Analisador Semântico

Adrian Garcia Valdes

04 de junho de 2023

1. Rodando o projeto

Para rodar o projeto basta apenas rodar o seguinte comando na pasta raiz do projeto:

node ./src/trab2.js

2. Contextualização

Este código é um analisador semântico escrito em JavaScript que processa um código fonte escrito em uma linguagem de programação fictícia. O objetivo do analisador semântico é verificar se o código fonte segue as regras semânticas da linguagem, ou seja, se as variáveis são declaradas antes de serem utilizadas, se as atribuições são feitas corretamente, entre outras verificações relacionadas à semântica do código.

3. Explicando o código principal

Inicialmente o programa lê um código escrito em uma linguagem fictícia usando a *lib* nativa do javascript “fs” (utilizada para manipulação de arquivos). O retorno é uma string grande com todo o conteúdo do arquivo.

Em seguida com a string armazenada em uma variável ela é passada para a função da lógica principal de análise semântica, onde a partir dessa string se cria uma variável do tipo array gerada a

partir do método “split()”. Basicamente cada elemento do array gerado vai ser uma linha do código em questão.

A partir daí, com o array de linhas do código se faz uma iteração para cada linha, onde a mesma é analisada para ver baseado no seu conteúdo qual tipo de operação realizar. Inicialmente se cria uma pilha de escopos vazia onde serão adicionados os escopos encontrados.

Utilizou-se a segunda abordagem sugerida pela próprio enunciado do trabalho, onde se criou uma pilha de escopos, e para cada escopo identificado se adiciona todas as variáveis lidas naquele escopo, após a finalização de um bloco o escopo em questão é removido da pilha

Se a linha contém a palavra "BLOCO", é adicionado um novo escopo à pilha de escopos. Se a linha contém a palavra "FIM", é removido o escopo mais recente da pilha. Se a linha contém a palavra "NUMERO" ou "CADEIA", a função “*extractVariable*” é chamada para extrair as variáveis do tipo correspondente. Caso contrário, a função “*changeVariableValueOnAtribution*” é chamada para tratar as atribuições de valores às variáveis. Se a linha contém a palavra "PRINT", a função “*printVariable*” é chamada para processar a instrução de impressão.

O output é basicamente cada PRINT lido no código carregado para cada variável em questão, e verificar os erros semânticos desse código. Podemos ver na imagem a baixo o output:

Obs: As tratativas de erro são feitas pelas funções auxiliares que serão explicadas na secção a seguir.

4. Funções auxiliares

No programa temos 5 funções auxiliares:

1. ***extractVariable***: Essa função é responsável por extrair as variáveis do código fonte e adicioná-las ao escopo atual. Ela utiliza expressões regulares para fazer a extração. A função recebe como parâmetros a pilha de escopos, o tipo da variável e a linha de código onde a variável está sendo declarada.

2. ***printVariable***: Essa função processa as instruções de impressão (PRINT) no código. Ela recebe a pilha de escopos e a linha de código contendo a instrução de impressão. A função verifica se a variável a ser impressa está declarada no escopo atual e imprime o seu valor, caso exista. Caso não é printada no console uma mensagem de erro com a linha em que foi encontrado o erro.

3. ***isNumber***: Essa função verifica se uma variável é um número. Ela utiliza a função isNaN do JavaScript para fazer a verificação.

4. ***changeVariableValueOnAtribution***: Essa função trata as atribuições de valores às variáveis após as declarações. Ela utiliza expressões regulares para identificar as atribuições e realiza as verificações de tipo e existência das variáveis no escopo atual. Caso o tipo não seja compatível com a declaração inicial (caso se tente atribuir um valor de string a uma variável do tipo number) uma mensagem de erro é impressa e a atribuição é ignorada.

5. Conclusão

Em conclusão, o analisador semântico implementado neste código em JavaScript é capaz de realizar a verificação de diversas regras semânticas em um código fonte escrito em uma linguagem fictícia. O analisador extrai e armazena as variáveis declaradas, verifica a correta atribuição de valores a essas variáveis, e processa as instruções de impressão. Além disso, o uso de uma pilha de escopos permite lidar com a hierarquia de escopos presentes no código, garantindo que as variáveis sejam acessadas corretamente.

O analisador semântico demonstra a importância de realizar a análise semântica em um compilador ou interpretador, pois é nessa etapa que são verificadas as regras relacionadas ao significado e interpretação do código. A detecção de erros semânticos durante essa etapa permite evitar comportamentos indesejados ou imprevisíveis do programa em tempo de execução.

Embora o código analisado seja específico para uma linguagem fictícia, as técnicas e conceitos utilizados podem ser aplicados em diferentes linguagens de programação, adaptando-se às regras semânticas específicas de cada uma. Um analisador semântico robusto desempenha um papel fundamental na validação do código fonte e na melhoria da qualidade e confiabilidade do software desenvolvido.