Week 14 - Day 1 (Asserts, etc)

Table of Contents

[CS 403 - 001 Spring 2016](/CS403-001/)

[About](/CS403-001/about/)

# Week 14 - Day 1 (Asserts, etc)

Apr 19, 2016

Download Word (docx):

## Navigate using audio

## Asserts

* Audio 0:01:47.287221

#include <assert.h>  
  
function f(x) {  
 // precondition  
 assert(x >= 0);  
 //.  
 //.  
 //.  
 // postcondition  
 assert(result < );  
}

* Audio 0:02:50.569008
* Final will be open notes and open book
  + Whatever you want
    - Audio 0:03:30.949634

x = x - 1;  
  
// double bracket statements represent conditions  
// precondition  
<<x > 1>>  
// postcondition  
<<x > 0>>  
// is the program correct?  
// Audio 0:05:01.943301

We’re going to run the program in reverse in order to check correctness

// Audio 0:05:52.521103   
// start with x > 0  
// going backwards  
x - 1 > 0  
// this is the xame as  
x > 1  
// which is what we sought to proove

* Audio 0:07:41.389287

<<x <= 100>>  
x > 1

-> x == 100 -> x > 1

P implies Q where x replaces x-1

P -> Q x\x-1

* What is your weakest precondition?

<<x > 1>>  
 <<x == 100>>

* It’s x > 1 because it’s more general. You’re allowing more stuff in
  + Audio 0:10:43.221994

What is the weakest precondition for the following postcondition?

<<x < 5>>

x = x - 1  
  
<<x < 6>>

Weakest possible pre-condition is “true”

x = x - 1  
y = y \* 2  
  
<<x==0 && y > 1>>  
  
<<x==1 && y\*2 > 1>>

// Audio 0:16:02.064413   
x = x -1;  
y = y + x;  
<<x == 0 && y > 20>>  
  
<<x > 0 && y + x > 20>>  
  
<<x-1 > 0 && y + x - 1 > 20>> // weakest

//<<P>>  
if (E)  
 S;  
else  
 T;  
//<Q>>

// Audio 0:20:04.941081   
//<<P>>  
if (E) S;  
else T;  
//<<Q>>

Proof rule:

P && E - > Q (pushback) S && P && !E -> Q (pushback) T

* Audio 0:21:59.918531

if ((fp = fopen(filename, "r")) != 0)

Audio 0:24:45.279192

if (x>1)  
 x=-x  
else  
 x=x+1  
  
//<<x > 0>>

//<<true>>  
if (x<0)  
 x=-x  
else  
 x=x+1  
  
//<<x > 1>>

Audio 0:28:20.419273

* true && x < 0 -> -x > 1
* true && x >= 0 -> Q (pushback) x + 1 > 1
  + x < 0 -> x < -1 –> False
  + x > 0 -> x > 0 –> true

TAUTOLOGY

while(x>0)  
{  
 x = x-1;  
 y = y+ 1;  
}  
// (#x means x's original value)  
// <<y==#x && x==0>>  
// Audio 0:37:16.326192   
// x + y == #x + #y  
// y has gotten smaller by the same amount that x got larger  
// Therefore, we have preserved the amount and it remains unchanged

Next time we’ll do proof rule for while loop and try to find the weakest precondition for if and while loop

## CS 403 - 001 Spring 2016

* CS 403 - 001 Spring 2016
* [jmbeach1@crimson.ua.edu](mailto:jmbeach1@crimson.ua.edu)
* jmbeach

Website for notes and study material for CS 403 (Programming Languages) at The University of Alabama Spring 2016