ECE 198 Fall 2019 Jin Yucheng (yucheng9)
Homework 4 Problem 1-4

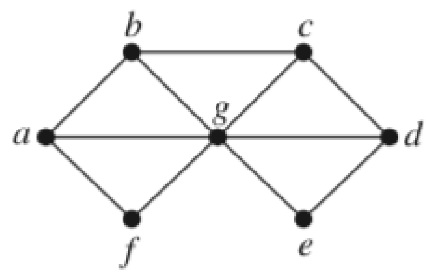
Solution:

(1) Prove that the iterative program for finding F_n given in Section 5.4 is correct: Fib(x): if $x =$ 0, return 0; if $x = 1$, return 1; else, return Fib$(x-1)$ + Fib$(x-2)$.

- Basis Step: when $x = 0$, Fib(0) successfully returns 0 and when $x = 1$, Fib(1) successfully returns 1, Fib(x) is correct when $x = 0$ or 1.
- Inductive Step: Assume for non-negative integers k and n ($n \geq 2$), for each k, such that $k \leq n$, Fib(k) is correct. Therefore, Fib$(n+1)$ will return Fib(n) + Fib$(n-1)$, by the inductive hypothesis, Fib(n) and Fib$(n-1)$ correctly compute F_n and F_{n-1}, therefore, Fib$(n+1)$ correctly computes F_{n+1}.

So the iterative program for finding F_n given in Section 5.4 is correct.

(2) Find the chromatic number of the graph below and prove your result with two-way bounding



The chromatic number of the above graph is 3.

- The chromatic number ≤ 3: since if we assign red to g, green to b, d, f, and blue to a, c, e, then no adjacent vertices have the same color, so the chromatic number is at most 3.
- The chromatic number ≥ 3: since b is adjacent to g, b and g must have different colors; g is adjacent to a, a and g must have different colors; a is adjacent to b, a and b must have different colors. Therefore, a, b, g all have different colors pairwisely, then the chromatic number is at least 3.

(3) Let G_1 and G_2 be CFGs with languages $L(G_1)$ and $L(G_2)$ respectively. Prove there exists a CFG with language $L(G_1) \cup L(G_2)$.

Denote language $L(G_1)$ generated by CFG G_1 as $\{V_1, T_1, S_1, P_1\}$, language $L(G_2)$ generated by CFG G_2 as $\{V_2, T_2, S_2, P_2\}$. Denote G as $\{V, T, S, P\}$, where,

- $V = V_1 \cup V_2 \cup S$, where S denotes the new start symbol.
- $T = T_1 \cup T_2$.
- S is the new start symbol.
- $P = P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}$.

Then $L(G)$ is the union of $L(G_1)$ and $L(G_2)$, therefore, there exists a CFG G with $L(G) = L(G_1) \cup L(G_2)$.

(4) Prove that if FSM A accepts language L_a and FSM B accepts language L_b, there exists a FSM that accepts language $L_c = \{xy \mid x \in L_a \text{ and } y \in L_b\}$ by constructing that FSM and using two-way bounding to show that your FSM accepts exactly L_c.

Denote FSM A as $\{S_a, I_a, O_a, f_a, g_a, s_{0,a}\}$, FSM B as $\{S_b, I_b, O_b, f_b, g_b, s_{0,b}\}$.
Denote FSM C as $\{S_c, I_c, O_c, f_c, g_c, s_{0,a}\}$, where,

- $S_c = S_a \cup S_b$.
- $I_c = I_a \cup I_b$.
- $O_c = O_a \cup O_b$.
- $f_c = f_a \cup f_b \cup f_{new}$, where the states of f_{new} are states $s_i \in S_a$, such that there exists an input j, and another state $s_k \in S_a$, that $f(s_k, j) = s_i$ and $g_a(s_k, j) = 1$; the inputs of f_{new} are elements in I_b, denote l as one element in I_b. $g_c = g_{a'} \cup g_b \cup g_{new}$, where $g_{a'}$ assigns 0 for each state and input pair in FSM A.
  $f_{new}(s_i, l) = f_b(s_{0,b}, l)$, $g_{new}(s_i, l) = g_b(s_{0,b}, l)$ (actually f_c, is not a total function because there are some implicitly rejected states).
- $s_{0,a}$ is the start state.

Then prove C accepts exactly L_c.

- for every string $w = xy$ such that $x \in L_a$ and $y \in L_b$, since if $x \in L_a$, then the last output bit is 1, because all states $s_i \in S_a$ with a transition function $f(s_k, j) = s_i$ and an output function $g_a(s_k, j) = 1$ is assigned $f_{new}(s_i, l) = f_b(s_{0,b}, l)$ for every input l in I_b, so all strings accepted by A with at least one input l in I_b can transit to B, and B accepts all strings in L_b, therefore C accepts every string $w = xy$ such that $x \in L_a$ and $y \in L_b$.
- for every string $w = xy$ such that $x \notin L_a$ or $y \notin L_b$, because if $x \notin L_a$, then w is not accepted by L_a, and there is no transition function that can transit the state upto input string xl ($l \in I_b$) to B; if $y \notin L_b$, then the last output digit must be 0, therefore C rejects every string $w = xy$ such that $x \notin L_a$ or $y \notin L_b$.

■