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

Solutions to In-Class Problems — Week 5, Mon

Problem 1. By now you are very familiar with the 6.042 icon that appears on every lecture slide and course webpage. This icon is a picture of a game called the **Fifteen Puzzle**. In this problem you will establish a basic property of the Fifteen Puzzle using the method of invariants, and, if this exercise works as planned, you will come to appreciate why this icon was chosen as the 6.042 logo.

The Fifteen Puzzle consists of sliding square tiles numbered $1, \ldots, 15$ held in a 4×4 frame with one empty square. Any tile adjacent to the empty square can slide into it.

The standard initial position is

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

We would like to reach the target position (known in my youth as "the impossible" — ARM):

15	14	13	12
11	10	9	8
7	6	5	4
3	2	1	

A state machine model of the puzzle has states consisting of a 4×4 matrix with 16 entries consisting of the integers $1, \ldots, 15$ as well as one "empty" entry—like each of the two arrays above.

The state transitions correspond to exchanging the empty square and an adjacent numbered tile. For example, an empty at position $(2, 2)$ can exchange position with tile above it, namely, at position $(1, 2)$:

n_1	n_2	n_3	n_4		n_1		n_3	n_4
n_5		n_6	n_7	$\longrightarrow$	n_5	n_2	n_6	n_7
n_8	n_9	n_{10}	n_{11}		n_8	n_9	n_{10}	n_{11}
n_{12}	n_{13}	n_{14}	n_{15}		n_{12}	n_{13}	n_{14}	n_{15}

We will use the invariant method to prove that there is no way to reach the target state starting from the initial state.

We begin by noting that a state can also be represented as a pair consisting of two things:

1. a list of the numbers $1, \ldots, 15$ in the order in which they appear—reading rows left-to-right from the top row down, ignoring the empty square, and
2. the coordinates of the empty square—where the upper left square has coordinates $(1, 1)$, the lower right $(4, 4)$.

(a) Write out the "list" representation of the start state and the "impossible" state.

Solution. start: ((1 2 ... 15), (4,4)),
impossible: ((15 14 ... 1), (4,4)), ■

Let L be a list of the numbers $1, \ldots, 15$ in some order. A pair of integers is an *out-of-order pair* in L when the first element of the pair both comes *earlier* in the list and *is larger*, than the second element of the pair. For example, the list $1, 2, 4, 5, 3$ has two out-of-order pairs: (4,3) and (5,3). The increasing list $1, 2 \ldots n$ has no out-of-order pairs.

Let a state, S, be a pair $(L, (i, j))$ described above. We define the *parity* of S to be the mod 2 sum of the number of out-of-order pairs in L and the row-number of the empty square (*i.e.*, $(L+i) \bmod 2$).

(b) Verify that the parity of the start state and the target state are different.

Solution. The parity of the start state is

$$(0 + 4) \bmod 2 = 0.$$

The parity of target is

$$((15 \cdot 14/2) + 4) \bmod 2 = 1.$$

■

(c) Show that the parity of a state is invariant under transitions. Conclude that "the impossible" is impossible to reach.

Solution. To show that the parity is constant, consider how moves may affect the parity. There are only 4 moves types of moves: a move to the left, a move to the right, a move to the row above, or a move to the row below.

Note that horizontal moves change nothing, and vertical moves both change i by 1, and move a tile three places forward or back in the list, L. To consider how the parity is changed in this case, we need to consider only the 3 pairs in L that are between the tile's old and new position. (The other pairs are not effected by the tile's move). This reverses the order of three pairs in L, changing the number of inversions by 3 or 1, but always by an odd amount.

To confirm this last remark, note that if the 3 pairs were all out of order or all in order before, the amount is changed by 3. If two pairs were out of order and 1 pair was in order or if one pair was out of order and two were in order, this will change the amount by 1. So the sum of i and the number of out-of-order pairs changes by an even amount (either 1+3 or 1+1), which implies that its parity remains the same. Since the initial state has parity 0 (even), all states reachable from the initial state must have parity 0, so the target state with parity 1 can't be reachable. ■

By the way, if two states have the same parity, then in fact there *is* a way to get from one to the other. If you like puzzles, this is a good one to think about on your own about after class.

Problem 2. The most straightforward way to compute the bth power of a number, a, is to multiply a by itself b times. This of course requires $b-1$ multiplications. There is another way to do it using considerably fewer multiplications. This algorithm is called *Fast Exponentiation*:

Given inputs $a \in \mathbb{R}, b \in \mathbb{N}$, initialize registers x, y, z to $a, 1, b$ respectively, and repeat the following sequence of steps until termination:

- if $z = 0$ **return** y and terminate
- $r :=$ remainder$(z, 2)$
- $z :=$ quotient$(z, 2)$
- if $r = 1$, then $y := xy$
- $x := x^2$

We claim this algorithm always terminates and leaves $y = a^b$.

(a) Model this algorithm with a state machine, carefully defining the states and transitions.

Solution.
1. $Q ::= \ \mathbb{R} \times \mathbb{R} \times \mathbb{N}$,
2. $Q_0 ::= \ \{(a, 1, b)\}$,
3. transitions

$$(x, y, z) \to \begin{cases} (x^2, y, \text{quotient}(z, 2)) & \text{if } z \text{ is positive and even,} \\ (x^2, xy, \text{quotient}(z, 2)) & \text{if } z \text{ is positive and odd.} \end{cases}$$

■

(b) Let $d ::= a^b$. Verify that the following predicate, P, is an invariant:

$$P((x, y, z)) ::= \ [yx^z = d].$$

Solution. See Week 5 Notes, §5.3. ■

(c) Prove that the algorithm is partially correct: if it halts, it does so with $y = d$.

Solution. See Week 5 Notes, §5.3. ■

(d) Prove that the algorithm terminates.

Solution. See Week 5 Notes, §5.3. ■

(e) In fact, prove that it requires at most $2 \log_2 b$ multiplications for the Fast Exponentiation algorithm to compute a^b for $b > 1$.

Solution. The value of z is initially b and gets halved at least at every other step. So it can't be halved more than $\log_2 b$ times before hitting zero. ■