Atelier B

Obligations de preuve

Manuel de référence

version 3.7

ATELIER B
Obligations de preuve Manuel de référence
version 3.7

Document établi par CLEARSY.

CLEARSY
Maintenance ATELIER B
Parc de la Duranne
320 avenue Archimède
Les Pléiades III - Bât.A
13857 Aix-en-Provence Cedex 3
France

Tél 33 (0)4 42 37 12 99
Fax 33 (0)4 42 37 12 71
email : maintenance.atelierb@clearsy.com

Table des matières

Chapitre 1

Glossaire

Développement vertical : Ensemble des composants B liés par une clause **REFINES**.

Exemple :

MACHINE	**REFINEMENT**	**REFINEMENT**
MA	MA_1	MA_2
	REFINES	**REFINES**
	MA	MA_1
...	...	...

Le développement vertical est ici formé des composants MA, MA_1 et MA_2.

Instanciation : Affectation d'une valeur aux paramètres d'une machine abstraite lors d'une inclusion/importation. Les paramètres de la machine abstraite instanciée sont appelés les *paramètres formels*, et les valeurs qui leur sont affectés les *paramètres effectifs*.

Exemple :

MACHINE	**MACHINE**
M1	M2(INTV, val)
INCLUDES	**CONSTRAINTS**
M2(1 .. 100, 35)	val $\in$ INTV
...	...

Paramètre effectif : Voir « Instanciation ».

Paramètre formel : Voir « Instanciation ».

Prédicat : Un prédicat est une expression logique qui se lit comme une affirmation en français. Une telle expression peut être exacte ou inexacte. Sont des prédicats les équations, les inéquations, les inégalités, les tests d'appartenance ou d'inclusion. Sont également des prédicats la conjonction de deux prédicats, la disjonction de deux prédicats, et la négations d'un prédicat.

Exemple :

Les expressions suivantes sont des prédicats :

$x = 3$	« *x est égal à 3* »
$5 < 2$	« *5 est strictement inférieur à 2* »
$x \in \{1, 2, 4\}$	« *x appartient à l'ensemble $\{1, 2, 4\}$* »
$x + y^2 = 0 \ \vee \ y < x$	« *$x + y^2$ est égal à 0 ou y est strictement inférieur à x* »

Les expressions suivantes **ne** sont **pas** des prédicats :

$x + y$	« *la somme de* x *et de* y »
$f(2)$	« *la valeur de* f *en 2* »
$\{1, 2, 4\}$	« *l'ensemble* $\{1, 2, 4\}$ »
$A \cup B$	« *l'union de* A *et de* B »

Processus de Vérification : Le processus de Vérification consiste à vérifier la conformité du produit relativement à ses spécifications, tout au long de son développement (« *Construisons nous le produit correctement ?* »).

Chapitre 2

Introduction

Nous décrivons dans ce document les obligations de preuve de la théorie de B :

Définition :

> Une *obligation de preuve* est une formule mathématique à démontrer afin d'assurer qu'un composant B est *correct*.

La théorie B indique quelles sont les obligations de preuve à démontrer pour assurer la *correction* d'un composant B donné. Dans cette optique, les obligations de preuve sont une aide au processus de Vérification.

Les obligations de preuves décrites dans ce document sont des formules mathématiques. Pour bien comprendre leur portée, il est nécessaire d'avoir de bonnes notions de B et de logique mathématique.

Les obligations de preuve produites par le générateur d'obligations de preuve de l'atelier B ne sont pas exactement celles qui sont décrites dans ce document. En effet, le générateur d'obligations de preuve transforme les formules théoriques en formules uniformes (plus nombreuses et plus simples) propres à être utilisées efficacement par le prouveur de l'atelier B.

2.1 Forme générale des obligations de preuve

Les obligations de preuve B sont auto-suffisantes, c'est-à-dire qu'aucune information implicite ne doit être utilisée pour leur démonstration. Toutes les obligations de preuve ont la structure suivante :

$$\begin{array}{l} H \\ \Rightarrow \\ P \end{array}$$

où P et H sont des prédicats. Cette formule signifie qu'il faut démontrer le but P sous l'hypothèse H, H étant généralement une conjonction de prédicats.

En B, le prédicat P et certaines hypothèses de H sont construits par application d'une (ou plusieurs) substitution(s) à un prédicat. B étant un langage mathématique, les substitutions et les prédicats considérés sont directement extraits du source B.

Le lecteur pourra se reporter au manuel de référence du langage B pour obtenir la signification de l'application d'une substitution à un prédicat. Notons que l'application d'une substitution « appel d'opération » se fait par remplacement de cet appel par le corps de l'opération spécifié dans sa machine abstraite d'origine.

2.2 Exemple introductif

Considérons la machine « Exemple » ci-dessous : elle contient une variable d'état « val », qui est un entier naturel non borné ; elle définit également une opération « incrément » dont le rôle est d'ajouter 1 à l'état « val » :

MACHINE
Exemple
VARIABLES
val
INVARIANT
$val \in \mathbb{N}$
OPERATIONS
increment $\widehat{=}$
BEGIN
$val := val + 1$
END
END

La sémantique de B demande que les services offerts par un composant (ici le service est l'opération « increment ») n'invalident jamais l'invariant. Ici, cela signifie que l'application de l'opération « increment » doit préserver le prédicat $val \in \mathbb{N}$. L'obligation de preuve correspondante est :

$$\begin{array}{l} val \in \mathbb{N} \\ \Rightarrow \\ [val := val + 1]val \in \mathbb{N} \end{array}$$

Par définition de la substitution := nous obtenons

$$\begin{array}{l} val \in \mathbb{N} \\ \Rightarrow \\ val + 1 \in \mathbb{N} \end{array}$$

Si cette formule est vraie (ainsi que toutes les autres obligations de preuve associées à la machine Exemple), alors le composant B ci-dessus est correct.

2.3 Portée des obligations de preuve

Certains concepts comme la faisabilité des substitutions ou la réalisabilité des opérations n'appartiennent pas à cette définition de la correction en B. À une exception près, les preuves existentielles des entités définies dans une spécification B ne sont pas demandées

par la théorie. La justification de ceci est que les preuves existentielles, généralement difficiles, donc coûteuses, seront effectuées par exhibition d'un cas particulier (l'implantation) à l'issue du processus de développement B.

L'exception dont nous parlions est le cas des constantes raffinables (clause **ABSTRACT_CONSTANTS**), jamais valuées, et dont la faisabilité doit être explicitement démontrée.

Étant à présent établi que la notion d'obligation de preuve B est liée à celle de correction, c'est sous cette dernière perspective que nous présentons ici les obligations de preuve théoriques du langage B.

2.4 Survol des obligations de preuve

L'exposé des obligations de preuve est décomposé en trois parties : les machines abstraites, les raffinements et les implantations.

Nous n'abordons pas ici le problème de la correction syntaxique qui est un présupposé à la correction sémantique. Les composants B mentionnés dans la suite sont donc supposés syntaxiquement corrects (notion de « type check » du B BOOK).

Les obligations de preuve relatives à une machine abstraite concernent :
- la correction de l'instanciation lors de l'inclusion de machines (clause **INCLUDES**) : les paramètres effectifs d'instanciation doivent vérifier les contraintes des paramètres de la machine incluse ;
- la correction des assertions ;
- la correction de l'initialisation : l'initialisation doit établir l'invariant de la machine (l'invariant de la machine doit être vrai après application de la substitution de l'initialisation) ;
- la correction des opérations : les opérations doivent préserver l'invariant (l'invariant de la machine doit être vrai après application de la substitution de l'opération, sachant que l'invariant était vrai avant) ; les opérations doivent établir leur postcondition ;

Les quatre catégories d'obligations de preuve des machines abstraites se retrouvent pour les raffinements. Cependant, dans le cas du raffinement la correction de l'initialisation et des opérations fait intervenir l'initialisation/l'opération abstraite afin de montrer la pertinence du raffinement. De plus, dans le cas de la correction d'une opération, il est inutile d'établir à nouveau la postcondition : la correction du raffinement suffit.
Les obligations de preuve relatives à un raffinement concernent donc :
- la correction de l'instanciation (obligation de preuve similaire à celle d'une machine abstraite) ;
- la correction des assertions ;
- la correction de l'initialisation : l'initialisation doit établir l'invariant du raffinement (propriétés des nouvelles variables **et** invariant de liaison) **sans contredire l'initialisation spécifiée** ;
- la correction des opérations : les opérations doivent préserver l'invariant du raffinement (propriétés des nouvelles variables **et** invariant de liaison) **sans contredire l'opération spécifiée** ;

Aux quatre catégories d'obligations de preuve précédentes, s'ajoutent, pour les implantations les obligations de preuve des valuations et les obligations de preuve des opérations locales. La pertinence de l'implantation pour l'initialisation et les opérations donne lieu à

des obligations de preuve semblables à celles des raffinements :
- la correction de l'instanciation lors de l'importation de (obligation de preuve similaire à celle de l'inclusion pour la machine abstraite) ;
- la correction de la valuation des constantes et ensembles abstraits : les valuations doivent vérifier les propriétés (explicites et implicites) des constantes et ensembles abstraits ;
- la correction des assertions ;
- la correction de l'initialisation : l'initialisation doit établir l'invariant de l'implantation (propriétés des nouvelles variables **et** invariant de liaison) **sans contredire l'initialisation spécifiée** ;
- la correction des opérations : les opérations doivent préserver l'invariant de l'implantation (propriétés des nouvelles variables **et** invariant de liaison) **sans contredire l'opération spécifiée** ;
- la correction des spécifications d'opérations locales : les opérations locales doivent préserver l'invariant des machines importées ; elles doivent établir leur postcondition ;
- la correction des implémentations d'opérations locales : les implémentations d'opérations locales doivent établir la préservation des variables de l'implantation et des variables importées **sans contredire l'opération locale spécifiée**.

Chapitre 3

Correction de la machine abstraite

3.1 Correction des inclusions

Présentation :

L'inclusion de machines en B est *correcte* lorsque les contraintes des paramètres des paramètres formels de la machine incluse sont vérifiées par les paramètres effectifs spécifiés lors de l'instanciation.

L'instanciation des machines référencées dans la clause **INCLUDES** peut être réalisée à partir :
- des ensembles et des constantes de la machine abstraite, des machines référencées dans la clause **USES** et des machines référencées dans la clause **SEES**,
- des paramètres formels de la machine abstraite et des machines référencées dans la clause **USES**.

Les propriétés de ces différentes entités sont spécifiées par des prédicats dans la clause **PROPERTIES** de la machine abstraite, des machines référencées dans la clause **SEES** et des machines référencées dans la clause **USES**, et dans la clause **CONSTRAINTS** de la machine abstraite et des machines référencées dans la clause **USES**. Ces différents prédicats devront donc figurer dans les hypothèses relatives à la correction des inclusions.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction d'une inclusion, à démontrer pour chaque inclusion, est formée des hypothèses suivantes :
- Contraintes des paramètres de la machine abstraite et des machines référencées dans la clause **USES**.
- Propriétés des constantes de la machine abstraite, des machines référencées dans la clause **SEES** et des machines référencées dans la clause **USES**.

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :
- Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
- Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
- Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

> La contrainte (instanciée) de la machine incluse.

Nous présentons en annexe § A.1 la formulation mathématique de cette même obligation de preuve.

Exemple : Considérons les deux machines suivantes :

MACHINE
MAIN(param10)
CONSTRAINTS
param10 < 10
INCLUDES
COUNT(param10)
END

MACHINE
COUNT(par)
CONSTRAINTS
par < 15
VARIABLES
tab
INVARIANT
tab ∈ (1 . . . par) → BOOL
END

L'obligation de preuve relative à l'inclusion de la machine COUNT ci-dessus doit permettre d'établir que la contrainte de cette machine est bien vérifiée par l'instanciation fournie. Nous obtenons donc :

param10 < 10 « *Contraintes des paramètres de la machine MAIN* »
⇒
[par := param10](par < 15) « *Instanciation du paramètre par param10* »

Ce qui devient, par définition de la substitution :=

param10 < 10
⇒
param10 < 15

3.2 Correction des assertions

Présentation :

Les assertions sont des lemmes pour les phases de preuves. C'est-à-dire que ce sont des théorèmes intermédiaires, déduisibles de l'invariant, utilisés pour établir la preuve de correction du composant. Ces prédicats seront ajoutés (par conjonction) en hypothèse des obligations de preuve à chaque fois que l'invariant figure dans ces hypothèses.

Les assertions de la machine doivent être démontrées à partir de l'invariant et des propriétés des entités manipulées : variables locales, variables des machines référencées dans les clauses **USES**, **INCLUDES**, et **SEES**, constantes, ensembles et paramètres.

L'ordre des assertions dans le texte de la spécification B est important puisque les assertions sont démontrées dans l'ordre, en ajoutant en hypothèses les assertions précédentes déjà démontrées.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction des assertions, à démontrer pour chaque assertion dans l'ordre, est formée des hypothèses suivantes :

- Contraintes des paramètres de la machine abstraite et des machines référencées dans la clause **USES**.
- Propriétés des constantes de la machine abstraite et des machines référencées dans les clauses **SEES**, **USES**, **INCLUDES**.
- Invariants et assertions des machines référencées dans la clause **INCLUDES**, auxquels l'instanciation correspondante a été appliquée.
- Invariant de la machine abstraite et des machines référencées dans la clause **USES**.
- Assertions des machines référencées dans la clause **USES**.
- Assertions précédentes (dans l'ordre du texte) de la machine abstraite.

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :

- Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
- Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
- Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

L'assertion.

Nous présentons en annexe § A.2 la formulation mathématique de cette même obligation de preuve.

Exemple : Considérons la machine abstraite suivante :

MACHINE
EXAM
VARIABLES
xx, yy, zz
INVARIANT
$xx < 0 \wedge$
$yy > 10 \wedge$
$zz = yy * xx$
ASSERTIONS
$zz < 0$;
$zz * xx > 0$
END

Les obligations de preuve relatives aux assertions sont les suivantes :

$xx < 0 \wedge$ « *Invariant de la machine* »
$yy > 10 \wedge$
$zz = yy * xx$
$\Rightarrow$
$zz < 0$ « *Première assertion de la machine* »

et

$$
\begin{array}{ll}
\mathrm{xx} < 0 \wedge & \text{« } \textit{Invariant de la machine} \text{ »} \\
\mathrm{yy} > 10 \wedge & \\
\mathrm{zz} = \mathrm{yy} * \mathrm{xx} \wedge & \\
\mathrm{zz} < 0 & \text{« } \textit{Première assertion de la machine} \text{ »} \\
\Rightarrow & \\
\mathrm{zz} * \mathrm{xx} > 0 & \text{« } \textit{Seconde assertion de la machine} \text{ »}
\end{array}
$$

3.3 Correction de l'initialisation

Présentation :

L'initialisation d'une machine abstraite est *correcte* lorsque celle-ci établit l'invariant. Après application de cette initialisation, l'invariant du composant doit être vrai ; le but à prouver est donc l'application de l'initialisation à l'invariant.

L'initialisation de la machine abstraite peut être spécifiée à partir :
- des ensembles et des constantes de la machine abstraite, des machines référencées dans les clauses **USES**, **INCLUDES** et **SEES**,
- des paramètres formels de la machine abstraite et des machines référencées dans les clauses **USES**,
- des variables des machines référencées dans les clauses **USES** et **INCLUDES**,
- des opérations de consultation et des variables concrètes des machines référencées dans la clause **SEES**.

Les propriétés de ces entités sont spécifiées par des prédicats dans les clauses **PROPERTIES** correspondantes, et dans les clauses **CONSTRAINTS** de la machine abstraite et des machines référencées dans la clause **USES**. Ces prédicats devront donc figurer dans les hypothèses relatives à la correction des inclusions. Les définitions et propriétés des variables sont spécifiées dans les clauses **INVARIANT** et **ASSERTIONS** des machines abstraites correspondantes, c'est pourquoi ces prédicats figurent également en hypothèses.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction de l'initialisation contient les hypothèses suivantes :
- Contraintes des paramètres de la machine abstraite et des machines référencées dans la clause **USES**.
- Propriétés des constantes de la machine abstraite, et des machines référencées dans les clauses **SEES**, **INCLUDES** et **USES**.
- Invariants et assertions des machines référencées dans la clause **INCLUDES**, auxquels l'instanciation correspondante a été appliquée. [1]
- Invariants et assertions des machines référencées dans les clauses **USES** et **SEES**.

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :
- Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
- Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.

[1] Notons qu'il existe un cas particulier (rarement rencontré) pour les groupes de machines reliées par des clauses USES et figurant simultanément dans une clause INCLUDES : dans ce cas les invariants de ces machines reliant des variables de plusieurs composants font partie du but à prouver plutôt que des hypothèses (cf. § A.3).

– Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

> L'invariant de la machine abstraite, après application des initialisations des machines référencées dans la clause **INCLUDES** puis de l'initialisation de la machine.

Nous présentons en annexe § A.3 la formulation mathématique de cette même obligation de preuve.

Exemple : Considérons la machine suivante :

MACHINE
OP01
VARIABLES
v1
INVARIANT
$v1 \in \mathbb{N} \land v1 - 1 \in \mathbb{N}$
INITIALISATION
$v1 := 2$
END

L'obligation de preuve relative à la correction de l'initialisation de la machine OP01 ci-dessus doit permettre d'établir l'invariant. Nous obtenons donc :

« *Aucune hypothèse dans cet exemple* »
$\Rightarrow$
$[v1 := 2](v1 \in \mathbb{N} \ \land \ v1 - 1 \in \mathbb{N})$

Ce qui devient, par définition de la substitution :=

$\Rightarrow$
$2 \in \mathbb{N} \ \land \ 2 - 1 \in \mathbb{N}$

3.4 Correction des opérations

Présentation :

Chaque opération est définie par un en-tête et une substitution généralisée. L'en-tête contient éventuellement des prédicats de typage des paramètres de sortie, qui constituent la postcondition de l'opération. L'effet produit par l'application d'une opération est défini comme l'application de la substitution à un prédicat donné. Dans l'optique de la démonstration de la correction d'une opération, le prédicat considéré est l'invariant de la machine et la postcondition de l'opération :

> Une opération d'une machine abstraite est *correcte* lorsque celle-ci préserve l'invariant, - c'est-à-dire que si l'invariant était vrai avant application de l'opération, alors il est toujours vrai après - et établit la postcondition.

L'obligation de preuve contient l'invariant du composant en hypothèse, et son but est l'invariant et la postcondition, après application de l'opération.

La substitution définissant une opération peut être spécifiée à partir :

– des ensembles et des constantes de la machine abstraite, et des machines référencées dans les clauses **USES**, **INCLUDES** et **SEES**,
– des paramètres formels de la machine abstraite et des machines référencées dans la clause **USES**,
– des variables de la machine abstraite, et des machines référencées dans les clauses **USES** et **INCLUDES**,
– des opérations de consultation et des variables concrètes des machines référencées dans la clause **SEES**.

Les définitions et propriétés des paramètres et constantes sont les prédicats spécifiés dans les clauses **PROPERTIES** et **CONSTRAINTS** des machines abstraites correspondantes. Ces différents prédicats figurent donc dans les hypothèses relatives à la correction de l'opération. Les définitions et propriétés des variables sont les prédicats spécifiés dans les clauses **INVARIANT** et **ASSERTIONS** des machines abstraites correspondantes, c'est pourquoi elles figurent également en hypothèses.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction d'une opération, à démontrer pour chaque opération, contient les hypothèses suivantes :
– Contraintes des paramètres de la machine abstraite et des machines référencées dans la clause **USES**.
– Propriétés des constantes de la machine abstraite, des machines référencées dans la clause **SEES**, des machines référencées dans la clause **INCLUDES** et des machines référencées dans la clause **USES**.
– Invariants et assertions de toutes les machines référencées dans la clause **INCLUDES**, auxquels l'instanciation correspondante a été appliquée.
– Invariants et assertions de la machine abstraite, des machines référencées dans les clauses **USES** et **SEES**,

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :
– Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
– Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
– Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

> L'invariant de la machine abstraite et la postcondition de l'opération, après application de la substitution définissant l'opération.

Nous présentons en annexe § A.4 la formulation mathématique de cette même obligation de preuve.

Exemple : Considérons la machine suivante :

MACHINE
OP02
VARIABLES
v1
INVARIANT
$v1 \in \mathbb{N} \ \wedge \ v1 - 1 \in \mathbb{N}$
INITIALISATION
$v1 := 2$
OPERATIONS
$(ss \in \mathbb{N}) \longleftarrow$ increment $\widehat{=}$
PRE
$v1 - 2 \in \mathbb{N}$
THEN
$v1, ss := v1+1, v1-1$
END
END

L'obligation de preuve relative à la correction de l'opération de la machine OP02 ci-dessus doit permettre de montrer la préservation de l'invariant et l'établissement de la postcondition. Nous obtenons donc :

$$
\begin{array}{l}
v1 \in \mathbb{N} \wedge \quad \text{« } \textit{Invariant de la machine} \text{ »} \\
v1 - 1 \in \mathbb{N} \\
\Rightarrow \\
[\text{PRE } v1 - 2 \in \mathbb{N} \text{ THEN } v1, ss := v1 + 1, v1 - 1 \text{ END}](v1 \in \mathbb{N} \ \wedge \ v1 - 1 \in \mathbb{N} \ \wedge \ ss \in \mathbb{N})
\end{array}
$$

Ce qui ce transforme, par définition de la substitution PRE en

$$
\begin{array}{l}
v1 \in \mathbb{N} \wedge \\
v1 - 1 \in \mathbb{N} \\
\Rightarrow \\
v1 - 2 \in \mathbb{N} \ \Rightarrow \ ([v1, ss := v1 + 1, v1 - 1](v1 \in \mathbb{N} \ \wedge \ v1 - 1 \in \mathbb{N} \ \wedge \ ss \in \mathbb{N}))
\end{array}
$$

Et par définition de la substitution :=

$$
\begin{array}{l}
v1 \in \mathbb{N} \wedge \\
v1 - 1 \in \mathbb{N} \\
\Rightarrow \\
v1 - 2 \in \mathbb{N} \ \Rightarrow \ (v1 + 1 \in \mathbb{N} \ \wedge \ v1 + 1 - 1 \in \mathbb{N} \ \wedge \ v1 - 1 \in \mathbb{N})
\end{array}
$$

Ce qui est logiquement équivalent à

$$
\begin{array}{l}
v1 \in \mathbb{N} \wedge \\
v1 - 1 \in \mathbb{N} \wedge \\
v1 - 2 \in \mathbb{N} \\
\Rightarrow \\
v1 + 1 \in \mathbb{N} \ \wedge \ v1 + 1 - 1 \in \mathbb{N} \ \wedge \ v1 - 1 \in \mathbb{N}
\end{array}
$$

Chapitre 4

Correction du raffinement

La correction des clauses **INCLUDES** et la preuve des assertions donnent lieu à des obligations de preuve semblables à celles des machines abstraites.

Comme pour les machines abstraites, la correction de l'initialisation et des opérations des raffinements est établie par établissement et préservation de l'invariant. Cependant l'invariant d'un raffinement est un invariant **de liaison** définissant les propriétés des nouvelles variables par rapport aux variables du composant raffiné. Il ne suffit plus d'appliquer une substitution à l'invariant, mais il faut appliquer conjointement la substitution du raffinement et la substitution du composant raffiné suivant la formule de la double négation (voir ci-dessous). Il est alors clair que la correction d'un raffinement n'est pas une simple vérification de cohérence interne, mais bien une assurance que le raffinement a été développé dans le respect de sa spécification.

Dans toutes les obligations de preuve décrites dans cette section les propriétés des constantes des raffinements antérieurs et de la machine abstraite sont composées des propriétés des constantes propres du composant et des propriétés des constantes des machines référencées dans la clause **INCLUDES**. De même, les invariants des raffinements antérieurs et de la machine abstraite sont composées des invariants propres et des invariants instanciés des machines référencées dans la clause **INCLUDES**.

4.1 Correction des inclusions

Présentation :

De même que pour les machines abstraites, l'inclusion de machines en B est *correcte* lorsque les contraintes des paramètres formels de la machine incluse sont vérifiées par les paramètres effectifs spécifiés lors de l'instanciation.

L'instanciation des machines référencées dans la clause **INCLUDES** peut être réalisée à partir :
– des ensembles et des constantes du développement vertical et des machines référencées dans la clause **SEES**,
– des paramètres formels de la machine abstraite.
Les propriétés de ces entités sont spécifiées dans les clauses **PROPERTIES** des composants correspondants, et dans la clause **CONSTRAINTS** de la machine abstraite. Ces clauses devront donc figurer dans les hypothèses relatives à la correction des inclusions.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction d'une inclusion, à démontrer pour chaque inclusion du raffinement, est formée des hypothèses suivantes :
- Contraintes des paramètres de la machine abstraite.
- Propriétés des constantes du développement vertical et des machines référencées dans la clause **SEES**.

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :
- Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
- Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
- Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

> La contrainte (instanciée) de la machine incluse.

Nous présentons en annexe § B.1 la formulation mathématique de cette même obligation de preuve.

Exemple : Considérons les composants suivants :

MACHINE
MAIN(param10)
CONSTRAINTS
param10 < 10
END

REFINEMENT
MAIN_1(param10)
REFINES
MAIN
CONSTANTS
vscal
PROPERTIES
vscal ∈ ℕ ∧
vscal < param10
/2
INCLUDES
COUNT(vscal)
END

MACHINE
COUNT(par)
CONSTRAINTS
par < 15
VARIABLES
tab
INVARIANT
tab ∈ (1 . . . par) →
BOOL
END

L'obligation de preuve relative à l'inclusion de la machine COUNT dans le raffinement MAIN_1 ci-dessus doit permettre d'établir que la contrainte de COUNT est bien vérifiée par l'instanciation fournie. Nous obtenons donc :

param10 < 10 ∧ « *Contraintes des paramètres de la machine abstraite* »
vscal < param10/2 ∧ « *Propriétés des constantes du raffinement* »
vscal ∈ ℕ
⇒
[par := vscal](par < 15) « *Contrainte de COUNT et son instanciation* »

Ce qui devient, par définition de la substitution :=

$$
\begin{array}{l}
\text{param10} < 10 \wedge \\
\text{vscal} < \text{param10}/2 \wedge \\
\text{vscal} \in \mathbb{N} \\
\Rightarrow \\
\text{vscal} < 15
\end{array}
$$

4.2 Correction des assertions

Présentation :

Les assertions sont des lemmes pour les phases de preuves. Ces prédicats seront ajoutés (par conjonction) en hypothèse des obligations de preuve à chaque fois que l'invariant figure dans ces hypothèses.

Les assertions du raffinement doivent être démontrées à partir de l'invariant et des propriétés des entités manipulées : variables du raffinement, variables concrètes du développement vertical, variables des machines référencées dans la clause **INCLUDES**, constantes et ensembles.

L'ordre des assertions dans le texte de la spécification B est important puisque les assertions sont démontrées dans l'ordre, en ajoutant en hypothèses les assertions précédentes déjà démontrées.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction des assertions, à démontrer pour chaque assertion dans l'ordre, est formée des hypothèses suivantes :
- Contraintes des paramètres de la machine abstraite,
- Propriétés des constantes du développement vertical, des machines référencées dans les clauses **SEES** et **INCLUDES**.
- Invariants et assertions des machines référencées dans la clause **INCLUDES**, auxquels l'instanciation correspondante a été appliquée.
- Invariants du développement vertical.
- Assertions des raffinements antérieurs et de la machine abstraite.
- Assertions précédentes (dans l'ordre du texte) du raffinement.

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :
- Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
- Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
- Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

L'assertion.

Nous présentons en annexe § B.2 la formulation mathématique de cette même obligation de preuve.

Exemple : Considérons les composants suivants :

MACHINE
EXAM
VARIABLES
xx, yy
INVARIANT
$xx < 0 \wedge$
$yy > 10 \wedge$
END

REFINEMENT
EXAM_1
REFINES
EXAM
VARIABLES
zz
INVARIANT
$zz \in \mathbb{Z} \wedge$
$zz = yy * xx$
ASSERTIONS
$zz < 0$
END

L'obligation de preuve relative à l'assertion est la suivante :

$xx < 0 \wedge$ « *Invariant de la machine* »
$yy > 10 \wedge$
$zz \in \mathbb{Z} \wedge$ « *Invariant du raffinement* »
$zz = yy * xx$
$\Rightarrow$
$zz < 0$ « *Assertion du raffinement* »

4.3 Correction de l'initialisation

Présentation :

L'initialisation d'un raffinement est *correcte* lorsque celle-ci établit l'invariant du raffinement, *sans contredire l'initialisation du composant raffiné*. Comprenons dans cette définition que la nouvelle initialisation ne doit pas produire exactement les même résultats que l'ancienne, mais plutôt que les nouvelles valeurs de l'initialisation ne doivent pas être en contradiction avec les anciennes. Généralement cela signifie que le domaine des valeurs des variables communes peut être restreint.

Par exemple, si l'initialisation d'une variable *vv* de la machine abstraite spécifie une valeur initiale dans l'intervalle 1..3, l'initialisation du raffinement pourra affecter la valeur initiale 1 à *vv*.

Lorsqu'une variable abstraite n'est pas conservée dans le raffinement, elle est généralement raffinée par l'intermédiaire de l'invariant de liaison, en une nouvelle variable. Dans ce cas, l'initialisation de la nouvelle variable doit être effectuée sans contredire l'initialisation de la variable abstraite.

Par exemple, l'initialisation d'une variable abstraite SS est l'ensemble vide $\emptyset$, et l'invariant de liaison spécifie que la nouvelle variable num est le cardinal de SS, alors l'initialisation de num est 0.

Le but à prouver est construit à partir de l'invariant : à la contraposée de cet invariant on applique l'initialisation du composant raffiné (notamment pour initialiser les variables abstraites, en liaison avec les variables concrètes) ; puis, à la contraposée du prédicat obtenu on applique l'initialisation du raffinement.

L'initialisation du raffinement peut être spécifiée à partir :
- des ensembles et des constantes du développement vertical, et des machines référencées dans les clauses **SEES** et **INCLUDES**,
- des paramètres formels de la machine abstraite,
- des variables des machines référencées dans la clause **INCLUDES**,
- des variables concrètes et des opérations de consultation des machines référencées dans la clause **SEES**.

Les définitions et propriétés des paramètres et constantes sont spécifiées dans les clauses **PROPERTIES** et **CONSTRAINTS** des machines abstraites correspondantes. Ces différentes clauses figurent donc dans les hypothèses relatives à la correction de l'initialisation. Les définitions et propriétés des variables sont spécifiées dans les clauses **INVARIANT** et **ASSERTIONS** des composants abstraits correspondants, c'est pourquoi elles figurent également en hypothèses.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction de l'initialisation contient les hypothèses suivantes :
- Contrainte de la machine abstraite,
- Propriétés des constantes du développement vertical, et des machines référencées dans les clauses **SEES** et **INCLUDES**,
- Invariants et assertions de toutes les machines référencées dans la clause **INCLUDES**, auxquels l'instanciation correspondante a été appliquée.
- Invariants et assertions des machines référencées dans la clause **SEES**.

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :
- Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
- Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
- Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

> L'invariant du raffinement, après contraposition, puis application de l'initialisation du composant raffiné, contraposition, puis application de l'initialisation du raffinement.

Nous présentons en annexe § B.3 la formulation mathématique de cette même obligation de preuve.

Exemple : Considérons les composants suivants :

MACHINE
OP01
VARIABLES
v1
INVARIANT
v1 ∈ 0 . . 10
INITIALISATION
ANY valeur WHERE
valeur ∈ 1 . . 5
THEN
v1 := valeur
END
END

REFINEMENT
OP01_1
REFINES
OP01
VARIABLES
v2
INVARIANT
v2 = 2 * v1
INITIALISATION
v2 := 2
END

L'obligation de preuve relative à la correction de l'initialisation du raffinement OP01_1 ci-dessus est construite comme suit : la contraposée de l'invariant est :

$$v2 \neq 2 * v1$$

L'initialisation du composant raffiné, appliquée à ce prédicat est

$$[\text{ANY valeur WHERE valeur} \in 1 . . 5 \text{THEN v1} := \text{valeur END}](v2 \neq 2 * v1)$$

Ce qui devient, par définition de la substitution ANY :

$$\forall \text{valeur}.(\text{valeur} \in 1 . . 5 \quad \Rightarrow \quad v2 \neq 2 * \text{valeur})$$

La contraposée de ce dernier prédicat est

$$\exists \text{valeur} . (\text{valeur} \in 1 . . 5 \quad \wedge \quad v2 = 2 * \text{valeur})$$

L'application de l'initialisation du raffinement nous permet d'instancier v2 par 2, nous obtenons alors l'obligation de preuve à prouver

$$\Rightarrow$$
$$\exists \text{valeur} . (\text{valeur} \in 1 . . 5 \quad \wedge \quad 2 = 2 * \text{valeur})$$

4.4 Correction des opérations

Présentation :

Chaque opération du raffinement est une nouvelle version (plus concrète) d'une opération précédemment spécifiée. Les en-têtes des deux opérations sont identiques, seule la substitution généralisée définissant l'effet de l'opération est modifiée.

> Une opération d'un raffinement est *correcte* lorsqu'elle préserve l'invariant *sans contredire l'opération spécifiée*, et lorsque sa précondition est moins restrictive que la précondition spécifiée.

Comprenons dans cette définition que la nouvelle opération ne doit pas produire exactement les mêmes résultats que l'ancienne, mais plutôt que les effets de la nouvelle opération ne doivent pas être en contradiction avec les effets spécifiés dans l'opération abstraite. Le but de l'obligation de preuve va ainsi être construit par une double négation.

Cette définition signifie que le domaine de valeurs des variables de sortie de l'opération peut être restreint.

Par exemple, si l'opération abstraite retourne une valeur dans l'intervalle 1..10, alors la nouvelle opération pourra retourner une valeur dans l'intervalle 2..4.

Comme pour les opérations de machines abstraites le but à prouver se base sur l'invariant du composant ; dans le cas des raffinements on lui ajoute (par conjonction) un prédicat d'égalité entre les variables de sorties de l'opération du raffinement, et les variables de sorties renommées de l'opération spécifiée. Le but à prouver est alors composé
– de l'application de l'opération raffinée à la négation de l'application de l'opération abstraite à la négation de l'invariant.

L'exemple ci-après clarifie cette définition complexe.

La substitution définissant une opération peut être spécifiée à partir :
– des ensembles et des constantes du développement vertical, et des machines référencées dans les clauses **INCLUDES** et **SEES**,
– des paramètres formels de la machine abstraite,
– des variables concrètes du développement vertical,
– des variables des machines référencées dans la clause **INCLUDES**,
– des variables concrètes et des opérations de consultation des machines référencées dans la clause **SEES**.

Les définitions et propriétés des paramètres et constantes sont spécifiées dans les clauses **PROPERTIES** et **CONSTRAINTS** des composants correspondants. Ces différentes clauses figurent donc dans les hypothèses relatives à la correction de l'opération. Les définitions et propriétés des variables sont spécifiées dans les clauses **INVARIANT** et **ASSERTIONS** des composants correspondants, c'est pourquoi elles figurent également en hypothèses.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction d'une opération, à démontrer pour chaque opération, contient les hypothèses suivantes :
– Contraintes des paramètres de la machine abstraite,
– Propriétés des constantes du développement vertical, et des machines référencées dans les clauses **SEES** et **INCLUDES**,
– Invariants et assertions des machines référencées dans la clause **INCLUDES**, auxquels l'instanciation correspondante a été appliquée,

– Invariants et assertions du développement vertical et des machines référencées dans la clause **SEES**,
– Précondition de l'opération dans la machine abstraite.

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :

– Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
– Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
– Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

> l'opération du raffinement appliquée à la négation de l'application de l'opération abstraite sur la négation de l'invariant.

Nous présentons en annexe § B.4 la formulation mathématique de cette même obligation de preuve.

Exemple : Considérons les composants suivants :

```
MACHINE
    OP01
VARIABLES
    v1
INVARIANT
    v1 ∈ ℕ
INITIALISATION
    v1 := 0
OPERATIONS
    /* L'opération retourne la nouvelle valeur de v1,
       plus grande que la précédente */
    out ⟵ valv1 ≙
        BEGIN
            ANY valeur WHERE
                valeur > v1
            THEN
                out, v1 := valeur, valeur
            END
        END
END
```

```
REFINEMENT
    OP01_1
REFINES
    OP01
VARIABLES
    v2
INVARIANT
    v2 > v1
INITIALISATION
    v2 := 1
OPERATIONS
    /* La nouvelle valeur de v1 est déjà stockée dans v2,
       il suffit donc de renvoyer v2 et de préparer v2 pour l'appel suivant */
    out ⟵ valv1 ≙
        BEGIN
            out, v2 := v2, v2+1
        END
END
```

La première opération **valv1** renvoie des valeurs toujours plus grandes. La seconde opération renvoie toujours le successeur (arithmétique) du retour précédent. L'obligation de preuve doit donc permettre de montrer que le successeur arithmétique est bien une valeur plus grande que la précédente.

L'obligation de preuve relative à la correction de l'opération **valv1** du raffinement OP01_1 ci-dessus est construite comme suit : l'invariant en conjonction avec le prédicat d'égalité entre le paramètre de sortie de l'opération dans OP01 et le paramètre de sortie de l'opération dans OP01_1 est :

$$\text{v2} > \text{v1} \ \wedge \ out = out'$$

Ce prédicat représente ce qui doit toujours être vrai.

La contraposée de cette formule est

$$\text{v2} \leq \text{v1} \ \vee \ out \neq out'$$

Ce prédicat représente ce qui ne doit jamais se produire.

L'opération du composant raffiné (la machine abstraite ici), appliquée à ce prédicat est

$$[\text{ANY valeur WHERE valeur} > \text{v1 THEN out, v1} := \text{valeur, valeur END}](\text{v2} \leq \text{v1} \ \vee \ out \neq out')$$

Ce qui devient, par définition des substitutions ANY et := :

$$\forall \ \text{valeur}.(\text{valeur} > \text{v1} \ \Rightarrow \ (\text{v2} \leq \text{valeur} \ \vee \ \text{valeur} \neq out'))$$

Ce prédicat représente l'effet de l'opération spécifiée sur ce qui ne doit jamais se produire.

La contraposée de ce dernier prédicat est

$$\exists \, \text{valeur} \ . \ (\text{valeur} > \text{v1} \ \wedge \ (\text{v2} > \text{valeur} \ \wedge \ \text{valeur} = out'))$$

L'équation « valeur = out' » nous permet de simplifier logiquement ce prédicat en

$$out' > \mathrm{v1} \ \wedge \ \mathrm{v2} > out'$$

Ce prédicat représente ce qui n'est pas établi par l'opération spécifiée sur ce qui ne doit jamais se produire.

L'application de l'opération du raffinement nous permet d'instancier out' par v2 et v2 par v2+1 nous obtenons alors l'obligation de preuve suivante :

$$\begin{array}{l} \mathrm{v1} \in \mathbb{N} \wedge \\ \mathrm{v2} > \mathrm{v1} \\ \Rightarrow \\ \mathrm{v2} > \mathrm{v1} \ \wedge \ \mathrm{v2} + 1 > \mathrm{v2} \end{array}$$

Remarquons que v1 est la valeur renvoyée par le précédent appel de l'opération, v2 est la valeur qui va être renvoyée par l'appel courant, et v2+1 est la valeur qui sera renvoyée par le prochain appel. Nous avons bien entre ces trois valeurs la relation d'ordre spécifiée dans l'opération **valv1** de la machine abstraite.

Chapitre 5

Correction de l'implantation

Comme pour la correction du raffinement, la correction d'une implantation n'est pas une simple vérification de cohérence interne, mais une assurance que l'implantation a été développée dans le respect de sa spécification. Les obligations de preuve relatives à la correction des implantations sont similaires à celles des raffinements, auxquelles s'ajoute l'obligation de preuve d'existence des constantes raffinables.

Dans les obligations de preuve décrites dans cette section les propriétés des constantes des raffinements antérieurs et de la machine abstraite sont composées des propriétés des constantes propres de chacun de ces composants et des propriétés des constantes des machine référencées dans la clause **INCLUDES** par ces composants. De même, les invariants des raffinements antérieurs et de la machine abstraite sont composées des invariants propres et des invariants instanciés des machines référencées dans la clause **INCLUDES**.

5.1 Correction des importations

Présentation :

De même que pour l'inclusion dans les machines abstraites, l'importation de machines en B est *correcte* lorsque les contraintes des paramètres des paramètres formels de la machine importée sont vérifiées par les paramètres effectifs spécifiés lors de l'instanciation.

L'instanciation des machines référencées dans la clause **IMPORTS** peut être réalisée à partir :
- des ensembles et des constantes du développement vertical et des machines référencées dans la clause **SEES**,
- des paramètres formels de la machine abstraite.

Les propriétés de ces entités sont spécifiées dans les clauses **PROPERTIES** des composants correspondants, et dans la clause **CONSTRAINTS** de la machine abstraite. Ces clauses devront donc figurer dans les hypothèses relatives à la correction des importations.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction d'une importation, à démontrer pour chaque importation de l'implantation, est formée des hypothèses suivantes :
- Contraintes des paramètres de la machine abstraite.
- Propriétés des constantes du développement vertical et des machines référencées dans la clause **SEES**.

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :
– Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
– Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
– Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.
Sous ces hypothèses, le but à démontrer est :

> La contrainte (instanciée) de la machine importée.

Nous présentons en annexe § C.1 la formulation mathématique de cette même obligation de preuve.

5.2 Correction des valuations

Présentation :

La valuation des constantes et des ensembles abstraits définis dans la machine abstraite, dans les raffinements et dans l'implantation doit vérifier les propriétés de ces différentes entités.

La valuation des constantes et ensembles peut être réalisée à partir :
– des ensembles et des constantes des machines référencées dans les clauses **IMPORTS** et **SEES**.
Les propriétés de ces entités sont spécifiées dans les clauses **PROPERTIES** des machines référencées dans les clauses **IMPORTS** et **SEES**. Ces clauses devront donc figurer dans les hypothèses relatives à la correction des valuations.

Description de l'obligation de preuve :

L'obligation de preuve relative à la correction des constantes contient les hypothèses suivantes :
– Propriétés des constantes des machines référencées dans la clause **SEES**,
– Propriétés des constantes des machines référencées dans la clause **IMPORTS**.
A ces hypothèses s'ajoutent les hypothèses implicites suivantes :
– Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
– Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.
Sous ces hypothèses, le but à démontrer est :

> L'existence des constantes raffinables pour la conjonction des propriétés des constantes de la machine abstraite, des raffinements et de l'implantation auxquelles la substitution de valuation à été appliquée.

Nous présentons en annexe § C.2 la formulation mathématique de cette même obligation de preuve.

Exemple : Considérons les composants suivants :

MACHINE
OP01
CONSTANTS
S1, c1
ABSTRACT_CONSTANTS
c2
PROPERTIES
$c1 \in 0 \,..\, 10 \wedge$
$c2 \in S1$
END

IMPLEMENTATION
OP01_1
REFINES
OP01
VALUES
$S1 = 1 \,..\, 5\,;$
$c1 = 0$
END

L'obligation de preuve relative à la correction de la valuation est :

$$\Rightarrow$$
$$\exists\, c2 \;.\; [S1, c1 := 1 \,..\, 5, 0](c1 \in 0 \,..\, 10 \;\wedge\; c2 \in S1)$$

Par application de la substitution nous obtenons

$$\Rightarrow$$
$$\exists\, c2 \;.\; (0 \in 0 \,..\, 10 \;\wedge\; c2 \in 1 \,..\, 5)$$

5.3 Correction des assertions

Présentation :

Les assertions de l'implantation doivent être démontrées à partir de l'invariant et des propriétés des entités manipulées : variables locales, variables abstraites, variables des machines référencées dans les clauses **IMPORTS** et **SEES**, constantes et ensembles.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction d'une assertion, à démontrer pour chaque assertion dans l'ordre du texte, est formée des hypothèses suivantes :
– Contraintes des paramètres de la machine abstraite,
– Propriétés des constantes du développement vertical et des machines référencées dans les clauses **SEES** et **IMPORTS**.
– Invariants et assertions des machines référencées dans la clause **IMPORTS**, auxquels l'instanciation correspondante a été appliquée.
– Invariant du développement vertical.
– Assertions des raffinements antérieurs et de la machine abstraite.
– Assertions précédentes (dans l'ordre du texte) de l'implantation.
A ces hypothèses s'ajoutent les hypothèses implicites suivantes :

– Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
– Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
– Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

L'assertion.

Nous présentons en annexe § C.3 la formulation mathématique de cette même obligation de preuve.

5.4 Correction de l'initialisation

Présentation :

L'initialisation d'une implantation est *correcte* lorsque celle-ci établit l'invariant de l'implantation, *sans contredire l'initialisation du composant implanté*. Comprenons dans cette définition que la nouvelle initialisation ne doit pas produire exactement les même résultats que l'ancienne, mais plutôt que les nouvelles valeurs de l'initialisation ne doivent pas être en contradiction avec les anciennes. Généralement cela signifie que les domaines de valeurs des variables communes (les « concrete_variables » de tout le développement vertical) peuvent être restreints.

Le but à prouver est construit à partir de l'invariant : à la contraposée de cet invariant on applique l'initialisation du composant raffiné (notamment pour initialiser les variables abstraites, en liaison avec les variables concrètes) ; puis, à la contraposée du prédicat obtenu on applique l'initialisation de l'implantation.

L'initialisation de l'implantation peut être spécifiée à partir :
– des ensembles et des constantes du développement vertical et des machines référencées dans les clauses **IMPORTS** et **SEES**,
– des paramètres formels de la machine abstraite,
– des variables des machines référencées dans la clause **IMPORTS**,
– des variables concrètes et des opérations de consultation des machines référencées dans la clause **SEES**.

Les définitions et propriétés des paramètres et constantes sont spécifiées dans les clauses **PROPERTIES** et **CONSTRAINTS** des machines abstraites correspondantes. Ces différentes clauses figurent donc dans les hypothèses relatives à la correction de l'initialisation. Les définitions et propriétés des variables sont spécifiées dans les clauses **INVARIANT** et **ASSERTIONS** des machines abstraites correspondantes, c'est pourquoi elles figurent également en hypothèses.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction de l'initialisation contient les hypothèses suivantes :
– Contraintes des paramètres de la machine abstraite,
– Propriétés des constantes du développement vertical, et des machines référencées dans les clauses **SEES** et **IMPORTS**.
– Invariants et assertions des machines référencées dans la clause **IMPORTS**, auxquels l'instanciation correspondante a été appliquée.

– Invariants et assertions des machines référencées dans la clause **SEES**.

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :

– Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
– Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
– Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

> L'invariant de l'implantation, après contraposition, puis application de l'initialisation du composant raffiné, contraposition, puis application de l'initialisation de l'implantation.

Nous présentons en annexe § C.4 la formulation mathématique de cette même obligation de preuve.

Exemple : Considérons les composants suivants :

MACHINE
OP01
VARIABLES
v1
INVARIANT
$v1 \in 0 \,..\, 10$
INITIALISATION
$v1 :\in 1..5$
END

IMPLEMENTATION
OP01_i
REFINES
OP01
VARIABLES
v2
INVARIANT
$v2 = 2 * v1$
INITIALISATION
$v2 := 2$
END

L'obligation de preuve relative à la correction de l'initialisation de l'implantation OP01_i ci-dessus est construite comme suit : la contraposée de l'invariant de liaison est :

$$v2 \neq 2 * v1$$

L'initialisation de la spécification, appliquée à ce prédicat est

$$[v1 :\in 1 \,..\, 5](v2 \neq 2 * v1)$$

Ce qui devient, par définition de la substitution $:\in$:

$$\forall\, v1\_0.(v1\_0 \in 1 \,..\, 5 \quad \Rightarrow \quad v2 \neq 2 * v1\_0)$$

La contraposée de ce dernier prédicat est

$$\exists\, v1\_0 \,.\, (v1\_0 \in 1 \,..\, 5 \quad \wedge \quad v2 = 2 * v1\_0)$$

L'application de l'initialisation du raffinement nous permet d'instancier v2 par 2, nous obtenons alors l'obligation de preuve

$$\Rightarrow \\ \exists\, v1\_0 \,.\, (v1\_0 \in 1 \,..\, 5 \quad \wedge \quad 2 = 2 * v1\_0)$$

5.5 Correction des opérations

Présentation : Chaque opération de l'implantation est une nouvelle version (concrète) d'une opération précédemment spécifiée. Les en-têtes des deux opérations sont identiques, seule la substitution généralisée définissant l'effet de l'opération est modifiée.

> Une opération d'une implantation est *correcte* lorsqu'elle préserve l'invariant *sans contredire l'opération raffinée.*

Comprenons dans cette définition que la nouvelle opération ne doit pas produire exactement les mêmes résultats que l'ancienne, mais plutôt que les effets de la nouvelle opération ne doivent pas être en contradiction avec les effets spécifiés dans l'opération abstraite. Le but de l'obligation de preuve va ainsi être construit par une double négation.

Comme pour les opérations des raffinements le but à prouver est basé sur l'invariant (de liaison) de l'implantation en conjonction avec un prédicat d'égalité entre les variables de sorties de l'opération de l'implantation raffinement, et les variables de sorties renommées de l'opération spécifiée. Le but à prouver est alors composé
– de l'application de l'opération de l'implantation à la négation de l'application de l'opération abstraite à la négation de l'invariant.

La substitution définissant une opération peut être spécifiée à partir :
– des ensembles et des constantes du développement vertical et des machines référencées dans les clauses **IMPORTS** et **SEES**,
– des paramètres formels de la machine abstraite,
– des variables concrètes (**uniquement**) du développement vertical et des machines référencées dans les clauses **IMPORTS** et **SEES**.

Les définitions et propriétés des paramètres et constantes sont spécifiées dans les clauses **PROPERTIES** et **CONSTRAINTS** des composants correspondants. Ces différentes clauses figurent donc dans les hypothèses relatives à la correction de l'opération. Les définitions et propriétés des variables sont spécifiées dans les clauses **INVARIANT** et **ASSERTIONS** des composants correspondants, c'est pourquoi elles figurent également en hypothèses.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction d'une opération, à démontrer pour chaque opération, contient les hypothèses suivantes :
– Contraintes des paramètres de la machine abstraite,
– Propriétés des constantes du développement vertical et des machines référencées dans les clauses **SEES** et **IMPORTS**,
– Invariants et assertions des machines référencées dans la clause **IMPORTS**, auxquels l'instanciation correspondante a été appliquée,
– Invariants et assertions du développement vertical, et des machines référencées dans la clause **SEES**,
– Précondition de l'opération dans la machine abstraite.

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :
– Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
– Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
– Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

l'opération de l'implantation appliquée à la négation de l'application de l'opération abstraite sur la négation de l'invariant.

Nous présentons en annexe § C.5 la formulation mathématique de cette même obligation de preuve.

5.6 Correction des spécifications d'opérations locales

Présentation : Des opérations locales peuvent être utilisées en implantation afin de factoriser l'écriture de projets B. Les opérations locales sont spécifiées et implémentées dans une même implantation et ne sont utilisables qu'au sein de cette implantation.

> Une spécification d'opération locale d'une implantation est *correcte* lorsqu'elle préserve l'invariant des machines importées par l'implantation et établit sa postcondition.

Une opération locale doit préserver l'invariant des machines importées, car elle peut modifier directement leurs variables. Par contre, une opération locale ne préserve pas forcément l'invariant de l'implantation.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction d'une spécification d'opération locale, à démontrer pour chaque opération locale, contient les hypothèses suivantes :
- Contraintes des paramètres de la machine abstraite,
- Propriétés des constantes du développement vertical et des machines référencées dans les clauses **SEES** et **IMPORTS**,
- Invariants et assertions des machines référencées dans la clause **IMPORTS**, auxquels l'instanciation correspondante a été appliquée,
- Invariants et assertions des machines référencées dans la clause **SEES**,
- Typage B des variables concrètes de l'implantation,
- Précondition de la spécification de l'opération locale.

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :
- Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
- Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
- Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

> Le corps de l'opération locale, sans son éventuelle précondition, préserve l'invariant des machines importées et établit la postcondition.

Nous présentons en annexe § C.6 la formulation mathématique de cette même obligation de preuve.

5.7 Correction des implémentations d'opérations locales

Présentation : Des opérations locales spécifiées dans la clause **LOCAL_OPERATIONS** sont implémentées dans la clause **OPERATIONS** de la même implantation.

> Une implémentation d'opération locale est *correcte* lorsqu'elle préserve le prédicat d'égalité entre les variables modifiables par la spécification de l'opération locale et ces mêmes variables de l'implantation *sans contredire* la spécification de l'opération locale.

Description de l'obligation de preuve :

L'obligation de preuve définissant la correction d'une spécification d'opération locale, à démontrer pour chaque opération locale, contient les hypothèses suivantes :
- Contraintes des paramètres de la machine abstraite,
- Propriétés des constantes du développement vertical et des machines référencées dans les clauses **SEES** et **IMPORTS**,
- Invariants et assertions des machines référencées dans la clause **IMPORTS**, auxquels l'instanciation correspondante a été appliquée,
- Invariants et assertions des machines référencées dans la clause **SEES**,
- Typage B des variables concrètes de l'implantation,
- Précondition de la spécification de l'opération locale,
- Invariant implicite d'égalité des variables concrètes de l'implantation ainsi que des variables (abstraites et concrètes) des machines importées.

A ces hypothèses s'ajoutent les hypothèses implicites suivantes :
- Tout ensemble abstrait est défini comme une sous-partie non vide des entiers relatifs.
- Tout paramètre formel ensembliste est défini comme une sous-partie non vide des entiers relatifs.
- Tout ensemble énuméré est défini comme l'ensemble composé de tous ses éléments, et les éléments sont deux à deux distincts.

Sous ces hypothèses, le but à démontrer est :

> L'implémentation de l'opération locale appliquée à la négation de l'application de la spécification de l'opération locale sur la négation de l'invariant implicite d'égalité des variables de l'implantation, des machines importées et des paramètres de sortie de l'opération locale.

Nous présentons en annexe § C.7 la formulation mathématique de cette même obligation de preuve.

Annexes

Annexe A

Obligations de preuve des machines abstraites

Les machines suivantes introduisent les conventions de nommages que nous utiliserons pour décrire les obligations de preuve liées à la machine M1 :

MACHINE M1(X_1,x_1) **CONSTRAINTS** C_1 **SEES** Ms **SETS** S_1 ; $T_1 = \{a_1, b_1\}$ **ABSTRACT_CONSTANTS** ac_1 **CONCRETE_CONSTANTS** cc_1 **PROPERTIES** P_1 **INCLUDES** Mi1(N_{i_1}, n_{i_1}), Mi2(N_{i_2}, n_{i_2}) **USES** Mu **ABSTRACT_VARIABLES** av_1 **CONCRETE_VARIABLES** cv_1 **INVARIANT** $I_1 \wedge L_1(av_u, cv_u)$ **INITIALISATION** U_1 **ASSERTIONS** J_1 **OPERATIONS** $u_1 \leftarrow op_1(w_1) \mathrel{\widehat{=}}$ PRE Q_1 THEN V_1 END **END**	**MACHINE** Ms(X_s,x_s) **CONSTRAINTS** C_s **SETS** S_s ; $T_s = \{a_s, b_s\}$ **ABSTRACT_CONSTANTS** ac_s **CONCRETE_CONSTANTS** cc_s **PROPERTIES** P_s **ABSTRACT_VARIABLES** av_s **CONCRETE_VARIABLES** cv_s **INVARIANT** I_s **INITIALISATION** U_s **ASSERTIONS** J_s **OPERATIONS** $u_s \leftarrow op_s(w_s) \mathrel{\widehat{=}}$ PRE Q_s THEN V_s END **END**	**MACHINE** Mu(X_u,x_u) **CONSTRAINTS** C_u **SETS** S_u ; $T_u = \{a_u, b_u\}$ **ABSTRACT_CONSTANTS** ac_u **CONCRETE_CONSTANTS** cc_u **PROPERTIES** P_u **ABSTRACT_VARIABLES** av_u **CONCRETE_VARIABLES** cv_u **INVARIANT** I_u **INITIALISATION** U_u **ASSERTIONS** J_u **OPERATIONS** $u_u \leftarrow op_u(w_u) \mathrel{\widehat{=}}$ PRE Q_u THEN V_u END **END**

MACHINE	**MACHINE**
Mi1(X_{i_1},x_{i_1})	Mi2(X_{i_2},x_{i_2})
CONSTRAINTS	**CONSTRAINTS**
C_{i_1}	C_{i_2}
SETS	**SETS**
S_{i_1} ;	S_{i_2} ;
$T_{i_1} = \{a_{i_1}, b_{i_1}\}$	$T_{i_2} = \{a_{i_2}, b_{i_2}\}$
ABSTRACT_CONSTANTS	**ABSTRACT_CONSTANTS**
ac_{i_1}	ac_{i_2}
CONCRETE_CONSTANTS	**CONCRETE_CONSTANTS**
cc_{i_1}	cc_{i_2}
PROPERTIES	**PROPERTIES**
P_{i_1}	P_{i_2}
USES	
Mi2	
ABSTRACT_VARIABLES	**ABSTRACT_VARIABLES**
av_{i_1}	av_{i_2}
CONCRETE_VARIABLES	**CONCRETE_VARIABLES**
cv_{i_1}	cv_{i_2}
INVARIANT	**INVARIANT**
$I_{i_1} \wedge L_{i_1}(cv_{i_2},av_{i_2})$	I_{i_2}
INITIALISATION	**INITIALISATION**
U_{i_1}	U_{i_2}
ASSERTIONS	**ASSERTIONS**
J_{i_1}	J_{i_2}
OPERATIONS	**OPERATIONS**
$u_{i_1} \leftarrow op_{i_1}(w_{i_1}) \mathrel{\widehat{=}}$	$u_{i_2} \leftarrow op_{i_2}(w_{i_2}) \mathrel{\widehat{=}}$
PRE	PRE
Q_{i_1}	Q_{i_2}
THEN	THEN
V_{i_1}	V_{i_2}
END	END
END	**END**

Nous utilisons les notations suivantes :

$$A_1 \stackrel{\text{def}}{=} C_1 \ \wedge \ X_1 \in \mathbb{P}_1(INT)$$
$$B_1 \stackrel{\text{def}}{=} P_1 \ \wedge \ S_1 \in \mathbb{P}_1(INT) \ \wedge \ T_1 \in \mathbb{P}_1(INT) \ \wedge \ T_1 = \{a_1, b_1\} \ \wedge \ a_1 \neq b_1$$

De même pour les machines Mu, Ms, Mi1 et Mi2.

A.1 Inclusion dans une machine abstraite

Formule de l'obligation de preuve :

L'obligation de preuve ci-dessous est à démontrer pour chaque machine incluse (Mi1 et

Mi2), nous la présentons ici pour Mi1 :

$A_1 \wedge$ « *Contraintes des paramètres du composant* »

$A_u \wedge$ « *Contraintes des paramètres des composants utilisés* »

$B_1 \wedge$ « *Propriétés des constantes du composant* »

$B_u \wedge$ « *Propriétés des constantes des composants utilisés* »

B_s « *Propriétés des constantes des composants vus* »

$\Rightarrow$

$[X_{i_1}, x_{i_1} := N_{i_1}, n_{i_1}]C_{i_1}$ « *Contrainte instanciée de la machine incluse* »

A.2 Assertion dans une machine abstraite

L'assertion J_1 est une suite de prédicats que nous noterons J_{1_1}, J_{1_2}, $\cdots$, J_{1_k}.

Formule de l'obligation de preuve :
L'obligation de preuve ci-dessous est à démontrer pour chaque assertion de J_1 ; nous la présentons ici pour l'assertion J_{1_j} pour $1 \leq j \leq k$:

$A_1 \wedge$ « *Contraintes des paramètres du composant* »

$A_u \wedge$ « *Contraintes des paramètres des composants utilisés* »

$B_1 \wedge$ « *Propriétés des constantes du composant* »

$B_u \wedge$ « *Propriétés des constantes des composants utilisés* »

$B_s \wedge$ « *Propriétés des constantes des composants vus* »

$B_{i_1} \wedge B_{i_2} \wedge$ « *Propriétés des constantes des composants inclus* »

$[X_{i_1}, x_{i_1} := N_{i_1}, n_{i_1}](I_{i_1} \wedge L_{i_1} \wedge J_{i_1}) \wedge$ « *Invariants et assertions* »
$[X_{i_2}, x_{i_2} := N_{i_2}, n_{i_2}](I_{i_2} \wedge J_{i_2}) \wedge$ « *des composants inclus* »

$(I_u \wedge J_u) \wedge$ « *Invariants et assertions des composants utilisés* »

$(I_1 \wedge L_1) \wedge$ « *Invariant de la machine* »

$J_{1_1} \wedge \cdots \wedge J_{1_{j-1}}$ « *Assertions précédentes* »

$\Rightarrow$

J_{1_j} « *Assertion à prouver* »

A.3 Initialisation dans une machine abstraite

Formule mathématique de l'obligation de preuve :

$A_1 \wedge$ « *Contraintes des paramètres du composant* »

$A_u \wedge$ « *Contraintes des paramètres des composants utilisés* »

$B_1 \wedge$ « *Propriétés des constantes du composant* »

$B_u \wedge$ « *Propriétés des constantes des composants utilisés* »

$B_s \wedge$ « *Propriétés des constantes des composants vus* »

$B_{i_1} \wedge B_{i_2} \wedge$ « *Propriétés des constantes des composants inclus* »

$(I_u \wedge J_u) \wedge$ « *Invariants et assertions des composants utilisés* »

$[X_{i_1}, x_{i_1} := N_{i_1}, n_{i_1}](I_{i_1} \wedge J_{i_1}) \wedge$ « *Invariants et assertions* »
$[X_{i_2}, x_{i_2} := N_{i_2}, n_{i_2}](I_{i_2} \wedge J_{i_2}) \wedge$ « *des composants inclus* »

$(I_s \wedge J_s)$ « *Invariants et assertions des composants vus* »

$\Rightarrow$

« *Invariant après initialisations des machines incluses puis de la machine considérée* »
$[[X_{i_1}, x_{i_1} := N_{i_1}, n_{i_1}]U_{i_1} ; [X_{i_2}, x_{i_2} := N_{i_2}, n_{i_2}]U_{i_2} ; U_1](I_1 \wedge L_{i_1})$

A.4 Opérations dans une machine abstraite

Soit D_1 les prédicats de typage extraits de la définition des paramètres de sortie u_1.

Formule mathématique de l'obligation de preuve :

$A_1 \wedge$ « *Contraintes des paramètres du composant* »

$A_u \wedge$ « *Contraintes des paramètres des composants utilisés* »

$B_1 \wedge$ « *Propriétés des constantes du composant* »

$B_u \wedge$ « *Propriétés des constantes des composants utilisés* »

$B_s \wedge$ « *Propriétés des constantes des composants vus* »

$B_{i_1} \wedge B_{i_2} \wedge$ « *Propriétés des constantes des composants inclus* »

$(I_u \wedge J_u) \wedge$ « *Invariants et assertions des composants utilisés* »

$[X_{i_1}, x_{i_1} := N_{i_1}, n_{i_1}](I_{i_1} \wedge L_{i_1} \wedge J_{i_1}) \wedge$ « *Invariants et assertions* »
$[X_{i_2}, x_{i_2} := N_{i_2}, n_{i_2}](I_{i_2} \wedge J_{i_2}) \wedge$ « *des composants inclus* »

$(I_s \wedge J_s) \wedge$ « *Invariants et assertions des composants vus* »

$(I_1 \wedge L_1 \wedge J_1) \wedge$ « *Invariant et assertion de la machine* »

Q_1 « *Précondition de l'opération* »

$\Rightarrow$

$[V_1](I_1 \wedge L_{i_1} \wedge D_1)$ « *Opération appliquée à l'invariant et à la postcondition* »

Annexe B

Obligations de preuve des raffinements

Les machines suivantes introduisent les conventions de nommages que nous utiliserons pour décrire les obligations de preuve liées au raffinement Rn. Nous notons M1 la machine abstraite, et R2, ..., Rn-1 les raffinements antérieurs à Rn.

REFINEMENT	**MACHINE**	**MACHINE**
$\mathrm{Rn}(\mathrm{X}_1,\mathrm{x}_1)$	$\mathrm{M1}(\mathrm{X}_1,\mathrm{x}_1)$	$\mathrm{Ms}(\mathrm{X}_s,\mathrm{x}_s)$
REFINES		
Rn-1		
	CONSTRAINTS	**CONSTRAINTS**
	C_1	C_s
SEES		
Ms		
SETS	**SETS**	**SETS**
S_n ;	S_1 ;	S_s ;
$\mathrm{T}_n = \{a_n, b_n\}$	$\mathrm{T}_1 = \{a_1, b_1\}$	$\mathrm{T}_s = \{a_s, b_s\}$
ABSTRACT_CONSTANTS	**ABSTRACT_CONSTANTS**	**ABSTRACT_CONSTANTS**
ac_n	ac_1	ac_s
CONCRETE_CONSTANTS	**CONCRETE_CONSTANTS**	**CONCRETE_CONSTANTS**
cc_n	cc_1	cc_s
PROPERTIES	**PROPERTIES**	**PROPERTIES**
P_n	P_1	P_s
INCLUDES	**INCLUDES**	
$\mathrm{Mi1}(\mathrm{N}_{i_1}, \mathrm{n}_{i_1})$,	Minc_1	
$\mathrm{Mi2}(\mathrm{N}_{i_2}, \mathrm{n}_{i_2})$		
ABSTRACT_VARIABLES	**ABSTRACT_VARIABLES**	**ABSTRACT_VARIABLES**
av_n	av_1	av_s
CONCRETE_VARIABLES	**CONCRETE_VARIABLES**	**CONCRETE_VARIABLES**
cv_n	cv_1	cv_s
INVARIANT	**INVARIANT**	**INVARIANT**
I_n	I_1	I_s
INITIALISATION	**INITIALISATION**	**INITIALISATION**
U_n	U_1	U_s
ASSERTIONS	**ASSERTIONS**	**ASSERTIONS**
J_n	J_1	J_s
OPERATIONS	**OPERATIONS**	**OPERATIONS**
$\mathrm{u}_1 \leftarrow \mathrm{op}_1(\mathrm{w}_1) \mathrel{\widehat{=}}$	$\mathrm{u}_1 \leftarrow \mathrm{op}_1(\mathrm{w}_1) \mathrel{\widehat{=}}$	$\mathrm{u}_s \leftarrow \mathrm{op}_s(\mathrm{w}_s) \mathrel{\widehat{=}}$
PRE	PRE	PRE
Q_n	Q_1	Q_s
THEN	THEN	THEN
V_n	V_1	V_s
END	END	END
END	**END**	**END**

Les composants M1, R2, ..., Rn-1 font respectivement une inclusion des machines Minc_1, Minc_2, ..., Minc_{n-1}. Nous ne faisons pas ici de description de ces machines par souci de concision. Par contre, nous décrivons les machines $\mathrm{M1inc}_n$ et $\mathrm{M2inc}_n$ incluses par le raffinement Rn (notamment à cause de la partie L_{i_1} de l'invariant de $\mathrm{M1inc}_n$).

La machine abstraite M1 ci-dessus est différente de la machine M1 de l'appendice A car elle ne contient pas de clause **USES**.

MACHINE	**MACHINE**
Mi1(X_{i_1},x_{i_1})	Mi2(X_{i_2},x_{i_2})
CONSTRAINTS	**CONSTRAINTS**
C_{i_1}	C_{i_2}
SETS	**SETS**
S_{i_1} ;	S_{i_2} ;
$T_{i_1} = \{a_{i_1}, b_{i_1}\}$	$T_{i_2} = \{a_{i_2}, b_{i_2}\}$
ABSTRACT_CONSTANTS	**ABSTRACT_CONSTANTS**
ac_{i_1}	ac_{i_2}
CONCRETE_CONSTANTS	**CONCRETE_CONSTANTS**
cc_{i_1}	cc_{i_2}
PROPERTIES	**PROPERTIES**
P_{i_1}	P_{i_2}
USES	
Mi2	
ABSTRACT_VARIABLES	**ABSTRACT_VARIABLES**
av_{i_1}	av_{i_2}
CONCRETE_VARIABLES	**CONCRETE_VARIABLES**
cv_{i_1}	cv_{i_2}
INVARIANT	**INVARIANT**
$I_{i_1} \wedge L_{i_1}(v_{i_2})$	I_{i_2}
INITIALISATION	**INITIALISATION**
U_{i_1}	U_{i_2}
ASSERTIONS	**ASSERTIONS**
J_{i_1}	J_{i_2}
OPERATIONS	**OPERATIONS**
$u_{i_1} \leftarrow op_{i_1}(w_{i_1}) \mathrel{\widehat{=}}$	$u_{i_2} \leftarrow op_{i_2}(w_{i_2}) \mathrel{\widehat{=}}$
PRE	PRE
Q_{i_1}	Q_{i_2}
THEN	THEN
V_{i_1}	V_{i_2}
END	END
END	**END**

Nous utilisons les notations suivantes :

« *Propriétés explicites et implicites de la machine* »

$$B_1 \stackrel{\text{def}}{=} B_{inc1} \wedge P_1 \wedge S_1 \in \mathbb{P}_1(INT) \wedge T_1 \in \mathbb{P}_1(INT) \wedge T_1 = \{a_1, b_1\} \wedge a_1 \neq b_1$$

$$\vdots$$

« *Propriétés explicites et implicites du raffinement antérieur* »

$$B_{n-1} \stackrel{\text{def}}{=} B_{incn-1} \wedge P_{n-1} \wedge S_{n-1} \in \mathbb{P}_1(INT) \wedge T_{n-1} \in \mathbb{P}_1(INT) \wedge T_{n-1} = \{a_{n-1}, b_{n-1}\} \wedge a_{n-1} \neq b_{n-1}$$

« *Propriétés explicites et implicites du raffinement considéré* »

$$B_n \stackrel{\text{def}}{=} P_n \wedge S_n \in \mathbb{P}_1(INT) \wedge T_n \in \mathbb{P}_1(INT) \wedge T_n = \{a_n, b_n\} \wedge a_n \neq b_n$$

De plus, dans les formules des obligations de preuve présentées ci-dessous les invariants $I_1, \ldots, I_{n-1}$ sont constitués de la conjonction de l'invariant du composant correspondant (M1 ... Rn-1) et des éventuels invariants inclus instanciés.

B.1 Inclusion dans un raffinement

Formule de l'obligation de preuve :

L'obligation de preuve ci-dessous est à démontrer pour chaque machine incluse (Mi1 et Mi2), nous la présentons ici

pour Mi1 :

$A_1 \wedge$ « *Contraintes des paramètres de la machine* »

$B_1 \wedge \ldots \wedge B_{n-1} \wedge$ « *Propriétés des constantes des raffinements antérieurs* »

$B_n \wedge$ « *Propriétés des constantes du raffinement* »

B_s « *Propriétés des constantes des composants vus* »

$\Rightarrow$

$[X_{i_1}, x_{i_1} := N_{i_1}, n_{i_1}]C_{i_1}$ « *Contrainte instanciée* »

B.2 Assertion dans un raffinement

L'assertion J_n est une suite de prédicats que nous noterons J_{n_1}, J_{n_2}, $\cdots$, J_{n_k}.

Formule de l'obligation de preuve :
L'obligation de preuve ci-dessous est à démontrer pour chaque assertion de J_n :

$A_1 \wedge$ « *Contraintes des paramètres de la machine* »

$B_1 \wedge \ldots \wedge B_{n-1} \wedge$ « *Propriétés des constantes des raffinements antérieurs* »

$B_n \wedge$ « *Propriétés des constantes du raffinement* »

$B_s \wedge$ « *Propriétés des constantes des composants vus* »

$B_{i_1} \wedge B_{i_2} \wedge$ « *Propriétés des constantes des composants inclus* »

$[X_{i_1}, x_{i_1} := N_{i_1}, n_{i_1}](I_{i_1} \wedge L_{i_1} \wedge J_{i_1}) \wedge$ « *Invariants et assertions* »
$[X_{i_2}, x_{i_2} := N_{i_2}, n_{i_2}](I_{i_2} \wedge J_{i_2}) \wedge$ « *des composants inclus* »

$(I_1 \wedge J_1) \wedge \ldots \wedge (I_{n-1} \wedge J_{n-1}) \wedge$ « *Invariants et assertions des raffinements antérieurs* »

$I_n \wedge$ « *Invariant du raffinement* »

$J_{n_1} \wedge \cdots \wedge J_{n_{j-1}}$ « *Assertions précédentes* »

$\Rightarrow$

J_{n_j} « *Assertions à prouver* »

B.3 Initialisation dans un raffinement

Formule mathématique de l'obligation de preuve :

$A_1 \wedge$ « *Contraintes des paramètres de la machine* »

$B_1 \wedge \ldots \wedge B_{n-1} \wedge$ « *Propriétés des constantes des raffinements antérieurs* »

$B_n \wedge$ « *Propriétés des constantes du raffinement* »

$B_s \wedge$ « *Propriétés des constantes des composants vus* »

$B_{i_1} \wedge B_{i_2} \wedge$ « *Propriétés des constantes des composants inclus* »

$[X_{i_1}, x_{i_1} := N_{i_1}, n_{i_1}](I_{i_1} \wedge L_{i_1} \wedge J_{i_1}) \wedge$ « *Invariants et assertions* »
$[X_{i_2}, x_{i_2} := N_{i_2}, n_{i_2}](I_{i_2} \wedge J_{i_2}) \wedge$ « *des composants inclus* »

$I_s \wedge J_s$ « *Invariants et assertions des composants vus* »

$\Rightarrow$

« *Initialisation des composants inclus puis du raffinement appliquées à la négation de* »
« *l'application de l'initialisation raffinée appliquée à la négation de l'invariant* »
$[[X_{i_1}, x_{i_1} := N_{i_1}, n_{i_1}]U_{i_1} ; [X_{i_2}, x_{i_2} := N_{i_2}, n_{i_2}]U_{i_2} ; U_n] \neg [U_{n-1}] \neg (I_n \wedge L_{i_1})$

B.4 Opérations dans un raffinement

Formule mathématique de l'obligation de preuve :

$A_1 \wedge$ « *Contraintes des paramètres de la machine* »

$B_1 \wedge \ldots \wedge B_{n-1} \wedge$ « *Propriétés des constantes des raffinements antérieurs* »

$B_n \wedge$ « *Propriétés des constantes du raffinement* »

$B_s \wedge$ « *Propriétés des constantes des composants vus* »

$B_{i_1} \wedge B_{i_2} \wedge$ « *Propriétés des constantes des composants inclus* »

$[X_{i_1}, x_{i_1} := N_{i_1}, n_{i_1}](I_{i_1} \wedge L_{i_1} \wedge J_{i_1}) \wedge$ « *Invariants et assertions* »
$[X_{i_2}, x_{i_2} := N_{i_2}, n_{i_2}](I_{i_2} \wedge J_{i_2}) \wedge$ « *des composants inclus* »

$I_s \wedge J_s \wedge$ « *Invariants et assertions des composants vus* »

$(I_1 \wedge J_1) \wedge \ldots \wedge (I_n \wedge J_n) \wedge$ « *Invariants et assertions du développement vertical* »

Q_1 « *Précondition de l'opération abstraite* »

$\Rightarrow$

$Q_n \wedge$ « *Précondition de l'opération du raffinement* »

« *Opération du raffinement appliquée à la négation de l'opération spécifiée appliquée à la négation de l'invariant* »
$[[u_1 := u_1']V_n] \neg [V_{n-1}] \neg (I_n \wedge u_1 = u_1')$

Remarque sur cette obligation de preuve : afin de vérifier que le résultat de l'opération du raffinement est le même que le résultat de l'opération du raffinement antérieur un prédicat $u_1 = u_1'$ est ajouté à l'invariant I_n. u_1 va prendre pour valeur le résultat de l'opération V_{n-1} ; à l'aide de la substitution $[u_1 := u_1']$ appliquée à V_n, u_1' va prendre pour valeur le résultat de l'opération V_n.

Annexe C

Obligations de preuve des implantations

Les machines suivantes introduisent les conventions de nommages que nous utiliserons pour décrire les obligations de preuve liées à l'implantation Mn. Nous notons M1 la machine abstraite, et R2, ..., Rn-1 les raffinements antérieurs à Mn. Ces composants font respectivement une inclusion des machines Minc1, Minc2, ..., Mincn-1.

IMPLEMENTATION $\mathrm{Mn}(\mathrm{X}_1,\mathrm{x}_1)$ **REFINES** R_{n-1} **SEES** Ms **SETS** S_n ; $\mathrm{T}_n = \{a_n, b_n\}$ **CONCRETE_CONSTANTS** cc_n **PROPERTIES** P_n **VALUES** $\mathrm{S}_1 = \mathrm{E}_1$; . . . $\mathrm{S}_n = \mathrm{E}_n$; $\mathrm{cc}_1 = \mathrm{d}_1$; . . . $\mathrm{cc}_n = \mathrm{d}_n$ **IMPORTS** $\mathrm{Mi}(\mathrm{N}_i,\ \mathrm{n}_i)$ **CONCRETE_VARIABLES** cv_n **INVARIANT** I_n **INITIALISATION** U_n **ASSERTIONS** J_n **LOCAL_OPERATIONS** $\mathrm{u}_{l1} \leftarrow \mathrm{op}_{l1}(\mathrm{w}_{l1}) \mathrel{\widehat{=}}$ PRE Q_{l1} THEN V_{l1} END **OPERATIONS** $\mathrm{u}_{l1} \leftarrow \mathrm{op}_{l1}(\mathrm{w}_{l1}) \mathrel{\widehat{=}}$ V_{l2} ; $\mathrm{u}_1 \leftarrow \mathrm{op}_1(\mathrm{w}_1) \mathrel{\widehat{=}}$ V_n **END**	**MACHINE** $\mathrm{M1}(\mathrm{X}_1,\mathrm{x}_1)$ **CONSTRAINTS** C_1 **SETS** S_1 ; $\mathrm{T}_1 = \{a_1, b_1\}$ **ABSTRACT_CONSTANTS** ac_1 **CONCRETE_CONSTANTS** cc_1 **PROPERTIES** P_1 **INCLUDES** $\mathrm{Minc1}$ **ABSTRACT_VARIABLES** av_1 **CONCRETE_VARIABLES** cv_1 **INVARIANT** I_1 **INITIALISATION** U_1 **ASSERTIONS** J_1 **OPERATIONS** $\mathrm{u}_1 \leftarrow \mathrm{op}_1(\mathrm{w}_1) \mathrel{\widehat{=}}$ PRE Q_1 THEN V_1 END **END**	**MACHINE** $\mathrm{Ms}(\mathrm{X}_s,\mathrm{x}_s)$ **CONSTRAINTS** C_s **SETS** S_s ; $\mathrm{T}_s = \{a_s, b_s\}$ **ABSTRACT_CONSTANTS** ac_s **CONCRETE_CONSTANTS** cc_s **PROPERTIES** P_s **ABSTRACT_VARIABLES** av_s **CONCRETE_VARIABLES** cv_s **INVARIANT** I_s **INITIALISATION** U_s **ASSERTIONS** J_s **OPERATIONS** $\mathrm{u}_s \leftarrow \mathrm{op}_s(\mathrm{w}_s) \mathrel{\widehat{=}}$ PRE Q_s THEN V_s END **END**	**MACHINE** $\mathrm{Mi}(\mathrm{X}_i,\mathrm{x}_i)$ **CONSTRAINTS** C_i **SETS** S_i ; $\mathrm{T}_i = \{a_i, b_i\}$ **ABSTRACT_CONSTANTS** ac_i **CONCRETE_CONSTANTS** cc_i **PROPERTIES** P_i **ABSTRACT_VARIABLES** av_i **CONCRETE_VARIABLES** cv_i **INVARIANT** I_i **INITIALISATION** U_i **ASSERTIONS** J_i **OPERATIONS** $\mathrm{u}_i \leftarrow \mathrm{op}_i(\mathrm{w}_i) \mathrel{\widehat{=}}$ PRE Q_i THEN V_i END **END**

Dans cette section, nous utilisons les notations suivantes :

« *Propriétés explicites et implicites de la machine* »

$$B_1 \overset{\text{def}}{=} B_{inc1} \wedge \mathrm{P}_1 \wedge S_1 \in \mathbb{P}_1(INT) \wedge T_1 \in \mathbb{P}_1(INT) \wedge T_1 = \{a_1, b_1\} \wedge a_1 \neq b_1$$

$$\vdots$$

« *Propriétés explicites et implicites du composant raffiné* »

$$B_{n-1} \overset{\text{def}}{=} B_{incn-1} \wedge \mathrm{P}_{n-1} \wedge S_{n-1} \in \mathbb{P}_1(INT) \wedge T_{n-1} \in \mathbb{P}_1(INT) \wedge T_{n-1} = \{a_{n-1}, b_{n-1}\} \wedge a_{n-1} \neq b_{n-1}$$

« *Propriétés explicites et implicites de l'implantation* »

$$B_n \overset{\text{def}}{=} \mathrm{P}_n \wedge S_n \in \mathbb{P}_1(INT) \wedge T_n \in \mathbb{P}_1(INT) \wedge T_n = \{a_n, b_n\} \wedge a_n \neq b_n$$

C.1 Importation dans une implantation

Formule de l'obligation de preuve :

$A_1 \wedge$ « *Contraintes des paramètres de la machine* »

$B_1 \wedge \ldots \wedge B_n \wedge$ « *Propriétés du développement vertical* »

B_s « *Propriétés des constantes des composants vus* »

$\Rightarrow$

$[X_i, x_i := N_i, n_i]C_i$ « *Contrainte instanciée de la machine importée* »

C.2 Valuation dans une implantation

Formule de l'obligation de preuve :

$B_s \wedge$ « *Propriétés des constantes des composants vus* »

B_i « *Propriétés des constantes des composants importés* »

$\Rightarrow$

« *Valuation des constantes appliquée aux propriétés* »
$\exists(ac_1, \ldots, ac_{n-1})\ .\ [S_1 := E_1; cc_1 := d_1; \ldots; S_n := E_n; cc_n := d_n](B_1 \wedge \cdots \wedge B_n)$

C.3 Assertion dans une implantation

L'assertion J_n est une suite de prédicats que nous noterons J_{n_1}, J_{n_2}, $\cdots$, J_{n_k}.

Formule de l'obligation de preuve :
L'obligation de preuve ci-dessous est à démontrer pour chaque assertion de J_n :

$A_1 \wedge$ « *Contraintes des paramètres de la machine* »

$B_1 \wedge \ldots \wedge B_n \wedge$ « *Propriétés du développement vertical* »

$B_s \wedge$ « *Propriétés des constantes des composants vus* »

$B_i \wedge$ « *Propriétés des constantes des composants importés* »

$[X_i, x_i := N_i, n_i](I_i \wedge J_i) \wedge$ « *Invariants et assertions des composants importés* »

$(I_1 \wedge J_1) \wedge \ldots \wedge (I_{n-1} \wedge J_{n-1}) \wedge$ « *Invariants et assertions des raffinements antérieurs* »

$I_n \wedge$ « *Invariant de l'implantation* »

$J_{n_1} \wedge \cdots \wedge J_{n_{j-1}}$ « *Assertions précédentes* »

$\Rightarrow$

J_{n_j} « *Assertion à prouver* »

C.4 Initialisation dans une implantation

Formule mathématique de l'obligation de preuve :

$A_1 \wedge$ « *Contraintes des paramètres de la machine* »

$B_1 \wedge \ldots \wedge B_n \wedge$ « *Propriétés du développement vertical* »

$B_s \wedge$ « *Propriétés des constantes des composants vus* »

$B_i \wedge$ « *Propriétés des constantes des composants importés* »

$[X_i, x_i := N_i, n_i](I_i \wedge J_i) \wedge$ « *Invariants et assertions des composants importés* »

$I_s \wedge J_s$ « *Invariants et assertions des composants vus* »

$\Rightarrow$

« *Initialisations des composants importées puis initialisation de l'implantation appliquées à la négation de l'initialisation spécifiée appliquée à l'invariant* »
$[[X_i, x_i := N_i, n_i]U_i ; [X_{i_2}, x_{i_2} := N_{i_2}, n_{i_2}]U_{i_2} ; U_n] \neg [U_{n-1}] \neg I_n$

C.5 Opérations dans une implantation

Formule mathématique de l'obligation de preuve :

$A_1 \wedge$ « *Contraintes des paramètres de la machine* »

$B_1 \wedge \ldots \wedge B_n \wedge$ « *Propriétés du développement vertical* »

$B_s \wedge$ « *Propriétés des constantes des composants vus* »

$B_i \wedge$ « *Propriétés des constantes des composants importés* »

$I_s \wedge J_s \wedge$ « *Invariants et assertions des composants vus* »

$[X_i, x_i := N_i, n_i](I_i \wedge J_i) \wedge$ « *Invariants et assertions des composants importés* »
$(I_1 \wedge J_1) \wedge \ldots \wedge (I_n \wedge J_n) \wedge$ « *Invariants et assertions du développement vertical* »

Q_1 « *Précondition de l'opération abstraite* »

$\Rightarrow$

« *Opération de l'implantation appliquée à la négation de l'opération spécifiée appliquée à la négation de l'invariant de liaison* »
$[[u_1 := u_1'] V_n] \neg [V_{n-1}] \neg (I_n \wedge u_1 = u_1')$

C.6 Spécification d'opération locale dans une implantation

Soit D_{l1} les prédicats de typage extraits de la définition des paramètres de sortie u_{l1}.

Formule mathématique de l'obligation de preuve :

$A_1 \wedge$ « *Contraintes des paramètres de la machine* »

$B_1 \wedge \ldots \wedge B_n \wedge$ « *Propriétés du développement vertical* »

$B_s \wedge$ « *Propriétés des constantes des composants vus* »

$B_i \wedge$ « *Propriétés des constantes des composants importés* »

$I_s \wedge J_s \wedge$ « *Invariants et assertions des composants vus* »

$[X_i, x_i := N_i, n_i](I_i \wedge J_i) \wedge$ « *Invariants et assertions des composants importés* »

$TypageB(vc_n) \wedge$ « *Typage B des variables concrètes de l'implantation* »

Q_{l1} « *Précondition de la spécification de l'opération locale* »

$\Rightarrow$

« *Opération locale appliquée aux invariants des machines importées et à la postcondition* »
$[V_{l1}][X_i, x_i := N_i, n_i](I_i \wedge D_{l1})$

C.7 Implémentation d'opération locale dans une implantation

Formule mathématique de l'obligation de preuve :

$A_1 \wedge$ « *Contraintes des paramètres de la machine* »

$B_1 \wedge \ldots \wedge B_n \wedge$ « *Propriétés du développement vertical* »

$B_s \wedge$ « *Propriétés des constantes des composants vus* »

$B_i \wedge$ « *Propriétés des constantes des composants importés* »

$I_s \wedge J_s \wedge$ « *Invariants et assertions des composants vus* »

$[X_i, x_i := N_i, n_i](I_i \wedge J_i) \wedge$ « *Invariants et assertions des composants importés* »

$TypageB(vc_n) \wedge$ « *Typage B des variables concrètes de l'implantation* »

$cv_n = cv'_n \wedge$ « *Invariant implicite d'égalité des variables concrètes de l'implantation* »
$av_i = av'_i \wedge$ « *Invariant implicite d'égalité des variables abstraites des machines importées* »
$cv_i = cv'_i \wedge$ « *Invariant implicite d'égalité des variables concrètes des machines importées* »

Q_{l1} « *Précondition de la spécification de l'opération locale* »

$\Rightarrow$

« *Implémentation de l'opération locale appliquée à la négation de la spécification de l'opération locale appliquée à la négation de l'invariant d'égalité des variables de l'implantation, des machines importées et des paramètres de sortie de l'opération locale.* »
$[[u_{l1} := u'_{l1}]V_{l2}]\neg\,[V_{l1}]\neg\,(cv_n = cv'_n \wedge av_i = av'_i \wedge cv_i = cv'_i \wedge u_{l1} = u'_{l1})$