CS/Math 240: Introduction to Discrete Mathematics

Reading 6 : Invariants

Author: Dieter van Melkebeek (updates by Beck Hasti and Gautam Prakriya)

6.1 Invariants

In this reading we look at how induction plays a role in establishing properties of discrete systems (systems that evolve in discrete steps.).

An *invariant* is a property of the system that holds after every step. Usually, the state of the system after step $t+1$ only depends on the state after step t. This suggests that we can use induction on the number of steps t to prove that a property is an invariant. In particular, we can show that

1. The system has the property initially ($t = 0$).

2. If the system has the property after t steps, it has the property after $t+1$ steps as well.

The predicate $P(t)$ for the inductive proof would say "The system has the property after t steps."

Invariants are very useful when reasoning about discrete processes and play an important role in proving program correctness.

6.1.1 Finding Invariants

It turns out that finding the right invariant for a particular problem is not always easy. Let's start with an example where finding an invariant is not that hard.

Suppose we have a robot which walks on a 2-dimensional grid. The rows and columns of the grid are labeled by integers. Our robot starts at position $(0,0)$, and can only move diagonally, one square at a time. We show all possible moves the robot can make from square (x,y) in Figure 6.1.

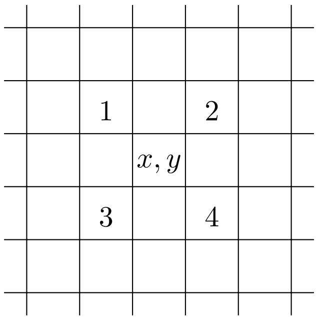


Figure 6.1: The robot can move from square (x,y) to the squares labeled by numbers 1 through 4.

Can the robot get to position $(8,9)$? Intuitively, the answer is no. If it were able to get to $(8,9)$, it could also get to $(7,8)$. From there, it could get to $(6,7)$, then to $(5,6)$, and eventually it would end up at $(0,1)$. But that seems unlikely.

The intuitive reasoning from the previous paragraph is a start, but by no means a proof that the robot can't get to $(8,9)$. We argue that the answer is "no" rigorously by finding an invariant and proving it holds by induction. Our "system" is just our robot, and the "state" of the system is the robot's position.

After trying a few moves and writing down the positions after those moves, we observe that whenever the robot is at position (x, y), either both x and y are odd, or they are both even. In other words, $x + y$ is always even. This suggests the following invariant: For any position (x, y) that is reached by the robot, $x + y$ is even. Let's prove this invariant by induction.

Theorem 6.1. *For any position (x, y) that is reached by the robot, $x + y$ is even.*

Proof. We prove this by induction on the number of steps taken by the robot. $P(t)$: "After any sequence of t steps, the sum of the coordinates of the position (x, y) reached by the robot is even ($x + y$ is even)." We will establish that $\forall t \in \mathbb{N}\ P(t)$ holds.

In the base case, $t = 0$, the robot is at position $(0, 0)$. Since $0 + 0 = 0$ is even, the base case holds.

For the inductive step, we need to argue $(\forall t \geq 0) P(t) \Rightarrow P(t + 1)$.

Assume that $P(t)$ holds. Consider any sequence of $t + 1$ steps $s_1, s_2 \ldots s_t, s_{t+1}$ taken by the robot. Let (x_t, y_t) be the position of the robot after the first t steps.

By our induction hypothesis, $x_t + y_t$ is even. Let (x_{t+1}, y_{t+1}) be the position of the robot after the $t + 1$- th step. We want to argue that $x_{t+1} + y_{t+1}$ is even. We argue by cases. There is one case for each possible move the robot can make from (x_t, y_t). The possible destinations are $(x_t - 1,\ y_t + 1)$, $(x_t + 1,\ y_t + 1)$, $(x_t - 1,\ y_t - 1)$ and $(x_t + 1,\ y_t - 1)$. (The possible destinations correspond to the four squares labeled by numbers 1 through 4 in Figure 6.1.)

Case 1: The robot moves to $(x_{t+1}, y_{t+1}) = (x_t - 1, y_t + 1)$. Then $x_{t+1} + y_{t+1} = x_t - 1 + y_t + 1 = x_t + y_t$, which is even by the induction hypothesis.

Case 2: The robot moves to $(x_{t+1}, y_{t+1}) = (x_t + 1, y_t + 1)$. Then $x_{t+1} + y_{t+1} = x_t + 1 + y_t + 1 = x_t + y_t + 2$. Now $x_t + y_t$ is even by the induction hypothesis, and 2 is also even, so $x_{t+1} + y_{t+1}$ is even.

Case 3: The robot moves to $(x_{t+1}, y_{t+1}) = (x_t - 1, y_t - 1)$. Then $x_{t+1} + y_{t+1} = x_t - 1 + y_t - 1 = x_t + y_t - 2$. Now $x_t + y_t$ is even by the induction hypothesis, and -2 is also even, so $x_{t+1} + y_{t+1}$ is even.

Case 4: The robot moves to $(x_{t+1}, y_{t+1}) = (x_t + 1, y_t - 1)$. Then $x_{t+1} + y_{t+1} = x_t + 1 + y_t - 1 = x_t + y_t$, which is even by the induction hypothesis.

Since the four cases above are the only possibilities, we have proved the inductive step, and thus also the theorem. $\square$

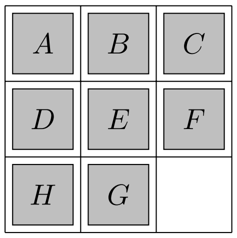
Note that $8 + 9 = 17$ is odd, so Theorem 6.1 implies that the robot cannot reach position $(8, 9)$. In fact, we can show that a position (x, y) can be reached if and only if the sum of the coordinates, $x + y$, is even. At Theorem 6.1, we argued the implication "if (x, y) can be reached then $x + y$ is even." To argue the other implication, that is, to argue that if $x + y$ is even, then the robot can reach (x, y), we would demonstrate a sequence of moves that takes the robot from $(0, 0)$ to (x, y). We can do this by induction, and leave the proof as an exercise to the reader.

We conclude this section with a remark. In the last example, our invariant completely describes the set of squares our robot can reach. As systems get more complicated, the invariants we come up with become complicated as well, and often fail to describe the system's behavior completely. But we may be lucky enough and come up with invariants that are sufficient for our purposes, that is, although they don't fully characterize the system, they describe the characteristics of the system we actually care about.

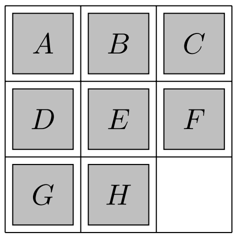
6.1.2 A More Complicated Invariant

Not all invariants are as simple. In the next example, it takes more work to find an invariant.

Consider the following puzzle, which we'll call the 8-puzzle. This is a simplification of a puzzle you probably saw when you were little. You have a 3×3 grid. Eight of the squares are filled with tiles with the first 8 letters on them as in Figure 6.2a, and the lower right square is blank. We can move a tile to the blank square from a neighboring square, which swaps the tile we move with the blank square. In Figure 6.2a, the only valid moves are swapping the tile labeled F with the blank square and swapping the tile labeled G with the blank square. Our goal is to rearrange the tiles so that the letters are "in order" and the bottom right corner is blank. This is shown in Figure 6.2b. Of course, we are only allowed to make valid moves.



(a) Starting arrangement.



(b) Desired arrangement.

Figure 6.2: Two arrangements in the 8-puzzle.

It turns out that no sequence of valid moves can transform the arrangement from Figure 6.2a to the arrangement in Figure 6.2b. In the following paragraphs, we illustrate the thought process that leads us to this conclusion.

Consider the sequence formed by listing the labels on the tiles row by row, and listing tiles in a row from left to right. We also have a term, $\square$, for the blank square. For example, the sequences that correspond to the arrangements in Figure 6.2a and Figure 6.2b are $A, B, C, D, E, F, H, G, \square$ and $A, B, C, D, E, F, G, H, \square$, respectively.

We say that a pair of positions (i, j) in our sequence is *in error* if $i < j$, there are letters at both positions, but the letter at position j precedes the letter at position i in alphabetical order. For example, the sequence corresponding to the arrangement in Figure 6.2a has exactly one pair of possitions in error, namely $(7, 8)$, and the sequence corresponding to the arrangement in Figure 6.2b has no pair of positions in error.

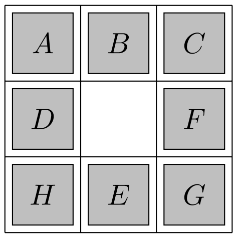
Let's see what happens to the sequence corresponding to an arrangement as we perform certain moves. In particular, we investigate how the number of pairs in error changes after a move. There are two kinds of moves we consider:

1. Row moves where the tile we move and the blank square are in the same row.
2. Column moves where the tile we move and the blank square are in the same column.

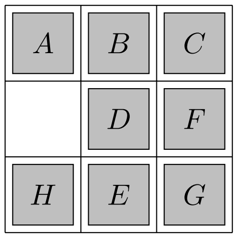
Consider how a move changes the sequence. Any move swaps the position of one tile with the blank square. Suppose the blank square is swapped with a tile labeled x. Then the positions of x and $\square$ switch after the move, and terms in all other positions in the sequence remain the same. Hence, a move of tile x can change the relative order of the letter x with other letters of our sequence, but can't change the relative order of two letters that correspond to tiles which stay put.

First consider a row move of tile x. By the way we constructed our sequence, it follows that x and $\square$ are adjacent terms in the sequence. If y is before x and $\square$ in the sequence, it will remain before x and $\square$ after the move because x and $\square$ just switch their positions in the sequence. Similarly, if y is after both x and $\square$ in the sequence, it will remain after x and $\square$ after the move. Since x and $\square$ are adjacent in the sequence, all other terms are either before them or after them, so we have shown that the relative order of no two letters changes after a row move. Hence, a row move of tile x causes no change in the number of pairs that are in error.

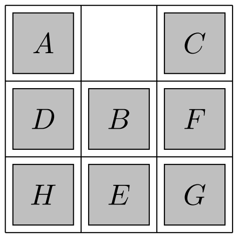
For example, the sequence representing the arrangement in Figure 6.3a is $A, B, C, D, \square, F, H, E, G$, and the sequence for the arrangement in Figure 6.3b is $A, B, C, \square, D, F, H, E, G$. As we can see, the only two terms in the sequence that changed their relative order correspond to the blank square and the tile that was moved. The relative order of no two letters changed.



(a) An arrangement obtained from the one in Figure 6.2a by moving tile G and then tile E



(b) An arrangement obtained from the on in Figure 6.3a by moving tile D. The move is a row move.



(c) An arrangement obtained from the one in Figure 6.3a by moving tile B. The move is a column move.

Figure 6.3: Row moves and column moves in an 8-puzzle.

Now consider a column move of tile x. By the way we constructed our sequence, there are exactly two terms, say a and b, between x and $\square$, and the sequence before the move looks like one of the rows in Figure 6.4 below. A column move takes us from one row to the other one (whether from the top one to the bottom one or the other way around depends on the relative position of tile x and the blank square before the move). Thus, a column move only changes the relative order of x and a and the relative order of x and b.

$$\cdots \quad x \quad a \quad b \quad \square \quad \cdots$$

$$\cdots \quad \square \quad a \quad b \quad x \quad \cdots$$

Figure 6.4: Two possibilities for the sequence before a column move of tile x. The dots indicate there are other terms before and after x, a, b and $\square$ in the sequence, and note that the relative order of x and those terms cannot change.

Suppose the sequence before the column move has the form of row 1 of Figure 6.4. Then if x precedes a (in alphabetical order), the pair (old position of x, position of a) was not in error, and also is not in error after the move. On the other hand, the pair (position of a, new position of x) was not in error before the move, but is in error after the move. So the move has increased the number of pairs involving a's position that are in error by one.

On the other hand, if a precedes x (in alphabetical order), the pair (old position of x, posotion of a) is in error, and the error goes away after the move. The pair (position of a, new position of x) was not in error before the move, and also isn't in error after the move. Thus, in this case, the move decreases the number of pairs involving a's position that are in error by one.

In either case, the number of pairs involving a's position that are in error changes by one. We can make the same argument about the number of pairs involving b's position. As we argued earlier, only pairs involving the positions of a, b and x can change their error states, so the net change in the number of pairs that change their error state is either -2, 0, or 2.

We can argue in a similar fashion if the sequence before the column move has the form of row 2 of Figure 6.4, which means that any column move changes the number of pairs in error by -2, 0, or 2.

For example, the sequence for the arrangement in Figure 6.3c is A, $\square$, C, D, B, F, H, E, G, which is different from the sequence A, B, C, D, $\square$, F, H, E, G for the arrangement in Figure 6.3a. In terms of Figure 6.4, we have $x = B$, $a = C$, $b = D$. We have changed the relative order of B with C, and the relative order of B with D. The move increased the number of pairs that were in error by 2 because pairs (3, 5) and (4, 5) are now in error.

We've seen that no matter what move we make, the change in the number of pairs in error is either -2, 0, or 2. All of these numbers are even. Hence, the invariant we deduce from the discussion above is that the *parity* of the number of pairs in error does not change. The parity of a number refers to whether the number is odd or even. For example 6 and 8 have the same parity . While 7 and 156 have different parities.

Theorem 6.2. *After any sequence of moves, the parity of the number of pairs in error is the same as initially.*

Proof. We use induction to prove the statement $(\forall n)P(n)$ where $P(n)$ is "after n moves, the parity of the number of pairs in error is the same as initially."

For the base case, notice that after zero moves we are still at the initial configuration, so the parity hasn't had a chance to change. Thus, the base case holds.

For the induction step, Assume that $P(k)$ holds for some $k \geq 0$. Consider any sequence of $k+1$ moves M_1, $M_2 \ldots M_k$, M_{k+1}. Let e_k be the number of pairs in errors after move M_k (after the first k moves). By the induction hypothesis. e_k has the same parity as the number of errors initially. Let e_{k+1} be the number of pairs in error after the $k+1$-th move M_{k+1}. We need to show that the parity of e_{k+1} is the same as in the initial configuration.

If M_{k+1} is a row move then $e_{k+1} = e_k$ (since row moves don't change the number of errors.).

On the other hand if M_{k+1} is a column move then the number of pairs in error $e_k + 1$ is either $e_k - 2$ or e_k or $e_k + 2$.

In both cases, the parity of e_{k+1} is the same as the parity of e_k. By the induction hypothesis the parity of e_{k+1} is the same as the parity of the number of errors initially. Which completes the induction step.

Therefore, by induction, for all n the parity of the number of pairs in error after n moves is the same as initially.

$\square$

Since the parity of the number of pairs in error is odd in the arrangement in Figure 6.2a, and even in the arrangement in Figure 6.2b, Theorem 6.2 implies that no sequence of moves can transform the configuration from Figure 6.2a to the configuration in figure 6.2b.

It took us some time before we formulated our invariant. Indeed, it takes some ingenuity to recognize a common pattern such as the one described in Theorem 6.2. The kind of creative thinking that led us to the formulation of Theorem 6.2 is very common. In any area, and for any problem we face, we may not see a solution to the problem right away, so we play with the problem, fail a few times, try again using a modification of an earlier strategy or using an entirely new strategy, and then finally recognize a pattern that helps us solve the problem.