

## CS 302 : Implementation of Programming Languages

### TUTORIAL 6 (Static Semantics - Declaration Processing ); March 22, 2017

Consider the following SDTS discussed in the class for declaration processing for Pascal like languages as a reference.

```

P --> M D      { addwidth(top(tableptr),top(offset)); pop (tableptr) ; pop (offset) }
M --> ε        { p := mktable (nil); push(p, tableptr); push (0, offset); nest := 1 }
D --> id : T    { enter (top(tableptr), id.name, T.type, T.width); top (offset) := top (offset) + T.width }
D --> procedure id ; N D ; S { p := top (tableptr); addwidth (p, top (offset)) ; pop (tableptr);
                                pop (offset); nest - - }
T --> integer { T.type } := integer; T.width := 4 }
T --> real    { T.type } := real; T.width := 8 }
T --> array [ num ] of T1 { T.width := num.val x T1.width; T.type := array (num.val, T1.type) }
T --> ↑ T1    { T.type := pointer (T1.type) ; T.width := 4 }
N --> ε        { p := mktable (top (tableptr)); enterproc(top(tableptr), id.name, nest, p);
                                push (p,tableptr); push (0, offset) ; nest ++; }

```

**P1.** Without constructing the parser explicitly, show the contents of the two stacks, the symbol tables and their interconnections when the parser reaches the point (i) marked by ( $\Rightarrow$  marked point 1) in the program fragment, and the point ii) marked by ( $\Rightarrow$  marked point 2) in the program given below when the SDTS translation scheme given above is used.

```

procedure p1;
  a : real;
  procedure p2;
    b : integer;
    procedure p3;
      b : real;
 $\Rightarrow$     marked point 1
      body of p3;
    procedure p4;
      a : integer;
 $\Rightarrow$     marked point 2
      body of p4;
    body of p2;
  body of p1;

```

**P2.** Identify all the possible semantic errors in processing of declarations of the types - bool, char, short, int, float, double and derived types over these basic types - arrays, pointers of language C and express the same in English. Write a grammar, mention the semantic attributes chosen and then write the STDS for handling processing of declarations and relevant semantic errors.

**P3.** Write a SDTS for handling **struct** declarations in C, including **nested structs**. The built-in types are int, char and derived types are arrays and pointers. The SDTS should address the important issues such as symbol table organization and semantic errors. Outline your objective in simple english first and then provide the SDTS.

**P4.** Do the assignment of problem P3 but replace **struct** by the **union** feature of C.

**P5.** Do the assignment of problem P3 but replace a **struct** by the **class** of C++ and the other types remaining same but of C++. Assume a class definition to have only data members and declaration(s) may be qualified as **public**, **private** or **protected**.

**P6.** Do the assignment of problem P3 but replace a **struct** by **enumerated** type of C.

**P7.** Do the assignment of problem P2 but add **const** objects of the types mentioned therein. Identify the requirements to handle const objects and semantic errors they could be involved in. Outline your objective in simple english first and then provide the SDTS.

**P8.** C++ has a special type called a **reference** type, which is similar to a pointer, yet quite different. Add this type to the types addressed in P2. Outline your objective in simple english first and then provide the SDTS.

**P9.** Declaration of a **derived class** of C++, single inheritance only, is the concern of this problem. Assume in the same spirit as P5 that a class definition has only data members and declaration(s) may be qualified as **public**, **private** or **protected**. Further the derivation may also be **public**, **private** or **protected**. Identify the issues in the semantic analysis of this feature and then provide a SDTS for the same.

\*\*\*\*\* End of Tutorial Sheet 6 \*\*\*\*\*

Released on March 21, 2017

Supratim Biswas