19CSE205 – Program Reasoning

Jevitha KP

Lecture 8-9 – Weakest Preconditions
- Assignment,
- Sequence of Assignments

Credits

- Adapted from :
  - Dr. Bharat Jayaraman, University of Buffalo, CSE449-459 Software verification course, Spring 2020.
  - 19CSE205 (2020-2021) - Dr. Vidhya Balasubramaniam

Weakest Precondition

- Overview
    - Formal reasoning about program correctness
    - Technique for proving correctness of sequential programs
    - Will study tools and techniques to verify the functional correctness for sequential programs.

Code based Verification

- How can we verify simple iterative programs interactively?
  - Use tools like Frama-C, Alt-Ergo
- How to generate verification conditions?
  - What predicate describes all valid inputs for which program S will complete (in a finite amount of time) with an output O
  - Define Weakest Preconditions
  - Write loop invariants

Code based Verification

- Define weakest preconditions
  - Deductive system by Edward Dijkstra
  - Provides an algorithmic solution to perform symbolic execution on program statements
  - in the backward direction in order to deduce the predicate that will guarantee a given postcondition.

Definitions

- Predicate
  - An expression that evaluates to either true or false
- Precondition
  - A predicate which, when it is true before execution of a statement/block, ensures postcondion is true after execution of that statement/block
- Postcondition
  - A predicate that evaluates true after execution a statement/block.

Definitions

- Symbolic Execution
    - o Uses symbolic values to variables to identify different execution paths of a program
- Deductive System
    - Uses axioms and rules to prove a theorem

Weakest Precondition

- Objective
- To prove that a program P is correct with respect to its contract which is stated as a pre-condition I and post-condition O.
- Weakest pre-conditions is a "backward flow" analysis, from output back to input.
- Given a code fragment P with post condition O, find the unique precondition which is the weakest precondition for P and O

Weakest Precondition

- The Weakest Precondition of a statement S w.r.t. a post-condition O is written as wp(S, O).
- If the input condition for S is I, then we want the following theorem to be true:

I ==> wp(,S O))

- Our goal is to verify if I –> WP(S,O) is valid

Defining Weakest Preconditions

1. wp(x = expr, O).
2. wp(S1 ; S2, O).
3a. wp(if (B) S1 else S2, O).
3b. wp(if (B) S1, O).
4. wp(while B do S, O).
5a. wp(break, O)
5b. wp(continue, O)
6. wp(skip)
7. wp(abort)

Assignment Axiom

- When S is an assignment statement, x = expr, the weakest precondition wp(x = expr, O) is defined as

  O [x <-- expr]

- i.e., replace all occurrences of x in O by expr.
- Example: If S is x = y * 5 and O is {x >= 20} then wp(S, O)

= {x >= 20} [x <-- y * 5]

= {y * 5 >= 20}

= {y >= 4}

Why “weakest” pre-condition?

- Re-consider: S is x = y * 5 and O is {x >= 20}.
- We derived: wp(S, O) = {x >= 20} [x <-- y * 5]
  = {y >= 4}
- Given the above S and O, input conditions such as
  {y = 4} or
  {y = 50} or
  {y >= 100} …

  will all yield the output condition O.
- However, the “weakest” (i.e., least restrictive) input condition is
  {y >= 4}

“Strong” vs “Weak” conditions

- A condition can be thought of as a set, i.e., the set of values that make the condition true. For example, {y >= 20} can be thought of as the set {20, 21, 22, …} assuming y:int.

- Stronger conditions yield smaller sets and weaker conditions yield larger sets. For example, we can say that {y >= 50} is a stronger condition than {y >= 20}.

- The strongest condition is false, and this corresponds to the empty set.
- The weakest condition is true, and this corresponds to the universal set – in our example, the set of all numbers.

Verify Expressions

- Use Alt-Ergo
- Given Input condition I, Statement S and Output Condition O
- Manually derive the weakest precondition for the given statement S - wp(S,O)
- For the given input condition I, using Alt-Ergo tool find whether I ⇒ wp(S,O)

Verify Expressions

- For previous example (assuming I is y=4)
- Alt-Ergo code is
- goal a:
- forall x,y: int.
- y=4 –> y>=4
- Here it is valid;
- if input was y =2, the verification fails

Verify Expressions - Exercise

- Given S: $x = y+1$, O : $x > 0$, derive the weakest precondition wp(S,O).
- For the input conditions I given below, check whether I ⇒ wp(S,O) in alt-ergo
- I : $y > 0$
- I : $y = 0$
- I : $y < 0$
- I : $y = 100000$
- I : $x = 0$
- I : $x < 0$
- I : $x > 0$

Verify Expressions – Exercise Solution

- Given S: x = y+1 , O : x > 0, derive the weakest precondition wp(S,O).
- WP (x=expr, O) = O[x<-- expr]
  - [x > 0] {x <-- y+1}
  - y+1>0
  - y>-1 - WP
- Alt-ergo: I --> wp(S,O) ; I: y>0

  goal a:
  forall x,y: int.
  y>0 -> y>-1
- Similarly try for other Input conditions given

Sequence of Assignments

Sequencing

- Given a statement sequence, S1 ; S2 ;

  wp(S1 ; S2 ; , O) = wp(S1, wp(S2, O))

- Weakest pre-conditions is a "backward flow" analysis, from output back to input.

Sequencing - Example

- Given a sequence of statements, S1 ; S2 ;

    wp(S1 ; S2 ; , O) = wp(S1, wp(S2, O))

Example:

If S is y = z – 4; x = y * 5; and

O is {x >= 20} . Find the WP.

Sequencing - Example

- If S is y = z – 4; x = y * 5; and O is {x >= 20} . Find the WP.
- wp(S, O) = wp(y = z – 4, wp(x = y * 5, {x >= 20}))

  = wp(y = z – 4, {y * 5 >= 20}) [replace x by y*5 in O]

  = wp(y = z – 4, {y >= 4})

  = {z - 4 >= 4} [replace y by z-4 in prev. condition]

  = **{z >= 8}** --> required weakest precondition

Longer Example

- @requires w = 5 && y = 7 <--- @requires represents pre-conditions
- @ensures w*w + x = y*y + z <-- @ensures represents post-conditions
- @program {

```
  x = 6;
  z = 8;
  w = w*2 - 5;
  x = (x-1)*(x-1);
  y = y-2;
  z = (z+2)*(z+2);
  z = z - 75;
}
```

- The example motivates need for automated tools.

Exercises

- Find the weakest preconditions for the following:
  - S1 = x=z+1; S2 = y=x+w; O is {y>5} ;
  - Check the validity in Alt-Ergo for I1: z >=0 & I2: z >=0; y>=0.
  - Give an input condition for which the above WP is valid

Exercises

- Find the weakest preconditions for the following:
  - S1 = x=z+1; S2 = y=x+w; O is {y>5} ;
  - Check the validity in Alt-Ergo for I1: z >=0 & I2: z >=0; y>=0.
  - Give an input condition for which the above WP is valid

WP (S2,O) = {y>5} (y=x+w) ==> x+w > 5
WP(S1, WP(S2,O)) = {x+w > 5} (x=z+1) ==> z+1+w > 5 ==> z+w > 4 -- WP

- I1 --> WP(S,O) ==> z >=0 -> z+w > 4 Alt-Ergo : I don't know
- I2 --> WP(S,O) ==> z >=0 /\ y>=0 -> z+w > 4 Alt-Ergo : I don't know
- I3 --> WP(S,O) ==> z > 2 /\ y>2 -> z+w > 4 Alt-Ergo : Valid
- I4 --> WP(S,O) ==> z > 1 /\ y>3 -> z+w > 4 Alt-Ergo : Valid

Exercises

- A thermal scanner reads the body temperature in Celsius. Due to Corona, the company will not allow a person into the building if body temperature is >103 degree F. Derive the WP for a person to enter the building. F=9 * Celsius/5 + 32.0

Exercises

- S1 : F=9 * Celsius/5 + 32.0
- O : F <= 103
- WP(S,O) = 9 * Celsius/5 + 32.0 <= 103 ==> Celsius <= 39.44
- I --> WP ==> c = 38.0 -> c <= 39.4
- Alt-ergo:
- goal a:
-     forall c,f : real.
-   c = 38.0 -> c <= 39.4 - Valid

Exercises

- Sankar assures a treat to Siddi after receiving his first month salary, if he gets a minimum of Rs.20000 as take home. His take home is calculated as (salary - 5000.00) * 0.20 - 1425.00. What is the WP for Siddi to receive his treat?

Exercises

- T = Take home ; S = Salary
- S: t = (s - 5000.00) * 0.20 - 1425.00
- O : {t >= 20000}
- WP(S,O) = (s - 5000.00) * 0.20 - 1425.00 >= 20000
- ==> s >= (((20000+1425))/0.2) + 5000
- WP(S,O) ==> s >= 1,12,125
- Alt-Ergo:
- goal a:
- forall s : int.
- s = 100000 -> s >= 112125 -- I don't know
- goal b:
- forall s : int.
- s = 200000 -> s >= 112125 -- Valid

AMRITA VISHWA VIDYAPEETHAM

Exercises

- Find the weakest preconditions for the following:
  - S1 : a=a+1; S2: b=b+1; O = {a *b=0); Verify the WP for I:$(a=-1) \wedge (b=1)$
  - S1 : x:=x+2; S2 : y=y-2; O= {x+y=0}
  - S1 : x = x + 5; S2: y = 2 * x; O is { y > 10 }
  - S1: y = x + 6; S2: z = x + y; O is { z ≤ 0 }
- Identify the input conditions that satisfy the weakest precondition for the above questions

Exercises

- Find the weakest preconditions for the following:
  - S1 : a=a+1; S2: b=b+1; O = {a *b=0}; Verify the WP for I:(a=−1)∧(b=1)

WP(S2,O) = {a*b=0} (b = b+1) = {a*(b+1) =0}
WP(S1, WP(S2,O)) = {a*(b+1) = 0} (a=a+1) = **(a+1)*(b+1) = 0 <-- WP**
I --> WP(S,O) ==> (a=-1) /\ (b=1) --> (a+1)*(b+1) = 0
Alt-Ergo:

```
goal a:
  forall a,b : int.
  (a=-1) and (b=1) -> (a+1)*(b+1) = 0
```

Valid (0.1500) (2 steps) (goal a)