Plan for Today

Ambiguous Grammars
- Due to associativity
- Due to precedence

Making Grammars Non-Ambiguous
- example related to PA3

What do the SableCC errors look like for ambiguous grammars?

Example Ambiguous Grammars: Associativity

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

Productions
stm = exp ;
exp =
{assign_rule}  [left]:exp assign [right]:exp
| {id_rule}  id
;

Example Ambiguous Grammars: Precedence

Productions
stm = exp ;
exp =
{or_rule}  [left]:exp or [right]:exp
| {and_rule}  [left]:exp and [right]:exp
| {true_rule}  true
| {false_rule}  false
;

Example Ambiguous Grammars: Both

Tokens
pow = ‘^’; dot = ‘.’;

Productions
stm = exp ;
exp =
{pow_rule}  [left]:exp pow [right]:exp
| {field_rule}  exp dot id
| {paren_rule}  lparen exp rparen
| {plus_rule}  [left]:exp plus [right]:exp
| {id_rule}  id
;

Precedence (high to low)
() id
^  +
Example Ambiguous Grammars: SableCC errors

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

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

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)
}
```