Python

Java

Théorie des langages de programmation Le projet (I)

LULUII

C++

Alain Chillès – 祁冲

ParisTech Shanghai Jiao Tong 上海交大—巴黎高科卓越工程师学院

26 octobre 2020 - 2020年10月26日 - 庚子九月初十

APL

Fortran

Plan Projet Description

Fortran

Objectif

- Écrire un compilateur pour un langage suffisamment simple permettant d'aborder les thèmes suivants :
 - Analyse lexicale
 - Analyse syntaxique
 - Analyse sémantique
 - Interprétation
 - Compilation en machine virtuelle
 - Compilation en machine réelle
- Le langage proposé est un langage inspiré du langage Forth (nous l'appellerons L_{AC}).

Fortran



ļ

Lexèmes

- Les identificateurs (toute chaîne de caractère d'au moins un caractère, composée à l'aide des lettres, chiffres et symboles de ponctuation, sauf le caractère blanc [noté □]).
- Les chaînes de caractères (toute chaîne de caractère commençant par " (ou " en début de ligne), ne contenant pas le caractère " à l'intérieur et se terminant par le caractère "). On fera attention qu'elle peut être sur plusieurs lignes et donc contenir des sauts de ligne.
- Les commentaires de ligne commençant par les caractères (ou) en début de ligne) et finissant en fin de ligne.
- Les commentaires multi-lignes commençant par les caractères L(u (ou (u en début de ligne) et se terminant par).

Ę

Première étape

- Définir le type lexème en C.
- Écrire un analyseur lexical qui à partir d'un fichier contenant un texte en L_{AC} :
 - Supprime les commentaires.
 - Produit une liste de lexèmes.

On produira donc trois fichiers :

- Un fichier analex.h
 - Un fichier analex.c
 - Un fichier principal L_AC.c
- Tester sur le fichier factorielle.lac (fourni).

Le fichier test

Fichier factorielle.lac

```
\ Fichier "factorielle.lac"
2
   ( Ce fichier est un "exemple" étudié pour
3

→ tester

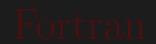
   l'analyseur lexical écrit en phase 1 du
    → projet)
5
   : fact ( n -- n!)
       dup 0=
7
       if
8
            drop 1
9
       else
10
           dup 1- recurse *
11
       then ;
12
```

Le fichier test

```
T A OTTOTT
```

```
Java
```

Fichier factorielle.lac



Plan

I y ULIVII

Java

C

 Prolog

Forth

Projet

Analyse lexicale

Bilan de ce qui est demandé

Pascal

APL

Fortran

Lisp

Projet 1 : définition

- Les lexèmes du langage sont :
 - les entiers naturels (N)
 - les chaînes de caractères (nom des mots L_{AC} (M) et chaînes de caractères en L_{AC} (S))
- Ce qui sépare les lexèmes, sauf dans les chaînes de caractères est le caractère blanc □
- Tous les commentaires doivent être supprimés lors de l'analyse lexicale

Il faut produire une liste (ou un tableau) de lexèmes. C'est l'analyseur lexical! Les fichiers à fournir seront

- Un fichier analex.h
- Un fichier analex.c
- Un fichier permettant de tester l'analyseur, appelé projet1.c

Projet 1: exemple

Fichier factorielle.lac

```
\begin{array}{l} \mathsf{M}(":") {\to} \mathsf{M}("\mathsf{go}") {\to} \mathsf{S}("\mathsf{Factorielle}") {\to} \mathsf{M}("\mathsf{count}") {\to} \mathsf{M}("\mathsf{type}") \\ {\to} \mathsf{M}("\mathsf{dup}") {\to} \mathsf{M}(".") {\to} \mathsf{S}("\mathsf{vaut}_{\sqcup} : \mathsf{n}") {\to} \mathsf{M}("\mathsf{count}") \\ {\to} \mathsf{M}("\mathsf{type}") {\to} \mathsf{M}("\mathsf{fact}") {\to} \mathsf{M}(".") {\to} \mathsf{M}("\mathsf{cr}") {\to} \mathsf{M}(";") \end{array}
```

Projet 1: contraintes

- C_{-}
- L'analyse lexicale sera faite à l'aide d'expressions régulières.
- Les codes doivent pouvoir être compilés sur Linux, sans warning ni error avec la commande

Terminal

\$ gcc -Wall projet1.c

