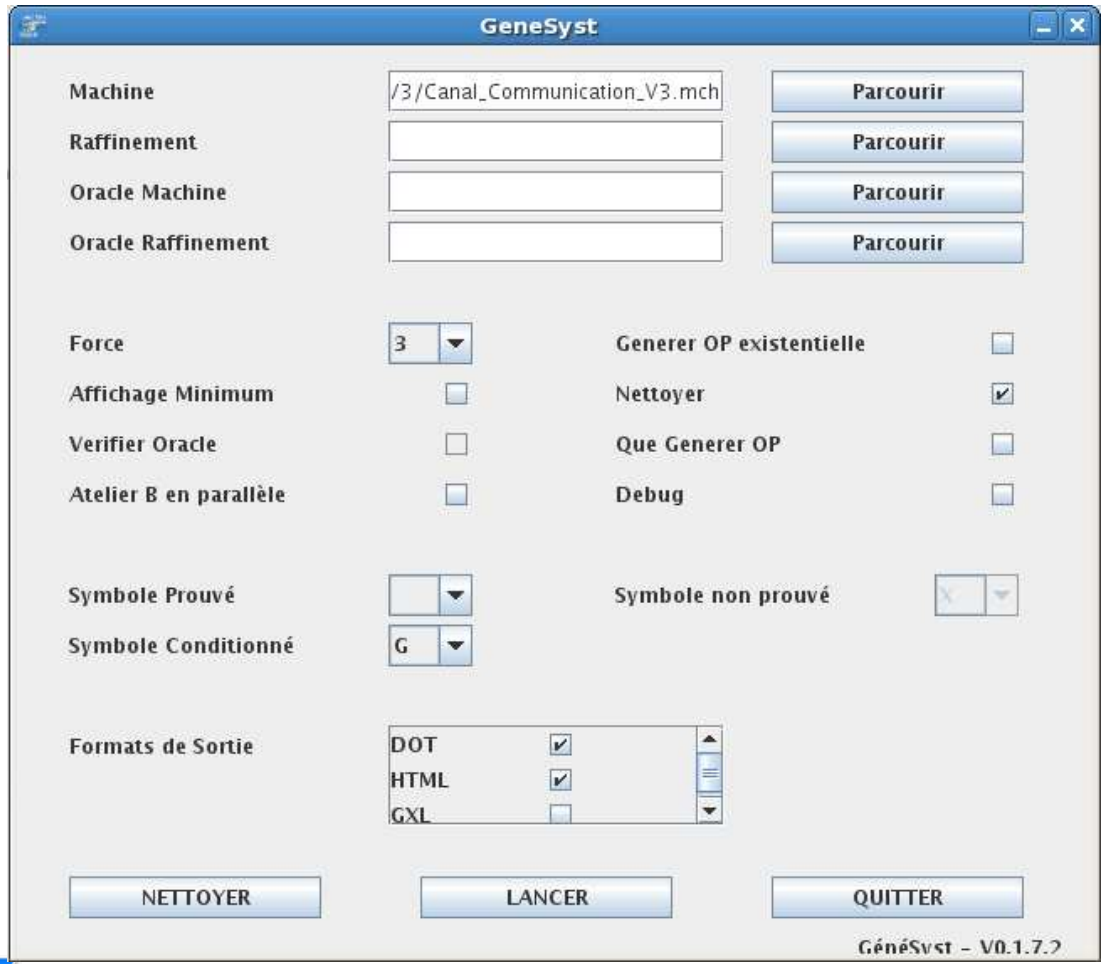
$$\{\text{Garde}(ev_1) \quad , \quad \text{Garde}(ev_2) \quad , \quad \dots \quad , \quad \text{Garde}(ev_n)\}$$

- États aux limites d'une variable *(Comportements aux limites)*

$$\{x = 0 \quad , \quad x \in 1..n-1 \quad , \quad x = n\}$$

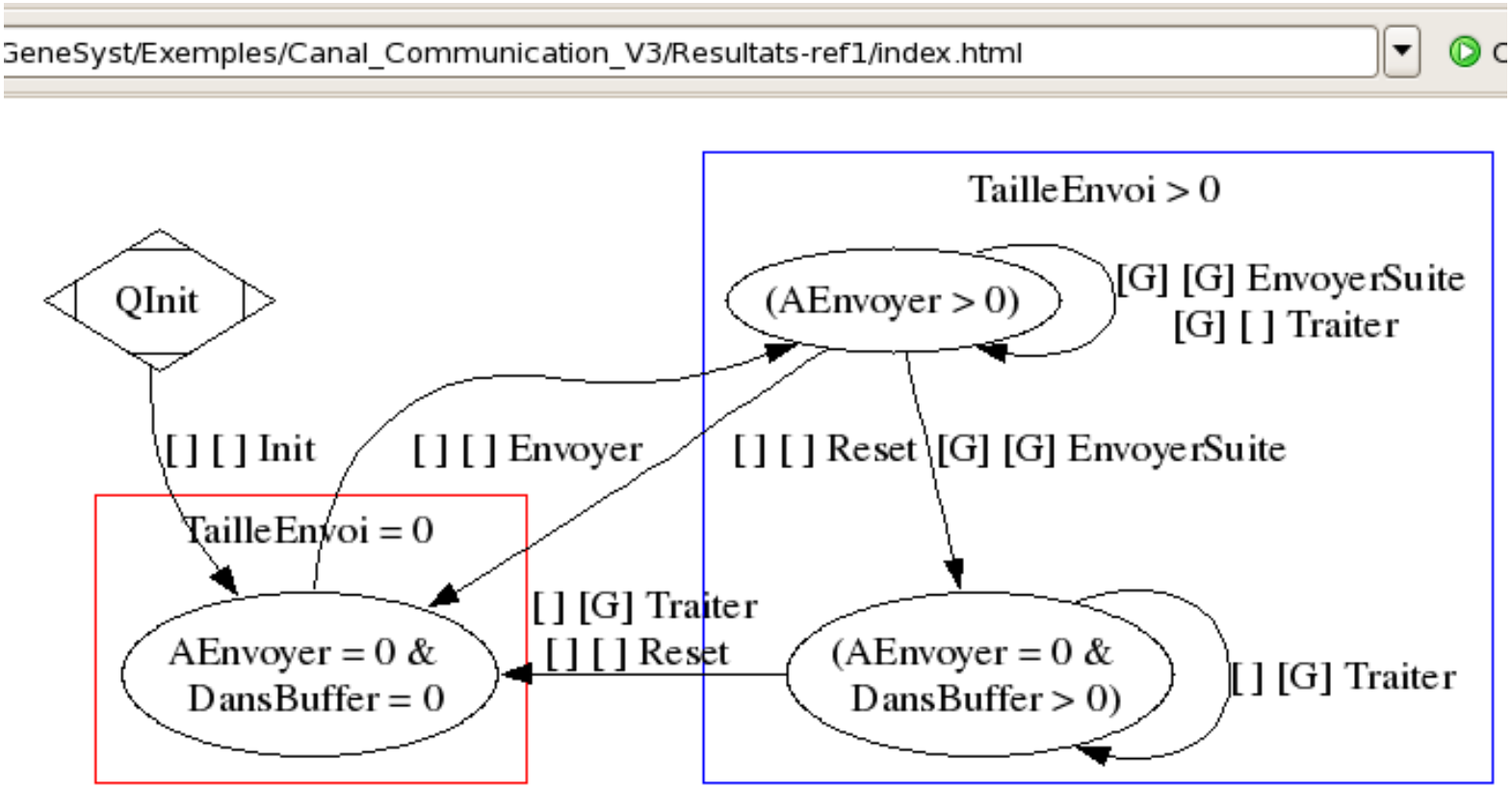
GénéSyst : construction de STEH

- L'utilisateur fournit le modèle B *(+ espace d'états en assertion)*
- Format des **STEH** produits : DOT, GXL ou HTML .



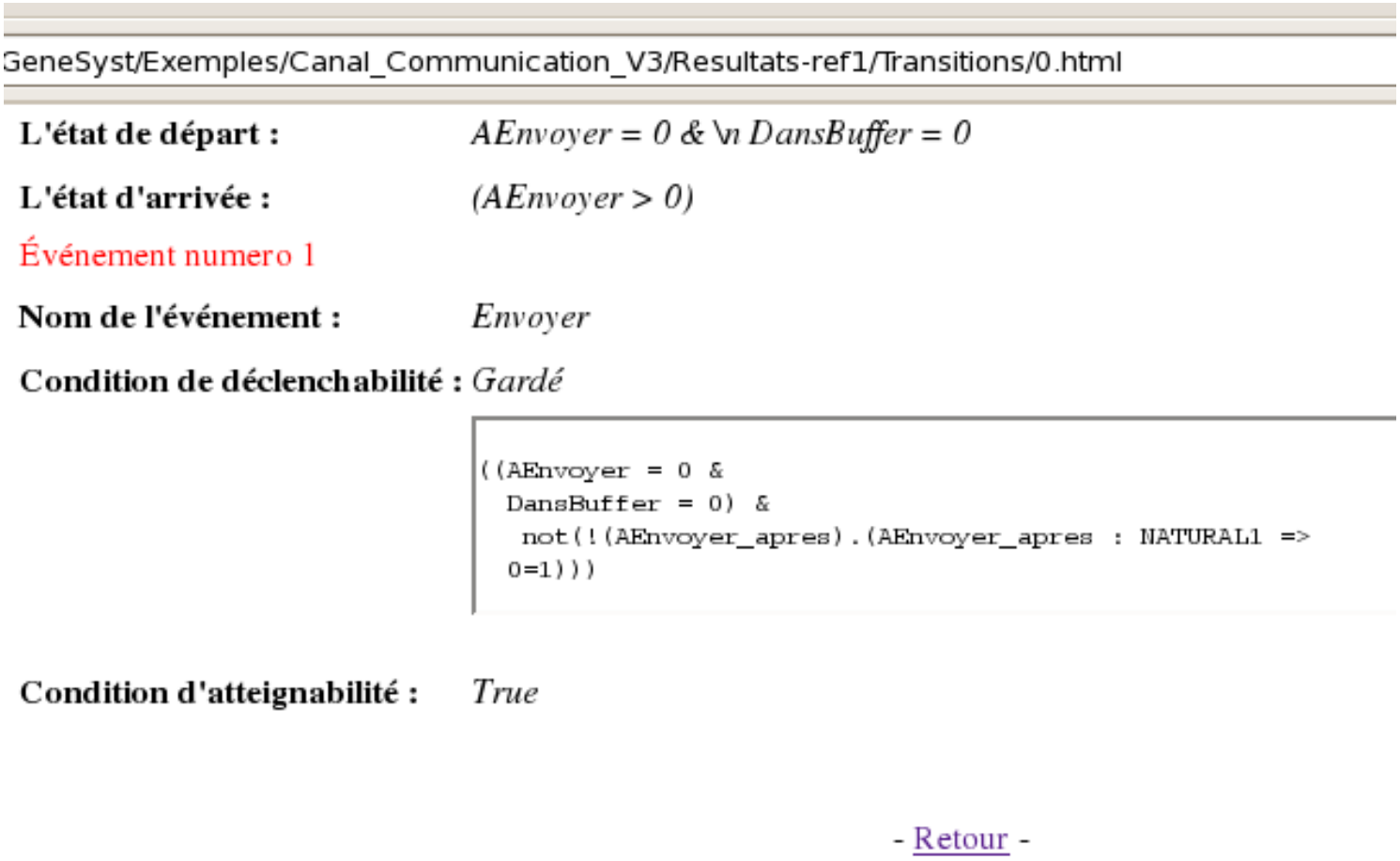
GénéSyst : construction de STEH

- L'utilisateur fournit le modèle B *(+ espace d'états en assertion)*
- Format des **STEH** produits : DOT, GXL ou HTML .



GénéSyst : construction de STEH

- L'utilisateur fournit le modèle B *(+ espace d'états en assertion)*
- Format des **STEH** produits : DOT, GXL ou HTML *(Interactif)*.


GeneSyst/Exemples/Canal_Communication_V3/Resultats-ref1/Transitions/0.html

L'état de départ : $AEnvoyer = 0 \ \& \backslash n \ DansBuffer = 0$

L'état d'arrivée : $(AEnvoyer > 0)$

Événement numero 1

Nom de l'événement : *Envoyer*

Condition de déclenchabilité : *Gardé*

```
((AEnvoyer = 0 &
  DansBuffer = 0) &
   not(!(AEnvoyer_apres).(AEnvoyer_apres : NATURAL1 =>
  0=1)))
```

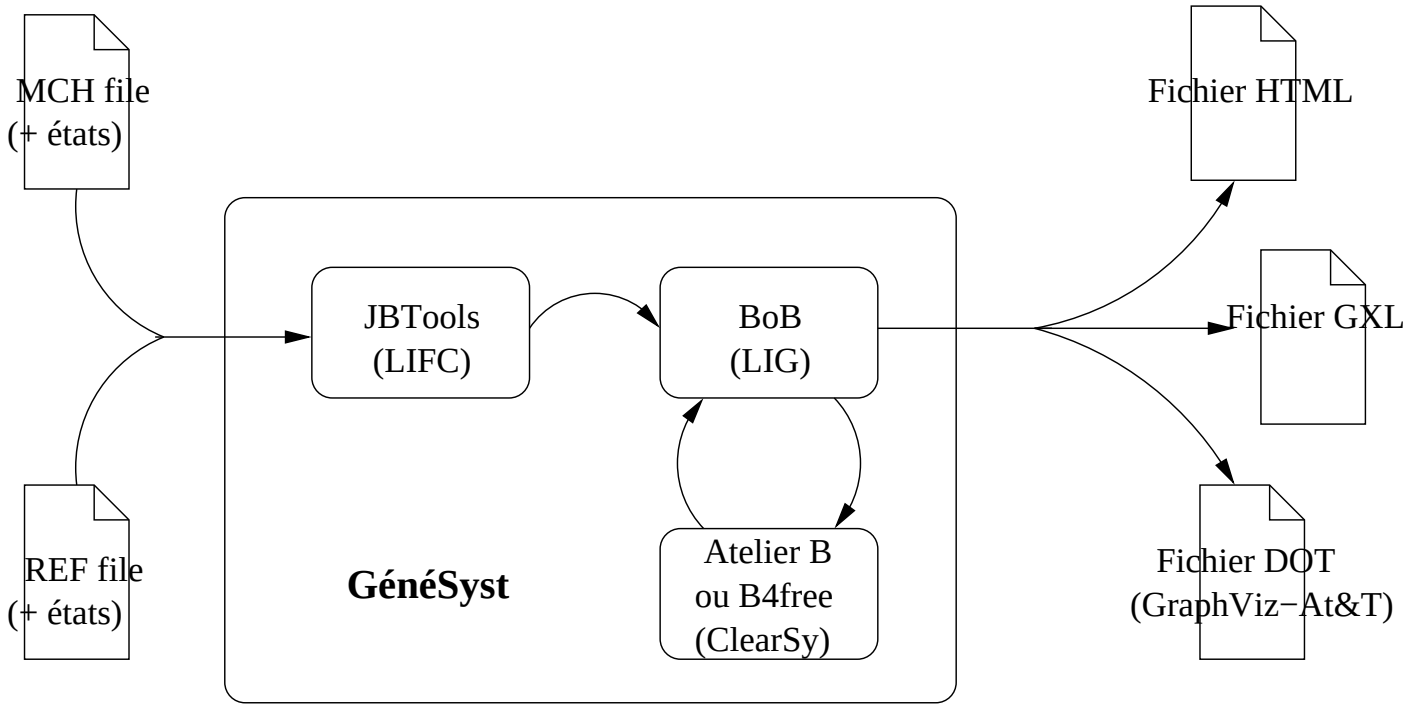
Condition d'atteignabilité : *True*

- Retour -

Expérimentations

Nom	Événements	Espace d'états (Heuristique utilisée)	Transitions	Obligations de preuve	Défauts de preuve
Centrale de réservation	2	3 *(Gardes)*	11	40	2
Écluse	6	4 *(Énumération)*	6	59	0
Machine à chocolat	5	3 *(Énumération)*	10	57	0
Premier raffinement	7	9 *(Limites)*	16	114	0
Parking	4	3 *(Énumération)*	4	32	0
Premier raffinement	4	4 *(Énumération)*	6	29	0
Ordonnanceur	6	4 *(Témoins)*	16	106	0
Canal de communication	3	2 *(Gardes)*	4	20	1
Premier raffinement	4	4 *(Limites)*	12	54	1
Second raffinement	5	5 *(Énumération)*	14	62	1
Canal de communication avec énumération des états	3	102 *(Énumération)*	304	21421	1
Total			21941	21875	6 (0,1%)

Présentation globale



- Diffusé sous la licence CeCILL
- `http://www-lsr.imag.fr/users/Nicolas.Stouls/`

Application au B classique

Traduction $\mathcal{T}$ des opérations en événements

- Opérations appelées depuis l'extérieur *(Contrôle externe)*

  $Res \longleftarrow Op(Params) \mathrel{\hat{=}}$ **PRE** P **THEN** S **END**

- Pré-conditions changées en gardes
  (Difficulté : simulation du passage des paramètres)

Approche par encapsulation des paramètres

$ev_{op} \mathrel{\hat{=}}$ **ANY** *params* **WHERE** P **THEN** **VAR** Res **IN** S **END** **END**

Approche par externalisation des paramètres

- États exprimables en termes des valeurs des paramètres

Application au B classique

Traduction $\mathcal{T}$ des opérations en événements

- Opérations appelées depuis l'extérieur *(Contrôle externe)*

  $Res \longleftarrow Op(Params) \mathrel{\hat{=}}$ **PRE** P **THEN** S **END**

- Pré-conditions changées en gardes
  (Difficulté : simulation du passage des paramètres)

Approche par encapsulation des paramètres

$ev_{op} \mathrel{\hat{=}}$ **ANY** *params* **WHERE** P **THEN** **VAR** Res **IN** S **END** **END**

Approche par externalisation des paramètres

- États exprimables en termes des valeurs des paramètres

Application au B classique

Traduction $\mathcal{T}$ des opérations en événements

- Opérations appelées depuis l'extérieur *(Contrôle externe)*

$$Res \longleftarrow Op(Params) \mathrel{\hat{=}} \textbf{PRE } P \textbf{ THEN } S \textbf{ END}$$

- Pré-conditions changées en gardes
  (Difficulté : simulation du passage des paramètres)

Approche par encapsulation des paramètres

$ev_{op} \mathrel{\hat{=}}$ **ANY** *params* **WHERE** P **THEN** **VAR** Res **IN** S **END** **END**

Approche par externalisation des paramètres

- États exprimables en termes des valeurs des paramètres

Sur-approximation des séquences d'appels possibles

- Exemple de séquence d'actions :

$$S_1 \quad ; \quad S_2$$

- **Sémantique en B événementiel** (Faisabilité) **:**
  Il existe V atteignable par S_1, qui vérifie la garde de S_2
  $$\exists V \cdot ([S_1]x = V \ \wedge \ [x := V]\text{Garde}(S_2))$$

- **Sémantique en B classique** (Terminaison) **:**
  Tout V atteignable par S_1, doit vérifier la pré-condition de S_2
  $$\forall V \cdot ([S_1]x = V \Rightarrow [x := V]\text{Garde}(S_2))$$

- **Intuitivement :**
  $\text{Traces}(t) \subseteq \text{Traces}(\mathcal{T}(t))$

Cas du raffinement B classique

- Particularités du raffinement B classique :
  - Pré-conditions préservées par raffinement
    Conditions de déclenchabilité préservées
  - Pas d'introduction d'opérations
- Si *espace d'états = intersection des gardes* :
  - Conditions de déclenchabilité nécessairement à true ou false
  - Transitions toutes factorisées sur super-état de départ

Application à la vérification de propriétés

- Vérification de propriétés comportementales
  - Utilisation du **STES** plutôt que le modèle B
    (Théorème d'équivalence des traces et des chemins)
  - Dirigé par les besoins de l'étude de cas DEMONEY
    - *Porte-monnaie pour carte à puce [Marlet et al.]*
    - *Modèle B développé au cours de cette thèse*
- 3 techniques de vérification expérimentées :
  - *Pour propriétés invariantes (Inspirée de ProB)*
  - *Pour propriétés décrites par des automates*
  - *Pour propriétés décrites par des formules SEPL*

Application à la vérification de propriétés

- Vérification de propriétés comportementales
  - Utilisation du **STES** plutôt que le modèle B
    (Théorème d'équivalence des traces et des chemins)
  - Dirigé par les besoins de l'étude de cas DEMONEY
    - *Porte-monnaie pour carte à puce [Marlet et al.]*
    - *Modèle B développé au cours de cette thèse*
- 3 techniques de vérification expérimentées :
  - *Pour propriétés invariantes* *(Inspirée de ProB)*
  - *Pour propriétés décrites par des automates*
  - *Pour propriétés décrites par des formules SEPL*

Application à la vérification de propriétés

- Vérification de propriétés comportementales
  - Utilisation du **STES** plutôt que le modèle B
    (Théorème d'équivalence des traces et des chemins)
  - Dirigé par les besoins de l'étude de cas DEMONEY
    - *Porte-monnaie pour carte à puce [Marlet et al.]*
    - *Modèle B développé au cours de cette thèse*
- 3 techniques de vérification expérimentées :
  - *Pour propriétés invariantes* *(Inspirée de ProB)*
  - *Pour propriétés décrites par des automates*
  - *Pour propriétés décrites par des formules SEPL*

SEPL : States/Events Properties Language

Proposition du langage SEPL

- Logique propositionnelle
- 4 atomes de description des événements
  Enabled(q_1, e) *e peut être déclenché depuis* q_1 ;
  AlwaysEnabled(q_1, e) *e est toujours déclenchable depuis* q_1 ;
  Crossable(q_1, e, q_2) *e peut atteindre* q_2 *depuis* q_1 ;
  AlwaysCrossable(q_1, e, q_2) *e peut toujours atteindre* q_2 *depuis* q_1
- Propriétés définies par rapport à un **STES**
- Quantification des noms d'états et d'événements
  (Sucre syntaxique)

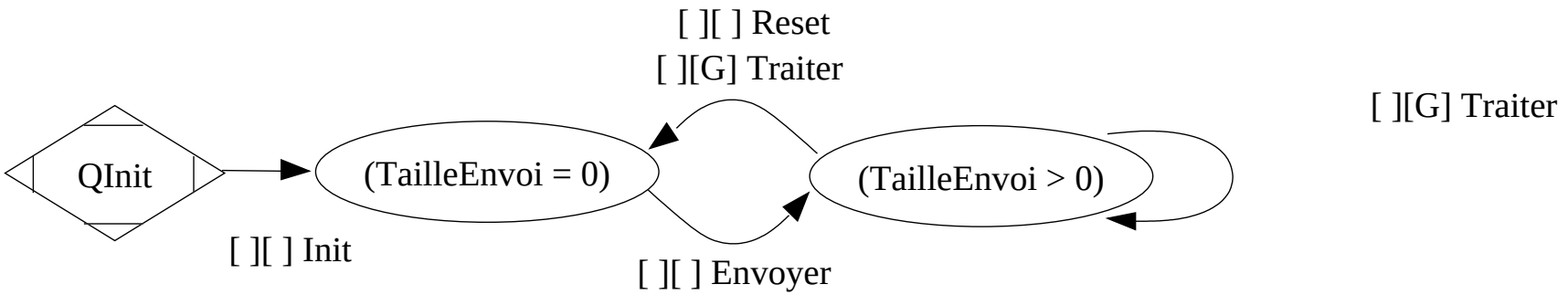
- Inspiré de JTPL [Trentelman *et al.*]
  Propriétés sur les états et les événements
  - modalités (until, unless)
  + état après une transition

Évaluation d'un atome sur un STES

- Sémantique définie sur le modèle B :
  Exemple : $Enabled(q_1, ev) \mathrel{\hat{=}} \exists x \cdot (\mathcal{D}ef(q_1) \wedge \text{Garde}(ev))$
- Proposition :
  évaluer les atomes syntaxiquement sur le **STES**
  (Permet de se ramener à un problème décidable)
- Exemple (dans le cas sans défaut de preuve) :

Condition syntaxique		**Atome**
$D \equiv true$	$\Leftrightarrow$	$AlwaysEnabled(q_1, e)$
$D \not\equiv false$	$\Leftrightarrow$	$Enabled(q_1, e)$
$A \equiv true \wedge D \equiv true$	$\Leftrightarrow$	$AlwaysCrossable(q_1, e, q_2)$
$A \not\equiv false \wedge D \not\equiv false$	$\Leftrightarrow$	$Crossable(q_1, e, q_2)$

Exemple de propriétés SEPL



Envoyer doit être suivi de *Traiter* ou de *Reset* :

$$\mathit{Crossable}(Q_1, \mathit{Envoyer}, Q_2) \Rightarrow (\mathit{Enabled}(Q_2, \mathit{Reset}) \vee \mathit{Enabled}(Q_2, \mathit{Traiter}))$$

Et rien d'autre :

$$\mathit{Crossable}(Q_1, \mathit{Envoyer}, Q_2) \wedge \mathit{Enabled}(Q_2, E) \Rightarrow (E = \mathit{Reset} \vee E = \mathit{Traiter})$$

Plan de l'exposé

Bilan

- Aide à la documentation et à la conception
  - Complémentarité des descriptions
    *Orientées données (B) et comportements (**STES**)*
  - Formalisme avec deux conditions de franchissement
  - Spécificité de l'approche :
    Suivi du concepteur tout au long du processus de raffinement
  - Définition des **STEH** orientée simplicité de compréhension
- Choix des états outillé *(GénéEtat – S. Hamdane et D. Bert)*
- Vérification de propriétés sur un modèle via son **STES**
- Outil utilisé dans le projet EDEMOI
  (Modélisation de la sécurité d'un aéroport)
- Outil utilisé dans le projet POSÉ
  (Validation d'un système par rapport à une politique de sécurité)

Travaux futurs

- Aide au développement :
  - Description des valeurs atteignables par une transition
    « *toutes les valuations de l'état F sont atteignables par ev depuis E* »
    $\Leftrightarrow \forall x' \cdot \exists x \cdot (E \ \wedge \ D \ \wedge \ A \ \wedge \ \neg[\text{Action}(ev)]\neg[x := x']F)$
  - Prise en compte de modèles modulaires
- *GénéSyst* :
  - Approche multi-vues pour conception / validation
    - Intégration dans la plateforme Rodin
    - Mise en place de liens avec des outils d'UML vers B
  - Améliorer le taux de preuves automatiques
    - Utilisation de tactiques utilisateur
      Sélection des hypothèses
    - Interfaçage avec différents prouveurs
      Adaptation à l'existant (ex : outil Why [Filliâtre])
      Nécessite d'axiomatiser certains symboles (ex : Barvey [Couchot et al.])

Avez vous des questions ?

Merci de votre attention.