

# Project Phase 1 Pascal Specifications

CS 441  
Fall 1992  
Craig A. Rich

language.pas

```
[environment]
module LANGUAGE;

  const MAX_STRING_LENGTH      = 30;
        MAX_alpha_LENGTH      = 7;
        MAX_PRODUCTIONS_LENGTH = 24;

  type NATURAL = 0..MAXINT;
       STRING  = varying [MAX_STRING_LENGTH] of CHAR;

  (*-----*)
  (* Grammar Symbols *)
  (*-----*)

  type SYMBOL_TYPE = (
    expr1,

    expr, atom_nt, list, cases, exprs, anonymous_function, named_function,
    case_nt, formals,

    T, id, NIL_t, leftparen, rightparen, QUOTE, CAR, CDR, CONS, EQ, ATOM_t,
    COND, LABEL_t, LAMBDA,

    endfile_$, lexical_error);

    GRAMMAR_SYMBOL_U_ENDFILE
      = expr1..endfile_$;
    GRAMMAR_SYMBOL
      = expr1..LAMBDA;
    NONTERMINAL
      = expr1..formals;
    TERMINAL
      = T..LAMBDA;
    PRODUCTION
      = record
        A: NONTERMINAL;
        alpha_LENGTH: 0..MAX_alpha_LENGTH;
        alpha: array [1..MAX_alpha_LENGTH] of GRAMMAR_SYMBOL
      end;

    PRODUCTIONS
      = array [0..MAX_PRODUCTIONS_LENGTH] of PRODUCTION;
    SET_OF_ITEMS
      = 0..60;

  (*-----*)
  (* Parse Tree Tokens *)
  (*-----*)

  type TOKEN = record
    SYMBOL: SYMBOL_TYPE
  end;

end.
```

lexical\_analyzer.pas

```
[inherit ('[cs.carich]language'), environment]
module LEXICAL_ANALYZER (INPUT);
    function NEXT_TOKEN: TOKEN;
end.
```

stack\_manager.pas

```
[inherit ('[cs.carich]language'), environment]
module STACK_MANAGER;
    procedure PUSH (X: TOKEN; I: SET_OF_ITEMS);
    procedure POP (DEPTH: NATURAL := 1);
    function TOP_X: SYMBOL_TYPE;
    function TOP_I: SET_OF_ITEMS;
end.
```

syntax\_analyzer.pas

```
[inherit ('[cs.carich]language', '[cs.carich]lexical_analyzer', 'stack_manager')]
program SYNTAX_ANALYZER (OUTPUT);
    var P: PRODUCTIONS;
    type ACTION_TYPE = (shift, reduce, error);
    type ACTION_ENTRY = record
        case ACTION: ACTION_TYPE of
            shift: (I: SET_OF_ITEMS);
            reduce: (k: 0..MAX_PRODUCTIONS_LENGTH);
            error: ();
        end;
    var M: array [SET_OF_ITEMS, GRAMMAR_SYMBOL_U_ENDFILE] of ACTION_ENTRY;
        :
        :
end.
```

Note: In each module, only the declarations of the visible (non-[hidden]) operations have been specified. [hidden] data structures and operations may be added as necessary.