I118 グラフとオートマトン理論 (Graphs and Automata)
テスト

2011 年 9 月 10 日（土） 3 限 (13:30–15:00)
上原 隆平 (uehara@jaist.ac.jp)

学生番号 (Student ID)________________________ 氏名 (Name)________________________

全 6 問から 5 問選んで，その解答を対応する問題の下の余白に書くこと．余白が足りない場合は裏を使ってもよい．各問 20 点で上位 5 問の合計を得点とする．Choose 5 of 6 problems and answer them. Write each answer in the blank after the problem. If it is not enough, use the back. Each answer is up to 20 points, and your score is the summation of top 5 of the answers.

Problem 1: N を非負整数の集合とする．つまり $N = \{0, 1, 2, 3, \ldots\}$．このとき，次の命題が間違っているか正しいかを明記して，それぞれの理由を説明せよ．(Let N denote the set of non negative integers. That is, we have $N = \{0, 1, 2, 3, \ldots\}$. Then, for each of the following propositions, specify that it is correct or wrong with the description of reason.)

1. $(\forall x \in N)(\exists y \in N)[x + y = 0]$
2. $(\forall x \in N)(\exists y \in N)[x \times y = 0]$
3. $(\exists x \in N)(\exists y \in N)[x \times y = 1]$
4. $(\exists x \in N)(\forall y \in N)[x \times y = 0]$

Problem 2: $G = (V, E)$ を無向グラフとする．頂点 v に接続している辺の本数をその頂点の**次数**といい，$\deg(v)$ で表す．このとき，$\sum_{v \in V} \deg(v) = 2\,|E|$ が成立する．(つまり頂点の次数をすべて加えると，辺の本数のちょうど 2 倍になる．) これを証明せよ．(Let $G = (V, E)$ be a graph. The *degree* of a vertex v, denoted by $\deg(v)$, is the number of edges incident to v. Then $\sum_{v \in V} \deg(v) = 2\,|E|$ holds. That is, the summation of all degrees makes twice of the number of edges in the graph. Prove it.)

Problem 3: 連結で閉路を持たないグラフを**木**という．n 頂点の木の辺の本数は必ず $n-1$ 本である．また頂点 v に接続している辺の本数をその頂点の**次数**という．木において，次数が 1 である頂点を**葉**という．2 頂点以上からなる木は必ず葉をもつことを証明せよ．ただし前問の $\sum_{v \in V} \deg(v) = 2|E|$ が成立することを既知の事実として使ってよい．(A connected acyclic graph is called a *tree*. Any tree with n vertices consists of $n-1$ edges. The *degree* of a vertex v, denoted by $\deg(v)$, is the number of edges incident to v. A vertex of degree 1 is called a *leaf*. Prove that any tree with at least 2 vertices has a leaf. You can use the equation $\sum_{v \in V} \deg(v) = 2|E|$ in the previous problem as a known fact.)

Problem 4: 決定性有限オートマトン $D = (Q, \Sigma, \delta, q, F)$ が与えられたとき，D が受理する言語 $L(D)$ の否定 $L(\bar{D}) = \{x \mid x \notin L(D)\}$ を受理する決定性有限オートマトン $\bar{D} = (Q', \Sigma, \delta', q', F')$ を構成せよ．Q', δ', q', F' を形式的に明確に記述すること．(Let $D = (Q, \Sigma, \delta, q, F)$ be a finite deterministic automaton. Then construct the finite deterministic automaton $\bar{D} = (Q', \Sigma, \delta', q', F')$ that accepts the language $L(\bar{D}) = \{x \mid x \notin L(D)\}$, which is the complement the language $L(D)$ accepted by D. Describe Q', δ', q', and F' clearly and formally.)

Problem 5: 決定性有限オートマトン D が与えられたとき，D が受理する言語 $L(D)$ の否定 $L(\bar{D}) = \{x \mid x \notin L(D)\}$ を受理する決定性有限オートマトン $\bar{D}$ が構成できるとする．また，2 つの決定性有限オートマトン D_1 と D_2 が与えられたとき，D_1 が受理する言語 $L(D_1)$ と D_2 が受理する言語 $L(D_2)$ の和集合 $L(D_1) \cup L(D_2)$ を受理する決定性有限オートマトン D' が構成できるとする．これらを仮定したとき，与えられた 2 つの決定性有限オートマトン D_1 と D_2 に対して，D_1 が受理する言語 $L(D_1)$ と D_2 が受理する言語 $L(D_2)$ の積集合 $L(D_1) \cap L(D_2)$ を受理する決定性有限オートマトン D'' が構成できることを示せ．(For any given finite deterministic automaton D, We suppose that we can construct the finite deterministic automaton $\bar{D}$ that accepts the complement $L(\bar{D}) = \{x \mid x \notin L(D)\}$ of the language $L(D)$ accepted by D. Moreover, for any given finite deterministic automata D_1 and D_2, we suppose that we can construct a finite deterministic automaton D' that accepts the union $L(D_1) \cup L(D_2)$ of $L(D_1)$ and $L(D_2)$. Then, show that for any given finite deterministic automata D_1 and D_2, we can construct a finite deterministic automaton D'' that accepts the intersection $L(D_1) \cap L(D_2)$ of $L(D_1)$ and $L(D_2)$.)

Problem 6: カッコ，つまり $\Sigma = \{(,)\}$ 上の言語を考える．いわゆるカッコ “(” と “)” のバランスがとれている列だけを生成したい．例えば ()，(())，()()，(()()) はバランスがとれているが，(，())，)(，)()(，())(() はバランスがとれていない．つまり**バランスのとれたカッコ列**とは

- “(” と “)” の個数が同じ
- 文字列を前から途中まで見たとき，[“(” の個数]−[“)” の個数]≥ 0 が成立する

を満たす Σ 上の文字列である．すべてのバランスのとれたカッコ列だけを生成する文脈自由文法 $G = (V, \Sigma, P, S)$ を与えよ．(We consider a language of parentheses, i.e., $\Sigma = \{(,)\}$. The language consists of sequences of balanced string over “(” and “).” For example, (), (()), ()(), and (()()) are balanced, but (, ()),)(,)()(, and ())(() are not. That is, a sequence is *balanced* if and only if

- the number of “(”s is equal to the number of “)”s, and
- [the number of “(”s]−[the number of “)”s]≥ 0 holds for any prefix subsequence of the sequence.

Give a context free grammar $G = (V, \Sigma, P, S)$ that only generates all balanced sequences.)