Relatório Algoritmo Early

Luís Eduardo Bertelli, Yuji Yamada Correa

[1]Departamento de Ciência da Computação
Universidade do Estado de Santa Catarina (UDESC-CCT)
Joinville – SC – Brasil

1. Introdução

A linguagem utilizada para a implementação do algoritmo de Early foi o Python, pelo fato de ser uma linguagem na qual os integrantes da dupla possuem maior domínio, além de permitir uma maior liberdade na tomada de certas decisões durante a implementação do trabalho.

Foi utilizado o pseudocódigo apresentado pelo professor em sala de aula, como referência para entender o funcionamento do algoritmo. No entanto, algumas verificações extras precisaram ser adicionadas para rodar o algoritmo com sucesso.

Nas seções seguintes, partimos para a descrição código proposto, apresentando o passo a passo para o seu funcionamento.

2. Pré Primeira Etapa

A entrada de dados é feita através de um arquivo txt contendo os elementos da gramática: Variáveis (V), Terminais (T), Produções (P) e Estado inicial (S). Esses dados foram armazenados da seguinte maneira:

- V = Dados armazenados em uma array, onde cada elemento representa uma variável da gramática. Exemplo: V = [‘E’,’F’,’T’].
- T = Dados armazenados em uma array, onde cada elemento representa um terminal da gramática. Exemplo: T = [’(’, ’)’, ’*’, ’+’, ’x’].
- P = Dados armazenados em uma array, onde cada elemento representa uma produção, cada produção é armazenada em uma tupla onde o primeiro elemento é a variável de origem e o segundo elemento é uma array com suas derivações. O array de derivações separa elemento a elemento. Exemplo: P = [(’E’, [’E’, ’+’, ’T’]), (’E’, [’T’])].
- S = Dado armazenado como str. Exemplo: S = E.

Com o armazenamento desses dados, é possível dar início às etapas de execução do algoritmo.

3. Primeira Etapa: construção do primeiro conjunto de produções

Para a construção do primeiro conjunto de produções, foi utilizado como referência para construção do algoritmo o pseudocódigo presente na Figura 1.

A seguinte lógica foi utilizada para construção dessa função:

Foi instanciado um array D0 que terá o primeiro conjunto de produção e um array verificador, onde ficarão salvas as variáveis que serão adicionadas no conjunto D0, para evitar produções repetidas. Através de um *for*, percorremos o array de produções e

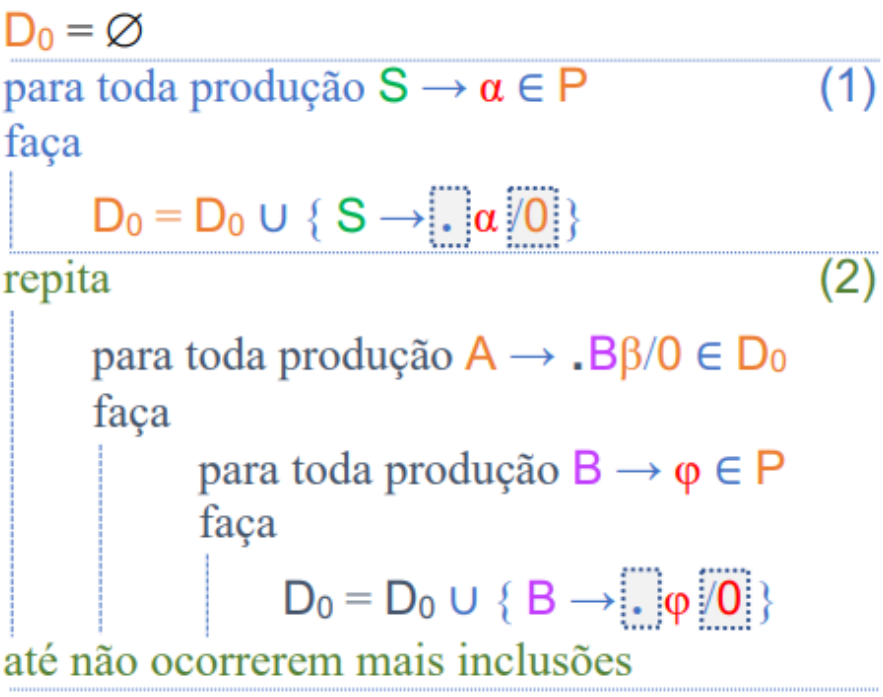

Figure 1. Pseudocódigo do algoritmo de Early

adicionamos ao conjunto D0 todas aquelas em que a variável derivada é igual ao estado inicial (S). Na hora de adicionarmos a D0, salvamos cada produção da seguinte forma: Um array com 3 elementos, sendo cada elemento outro array, o primeiro representa a variável derivada, o segundo representa as suas derivações e o terceiro possui um array com 2 elementos onde o primeiro é o índice da posição onde está posicionado o ponto e o segundo é em que conjunto essa regra foi adicionada. Exemplo: [[E],[‘E’,’+’,’T’],[0,0]].

Após isso, percorremos o conjunto D0 verificando o elemento indicado pelo ponto em cada regra de D0, se constatado que esse elemento é uma variável, verificamos se ela já está contida no array verificar, se não, adicionamos a variável ao array verificador e adicionamos a D0 mesmo formato já comentado as produções que derivam dessa variável.

Ao terminar de percorrer D0, retornamos ao array D0. Ainda antes de iniciar a próxima etapa, imprimimos a produção D0 no formato correto e solicitamos ao usuário que digite a palavra a ser verificada.

4. Segunda Etapa: construção dos demais conjuntos de produções

Para construção dos demais conjuntos de produção, foi utilizado como referência o pseudocódigo apresentado na Figura 2.

A seguinte lógica foi utilizada para o desenvolvimento dessa função:

Tendo a palavra a ser verificada, salvamos o tamanho dessa palavra, esse será o número de conjuntos gerado. As produções ficarão salvas no array D no seguinte formato, cada elemento do array será um outro array com 2 elementos, o primeiro representando qual o número desse conjunto (exemplo D0 é salvo o número 0) e o segundo as regras de produção desse conjunto, iniciaremos o código salvando a produção D0 em D.

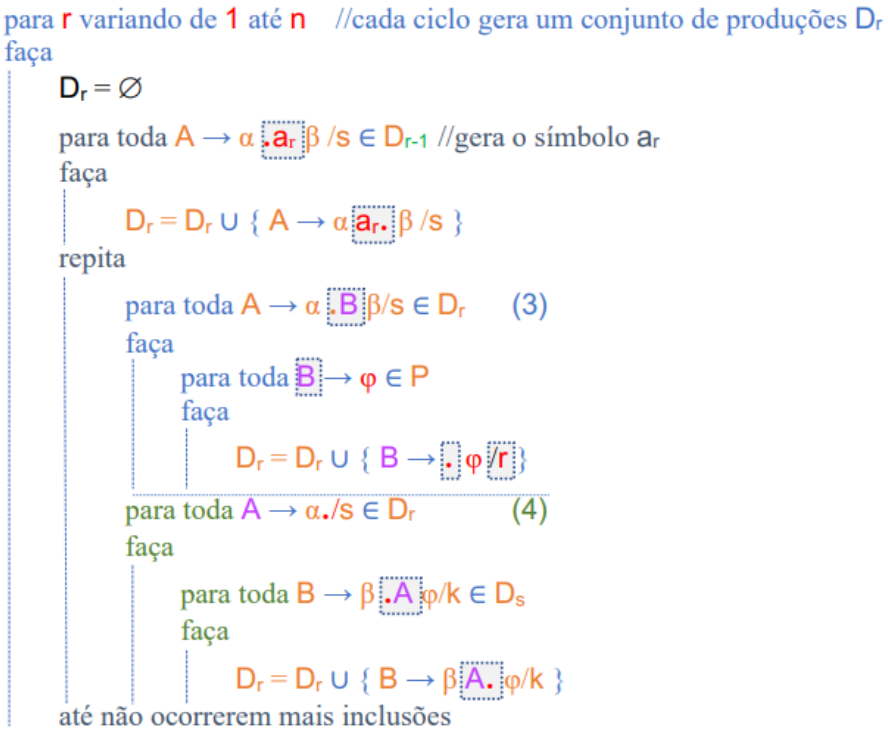

Figure 2. Pseudocódigo do algoritmo de Early

Iniciando as iterações, selecionaremos letra a letra da palavra até atingir o final. Para cada iteração é criado um array de produções, que salvará o conjunto antes de adicionar ao array D, e um array verificador, com o mesmo propósito do array verificador da função anterior, i.e, impedir regras repetidas de serem geradas.

Com isso, percorremos as produções do conjunto anterior para verificar se a letra da palavra que estamos verificando está presente em uma posição marcada por um ponto, antes de fazermos essa verificação fazemos um teste para avaliar se a produção que estamos analisando já chegou ou não até o final, para evitar acessar um índice inexistente. Se localizamos esse terminal em uma das regras do conjunto anterior, adicionamos ao array produções essa regra, adicionando +1 ao elemento do array que guarda a posição do ponto nessa produção, em outras palavras estamos passando o ponto para direita.

Após isso verificamos se a produção é vazia para evitar erros de lógica, se não for vazia, percorremos o array de produções. A primeira verificação que fazemos é para analisar se alguma produção chegou ao final, se sim, analisaremos o conjunto que essa produção foi gerada pelo índice salvo e adicionaremos ao array produções as regras que podem ter originado essa produção, adicionando +1 ao elemento que guarda o índice do ponto.

Se a verificação para avaliar se uma produção chegou ao final for falsa, faremos uma nova verificação para analisar se o elemento da produção na posição que aponta o índice do ponto é uma variável, se for, verificamos se essa variável já está no array verificador, se não estiver, adicionamos a variável a esse array e adicionaremos ao array produções as regras de produção em que a variável indicada pelo ponto é a variável

derivada, posicionando o índice do ponto na primeira posição e adicionando em que conjunto essa regra foi gerada.

Após terminar de percorrer o array produções, o mesmo é adicionado ao array D junto com o seu índice e a próxima iteração começa, até que a última letra da palavra tenha sido analisada. Ao final dessa função, retornamos o array D.

Com isso imprimimos na tela os conjuntos gerados e iniciamos a última etapa de verificar se a palavra pertence ou não à gramática.

5. Etapa Final: verificação da palavra

Para essa etapa, acessamos apenas a última produção gerada e verificamos se nessa produção está presente alguma regra onde a variável derivada é a mesma do estado inicial porém com o índice do ponto na última posição e o índice do conjunto onde foi gerado igual a 0, se encontramos essa produção do último conjunto, a palavra foi verificada e está presente na gramática, se não foi encontrada é constatado que a palavra não pertence a gramática.