

Notes for Recitation 3

1 Well-Ordering

Here is a statement that appears, at first glance, to be as useless as it is obvious.

Well-Ordering Principle. Every nonempty subset of the natural numbers has a smallest element.

The first glance is deceiving! As for obvious, note that the well-ordering principle would be false if the natural numbers were replaced by, say, the real numbers, the integers, or even the nonnegative rational numbers. So the well-ordering principle is capturing something special about the natural numbers.

As for useless, everything provable with ordinary or strong induction can equally well be proved with the well-ordering principle! Thus, well-ordering is as powerful as the most important proof technique used in computer science. Let's do an example to demonstrate its utility.

Theorem. *There is no solution over the positive integers to the equation:*

$$4a^3 + 2b^3 = c^3$$

Proof. We use contradiction and the well-ordering principle. Let S be the set of all positive integers a such that there exist positive integers b and c that satisfy the equation.

Assume for the purpose of obtaining a contradiction that S is nonempty. Then S contains a smallest element a_0 by the well-ordering principle. By the definition of S , there exist corresponding positive integers b_0 and c_0 such that:

$$4a_0^3 + 2b_0^3 = c_0^3$$

The left side of this equation is even, so c_0^3 is even, so c_0 is even. Thus, there exists an integer c_1 such that $c_0 = 2c_1$. Substituting into the preceding equation and then dividing both sides by 2 gives:

$$2a_0^3 + b_0^3 = 4c_1^3$$

Now b_0^3 must be even, so b_0 is even. Thus, there exists an integer b_1 such that $b_0 = 2b_1$. Substituting into the preceding equation and dividing both sides by 2 again gives:

$$a_0^3 + 4b_1^3 = 2c_1^3$$

From this equation, we know that a_0^3 is even, so a_0 is even. Thus, there exists an integer a_1 such that $a_0 = 2a_1$. Substituting into the previous equation one last time and dividing by 2 one last time gives:

$$4a_1^3 + 2b_1^3 = c_1^3$$

Evidently, $a = a_1$, $b = b_1$, and $c = c_1$ is another solution to the original equation, and so a_1 is an element of S . But this is a contradiction, because $a_1 < a_0$ and a_0 was defined to be the smallest element of S . Therefore, our assumption was wrong, and the original equation has no solutions over the positive integers. \square

This argument is quite similar to the proof that $\sqrt{2}$ is irrational. In fact, looking back, we implicitly relied on the well-ordering principle in that proof when we claimed that a rational number could be written as a fraction in *lowest terms*. We've been using the well-ordering principle on the sly since Day 1!

Induction, strong induction, and well-ordering are logically equivalent, so the best choice for a particular application is the one that you think gives the clearest proof. In practice, induction and strong induction are more commonly used than well-ordering. But for the occasional problem, like proving that $\sqrt{2}$ is irrational, an argument based on well-ordering may be much nicer.

2 State Machines

Recall from Lecture 3 (9/14) that an *invariant* is a property of a system (in lecture, that system was the 8-puzzle) that does not change, regardless of the system's behavior. Now we'll take a look at how we can make use of an important modeling tool — a state machine — to analyze systems. We'll see that finding invariants of state machines can often help us prove propositions.

A *state machine* is an abstract model of a step-by-step process. The model consists of a collection of *states* and *transitions* between those states. More formally, we can define a state machine as

- a set of states: Q
- a designated start state: $q_0 \in Q$
- a set of allowed transitions between states: $\delta \subseteq Q \times Q$.

2.1 Example: Bounded Counter

Take, for example, a bounded counter that counts from 0 to 99 and overflows at 100. A diagram representing the corresponding state machine is shown in Figure 1. Note that in this example, there is a finite number of states. However, in general, the set of states might be infinite.

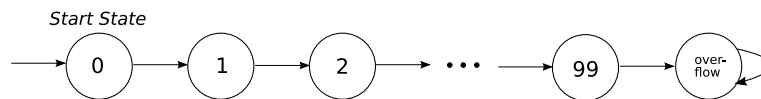


Figure 1: State machine for the bounded counter example. In the figure, the state labeled 0 is the start state. The self loop in the overflow state means that once the machine overflows, there is no way for it to transition out of this state.

2.2 Example: Unbounded Counter

An unbounded counter is similar to a bounded counter except that there is an infinite number of states and no overflow. See Figure 2.

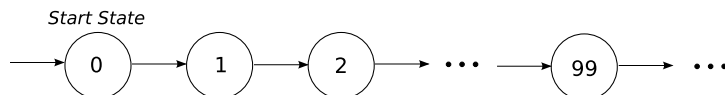


Figure 2: State machine for the unbounded counter example.

3 Problem: The Temple of Forever

Each monk entering the Temple of Forever is given a bowl with 15 red beads and 12 green beads. Each time the Gong of Time rings, a monk must do one of two things:

1. *Exchange*: If he has at least 3 red beads in his bowl, then he may exchange 3 red beads for 2 green beads.
2. *Swap*: He may replace each green bead in his bowl with a red bead and replace each red bead in his bowl with a green bead. That is, if he starts with i red beads and j green beads, then after he performs this operation, he will have j red beads and i green beads.

A monk may leave the Temple of Forever only when he has exactly 5 red beads and 5 green beads in his bowl.

Let's look at how we can represent this problem as a state machine.

- What do the states of the machine look like?

Solution. We can use variables such as r to represent the number of red beads and g to represent the number of green beads. Then the states can be represented as pairs (r, g) for $r \geq 0, g \geq 0$.

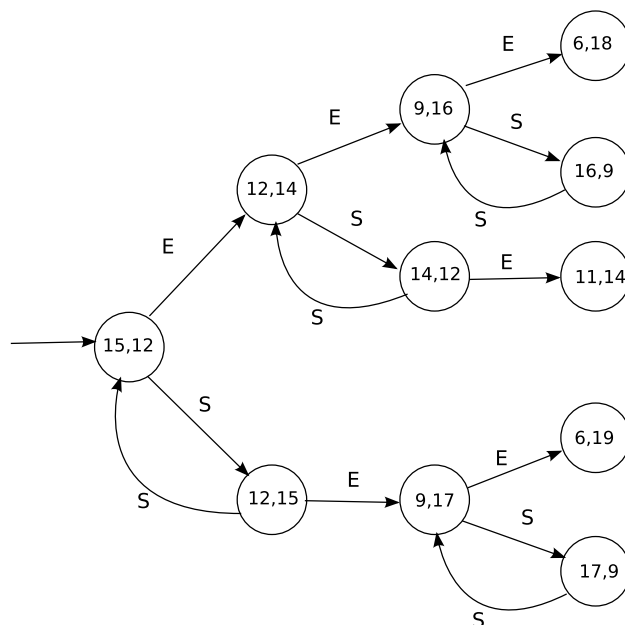
- Use the notation you developed above to represent the allowable transitions in the state machine.

Solution. There are two possible transitions:

1. trade 3 red for 2 green (*exchange*): $(r, g) \rightarrow (r - 3, g + 2), r \geq 3$
2. swap red and green (*swap*): $(r, g) \rightarrow (g, r)$

- Expand the state machine diagram to the first three or four levels. Label the transitions according to the operation type (E for *exchange* or S for *swap*).

Solution.



3.1 Invariants and State Machines

The Temple of Forever machine models every possible way of exchanging beads at each time step according to the rules. We would like to know whether the machine ever reaches the state $(5, 5)$. By starting at the start state and following the transition arrows through the state diagram, we can trace the possible paths, or *executions*, of the machine. So another way of stating our question is whether the state we are interested in appears in some execution of the machine. If so, then we can say that that state is *reachable*.

Definition 1. A state is called *reachable* if there is a path to it starting from the start state, that is, if it appears in some execution.

A useful technique for analyzing reachability is the identification of *invariants* of the machine. An invariant will hold for all reachable states of the machine. More formally,

Definition 2. An invariant for a state machine is a predicate P on state machines such that $P(q_0)$ holds (where q_0 is the start state of the machine), and whenever $P(q)$ is true of a state q , and $q \rightarrow r$ for some state r , then $P(r)$ holds.

Since, by definition, an invariant holds for the start state and for all transitions of the state machine, then we know (by induction) that the invariant must hold for all reachable states. Therefore, if we know some property to be an invariant, and we know that a certain state violates that property, then we can say that this state is unreachable.

Now we'll show that no monk can ever escape the Temple of Forever because the state $(5, 5)$ violates an invariant of the Temple of Forever machine.

Theorem 1. No one ever leaves the Temple of Forever.

Prove this theorem by induction. Begin by searching for an invariant that holds initially and is maintained by each operation, but would be violated by the condition required for departure.

Solution.

Proof. We use induction on the number of gong rings. Let $P(n)$ be the proposition that after n rings, the number of red beads in the monk's bowl minus the number of green beads is equal to $5k + 2$ or $5k + 3$ for some integer k .

Base case: $P(0)$ is true because initially (after zero rings) the number of red beads minus the number of green beads is $15 - 12 = 5 \cdot 0 + 3$.

Inductive step: Now assume that $P(n)$ holds after n gong rings, where $n \geq 0$. Let r denote the number of red beads in the monk's bowl, and let g denote the number of green beads. In these terms, we are assuming that $r - g$ is equal to $5k + 2$ or $5k + 3$ for some integer k . After $n + 1$ gong rings, there are two cases to consider, depending on the monk's action:

1. If $r \geq 3$, then the monk may have exchanged 3 red beads for 2 green beads. Thus, the number of red beads minus the number of green becomes:

$$(r - 3) - (g + 2) = (r - g) - 5$$

This is equal to either $5(k - 1) + 2$ or $5(k - 1) + 3$, so $P(n + 1)$ is true.

2. Alternatively, the monk may have swapped every red bead for a green bead and vice versa. In this case, the number of reds minus the number of greens becomes $g - r$. If $r - g = 5k + 3$, then $g - r = 5(-k) - 3 = 5(-k - 1) + 2$. If $r - g = 5k + 2$, then $g - r = 5(-k) - 2 = 5(-k - 1) + 3$. Thus, $P(n + 1)$ is again true.

Therefore, $P(n)$ implies $P(n + 1)$ for all $n \geq 0$.

By the induction principle, $P(n)$ is true for all $n \geq 0$. Since the number of red beads minus the number of greens is always of the form $5k + 2$ or $5k + 3$ and the difference required to leave the temple does not match either form, no monk can ever leave the Temple of Forever. \square

Now let's take a look at a different property of the Temple of Forever machine.

Theorem 2. *There is a finite number of reachable states in the Temple of Forever machine.*

Prove this theorem. (Hint: First find an invariant that suggests an upper bound on the number of reachable states. Be sure to prove the invariant.)

Solution.

We begin by noting that the Temple of Forever machine exhibits the following invariant:

Lemma 3. *For all reachable states, the total number of red beads and green beads in the monk's bowl — $r + g$ — is at most 27.*

Proof. We use induction on the number of gong rings. Let $P(n)$ be the proposition that after n gong rings, $r + g \leq 27$.

Base case: $P(0)$ is true because initially (after zero rings) the number of red beads plus the number of green beads is $15 + 12 = 27$.

Inductive step: Now assume that $P(n)$ holds after n gong rings, where $n \geq 0$. Let r denote the number of red beads in the monk's bowl, and let g denote the number of green beads. In these terms, we are assuming that $r + g$ is at most 27 after n gong rings. After $n + 1$ gong rings, there are two cases to consider, depending on the monk's action:

1. If $r \geq 3$, then the monk may have exchanged 3 red beads for 2 green beads. Thus, the number of red beads plus the number of green becomes:

$$(r - 3) + (g + 2) = (r + g) - 1$$

Since, by the inductive hypothesis, $r + g \leq 27$, it follows that $(r + g) - 1 \leq 27$, and so $P(n + 1)$ is true.

2. Alternatively, the monk may have swapped every red bead for a green bead and vice versa. In this case, the number of reds plus the number of greens becomes $g + r = r + g$. Thus, $P(n + 1)$ is again true by the inductive hypothesis.

□

We now prove the theorem by showing that there is a finite upper bound on the number of reachable states.

Proof. We give a direct argument. Lemma 3 tells us $r + g \leq 27$. This implies that there is an upper bound on the number of reachable states, since there can be at most 28 ways for r and g to sum to 27, 27 ways to sum to 26, and so on. Therefore, there can be at most $28 + 27 + \dots + 1 = \frac{29 \cdot 28}{2} = 406$ states in the Temple of Forever machine. Hence, there is a finite number of reachable states in the Temple of Forever machine. □

Inside the Temple of Forever, the Gong of Time rings on. As you may well imagine, the monks begin to recognize that no matter how many ways they try to exchange or swap their beads, they always seem to end up in some state they've already been in before! For one or two monks, this realization is all they need to propel them instantly into a state of enlightenment. For the overwhelming majority, however, this knowledge does nothing but weaken their resolve. They just get depressed. Taking note of the mental state of this second group, the Keeper of the Temple makes an unannounced appearance and proclaims to the

group, “From now on, any monk who is able to visit 108 (108 being the mystical number that encompasses all of existence¹) unique states will be allowed to leave the Temple of Forever.”

Do the monks have any chance of leaving the Temple of Forever?

Theorem 4. *It is not possible to visit 108 unique states in the Temple of Forever machine.*

Prove this theorem. (Hint: Consider a proof by contradiction.)

Solution.

Proof. The proof is by contradiction. Assume that it *is* possible for a monk to visit 108 unique states in some execution of the Temple of Forever machine, and consider the sequence of moves that the monk must have made to visit these states. Each move in the sequence must be either an *exchange* or a *swap*, since these are the only legal moves. Now, whenever the monk performs an *exchange* operation, the sum $r + g$ decreases by one:

$$(r - 3) + (g + 2) = (r + g) - 1$$

In contrast, swaps do not have any effect on the sum. Furthermore, we know that the sum $r + g$ must be at least 3 to perform an exchange operation. Therefore, there can be at most 25 exchange operations in the sequence.

Now consider swap operations: between each pair of exchanges in the sequence, there may be an unlimited number of swaps. However, only a single swap can take the monk to a new state: if at step k the monk is in state (r, g) , then at step $k + 2$, he will return to the same state. Therefore, an upper bound on the number of unique states in any execution of the machine is $25 + 26 + 1 = 52$ (if swaps are inserted at both the beginning and end of the sequence). But then this contradicts the assumption that the monk visits 108 unique states, so no monk ever leaves the Temple of Forever.

□

What is the true maximal number of unique states a monk can visit in any execution of the Temple of Forever machine? How can this number be achieved?

Solution.

The true maximum is 52. To achieve this number, the monk can perform sequential swaps and exchanges until he reaches the state $(5, 2)$ via an exchange. At this point, the longest path goes to $(2, 4)$, via an exchange, instead of $(2, 5)$, via a swap. This is because the path leading to $(2, 5)$ ends at $(2, 5)$, whereas the path leading to $(2, 4)$ continues with swaps and exchanges, with the final state being $(2, 0)$ (arrived at via a swap). However, the monk can reach 52 unique states if, at state $(5, 2)$, he performs two swap in a row to pick up state $(2, 5)$. Alternatively, the monk can perform two swaps at the start state, picking up state $(12, 15)$, and then continue with 25 pairs of sequential exchange, swap operations until he reaches $(2, 0)$. This also generates a path with 52 unique states.

¹See <http://astrologyforthesoul.com/vp/mysticalnumber108.html>. Also consider: $42 + 24 + 42 = 108$.