Tuesday Dec. 4

Lecture 24

Review Sessions for Exam

LAS C

2pm ~ 4pm Thursday Dec. 6

11am ~ 1pm Friday Dec. 7

Confirm your attendance on Moodle !

Marks for:

Labs and Lab Test

Available around Exam day

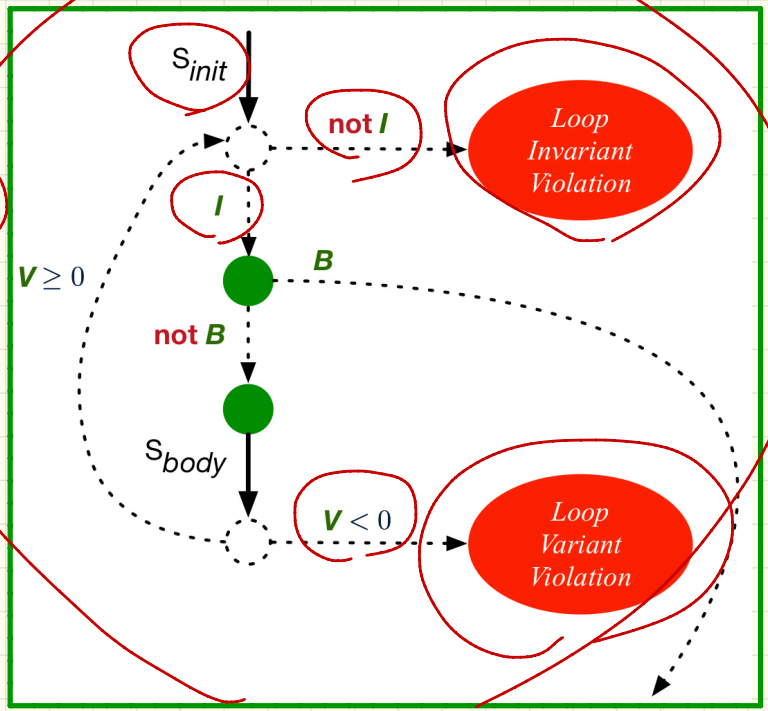
You will be able to speak to me about these shortly after the exam.

Contracts of Loops

Syntax

```
from
  S_init
invariant
  invariant_tag: I
until
  B
loop
  S_body
variant
  variant_tag: V
end
```

Runtime Checks

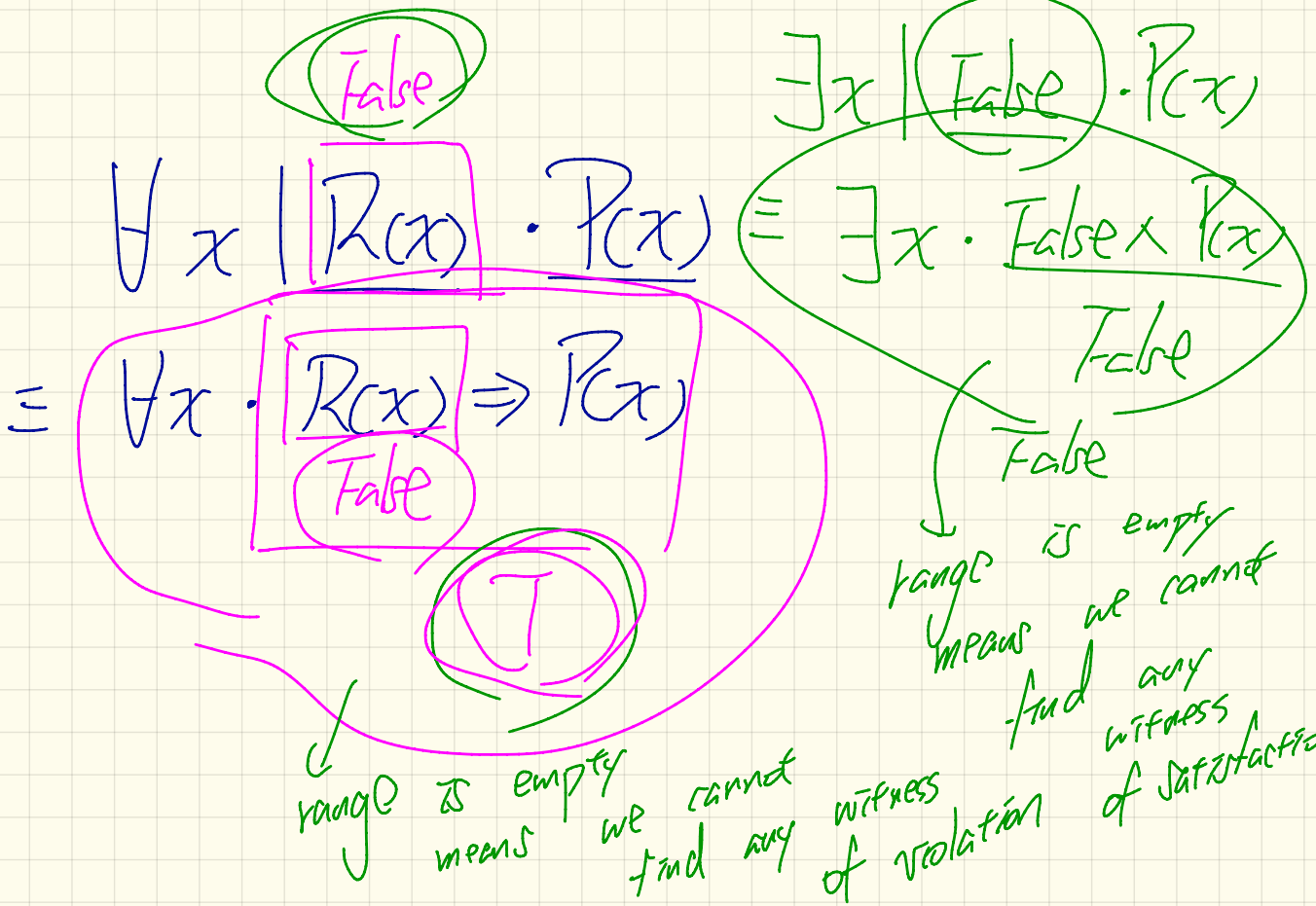


Finding Max : v1

```
find_max (a: ARRAY [INTEGER]): INTEGER
  local i: INTEGER
  do
    from
      i := a.lower ; Result := a[i]
    invariant
      loop_invariant: -- ∀j | a.lower ≤ j ≤ i • Result ≥ a[j]
      across a.lower |..| i as j all Result >= a [j.item] end
    until
      i > a.upper
    loop
      if a [i] > Result then Result := a [i] end
      i := i + 1
    variant
      loop_variant: a.upper - i + 1
    end
  ensure
    correct_result: -- ∀j | a.lower ≤ j ≤ a.upper • Result ≥ a[j]
      across a.lower |..| a.upper as j all Result >= a [j.item]
  end
end
```

After Iteration	i	Result	LI	Exit (i > a.upper)?	LV
Initialization	1	20	✓	×	●
1st	2	20	✓	×	●
2nd	3	20	×	●	●

Exercise: change i := i+1 of LI of Sbody.


False
$\forall x \mid R(x) \cdot P(x)$
$\equiv \forall x \cdot R(x) \Rightarrow P(x)$
False
T
range is empty
means we cannot
find any witness
of violation
$\exists x \mid \text{False} \cdot P(x)$
$\equiv \exists x \cdot \text{False} \wedge P(x)$
False
False
range is empty
means we cannot
find any
witness
of satisfaction

Finding Max : v2

20	10	40	30

```
find_max (a: ARRAY [INTEGER]): INTEGER
  local i: INTEGER
  do
    from
      i := a.lower ; Result := a[i]
    invariant
      loop_invariant: -- ∀j | a.lower ≤ j < i • Result ≥ a[j]
        across a.lower |..| (i - 1) as j all Result >= a [j.item] end
    until
      i > a.upper
    loop
      if a [i] > Result then Result := a [i] end
      i := i + 1
    variant
      loop_variant: a.upper - i
    end
  ensure
    correct_result: -- ∀j | a.lower ≤ j ≤ a.upper • Result ≥ a[j]
      across a.lower |..| a.upper as j all Result >= a [j.item]
  end
end
```

i ↑ LV ↓

After Iteration	i	Result	*LI*	Exit (*i* > *a.upper*)?	*LV*
Initialization	1	20	✓	×	–
1st	2	20	✓	×	2
2nd	3	20	✓	×	1
3rd	4	40	✓	×	0
4th	5	●	●	●	-1

Proof Obligations for Correct Loops

$I \wedge B \Rightarrow R$

$\{I \wedge \neg B\}\ S_{body}\ \{V \geq 0\}$

$\{I\}\ \neg B\ S_{body}\ \{I\}$

```
{Q}   from
        S_init
      invariant
        I
      until
        B
      loop
        S_body
      variant
        V
      end        {R}
```

$\{I \wedge \neg B\}\ S_{body}\ \{\ \}$

$V < \text{old } V$

- A loop is ***partially*** *correct* if:
  - Given precondition ***Q***, the initialization step S_{init} establishes ***LI*** *I*.

    $$\{Q\}\ S_{init}\ \{I\}$$

  - At the end of S_{body}, if not yet to exit, ***LI*** *I* is maintained.

    $$\{I \wedge \neg B\}\ S_{body}\ \{I\}$$

  - If ready to exit and ***LI*** *I* maintained, postcondition ***R*** is established.

    $$I \wedge B \Rightarrow R$$

- A loop *terminates* if:
  - Given ***LI*** *I*, and not yet to exit, S_{body} maintains ***LV*** *V* as non-negative.

    $$\{I \wedge \neg B\}\ S_{body}\ \{V \geq 0\}$$

  - Given ***LI*** *I*, and not yet to exit, S_{body} decrements ***LV*** *V*.

    $$\{I \wedge \neg B\}\ S_{body}\ \{V < V_0\}$$

Proof Obligations for Correct Loops: Example

```
find_max (a: ARRAY [INTEGER]): INTEGER
  local i: INTEGER
  do
    from
      i := a.lower ; Result := a[i]
    invariant
      loop_invariant: ∀j | a.lower ≤ j < i • Result ≥ a[j]
    until
      i > a.upper
    loop
      if a [i] > Result then Result := a [i] end
      i := i + 1
    variant
      loop_variant: a.upper − i + 1
    end
  ensure
    correct_result: ∀j | a.lower ≤ j ≤ a.upper • Result ≥ a[j]
  end
end
```

I

Initialization:

{True} i := a.lower ; Result := a[i] {I}

Before Termination:

Upon Termination:

Non-Negative Variant:

Decreasing Variant:

Prove

Establishment of Loop Invariant:

```
{ True }
  i := a.lower
  Result := a[i]
{ ∀j | a.lower ≤ j < i • Result ≥ a[j] }
```

$wp(\ i := a.lower\ ;\ Result := a[i]\ ,\ \forall j \mid a.l \le j < i \bullet R \ge a[j])$

$= \{$ wp rule for ; $\}$

$wp(\ i := a.lower\ ,\ wp(\ Result := a[i]\ ,\ \forall j \mid a.l \le j < i \bullet R \ge a[j]))$

$= \{$ wp rule for := $\}$ $[1 \to 1]$

$wp(\ i := a.lower\ ,\ \forall j \mid a.l \le j < i \bullet a[i] \ge a[j]\ )$

$= \{$ wp rule for := $\}$

$\forall j \mid a.l \le j < a.l \bullet a[a.l] \ge a[j] \equiv T$

Prove

Establishment of Postcondition upon Termination:

$$(\forall j \mid a.lower \le j < i \bullet Result \ge a[j]) \wedge i \ge a.upper \Rightarrow \forall j \mid a.lower \le j \le a.upper \bullet Result \ge a[j]$$

$P \Rightarrow Q$

$\Rightarrow (\forall x \mid Q \cdot R)$

$\Rightarrow (\forall x \mid P \cdot R)$

$(\forall x \mid 1 \le x \le 10 \cdot x^2 \le 100)$

$\Rightarrow (\forall x \mid 1 \le x \le 9 \cdot x^2 \le 100)$

$(\forall x \mid 1 \le x \le 9 \cdot x^2 \le 81)$

$\Rightarrow (\forall x \mid 1 \le x \le 10 \cdot x^2 \le 81)$

Prove

Loop Variant Stays Non-Negative Before Exit:

{ $(\forall j \mid a.lower < j < i \bullet Result \geq a[j]) \wedge \neg(i > a.upper)$ }

```
if a [i] > Result then Result := a [i] end
i := i + 1
```

{ $a.upper - i + 1 \geq 0$ }