

Initiation au B

mardi 15 mars 2016

14:28

Idee : → spécification fonctionnelle d'un **logiciel**

Éléments du logiciel à spécifier

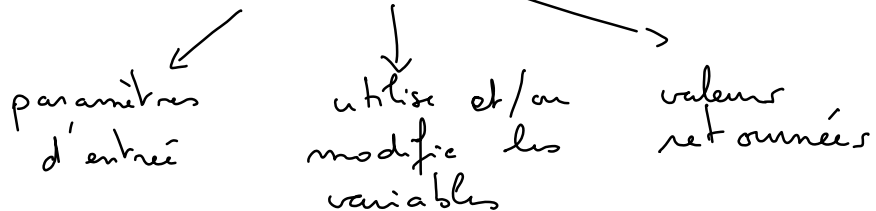
→ Contexte : constantes

propriétés vérifiées par ces
constantes (type, domaine)

→ État : variables

propriétés vérifiées par ces
variables (appelé **d'invariant**)

→ Opérations : fonctions que l'on peut
utiliser



Langage des propriétés : logique du premier
ordre sur les ensembles

Types primitifs : booléens $(TRUE/FALSE) \neq T, \perp$
 \mathbb{Z}
 \mathbb{N}

Types construits : - couples (avec, ou \mapsto)
- produit cartésien (ensemble de tous les couples)

- ensembles \rightarrow par extension
- \rightarrow par compréhension

Les structures de données usuelles sont définies à l'aide de ces primitives (comme en théorie des ensembles)

- relation = $\{ (x, y) \mid x R y \}$
- fonction = $\{ (x, y) \mid f(x) = y \}$
(une fonction est une relation)
 - partielle / totale
 - injective / bijective / surjective
- séquences (listes) = $\{ (n, x) \mid \begin{array}{l} n \in 1..|u| \\ u_n = x \end{array} \}$
- arbres = $\{ \text{chemin de } \mathcal{T} \text{ us comme une séquence} \}$

Opérations : chaque opération est décrite par :

- une **précondition** sur les paramètres et les variables
 \hookrightarrow l'opération ne doit être appelée que si cette précondition est vérifiée
- une substitution à effectuer
 (variables et valeurs de retour)
 devenant telles que

(formule logique des paramètres,

nouvelles et anciennes valeurs
des variables, valeurs de retour)

Le langage B définit de nombreuses constructions qui sont du sucre syntaxique exprimable à l'aide des éléments ci-dessus

Obligation de preuve

Le logiciel derrière le langage B gère
des propriétés logiques à prouver :

→ que les opérations, utilisées alors que la précondition est vraie, conservent l'invariant

→ que les constructions utilisées sont bien fondées

	$f(x)$: f est une fonction $x \in \text{dom } f$
	$x = \max E$: E a un majorant et est non vide

Raffinement

Afin d'arriver à une implémentation du logiciel, on effectue des raffinements.

↳ on précise de plus en plus comment sont effectuées les opérations

↳ on représente l'état par des variables plus proches de la machine réelle

1) 0 0 ... 0 auf der Ableitung.

↳ le roqier gemme est corrigé
de preuve qui montrent que le
raffinement respecte la spécification