Relatório Análise Semântica

**Acadêmicos:**

Darlan Nakamura

Rafael Rodrigues

# Introdução

Este relatório descreve as modificações realizadas a partir da entrega anterior, referente ao analisador sintático.

# Classes Adicionadas

Todos os não-terminais da gramática foram mapeados em classes que descrevem o funcionamento de cada uma. Segue abaixo alguns exemplos, com uma descrição abaixo de cada um.

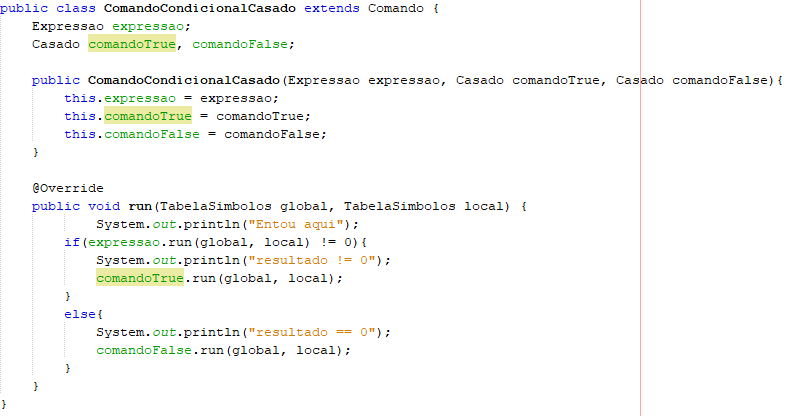


Figura 1: classe ComandoCondicionalCasado

Um comando condicional casado é composto por uma expressão e o comando que deve ser rodado caso ela seja verdadeira, assim como o comando para o caso dela resultar em falso. O método .run() indica seu funcionamento.



Figura 2: parte da classe Expressao

A classe expressão tem um funcionamento parecido, efetuando a operação (se expressaoSimple2 não for *null*), e retornando o inteiro resultante.

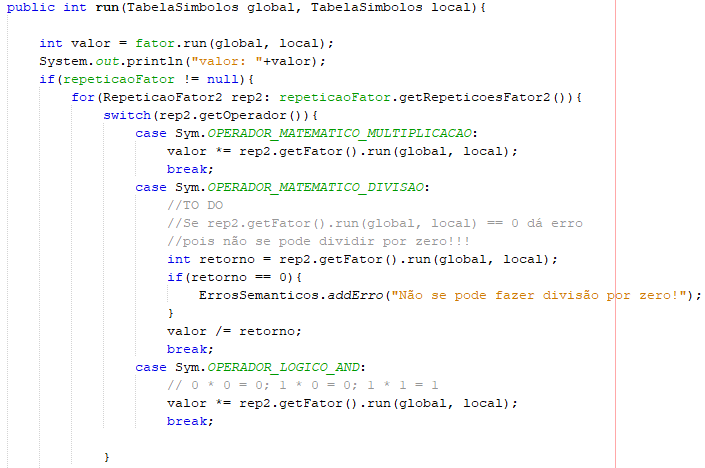


Figura 3: exemplo de erro semântico na classe Termo

Os erros semânticos são levantados dentro dos métodos .run(), ou em métodos invocados a partir do mesmo. Na Figura acima, é possível ver o erro semântico no caso de divisão por zero.

# Alterações no arquivo parser.cup

Cada produção da gramática resulta em um não-terminal. Dessa forma, foi necessário instanciar um objeto em cada produção, ou então repassar o objeto que foi criado anteriormente. Segue exemplos.

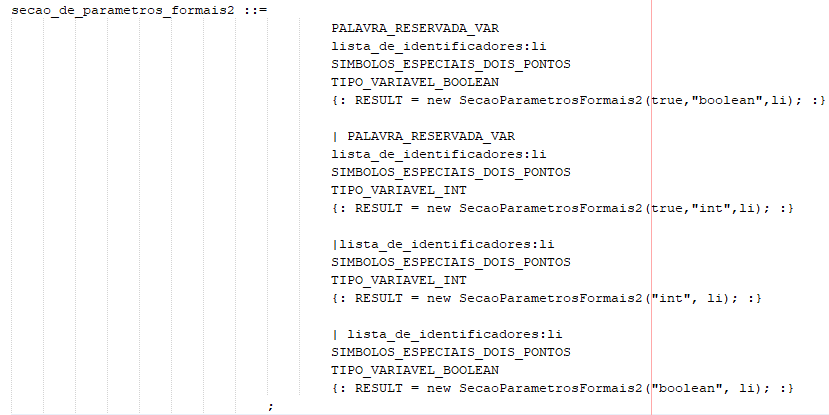


Figura 4: gramática para formar o não-terminal secao\_de\_parametros\_formais2

Na Figura acima, cada produção leva a uma instanciação com parâmetros diferentes.

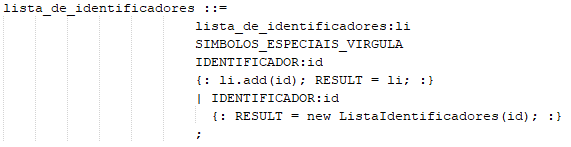


Figura 5: gramática para a criação de uma lista de identificadores na declaração de variáveis

Na Figura 5, quando um identificador é reconhecido, instancia-se uma nova lista de identificadores. No momento em que um segundo identificador é reconhecido, no caso de uma lista, o método .add() é invocado sobre a lista criada anteriormente, e o retorno é a instância da lista com o novo identificador adicionado.

# Alterações no arquivo language.lex

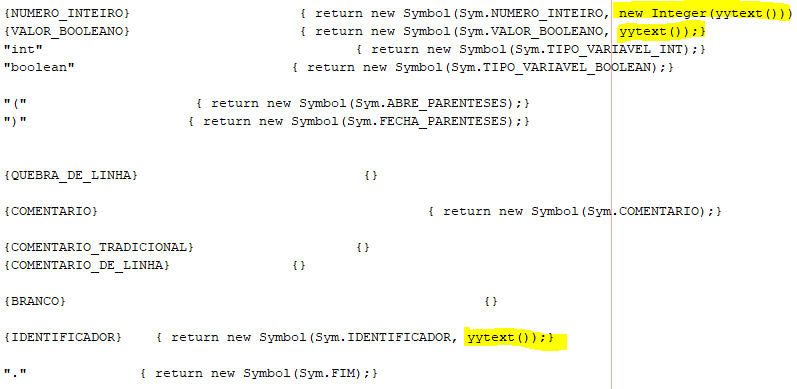


Figura 6: alterações realizadas no language.lex

Foi necessário adicionar os valores lidos no código fonte, de tal forma que o parser do CUP fosse capaz de reconhecer os valores de cada token.