CS205-Recitation 8

Xiaoxiao(Arist) He

Rutgers University

10-05-2022

Topics

- Strong induction
- Loop Invariants

Strong Induction Outlines

Recall weak induction:

- Prove one base case is correct.
- Induction step: Assume that for case n is correct. Then prove that $n+1$ case is correct by using the fact that n is correct.

However, strong induction assumes all cases up to n is correct. Then, we need to prove that $n+1$ is correct using the fact that previous n cases are correct.

Example of a strong induction proof

Proof: We need to show that $\forall n \in \mathbb{N} n \leq 1$ can be written as the sum of distinct powers of two by strong induction.

Example of a strong induction proof

Proof: We need to show that $\forall n \in \mathbb{N} n \leq 1$ can be written as the sum of distinct powers of two by strong induction.

Base case $n = 1$

When $n = 1$, then $2^0 = 1$

Example of a strong induction proof

Strong Induction Step

Assume that the claim is true for all $1 \leq n \leq k$. Then we can write $k + 1 = 2^i + M$ for some $M \geq 0$. Note that $M \leq 2^i$ by our choice of i. If $M = 0$, then we are done. Otherwise, $1 \leq M < 2^i$, M is the sum of some distinct power of two $M = 2^{j_1} + ... + 2^{j_t}$. Since $M < 2^i$, we have that $2^{j_1}...2^{j_t}$ all less than 2^i. Hence $k + 1 = 2^i + 2^{j_1} + ... + 2^{j_t}$ is a representation of $k + 1$ as sum of distinct power of two.

Loop Invariants

Next, proofs of correctness of **while** loops will be described. To develop a rule of inference for program segments of the type

```
while condition
    S
```

note that S is repeatedly executed until *condition* becomes false. An assertion that remains true each time S is executed must be chosen. Such an assertion is called a **loop invariant**. In other words, p is a loop invariant if $(p \wedge condition)\{S\}p$ is true.

Figure: Definition of loop invariants

Exercises

Find loop invariant and prove that the following code finds the minimum element of a $n-$element list A

$i = 1$
$min = A[0]$
while $i < n$ **do**
  if $A[i] \leq min$ **then**
    $min = A[i]$
  end if
  $i = i + 1$
end while
return min

Exercise

Let p be the assertion of min is the minimum of $A[0, ..., i-1], i \leq n$.
First, we need to prove that p is a loop invariant.
Suppose that $min = \min(A[0, ..., i-1]) i - 1 < n$, then we need to prove that $min_{new} = A[0, ..., i]$.
If $A[i] \leq min$, we make $min_{new} = A[i]$, which is the minimum of $A[0, ..., i]$, else $min_{new} = min$, which is also the minimum of $A[0, ..., i]$
Thus p is true at the end of the loop.

Remark

This looks familiar to the inductive step of induction proof.

Exercise

Then we can show that p is true for the beginning of the program, which means when $i = 1$, $min = A[0]$,
$\min A[0, ..., i-1] = \min A[0] = A[0] = min$.
The last check if the loop terminates: The initial value of i is 1, and after $n-1$ loops, the value of i is n, and the loop will terminate at then.

Attendance

https://go.rutgers.edu/9qksv8d5