

DA53: Compilation and Language Theory - Final Exam A2023

Duration: 2h00. No document allowed.
English recommended, French accepted.

Part 1: Introduction (5 points)

Question 1.1:

Consider the Python language. Is this programming language type-safe? Explain why.

Question 1.2:

What is a Syntax-Directed Definition (SDD)? Explain briefly the concept of SDD and its content.

Question 1.3:

What are the names of the different parts of the program memory in a standard execution environment? Explain briefly the role of each part of the program memory.

Part 2: Tiny-Caml (10 points)

In this part, you must consider a simplified version of the functional language Caml, that is explained below. In Tiny-Caml, the functions are defined as:

```
let <name> <param1> <param2> ... <paramn> = <expression>;;
```

where:

- <name> is an identifier that is the name of the function,
- <param_{*i*}> is the formal parameter of the function at the position *i*, and it is an identifier or a constant,
- <expression> is the body of the function as an expression composed of:
 - arithmetic expressions with floating point numbers, the operators +, -, *, /, and the parenthesis;
 - function calls of the form: <name> (<arg₁>, <arg₂>, ...).
 - “Conditional” statement:
if <condition> then <expression₁> else <expression₂>
This statement evaluates to condition expression. If it is evaluated to true, the value of <expression₁> is replied. Otherwise the value of <expression₂> is replied.

Remainder:

You may answer on your paper sheet to the three following questions in a single Syntax Directed Definition (SDD) that is containing both the grammar rules in Backus-Naur Form (BNF) and the SDD semantic rules.

Question 2.1:

Write the grammar rules (using the Backus Naur Form - BNF) for recognizing the nonterminal <expression> for the Tiny-Caml language. Your grammar will be analyzed with LL(0) approach. So that, you should avoid left-recursions and remove ambiguities from the rules.

Question 2.2:

Extend the previous BNF grammar for recognizing the “function declaration” for the Tiny-Caml language.

Question 2.3:

Write the semantic rules of the Syntax Direction Definition (SDD) to interpret the code and compute the numerical values for each element of the grammar. The goal of the SDD rules is to execute the Tiny-Caml code and reply the computed numerical value. To do so, you must follow the points:

1. List the attributes for each nonterminal. Are they synthesized or inherited?
2. Write the semantic rules for each nonterminal that is computing the values of the attributes.

Part 3: Code Generation (5 points)

In this part, you must focus on the generation of the form of three-address code that is named “quadruple.” The language to translate is the one described in the part 2.

Remainder:

In the generated three-address code, you may not define the addresses themselves, but use symbolic names in place.

Question 3.1:

In this question, you must assume that you have **NOT** obtained a syntax tree from the parsing stage. The generation of the quadruples must be directly put in the semantic rules of the SDD.

1. Define the additional attributes that are required to generate the “quadruples”.
2. Write the SDD to generate the “quadruples” for all the nonterminals of the BNF defined in Part 2.