SORBONNE UNIVERSITÉ

TME : Compilation du langage Mini-BASIC — semaine 1

8 février 2018

Objectif(s)

✯ Élaboration d'un compilateur complet pour un sous-ensemble de Basic.

Exercice 1 – Le langage Mini-BASIC

On considère la spécification du langage Mini-BASIC suivante.
— Un programme `BASIC_L2` niveau 1 contiendra une suite de lignes numérotées et rangées en ordre croissant.
— Une ligne est définie par un label (son numéro) et une des instructions suivantes.

Instruction	Action
`REM texte`	Commentaire.
`LET var = expr`	Affectation de la variable *var* par la valeur de *expr*.
`PRINT e_1; e_2; ...; e_n;`	Affiche les valeurs des expressions `e_1, ...,e_n` à la suite et fait un saut de ligne si la dernière expression n'est pas suivie d'un point-virgule.
`INPUT var`	Lit au clavier une valeur qui sera affectée à `var`.
`GOTO label`	Aller à l'instruction numérotée *label*
`IF expr GOTO label`	Si *expr* est vraie, aller à l'instruction numérotée *label*
`GOSUB label`	Appel de la sous-routine numérotée *label*
`RETURN`	Retourne à l'instruction suivante du dernier GOSUB rencontré
`END`	Fin du programme

Finalement, les expressions arithmétiques, relationnelles et logiques manipulées sont les suivantes :

Expression	Rôle
`e [*+/-] e'`	Expression arithmétique
`[-+] e`	Expression unaire
`( e )`	Expression parenthèsée
`e (<|=|>) e'`	Expressions relationnelles
`e (AND|OR) e'`	Expressions logiques binaires
`NOT e`	Expressions logiques unaires

En plus nous prenons en considération les aspects suivants :
— Les variables peuvent contenir des nombres (entiers ou flottants) ou bien des chaînes de caractères.
— On les distingue par leur nom.
— Une variable se terminant par un caractère `$` (dollar) contiendra des chaînes de caractères.

1. Le but de ce TME est de programmer un compilateur pour ce langage. Le choix du langage à utiliser pour programmer ce compilateur est libre mais nous vous conseillons OCaml. Vous pouvez démarrer du code du langage Calc donné en TD.
   Il est très important de ne pas vouloir tout faire d'un coup et de procéder de manière incrémentale en considérons des portions du langage de plus en plus grande. Par exemple :

— Étape 1 : Un langage avec seulement les nombres entiers et l'instruction `PRINT` : lexer, parser, AST, évaluation, vérification.
— Étape 2 : Ajouter la gestion des commentaires.
— Étape 3 : ...

L'important est de vérifier très souvent si tout marche et d'obtenir très vite un premier résultat.

Exercices supplémentaires

L'exercice suivant peut être fait avant de réaliser le compilateur Mini-BASIC pour s'échauffer. Vous pouvez également reprendre les exercices du TD afin de vous entrainer.

Exercice 2 – Expressions régulières

Traiter les exercices suivant en utilisant l'un (ou plusieurs si vous avez le temps) des outils suivants :
— `Str` en OCaml
— `regexp` en Java
— `regex` en C.

1. Ecrire un traducteur de dates du format anglais *mm.jj.aa* au format français *jj/mm/aa*, les dates pourront se trouver n'importe ou dans la chaine !

Exercice 3 – Utilisation de Genlex

1. Implantez un analyseur de Mini-BASIC en utilisant les flots et le module Genlex.

Exercice 4 – Etude de l'analyseur XML-light

Pour les projets nécessitant de faire une analyse syntaxique de langages à balises, il est possible d'utiliser xmllight développé en OCaml. Les sources sont à télécharger via l'URL suivante : `http://tech.motion-twin.com/xmllight`.

En utilisant les fonctions d'analyse syntaxique `Xml.parse_string` et `Xml.parse_file` construisez des valeurs XML que vous afficherez avec les fonctions `Xml.to_string` et `Xml.to_string_fmt`.