CmpE 260 - Principles of Programming Languages
Spring 2019
Assignment 1

due: 28.03.2019 - 16:59

Question 1 [35 points]

Suppose that, in a programming language, an assignment statement consists of a left-hand side (a single variable), equality sign, and a right-hand side. On the right-hand side, there may be variables, four types of binary operator (+, $-$, $*$, /), and parentheses. (The expressions on the right-hand side have the usual semantics.) Variables are single letters. The precedence order among operators is as follows (from highest to lowest): 1) * and /, 2) + and -.

Some example expressions are given below:

$$\begin{aligned} a &= b \\ a &= (b + (x * y)) \\ x &= ((d) + e/f) * x * r/(k + l) \end{aligned}$$

a) Write BNF description for assignment statements in this language. Do not use EBNF.

b) Add a binary exponentiation operator (ˆ) to the grammar given above and write BNF description of the modified grammar. This operator should have a higher precedence than +, -, / or *. In addition, ˆ has right precedence while the rest have left precedence.

c) Draw the parse tree of the given statement according to the modified grammar: $x = a\hat{}\,(b + c * d)$

Question 2 [35 points]

In a language, we have a loop construct called *repeat* whose syntax is given below:

$$\begin{aligned} < repeat > &\to repeat < st - list > until < bool > \\ < st - list > &\to < assign - st >< st - list > \mid < assign - st > \\ < bool > &\to < var > == < var > \\ < assign - st > &\to < var > = < var > \end{aligned}$$

The semantics is as follows: The body of the loop ($< st - list >$) is executed until the boolean condition ($< bool >$) is true. It is a posttest loop, thus the body is executed at least once (i.e. when the loop is initiated, first $< st - list >$ is executed, then $< bool >$ is checked. If $< bool >$ evaluates to true then the loop terminates, otherwise $< st-list >$ is executed again, then $< bool >$ is checked, and so on).

As can be seen, $< st - list >$ is a sequence of assignment statements (at least one); each assignment is a simple statement in the form of assigning a variable to another variable; and boolean condition is a simple statement in the form of checking equality of two variables.

Write denotational semantics description of the repeat construct. You must define every function in your description formally; do not assume the existence of any functions (except a function which returns the value of a given variable). Also, all possible errors should be handled.

Question 3 [30 points]

Water. Earth. Fire. Air.

Long ago, the four nations lived together in harmony. Then, everything changed when the Fire Nation attacked. Only the Avatar, master of all four elements, could stop them, but when the world needed him most, he vanished. A hundred years passed and my brother and I discovered the new Avatar, an airbender named Aang, and although his airbending skills are great, he has a lot to learn before he's ready to save anyone. But I believe Aang can save the world. (Link)

In our language, we have variable declarations in the following form:

$$\begin{aligned} katara < element >< element > ... \\ toph < element >< element > ... \\ zuko < element >< element > ... \\ aang < element >< element > ... \end{aligned}$$

That is, first elements of *katara*, then elements of *toph*, then elements of *zuko* and finally elements of *aang* are defined. All these four benders must exist in the definition and must be in the above order. For each bender, there is at least one element declared. $< element >$ is an identifier name and can be *Water (W), Earth (E), Fire (F), or Air (A)*. We have the following constraints:

- Each bender can only bend one type of element. Only the Avatar Aang can bend all types of elements.
  - **Katara**: Waterbending.
  - **Toph**: Earthbending.
  - **Zuko**: Firebending.
  - **Aang**: Avatar - Can bend all the elements - Waterbending, Earthbending, Firebending and Airbending.
- To restore the harmony, each bender must successfully bend the same amount of elements.
- Every bender can try to bend different elements than their specialization (e.g. Katara can try to bend Fire even though she is a waterbender), but only the elements that they have ability to bend counts toward our goal to restore the harmony.

Valid declaration:

katara W W E A W

toph E E E

zuko W F F E E F

aang A E F

Reasoning of the valid declaration:

- Katara is a waterbender. She tries to bend 3 water, 1 earth and 1 air elements. Only water elements are counted. So we have 3 elements from Katara.
- Toph is an earthbender. She tries to bend 3 earth elements. Only earth elements are counted. So we have 3 elements from Toph.
- Zuko is an firebender. He tries to bend 3 fire, 2 earth, 1 water elements. Only fire elements are counted. So we have 3 elements from Zuko.
- Aang is the Avatar and can bend all the elements. He tries to bend 1 air, 1 earth and 1 fire elements. All of the are counted. So we have 3 elements from Aang.
- All our benders have successfully bended the same amount of elements and the harmony is restored.

Invalid declaration:

katara W W E A W

toph E E E A F

zuko W F F E E

aang A E F

Reasoning of the invalid declaration:

- Zuko successfully bended only 2 elements while others bended 3 elements. Thus the harmony is not restored.

Design an attribute grammar for this purpose.

İki Kalp

İki kalp arasında en kısa yol:
Birbirine uzanmış ve zaman zaman
Ancak parmak uçlarıyla değebilen
İki kol.

Merdivenlerin oraya koşuyorum,
Beklemek gövde gösterisi zamanın;
Çok erken gelmişim seni bulamıyorum,
Bir şeyin provası yapılıyor sanki.

Kuşlar toplanmışlar göçüyorlar
Keşke yalnız bunun için sevseydim seni.

Cemal Süreya