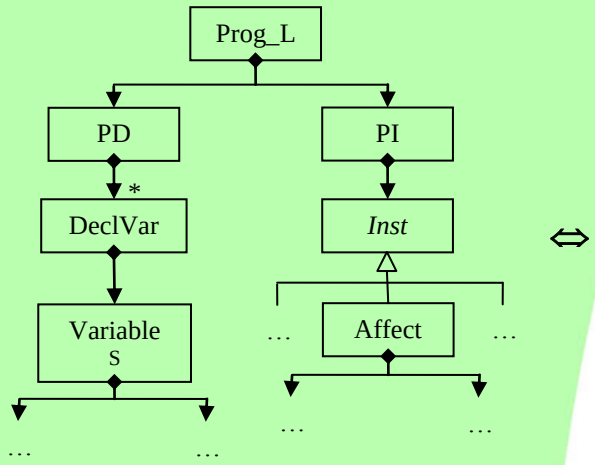
Modélisation en UML/OCL des langages et Processus de développement IDM

T.-T. Le Thi

Modélisation en UML de la grammaire d'un langage (de type procédural) :



⇔

Prog_L → PD, PI
PD → DeclVar*
DeclVar → Variable, Type_L
Variable → S
Type_L → Rel | Bool
Rel → Z
Bool → True | False
True →
False →
PI → Inst
Inst → Skip | Seq | Affect | Test | Boucle
Skip →
Affect → Exp, Inst
…
Exp → Variable | Const | Null | ExpBin
…

Modélisation des propriétés de typage :

Exp ::type() : Type_L

Null ::type() : Type_L = if self.exp.type().oclIsTypeOf(Rel)
then Bool()
else oclUndefined(Bool)
endif

…

Inst::typeOk() : Boolean

Affect::typeOk() : Boolean = (self.variable.type().oclIsTypeOf(Rel) and
self.inst.type().oclIsTypeOf(Rel)) xor
(self.variable.type().oclIsTypeOf(Bool) and
self.inst.type().oclIsTypeOf(Bool))

…

Modélisation des propriétés comportementales :

Environnement d'exécution :

EnvExec → VarVal*
VarVar → Var, ValSem
Var → S
ValSem → Entier | Booleen
Entier → Z
Booleen → B

Fonctions sémantiques :

Exp::eval(env : EnvExec) : ValSem
…
Inst::exec(env : Envexec) : EnvExec

Modélisation du triplet de Hoare ({ P } Inst { Q }) et Modélisation des propriétés axiomatiques :

Une interprétation comportementale du triplet de Hoare :

Inst ::tripletHoare(env : EnvExec) : Boolean = self.p.eval(env).b implies self.q.eval(self.exec(env)).b

Interprétation statique du triplet de Hoare (Calcul de l'assertion associée et appel à un Atelier B) :

Algorithme de la plus faible pré-condition (Séquence d'instructions)
Algorithme de substitution (Instruction d'affectation)
Correction partielle et totale (Instruction de boucle)

Application à la modélisation de propriétés comportementales et axiomatiques de diagrammes UML, en particulier les diagrammes d'activité, et à la vérification de la cohérence entre les activités de modélisation des exigences et de génération des codes issus des modèles.

T.T. Le Thi - IRIT / UPS Toulouse (lethi@irit.fr 05 61 55 67 82)