CS331 Assignment #2

Due Date: Oct 21, 2002

Ground rules: Be brief, complete and stick to what has been asked. If needed, you may cite results/proofs covered in class without reproducing them.

Do not copy solutions from others.

1. *[5 + 10 (+ bonus 10) marks]* We have studied PDAs in class. Suppose we use DFAs coupled with a queue to get QAs (queue automata). Assume that elments are inserted at the *tail* of a queue and removed from its *head*. A QA can then be described as $M = (Q, \Sigma, \Gamma, \delta, q_0, X_0, F)$ where Q, Σ, q_0 and F are the same as in a PDA. Γ is the queue alphabet, $X_0 \in \Gamma$ is the initial symbol in the queue, and $\delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \times Q \times \Gamma^*$ is the (possibly non-deterministic) transition function.

   Let $(q, a, X, r, YZW) \in \delta$. This tuple signifies that when M is in state q with a as the next input symbol and X as the element at the head of the queue, then the machine transitions to state r, removes X from the head of the queue, and inserts Y, Z and W (in that order) at its tail. If $a = \epsilon$, the state transition and queue updations happen without consuming any input symbol (analogous to an ϵ-transition in a PDA). If multiple moves are possible from state q on input a with X at the queue head (i.e., multiple tuples in δ with the first three components same), one move is chosen non-deterministically (as in a PDA). A transition (q, a, X, r, YZW) in a QA is represented pictorially as shown in Fig. 1.

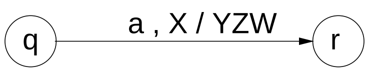

Figure 1:

   As with PDAs, an ID of M is a tuple (q, x, γ) where $q \in Q$ is the current state, $x \in \Sigma^*$ is the unconsumed part of the input, and γ

is the queue contents written (left-to-right) from head to tail. We say $(q, x, \gamma_1) \overset{\star}{\vdash} (r, s, \gamma_2)$ if there exists a sequence of zero or more moves that takes M from (q, x, γ_1) to (r, s, γ_2). A string $w \in \Sigma^*$ is accepted by M if $(q_0, w, X_0) \overset{\star}{\vdash} (q_f, \epsilon, \gamma_3)$, where $q_f \in F$ and $\gamma_3 \in \Gamma^*$. The language $L(M)$ of QA M is the set of all strings in Σ^* that are accepted by M.

(a) Give a QA M such that $L(M) = \{0^n 1^n 2^n \mid n \geq 1\}$. Explain (at least informally) why M accepts all strings in this language and nothing else.

(b) Show that for every PDA P, one can find a QA M such that $L(P) = L(M)$.
[Hint: You need to show that every move of P can be simulated using (possibly several) moves of M.]

(c) *[Bonus marks question:]* Given an arbitrary Turing Machine T, is it always possible to find a QA M such that $L(M) = L(T)$? Recall $L(T)$ is the set of strings accepted by T by entering a final state. Give justification for your answer.

2. *[10 + 10 marks]* In the following subquestions, you must determine whether the given languages are CFLs. An affirmative answer *must* be accompanied by a CFG or a PDA that accepts the language. A negative answer *must* be accompanied by a proof. The input alphabet Σ is $\{0, 1, 2\}$. The number of 0's, 1's and 2's in a string w are represented by $N_0(w)$, $N_1(w)$ and $N_2(w)$ respectively.

(a) $\{w \mid w \in \Sigma^*,\ N_0(w) - N_1(w) = N_2(w)\}$.

(b) $\{w \mid w \in \Sigma^*,\ |N_0(w) - N_1(w)| = N_2(w)\}$.

3. *[10 marks]* Let L be the CFL defined by the following grammar, where S and A are non-terminals, 0 and 1 are terminals, and S is the starting symbol:
$S \rightarrow 0S|0A|S1;\ A \rightarrow 0A1|1S0|1.$

Define a homomorphism h as follows: $h(0) = h(1) = 2$. Show that $h(L)$ is a regular language in $\mathbf{2}^*$.