Programming Language Processor Assignment 2

Answer the following questions and submit your report (Word or PDF) to tetsuya@shibaura-it.ac.jp before Dec. 15. The subject of your mail should be of the form "PLP Assignment 2".

Question 1

Modify the sytax diagram of PL/0' as follows, and explain the modified parts of the diagram.

- Introduce the following do-while statement to PL/0'.
- Introduce the following repeat-until statement to PL/0'.
- Introduce the following if-then-else statement to PL/0'.

do-while statement

Introduce the following do-while statement into PL/0'.

Production rule *statement* → **do** *statement* **while** *condition*

Action A statement '**do** *statement* **while** *condition*' works as follows

1. Execute *statement*.
2. If the value of *condition* is true, go to the step 1. Otherwise, exit this loop.

Fig.1 shows a sample program with a do-while statement.

```
var x;
begin
   x := 0;
   do begin
      write x;
      writeln;
      x := x + 1
   end
   while x < 3
end.
```

Figure 1: A sample program `do.pl0`

repeat-until statement

Introduce the following repeat-until statement into PL/0'.

Production rule *statement* → **repeat** *statement* **until** *condition*

Action A statement '**repeat** *statement* **until** *condition*' works as follows.

1. Execute *statement*.
2. If the value of *condition* is false, go to the step 1. Otherwise, exit this loop.

Fig.2 shows a sample program with a repeat-until statement.

```
var x;
begin
   x := 0;
   repeat begin
      write x;
      writeln;
      x := x + 1
   end
   until x=3
end.
```

Figure 2: A sample program `repeat.pl0`

if-then-else statement

Modify the syntax diagram so that PL/0' can accept the following if-then-else statement.

Production rule *statement* → **if** *condition* **then** *statement*$_1$ (**else** *statement*$_2$ | ϵ)

Action A statement '**if** *condition* **then** *statement*$_1$ (**else** *statement*$_2$ | ϵ)' works as follows.

1. Evaluate *condition*.
2. If the value of *condition* is true, execute *statement*$_1$.
3. If the value of *condition* is false and *statement*$_2$ exists, execute *statement*$_2$.

Description To resolve ambiguity of the grammar of PL/0', we use the following rule.

- When we find an **else**, we relate the **else** to the nearest **then** which has not be related to any **else** yet.

Fig.3 shows a sample program with if-then-else statements.

```
var x;
begin
   x := 0;
   while x<3 do begin
      if x < 1 then write 0
      else if x < 2 then write 1
      else write 2;
      writeln;
      x := x+1;
   end;
end.
```

Figure 3: A sample program `else.pl0`

Question 2

Modify your syntax diagram in Question 1 so that PL/0' can accept one-dimensional array, and explain the modified parts of the diagram. In addition, write a simple program tt array.pl0 which demonstrates how to use your array.

Hints You have to consider how to declare arrays, how to refer to array elements and how to assign values to array elements.

Question 3

Modify your syntax diagram of PL/0' in Question 1 so that PL/0' can accept procedure declarations and procedure calls. A procedure means a function without any return value like a void function in C.

We use the following statement to call a procedure with n arguments.

call *procedure*(arg_1, arg_2, ..., arg_n)

Explain the modified parts of your syntax diagram and write a simple program 'proc.pl0' which demonstrates how to declare and call procedures.