

L3 informatique, 2124-2025, Projet de LFC

Création d'un langage de script pour un système multi-agents

Corrigé de l'étape 3 et ETAPE 4, à rendre 1 semaine après votre TP3 (dates à la fin de ce fichier)

Corrigé de l'étape 3

La grammaire décrivant la syntaxe des fichiers d'entrée est la suivante :

Symboles terminaux (V_T)

ENV (Environnement)	CO (Crochet ouvrant)
NTYPA (NewTypeAgent)	CF (Crochet fermant)
NAG (NewAgent)	VG (Virgule)
NCT (NewContexte)	AO (Accolade ouvrante)
TINT (Type d'attribut int)	AF (Accolade fermante)
TDOUBL (Type d'attribut double)	DPT (Deux points)
TCAR (Type d'attribut caractère)	INT (Nombre entier)
TCH (Type d'attribut chaîne)	EG (Egal)
TBOOL (Type d'attribut booléen)	REEL (Nombre réel)
BOOL (Booléen)	CH (Chaîne de caractères)
IDF (Identificateur)	CAR (Caractère)

Symboles non terminaux (V_N)

- Program
- decla_env (déclaration d'environnement)
- suite_prog (suite du programme, après déclaration de l'environnement)
- instruction
- new_typ_agent (déclaration d'un nouveau type d'agent)
- liste_decla_attributs (liste de déclarations d'attributs)
- decla_attribut (déclaration d'un attribut)
- type_attribut
- new_agent (déclaration d'un nouvel agent)
- liste_affect_attributs (liste d'instanciations d'attributs)
- affect_attribut (instanciation d'un attribut)
- valeur_attribut
- new_context (déclaration d'un nouveau contexte)

Axiome

program

Règles de production

program \rightarrow decla_env suite_prog

decla_env \rightarrow ENV IDF CO INT VG INT CF

suite_prog \rightarrow instruction | instruction suite_prog

instruction \rightarrow new_typ_agent | new_agent | new_context

new_typ_agent \rightarrow NTYPA IDF AO liste_decla_attributs AF

liste_decla_attributs \rightarrow decla_attribut | decla_attribut VG liste_decla_attributs

decla_attribut → IDF DPT type_attribut

type_attribut → TINT | TDOUBL | TCAR | TCH | TBOOL

new_agent → NAG IDF DPT IDF CO INT VG INT CF AO liste_affect_attributs AF

liste_affect_attributs → affect_attribut | affect_attribut VG liste_affect_attributs

affect_attribut → IDF EG valeur_attribut

valeur_attribut → INT | REEL | CH | CAR | BOOL

new_context → NCT IDF CO INT CF | NCT IDF CO INT VG INT VG INT VG INT CF

Etape 4

Vous devez maintenant écrire le fichier YACC qui correspond à cette grammaire et modifier le fichier LEX afin de relier LEX et YACC, et ainsi obtenir un programme qui effectue à la fois les analyses lexicale et syntaxique. Si vous n'avez pas terminé la gestion de la table des symboles, rendez les fichiers LEX et YACC sans cette partie (repartez du corrigé du programme lex).

L'exécution du programme sur le fichier exemple2.txt, donné à l'étape 2 du projet, ne doit afficher aucune erreur. Vous pouvez tester votre programme sur ce même fichier après avoir inséré quelques erreurs syntaxiques afin de vérifier que celles-ci sont bien signalées.

Travail à rendre au plus tard 1 semaine après votre TP3, soit

- Groupe TP1 : 28 mars
- Groupe TP2 : 27 mars
- Groupe TP3 : 24 mars
- Groupe TP4 : 27 mars

Les fichiers à rendre sont à regrouper dans une archive zip. Déposez cette archive sur plubel en utilisant le lien « Dépôt étape 4 projet ».

Ecrivez votre numéro de groupe dans l'un des fichiers. Ajoutez un makefile ou un fichier texte dans lequel vous donnerez les commandes de compilation. Rendez un seul travail par groupe de projet.

Rappel

Il est conseillé de terminer le TP2 avant la fin du mois de mars.