**Applied Automata Theory** (SS 2011) Out: Wed, June 29 Due: Wed, July 13

# Exercise Sheet 10

Jun.-Prof. Roland Meyer, Georg Zetzsche Technische Universität Kaiserslautern

## Exercise 10.1

Consider the following C-like code, in which `int r()` is a function that randomly returns `0` or `1`. Present a pushdown system that models the program. Thereby, describe how you decided to represent the program counter and the value of variable `x`. *Hint:* In a first step, assign each line in the code a unique label.

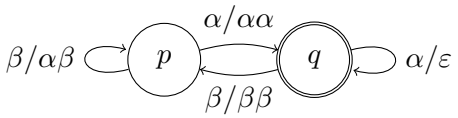
```
void m()
{
    x=1;
    if(r()==1) s();
}
```

```
void s()
{
    x=0;
    if(r()==1) m();
}
```

```
int x=0;
void main()
{
    m();
}
```

## Exercise 10.2

In the lecture, you learned about an algorithm to check whether there exists an accepting run starting at a given configuration of a BPDS. Use this algorithm to decide whether the following BPDS admits an accepting run starting at $(p, \beta\beta)$:




*Hint:* When searching for pairs $(q, \gamma)$ that satisfy conditions $(1')$ and $(2')$, start with $(q, \beta)$. It will already yield the desired result.

## Exercise 10.3

In the aforementioned algorithm, the set $\mathrm{pre}^+(C)$ is calculated by determining $\mathrm{pre}^*(C)$ and then $\mathrm{pre}(\mathrm{pre}^*(C))$. Thus, it was necessary to have a method to compute $\mathrm{pre}(C)$ for a given set $C$. Present a general description of such a method. *Hint:* Look at the example from the lecture.

## Exercise 10.4

Let $\Sigma$ be ranked alphabet. For a $\Sigma$-tree $t$, let $\mathrm{yield}(t)$ be the word obained from $t$ by reading the labels of the leaves from left to right. A context-free grammar is *non-erasing* if the empty word does not occur on the right side of any production.

(a) Show that for each regular tree language $L$, there is a non-erasing context-free grammar $G$ with $\text{yield}(L) = \mathsf{L}(G)$.

(a) Show that for each non-erasing context-free grammar $G$, there is a regular tree language $L$ with $\text{yield}(L) = \mathsf{L}(G)$.