

Prelude

*Happy
Valentine's Day!*

Have a chocolate.

James Sites & Elaine Regelson

Undergraduate Research Scholarship

- \$1000
- Do a research project with a faculty member and enroll for 3 research credits
- google: regelson scholarship

CS453 Lecture

Predictive Parsing

1

Plan for Today

Debugging shift/reduce errors

Recursive descent or predictive parsing

- example predictive parser
- FIRST and FOLLOW sets revisited
- constructing a predictive parser table

CS453 Lecture

Predictive Parsing

2

Example Ambiguous Grammars: SableCC errors

```
Productions
stm = exp ;
exp =
    {minus_rule} exp minus exp
    | {num_rule} num
    ;
```

```
Verifying identifiers.
java.lang.RuntimeException: [30,35] Redefinition of AMinusRuleExp.Exp.
```

```
Productions
stm = exp ;
exp =
    {minus_rule} [left]:exp minus [right]:exp
    | {num_rule} num
    ;
```

```
shift/reduce conflict in state [stack: PExp TMinus PExp *] on TMinus in {
  [ PExp = PExp * TMinus PExp ] (shift),
  [ PExp = PExp TMinus PExp * ] followed by TMinus (reduce)
}
```

CS453 Lecture

Predictive Parsing

3

Example Predictive Parser

```
S -> Mesh EOF
Mesh -> num NodeList num ElemList
NodeList -> ε | Node NodeList
Node -> node num real real // node_id, x, y
ElemList -> ε | Elem ElemList
Elem -> tri num num num num // elem id, 3 node ids
Elem -> sqr num num num num num // elemid, 4 node ids
```

```
void S() { switch(tok) {
  case NUM: Mesh(); eat EOF; break;
  default: error();
}}

void Mesh() { switch(tok) {
  case NUM: num_nodes = NUM.val; eat(NUM);
            NodeList();
            num_elem = NUM.val; eat(NUM); break;
  default: error();
}}

void NodeList() { switch(tok) {
  case NUM: break;
  case NODE: Node(); NodeList(); break;
  default: error();
}}
```

CS453 Lecture

Predictive Parsing

4

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
- $FIRST(X) = \cup FIRST(rhs_i)$, where X is a nonterminal and $X \rightarrow rhs_i$
 - 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 ($lhs \rightarrow rhs$) 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
```

```
S -> Mesh EOF
Mesh -> num NodeList num ElemList
NodeList -> ε | Node NodeList
Node -> node num real real // node_id, x, y
ElemList -> ε | Elem ElemList
Elem -> tri num num num num // elem id, 3 node ids
Elem -> sqr num num num num num // elemid, 4 node ids
```