CSCI 341 Problem Set 6

Stack Automata

Due Friday, October 17

Don't forget to check the webspace for hints and additional context for each problem!

Problem 1 (Pop-Push)**.** By induction on the length of the program, prove that every valid stack program is equivalent to either `skip` or a program of the following form:

$$\texttt{pop}\ \sigma_1.\texttt{pop}\ \sigma_2 \dots \texttt{pop}\ \sigma_n.\texttt{push}\ \tau_1.\texttt{push}\ \tau_2 \dots \texttt{push}\ \tau_m \qquad (*)$$

for some $n, m \in \mathbb{N}$.

Solution. □

Problem 2 (2 is better than 1)**.** Consider the language

$$L = \{w \mid w = w^{\mathrm{op}}\}$$

in the alphabet $A = \{a, b, c\}$. Design a stack automaton $\mathcal{S} = (Q, A, \Sigma, \delta, F)$ with a state $x \in Q$ such that $L = \mathcal{L}(\mathcal{S}, x)$.

Solution. □

Problem 3 (Algorithmic Balancing)**.** Let L be the set of balanced strings of brackets, where the bracket symbols are $A = \{(,), [,], \{,\}\}$, from Balancing Act. In that problem, you designed a grammar with a variable that derives L.

(1) Use the Grammar-to-Automaton construction to design a stack automaton with a state that accepts L.

(2) Can you think of a smaller stack automaton with a state that accepts L?

Solution. □

Problem 4 (Basic Stack Programs Unite)**.** Fix a set of stack symbols Σ. Let $p_1, p_2, \dots, p_n \in \{\texttt{push}\ \sigma, \texttt{pop}\ \sigma \mid \sigma \in \Sigma\}$, i.e., let them be basic stack programs. Assume that $p_1 \dots p_i = \texttt{skip}$ if and only if $i = 0$ or $i = n$. Now prove that

(1) $p_1 p_n = \texttt{skip}$

(2) $p_2 \dots p_{n-1} = \texttt{skip}$

Solution. □

Problem 5 (Intersection-product Construction)**.** Let $A = \{c, a, t, d, o, g\}$. Design a counter automaton with a state x that accepts the language L_{cat} of all words $w \in A^*$ such that the string "*cat*" appears in w more times than "*dog*" appears in w.

Solution. □

Problem 6 (Comes in 3s). Consider the language

$$L = \{a^n b^n \mid n \in \mathbb{N} \text{ and } n = 3k + 1 \text{ for some } k \in \mathbb{N}\}$$

(1) Use the Intersection-Product conctruction to design a stack automaton $\mathcal{S} = (Q, A, \Sigma, \delta, F)$ with a state $x \in Q$ such that $L = \mathcal{L}(\mathcal{S}, x)$.

(2) Design a grammar $\mathcal{G} = (X, A, R)$ with a variable that derives L.

Solution. □