Final Examination

Formal Language and Automata Theory 2009/01/10

1. (10 pts) Fill in the following table about Chomsky hierarchy.

language type	language class name	recognition machine model	grammar model
type 3		Finite Automata	
type 2	context free language		
type 1			
type 0			

2. [15 pts] Let A = { $x \in \{0,1\}^*$ | x contains an odd number of 1's}.

   (a) Design a DFA M (using transition diagram representation) to accept A. [5pts]

   (b) Find a right linear grammar G equivalent to M. [10pts]

3. [20 pts] Let B = $\{a^{2n}b^n \mid n \geq 0\}$.

   (a) List the first 4 shortest members of B. [4pts]

   (b) Design a CFG G such that L(G) = B. [8pts]

   (c) Design a PDA M to accept B. The machine can have only one state q and accepts by empty stack. Your answer should include 1. the input alphabet Σ, 2. the stack alphabet Γ and 3. the transition diagram representation of the transition relation δ.

4. [20 pts] Problem reduction:

(a) What does it mean that a language A is reducible to another language B ? Please give a formal **mathematical** definition for it. [5pts]

(b) Explain why if A is reducible to B and A is undecidable, then B must be undecidable. [5pts]

(c) Let H010 $=_{def}$ {"M" | Turing machine M *halts* on the input: `'010'`.}, and EH$=_{def}$ {"M" | Turing machine M halts on empty input ε.}. It is known that EH is undecidable. Show that H010 is also undecidable by reducing EH to H010. (Hint: Given a Turing machine M, could you construct a new machine M* s.t. M halts on ε iff M* halts on the input: `010` ?)

5. [15pts] It is known that the PCP problem is reducible to the Non-Disjoint problem of two context-free grammars (NDCFG). Namely, given a Post corresponding system C, we can generate from it two CFGs Gx and Gy such that C has a solution if and only if L(Gx)$\cap$ L(Gy) $\neq\emptyset$. Now suppose the given C is {(110,11),(110,10),(0,001),(10,01) }.

(a) What are the corresponding context-free grammars Gx and Gy ? (10 pts)

(b) Does C have a solution? Find a solution to C if it has one. (5pts)

6. [9pts] Let G be the context free grammar:

S → ASB | ε   A → S | aAS | ε   B → SbS | A | bb

(a) Find a grammar G_1 which has no ε-rule and $L(G_1) = L(G) - \{\varepsilon\}$. [5pts]

(b) Find a grammar G_2 which is equivalent to G_1 and has no unit productions

7. [6pts] Let G be the CFG: S → aAbB A→ab | BBA B →ASB | b

   Find a grammar G_3 which is equivalent to G and is in Chomsky normal form.

8. (15 pts) Answer whether the following language classes are closed under a given operation by filling in each cell of the following table with yes (0) or no (x).

Operation Language lass	Union	Intersection	complement	Concatenation	Kleene Star (i.e., *)
r.e.				O	O
Recursive				O	O
Context-free			X		
Regular					

9. [10 pts] True(O) or false(X) questions. Note: total = halts on all inputs; r.e. set = accepted by a TM; recursive set = accepted by a total TM.
   (a) Every context free language can be accepted by a total Turing machine.
   (b) $\{a^k b^k \mid k \geq 0\}$ is a context-free language.
   (c) If L is recursive, then so is its complement.
   (d) {“M” | M is a TM and | L(M) |> 20 } is r.e.
   (e) {“M “| M is a TM and L(M) ≠∅ } is recursive.
   (f) The problem of determining whether a given a PDA M1 and a given FA M2 accept the same language is decidable.
   (g) If L is r.e. but not recursive, then its complement is r.e.
   (h) 3-tape Turing machines are more powerful than 2-tape Turing machines.
   (i) 2-stack PDAs are more powerful than single stack PDAs.
   (j) {“G “| G is a CFG and L(G) ≠∅ } is recursive.

10. [15pts] As discussed in class, the Turing Language(TL) is defined to be the set:

    {e(M) ∈ {0,1,a,q,’,(,) }* | M is a standard Turing Machine(STM) }

    of all encodings e(M) of STMs. For each STM M = (Q,Σ,Γ, [, □, δ, s,t,r), let = $\lceil \log_2 |Q| \rceil$ and n = $\lceil \log_2 (|G|+2) \rceil$ be the minimum number of bits required to represent all states and all tape symbols together with move left/right actions, respectively, of M. For each state p, we designate a unique string e(p) = ‘q$b_1b_2\ldots b_m$’ to represent it, and for each symbol t ∈ $\Gamma\cup$ {L,R}, we designate a string e(t) = ‘a$c_1c_2\ldots c_n$’ to represent it, where all b_is and c_js are bits. Let $\alpha_1\alpha_2\ldots\alpha_\kappa$ be any sequence of all transitions of the machine where,for $1\leq j\leq k$, a_j = (p_j,a_j,q_j, b_j) ∈ QxΓxQx{Γ∪ {L,R}}. We then define e(M) = e(α_1)e(α_2)…e(α_κ) where e(α_j) = ‘(‘ e(p_j) ‘,’ e(a_j) ‘,’ e(q_j) ‘,’ e(b_j) ‘)’ be the concatenation of the encoding of all four instruction components separated by the symbol’,’ and enclosed by ‘(‘ and ‘)’. Now describe briefly how a 3-tape universal Turing machine U can be designed to simulate the execution of all STM M on any input w. In other words, for the machine U we should have the property: M accepts (rejects, loop) on input w with the result M(w) on the tape iff U accepts(rejects, loop) on input e(M) e(w) with the result U(e(M(w))) on the first tape of U.