Architecture de l'étape B

Equipe GL17

16 janvier 2017

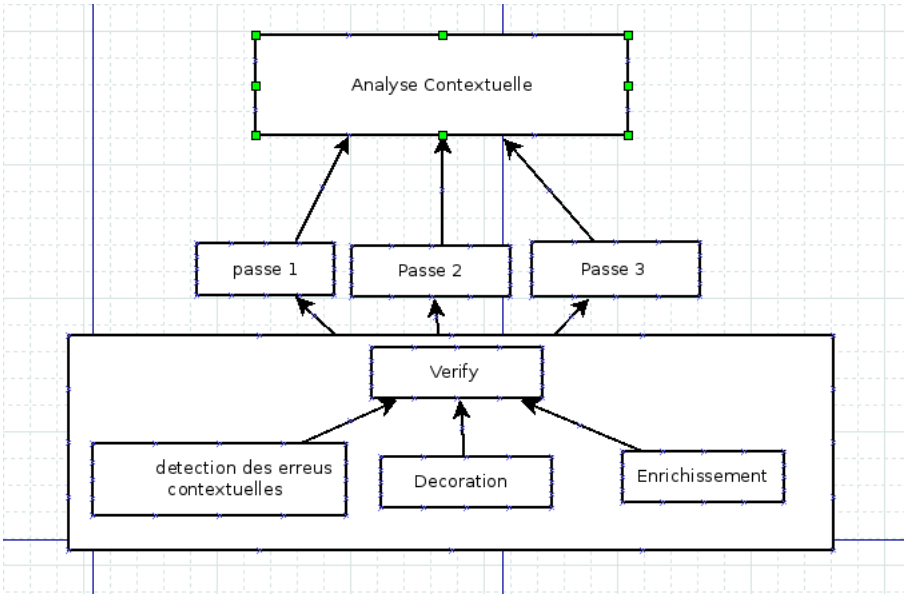
1 Etape B :Analyse contextuele

Les répertoires concernés sont :

+src/main/java/fr/ensimag/deca/context/*
+src/main/java/fr/ensimag/deca/tree/*
+src/main/java/fr/ensimag/deca/context/tools/symbolTable.java

1.1 Description :

Cette partie est censée décorer l'arbre abstrait généré par l'analyse syntaxique. Elle utilise pour cela la grammaire attibuée fournie dans la spécification. Elle effectue également la détection d'erreurs contextuelles, ainsi que l'enrichissement de l'arbre. Cela se fait en trois passes. Il est à noter que pour le langage sans-objet, on n'aura besoin que de la troisième passe.



1.2 Description de l'implémentation de chaque fonctionnalité :

+ Verify : on vérifie les expressions contextuellement à l'aide des fonctions verify que nous avons implémentées dans les classes .java du dossier tree. Les fonctions utilisées sont les suivantes : verifyExpr,verifyRvalue,verifyCondition. Ces dernières utilisent l'environnmentexp et environnementType(champ du paramètre compiler de type Decac-Compiler) pour effectuer leur travail.

+ EnvrionnmentType :une classe qu'on a créée dans java/.../context qui contient un dictionnaire qui associe à chaque Symbol(classe declarée dans tools/symboleTable.java) une TypeDefinition. Pour cette classe on implémente toutes les fonctions nécessaires pour l'intialisation de l'environnment prédefini, l'ajout d'un nouveau type(sera utilisé dans la partie avec objet), récuperation d'une définition d'un type qui existe déjà dans l'environnement.

+ Détection des erreurs : on la gère dans les fonctions verify. En faisant des tests sur les types des fils d'un neoud, on lève des erreurs contextuelles avec un message et la position de l'erreur dans le cas où ces types ne sont pas compatibles avec la définition de la règle en question.

+ Décoration et enrichissement : tous les paramètres nécessaires pour ces fonctionnalités sont disponibles après l'exécution du programme verifyProgram dans DecacCompiler.docompile(),car on intialise le type (ou la définition parfois si cela est nécessaire) à la fin de chaque traitement d'un noeud (fin de la fonction verify(Expr || Rvalue || Condition).