Plan for Today

Full Class Terminology Quiz

FIRST and FOLLOW sets

Predictive Parsing table

Error recovery

Predictive parsing as a specific subclass of recursive descent parsing
  - necessary to remove left-recursion
  - might have to left-factor
  - complexity comparisons with general parsing

FIRST and FOLLOW sets

nullable(X)
  - X is a nonterminal
  - nullable(X) is true if X can derive the empty string

FIRST
  - FIRST(z) = {z}, where z is a terminal
  - FIRST(X) = union of all FIRST(rhs_i), where X is a nonterminal and X -> rhs_i
  - FIRST(rhs_i) = union all of FIRST(sym) on rhs up to and including first nonnullable

FOLLOW(Y), only relevant when Y is a nonterminal
  - look for Y in rhs of rules (lhs -> rhs) and union all FIRST sets for symbols after Y up to and including first nonnullable
  - if all symbols after Y are nullable then also union in FOLLOW(lhs)
Constructing the Predictive Parser Table

Algorithm

for each X -> gamma
    for each T in FIRST(gamma)
        table[X,T] = X->gamma
    if gamma is nullable
        for each T in FOLLOW(X)
            table[X,T] = X->gamma

(1) svg -> SVG_START elem_list SVG_END EOF
(2a & b) elem_list -> elem_list elem | epsilon
(3) elem -> RECT_START ... ELEM_END
(4) elem -> CIRCLE_START ... ELEM_END
(5) elem -> LINE_START ... ELEM_END

Error Recovery

Goals

– Provide program with a list of as many errors as possible
– Provide USEFUL error messages
  – appropriate line and position information
  – guidance for fixing the error
– Avoid infinite loops or recursion
– Add minimal overhead to the processing of correct programs

Approaches

– Stop after first error
– Panic mode
– Phrase-level recovery
Modified MiniSVG Grammar

(1) svg -> SVG_START elem_list SVG_END

(2a & b) elem_list -> elem_list elem | epsilon

(3) elem -> RECT_START KW_X EQ NUM KW_Y EQ NUM KW_WIDTH EQ NUM KW_HEIGHT EQ NUM KW_FILL EQ COLOR ELEM_END

(4) | CIRCLE_START KW_CX EQ NUM KW_CY EQ NUM KW_R EQ NUM KW_FILL EQ COLOR ELEM_END

(5) | LINE_START KW_X1 EQ NUM KW_Y1 EQ NUM KW_X2 EQ NUM KW_Y2 EQ NUM KW_STROKE EQ COLOR ELEM_END

Example Parse Tree for Modified MiniSVG (PROBLEM!)
Predictive parser for Float Assignment Grammar

```c
void S() { switch (lookahead) {
  case ID:
    case EOF:// the 2 characters in the FIRST(StmList EOF)
      try { StmList(); match(EOF); } catch { panic(S); } break;
    default: panic(S); break;
}}
void StmList() { switch (lookahead) {
  case ID: // FIRST( Stm StmList ) = { ID }
    try { Stm(); StmList(); } catch { panic(StmList) } break;
  case EOF: // FOLLOW(StmList) = { EOF }
    break;
  default: panic(StmList); break;
}}
void Stm() { switch (lookahead) {
  case ID: try { match(ID); match(ASSIGN); match(FLOAT);
    } catch { panic(Stm); } break;
  default: panic(Stm); break;
}}
```