

PCS2046 - 2o Exercício Programa

Ricardo Luis de Azevedo da Rocha

¹Reconhecimento de Cadeias a Partir de uma Gramática Recursiva

ATIVIDADE DE PROGRAMAÇÃO

O objetivo didático desta atividade é experimentar concretamente os conceitos desenvolvidos em sala de aula a respeito de gramáticas, cadeias, etc. Além disso, deve ser usada uma linguagem funcional como paradigma de linguagem de programação - uma das linguagens *Elixir* ou *Clojure*.

O objetivo do exercício é implementar o algoritmo de reconhecimento de cadeias geradas por uma gramática de estrutura de frase *recursiva*, que foi definido em sala de aula - algoritmo para reconhecer cadeias a partir de gramáticas. No algoritmo um dos argumentos é a cadeia w a ser verificada, e o outro a gramática. Então, produz-se iterativamente um conjunto T_i contendo todas as formas sentenciais da gramática recursiva cujo comprimento ℓ seja $\ell \leq |w|$, até que o próximo conjunto $T_{i+1} = T_i$. Se $w \in T_{i+1}$ então a cadeia é aceita (isto é, foi gerada pela gramática), senão é rejeitada.

Para cumprir este objetivo sugere-se a seguinte seqüência de etapas de execução:

1. Construa uma função em *Elixir* ou *Clojure* que permita percorrer um conjunto recursivamente (recebido em uma lista) e, a cada chamada recursiva da sua função, retorne um dos elementos do conjunto. (Esta função deve ser chamada recursivamente)
2. Construa uma função em *Elixir* ou *Clojure* que permita a geração de cadeias (incluindo formas sentenciais e sentenças) em ordem de tamanho. A sua função deve receber as regras de uma gramática de estrutura de frase, recursiva, e deve solicitar um valor de tamanho para as cadeias. A partir destes dados a sua função deverá gerar todas as cadeias cujos tamanhos sejam menores ou iguais ao valor recebido. (use a função do item anterior como apoio)
3. Construa um sistema em *Elixir* ou *Clojure* que implemente o algoritmo de reconhecimento de cadeias a partir de uma determinada gramática. O seu sistema deverá utilizar a função desenvolvida no item anterior.

Obs:

- Em grupos de até duas pessoas.

Referências

- [1] Ulisses Almeida: Learn Functional Programming with Elixir. The Pragmatic Bookshelf (2018), (<https://pragprog.com/book/cdc-elixir/learn-functional-programming-with-elixir>)
- [2] Introdução a Elixir: <https://elixir-lang.org/getting-started/introduction.html>
- [3] Elixir site: <https://elixir-lang.org>
- [4] Alex Miller and Stuart Halloway and Aaron Bedra: Programming Clojure. The Pragmatic Bookshelf (2018), (<https://pragprog.com/book/shclop3/programming-clojure-third-edition>)
- [5] Clojure site: <https://clojure.org/>; Clojure.org