Massachusetts Institute of Technology
6.042J/18.062J, Fall '02: Mathematics for Computer Science
Professor Albert Meyer and Dr. Radhika Nagpal

Problem Set 5

Reading: Week 5 Notes

Problem 1. You are given two buckets, A and B, a water hose, a receptacle, and a drain. The buckets and receptacle are initially empty. The buckets are labeled with their respectively capacities, positive integers a and b. The receptacle can be used to store an unlimited amount of water, but has no measurement markings. Excess water can be dumped into the drain. Among the possible moves are:

1. fill a bucket from the hose,
2. pour from the receptacle to a bucket until the bucket is full or the receptacle is empty, whichever happens first,
3. empty a bucket to the drain,
4. empty a bucket to the receptacle,
5. pour from A to B until either A is empty or B is full, whichever happens first,
6. pour from B to A until either B is empty or A is full, whichever happens first.

(a) Model this scenario with a state machine. (What are the states? How does a state change in response to a move?)

(b) Prove that we can put $k \in \mathbb{N}$ gallons of water into the receptacle using the above operations if and only if $\gcd(a, b) \mid k$. *Hint:* Use the fact that if a, b are positive integers then there exist integers s, t such that $\gcd(a, b) = sa + tb$ (proven Week 5 Notes, §5.4).

Problem 2. The following algorithm, called the *Russian peasants algorithm*, can be used to multiply any two natural numbers x and y using only addition, left bit-shift (i.e., multiply by 2), and right bit-shift (i.e., divide by 2 and drop the remainder, if any) operations. The answer is accumulated in variable a; variables r and s are for temporary storage.

```
r := x;
s := y;
a := 0;
do until s = 0:
    if s is even then
        r := 2r;
        s := s/2;
    else
        a := a + r;
        r := 2r;
        s := (s − 1)/2;
```

The answer xy is the value left in the accumulator a when the procedure terminates.

(a) Model the algorithm as a state machine. That is, define the states Q, the start states Q_0, and the transitions.

(b) List the sequence of states that appears in an execution of the algorithm for inputs $x = 5$ and $y = 9$.

(c) Find an invariant that implies the algorithm is partially correct, that is, that $s = 0$ implies $a = xy$.

(d) Prove that the algorithm terminates.

Problem 3. Four VI-A students would like a position at a company. There are four companies, each with one position for a VI-A student. Below are the students' rankings of the companies and the companies' rankings of the students.

Student	Ranking of Companies
Nikos	Lotus, IBM, Compaq, Akamai
George	Compaq, Lotus, Akamai, IBM
Radhika	Compaq, Lotus, IBM, Akamai
Tina	Akamai, Compaq, Lotus, IBM

Company	Ranking of Students
Compaq	George, Tina, Nikos, Radhika
Lotus	George, Radhika, Nikos, Tina
IBM	Radhika, George, Nikos, Tina
Akamai	Tina, George, Radhika, Nikos

Based on the rankings, the VI-A office will assign one student to each company so the assignment forms a stable marriage set.

(a) There are VI-A student, S, and a company, C, that would form a rogue pair in *any* assignment in which S was not assigned to C. Give an example of such a pair (S, C), and briefly explain why your example has this property.

(b) Verify that in this case, the Mating Algorithm yields the same assignment whether Students are treated as Boys or as Girls.

(c) Explain why this implies that there is only one possible stable assignment (even including possible stable assignments that may not be produced by the Gale/Shapley Mating Algorithm).

Problem 4. The Stable Buddy Problem is a variant of the Stable Marriage Problem without the constraint that matched pairs must be Boy with Girl.

In contrast to the Boy-Girl Marriage Problem, there are buddy preferences where all buddy assignments are unstable. Give an example of four people and their preference rankings for which there is no stable way to assign buddies. Explain why.

Problem 5. Here is the generalization of the "choose-a-pair" game from Week 5, Friday Class Problems to "choose-a-triple." The rules are:

Player 1 chooses any triple in $\mathbb{N}^3$. Then, starting with Player 2, the players alternate moves, choosing as a move any triple, $\mathbf{t} \in \mathbb{N}^3$, such that no previous move is $\preceq_c \mathbf{t}$. A player wins when the other player chooses the origin (0,0,0).

For example, Player 1 might choose the 3-tuple (8, 9, 10). Possible subsequent choices might then be

$$(7, 8, 9), (0, 1, 67), (83, 0, 0), (1, 0, 0), (0, 0, 1)(0, 1, 0)$$

This finally leaves Player 2 with only the move (0,0,0), and the game now ends with his loss.

Prove that there is a winning strategy for the choose-a-triple game.