

Traduction de composants Scade/Lustre vers des Machines B

Florian THIBORD

M2-STL APR / UPMC

Année 2012-2013

- 1 Introduction
- 2 Scade
- 3 Machines B
- 4 Schémas de traduction
 - Machine Abstraite
 - Implantation
- 5 Exemples
 - Bound
 - Integr
 - Extab
- 6 Conclusion

Contexte

Stage réalisé au sein du projet CERCLES². Certification de **composants** réutilisables à l'aide de **méthodes formelles**.

Contexte

Stage réalisé au sein du projet CERCLES². Certification de **composants** réutilisables à l'aide de **méthodes formelles**.

- Composant : Programme + *Contrat*

Contexte

Stage réalisé au sein du projet CERCLES². Certification de **composants** réutilisables à l'aide de **méthodes formelles**.

- Composant : Programme + *Contrat*
- Méthode Formelle : raisonnement rigoureux sur un composant à l'aide d'une logique mathématique.

Description du travail

Composants développés avec **Scade**, développé par Esterel Technologies :

- Programme écrit avec des schémas-blocs
- Engendre du code pseudo-Lustre.

Description du travail

Composants développés avec **Scade**, développé par Esterel Technologies :

- Programme écrit avec des schémas-blocs
- Engendre du code pseudo-Lustre.

Méthode B, développé par J.R. Abrial, utilisée pour la validation des composants :

- Méthode basée sur le *raffinement de machines*
- Cadre du projet : **Machine abstraite** raffinée en **Machine implantation**

Description du travail

Composants développés avec **Scade**, développé par Esterel Technologies :

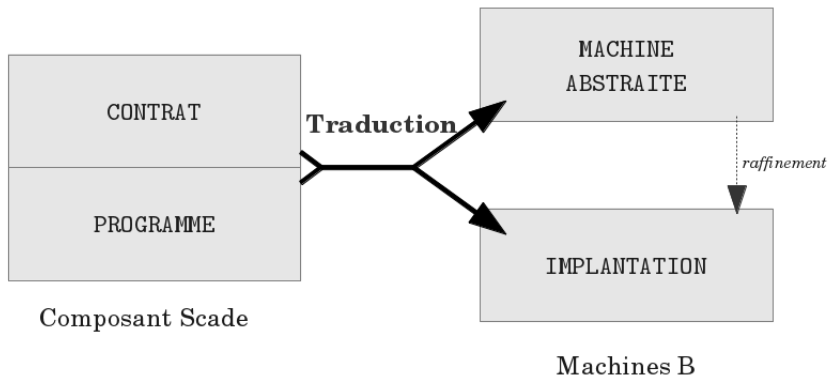
- Programme écrit avec des schémas-blocs
- Engendre du code pseudo-Lustre.

Méthode B, développé par J.R. Abrial, utilisée pour la validation des composants :

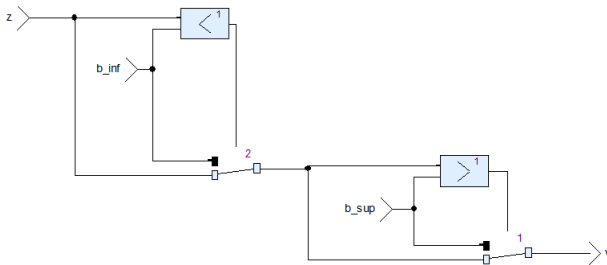
- Méthode basée sur le *raffinement de machines*
- Cadre du projet : **Machine abstraite** raffinée en **Machine implantation**

Ma tâche : développer un outil permettant de traduire un composant Scade en un couple de machines B.

Schéma général



Architecture d'un noeud Scade



Un programme Scade = un **noeud**

Architecture d'un noeud Scade

node bound(b_sup : int; z : int; b_inf : int) returns (v : int)

var

_L7 : int;

_L6 : bool;

_L5 : int;

_L4 : int;

_L3 : bool;

_L2 : int;

_L1 : int;

let

v = _L5;

_L1 = z;

_L2 = b_inf;

_L3 = _L7 > _L4;

_L4 = b_sup;

_L5 = if _L3 then (_L4) else (_L7);

_L6 = _L1 < _L2;

_L7 = if _L6 then (_L2) else (_L1);

tel

Le code pseudo lustre est constitué d'un ensemble d'équation atomiques.

3 familles d'équations :

- $v = \text{op}_{base}(x_1, \dots, x_n);$
- $v = \text{if } c \text{ then } x_1 \text{ else } x_2;$
- $v_1, \dots, v_p = \text{op}_{appel}(x_1, \dots, x_n);$

Un langage synchrone

Scade, un langage manipulant des flots de données. A chaque tic d'horloge :

- 1 ensemble de flots est reçu en entrée
- l'ensemble des équations du noeud sont résolues
- le noeud retourne un ensemble de flots correspondants au résultat

L'opérateur `fby`, prend 3 arguments : une variable, un délai et une valeur d'initialisation.

instant	0	1	2	...
v	10	20	30	...
z	0	10	20	...

Figure: $z = \text{fby}(v, 1, 0)$

Contrat avec Scade

Assertions écrites manuellement dans Scade. Elles sont de 2 type :

- **assume** A : $expr$ où $expr$ correspond à une condition sur une entrée
- **guarantee** G : $expr$ où $expr$ correspond à une condition sur une sortie

Contrat avec Scade

Assertions écrites manuellement dans Scade. Elles sont de 2 type :

- **assume** A : `expr` où `expr` correspond à une condition sur une entrée
- **guarantee** G : `expr` où `expr` correspond à une condition sur une sortie

Exemple sur le noeud Bound :

```
assume A_1 : b_inf <= 2000 and b_inf >= -2000 ;  
assume A_2 : b_sup <= 2000 and b_sup >= -2000 ;  
assume A_3 : z <= 2000 and z >= -2000 ;  
guarantee G_1 : v <= 2000 and v >= -2000 ;
```