Idée : → spécification fonctionnelle d'un logiciel

Éléments du logiciel à spécifier

→ Contexte : constantes

propriétés vérifiées par ces constantes (type, domaine)

→ Etat : variables

propriétés vérifiées par ces variables (appelé l'invariant)

→ Opérations : fonctions que l'on peut utiliser

- paramètres d'entrée
- utilise et/ou modifie les variables
- valeurs retournées

Langage des propriétés : logique du premier ordre sur les ensembles

Types primitifs : booléens (TRUE/FALSE) ≠ T, ⊥

$\mathbb{Z}$

$\mathbb{N}$

Types construits : - Couples (avec , ou $\mapsto$)

- produit cartésien (ensemble de tous les couples)

- ensembles → par extension

→ par compréhension

Les structures de donnée 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) = $\{(n,x) \mid \begin{matrix} n \in 1..|u| \\ u_n = x \end{matrix}\}$
- arbres = { chemin de T vus comme une séquence }

Opérations : chaque opération est décrite par :

- une précondition sur les paramètres et les variables
  ↳ l'opération ne doit être appelée que si cette précondition est vérifiée
- une substitution à effectuer

(variables et valeurs de retour)

deviennent 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 dérivé le langage B génère 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 $x \in \mathrm{dom}\, f$
- $x = \max E$ : $E$ 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

↳ le logiciel génère des obligations de preuve qui montrent que le raffinement respecte la spécification