Logical Quantifiers: Examples

$$\forall i \bullet i \in \mathbb{N} \Rightarrow i \geq 0$$

$$\forall i \bullet i \in \mathbb{Z} \Rightarrow i \geq 0$$

$$\forall i, j \bullet i \in \mathbb{Z} \wedge j \in \mathbb{Z} \Rightarrow i < j \vee i > j$$

$$\exists i \bullet i \in \mathbb{N} \wedge i \geq 0$$

$$\exists i \bullet i \in \mathbb{Z} \wedge i \geq 0$$

$$\exists i, j \bullet i \in \mathbb{Z} \wedge j \in \mathbb{Z} \wedge (i < j \vee i > j)$$

Logical Quantifiers: Examples

How to prove $\forall i \bullet R(i) \Rightarrow P(i)$?

How to prove $\exists i \bullet R(i) \land P(i)$?

How to disprove $\forall i \bullet R(i) \Rightarrow P(i)$?

How to disprove $\exists i \bullet R(i) \land P(i)$?

Prove/Disprove Logical Quantifications

- Prove or disprove: $\forall x \bullet (x \in \mathbb{Z} \land 1 \leq x \leq 10) \Rightarrow x > 0$.

- Prove or disprove: $\forall x \bullet (x \in \mathbb{Z} \land 1 \leq x \leq 10) \Rightarrow x > 1$.

- Prove or disprove: $\exists x \bullet (x \in \mathbb{Z} \land 1 \leq x \leq 10) \land x > 1$.

- Prove or disprove that $\exists x \bullet (x \in \mathbb{Z} \land 1 \leq x \leq 10) \land x > 10$?

Predicate Logic: Exercise 1

Consider the following predicate:

$$\forall x, y \bullet x \in \mathbb{N} \land y \in \mathbb{N} \Rightarrow x * y > 0$$

Choose **all** statements that are correct.

1. It is a theorem, provable by (5, 4).
2. It is a theorem, provable by (2, 3).
3. It is not a theorem, witnessed by (5, 0).
4. It is not a theorem, witnessed by (12, -2).
5. It is not a theorem, witnessed by (12, 13).

Predicate Logic: Exercise 2

Consider the following predicate:

$$\exists\, x, y \bullet x \in \mathbb{N} \land y \in \mathbb{N} \land x * y > 0$$

Choose **all** statements that are **correct**.

1. It is a theorem, provable by (5, 4).
2. It is a theorem, provable by (2, 3).
3. It is a theorem, provable by (-2, -3).
4. It is not a theorem, witnessed by (5, 0).
5. It is not a theorem, witnessed by (12, -2).
6. It is not a theorem, witnessed by (12, 13).

Predicate Logic: Exercise 3

Consider the following logical quantification:

!x,y.x:NAT&y:NAT=>x+y>=10&x+y<20 $\forall x, y \bullet x \in \mathbb{N} \wedge y \in \mathbb{N} \Rightarrow x + y \geq 10 \wedge x + y < 20$

Convert the above predicate to an equivalent one using the other logical quantifier.

Note the following constraints on your answer:

- Only put pairs of parentheses **when necessary**.
- Like the above predicate, there should be **no** white spaces.
- Like the above predicate, numerical constants (i.e., 10, 20) must appear as the right operands of the relational expressions (e.g., x + y >= 10).
- Relational expressions should be simplified whenever possible, e.g., write x >= 20 rather than not(x < 20).

Be cautious about the spellings: this question will be graded **automatically** and no partial marks will be give to spelling mistakes.

Answer:

Interpreting a Formula: Parse Trees (1)

ϕ	$::=$	$\top$	[*true*]
	$\mid$	$\bot$	[*false*]
	$\mid$	p	[propositional atom]
	$\mid$	$(\neg\phi)$	[logical negation]
	$\mid$	$(\phi \wedge \phi)$	[logical conjunction]
	$\mid$	$(\phi \vee \phi)$	[logical disjunction]
	$\mid$	$(\phi \Rightarrow \phi)$	[logical implication]
	$\mid$	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	$\mid$	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	$\mid$	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	$\mid$	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	$\mid$	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	$\mid$	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

$$\mathbf{F}\ p \wedge \mathbf{G}\ q \Rightarrow p\ \mathbf{U}\ r$$

Interpreting a Formula: Parse Trees (2)

ϕ	$::=$	$\top$	[*true*]
	$\mid$	$\bot$	[*false*]
	$\mid$	p	[propositional atom]
	$\mid$	$(\neg\phi)$	[logical negation]
	$\mid$	$(\phi \wedge \phi)$	[logical conjunction]
	$\mid$	$(\phi \vee \phi)$	[logical disjunction]
	$\mid$	$(\phi \Rightarrow \phi)$	[logical implication]
	$\mid$	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	$\mid$	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	$\mid$	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	$\mid$	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	$\mid$	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	$\mid$	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

$$\mathbf{F}\,(p \wedge \mathbf{G}\,q \Rightarrow p\,\mathbf{U}\,r)$$

Interpreting a Formula: Parse Trees (3)

ϕ	$::=$	$\top$	[*true*]
	$\mid$	$\bot$	[*false*]
	$\mid$	p	[propositional atom]
	$\mid$	$(\neg\phi)$	[logical negation]
	$\mid$	$(\phi \wedge \phi)$	[logical conjunction]
	$\mid$	$(\phi \vee \phi)$	[logical disjunction]
	$\mid$	$(\phi \Rightarrow \phi)$	[logical implication]
	$\mid$	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	$\mid$	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	$\mid$	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	$\mid$	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	$\mid$	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	$\mid$	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

$$\mathbf{F}\ p \wedge (\mathbf{G}\ q \Rightarrow p\ \mathbf{U}\ r)$$

Interpreting a Formula: Parse Trees (4)

ϕ	::=	$\top$	[*true*]
	|	$\bot$	[*false*]
	|	p	[propositional atom]
	|	$(\neg\phi)$	[logical negation]
	|	$(\phi \wedge \phi)$	[logical conjunction]
	|	$(\phi \vee \phi)$	[logical disjunction]
	|	$(\phi \Rightarrow \phi)$	[logical implication]
	|	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	|	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	|	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	|	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	|	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	|	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

$$\mathbf{F}\ p \wedge ((\mathbf{G}\ q \Rightarrow p)\ \mathbf{U}\ r)$$

Interpreting a Formula: LMD (1)

ϕ	$::=$	$\top$	[*true*]
	$\mid$	$\bot$	[*false*]
	$\mid$	p	[propositional atom]
	$\mid$	$(\neg\phi)$	[logical negation]
	$\mid$	$(\phi \wedge \phi)$	[logical conjunction]
	$\mid$	$(\phi \vee \phi)$	[logical disjunction]
	$\mid$	$(\phi \Rightarrow \phi)$	[logical implication]
	$\mid$	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	$\mid$	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	$\mid$	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	$\mid$	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	$\mid$	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	$\mid$	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

$$\mathbf{F}\ p \wedge \mathbf{G}\ q \Rightarrow p\ \mathbf{U}\ r$$

Interpreting a Formula: LMD (2)

ϕ	$::=$	$\top$	[*true*]
	$\mid$	$\bot$	[*false*]
	$\mid$	p	[propositional atom]
	$\mid$	$(\neg\phi)$	[logical negation]
	$\mid$	$(\phi \wedge \phi)$	[logical conjunction]
	$\mid$	$(\phi \vee \phi)$	[logical disjunction]
	$\mid$	$(\phi \Rightarrow \phi)$	[logical implication]
	$\mid$	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	$\mid$	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	$\mid$	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	$\mid$	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	$\mid$	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	$\mid$	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

$$\mathbf{F}\,(p \wedge \mathbf{G}\,q \Rightarrow p\,\mathbf{U}\,r)$$

Interpreting a Formula: LMD (3)

ϕ	$::=$	$\top$	[*true*]
	$\mid$	$\bot$	[*false*]
	$\mid$	p	[propositional atom]
	$\mid$	$(\neg\phi)$	[logical negation]
	$\mid$	$(\phi \wedge \phi)$	[logical conjunction]
	$\mid$	$(\phi \vee \phi)$	[logical disjunction]
	$\mid$	$(\phi \Rightarrow \phi)$	[logical implication]
	$\mid$	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	$\mid$	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	$\mid$	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	$\mid$	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	$\mid$	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	$\mid$	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

$$\mathbf{F}\ p \wedge (\mathbf{G}\ q \Rightarrow p\ \mathbf{U}\ r)$$

Interpreting a Formula: LMD (4)

ϕ	$::=$	$\top$	[*true*]
	$\mid$	$\bot$	[*false*]
	$\mid$	p	[propositional atom]
	$\mid$	$(\neg\phi)$	[logical negation]
	$\mid$	$(\phi \wedge \phi)$	[logical conjunction]
	$\mid$	$(\phi \vee \phi)$	[logical disjunction]
	$\mid$	$(\phi \Rightarrow \phi)$	[logical implication]
	$\mid$	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	$\mid$	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	$\mid$	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	$\mid$	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	$\mid$	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	$\mid$	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

$$\mathbf{F}\ p \wedge ((\mathbf{G}\ q \Rightarrow p)\ \mathbf{U}\ r)$$

Interpreting a Formula: RMD (1)

ϕ	$::=$	$\top$	[*true*]
	$\mid$	$\bot$	[*false*]
	$\mid$	p	[propositional atom]
	$\mid$	$(\neg\phi)$	[logical negation]
	$\mid$	$(\phi \wedge \phi)$	[logical conjunction]
	$\mid$	$(\phi \vee \phi)$	[logical disjunction]
	$\mid$	$(\phi \Rightarrow \phi)$	[logical implication]
	$\mid$	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	$\mid$	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	$\mid$	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	$\mid$	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	$\mid$	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	$\mid$	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

$$\mathbf{F}\ p \wedge \mathbf{G}\ q \Rightarrow p\ \mathbf{U}\ r$$

Interpreting a Formula: RMD (2)

ϕ	$::=$	$\top$	[*true*]
	$\mid$	$\perp$	[*false*]
	$\mid$	p	[propositional atom]
	$\mid$	$(\neg \phi)$	[logical negation]
	$\mid$	$(\phi \wedge \phi)$	[logical conjunction]
	$\mid$	$(\phi \vee \phi)$	[logical disjunction]
	$\mid$	$(\phi \Rightarrow \phi)$	[logical implication]
	$\mid$	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	$\mid$	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	$\mid$	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	$\mid$	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	$\mid$	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	$\mid$	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

$$\mathbf{F}\,(p \wedge \mathbf{G}\,q \Rightarrow p\,\mathbf{U}\,r)$$

Interpreting a Formula: RMD (3)

ϕ	::=	$\top$	[*true*]
	$\mid$	$\bot$	[*false*]
	$\mid$	p	[propositional atom]
	$\mid$	$(\neg\phi)$	[logical negation]
	$\mid$	$(\phi \wedge \phi)$	[logical conjunction]
	$\mid$	$(\phi \vee \phi)$	[logical disjunction]
	$\mid$	$(\phi \Rightarrow \phi)$	[logical implication]
	$\mid$	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	$\mid$	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	$\mid$	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	$\mid$	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	$\mid$	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	$\mid$	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

$\mathbf{F}\ p \wedge (\mathbf{G}\ q \Rightarrow p\ \mathbf{U}\ r)$

Interpreting a Formula: RMD (4)

ϕ	$::=$	$\top$	[*true*]
	$\mid$	$\bot$	[*false*]
	$\mid$	p	[propositional atom]
	$\mid$	$(\neg\phi)$	[logical negation]
	$\mid$	$(\phi \wedge \phi)$	[logical conjunction]
	$\mid$	$(\phi \vee \phi)$	[logical disjunction]
	$\mid$	$(\phi \Rightarrow \phi)$	[logical implication]
	$\mid$	$(\mathbf{X}\,\phi)$	[ne**X**t state]
	$\mid$	$(\mathbf{F}\,\phi)$	[some **F**uture state]
	$\mid$	$(\mathbf{G}\,\phi)$	[all future states (**G**lobally)]
	$\mid$	$(\phi\,\mathbf{U}\,\phi)$	[**U**ntil]
	$\mid$	$(\phi\,\mathbf{W}\,\phi)$	[**W**eak-until]
	$\mid$	$(\phi\,\mathbf{R}\,\phi)$	[**R**elease]

$$\mathbf{F}\ p \wedge ((\mathbf{G}\ q \Rightarrow p)\ \mathbf{U}\ r)$$

Interpreting a Formula: PT vs. LMD vs. RMD

$$\mathbf{F}\ p \wedge \mathbf{G}\ q \Rightarrow p\ \mathbf{U}\ r$$

Deriving Subformulas from a Parse Tree

Enumerate all **subformulas** of:

$$\mathbf{F}\,(p \Rightarrow \mathbf{G}\, r) \vee ((\neg\, q)\ \mathbf{U}\ p)$$

Labelled Transition System (LTS)

$$\boxed{M = (S, \longrightarrow, L), \text{ given } P}$$

Q. Formulate **deadlock freedom**:

From any state, it is always possible to make progress.

Path Satisfaction: Logical Operations

A path satisfies a proposition
if its initial state ("current state") satisfies it.

$$s_1 \to s_2 \to \cdots \to s_{i-1} \to s_i \to s_{i+1} \to \cdots$$

$\pi \models p$

$\pi \models \top$

$\pi \models \bot$

$\pi \models \neg\phi$

$\pi \models \phi 1 \wedge \phi 2$

$\pi \models \phi 1 \vee \phi 2$

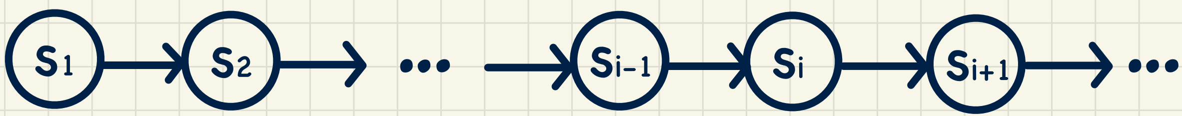
$\pi \models \phi 1 \Rightarrow \phi 2$

[redacted]

Path Satisfaction: Temporal Operations (1)

A path satisfies $X\phi$

if the next state (of the "current state") satisfies it.

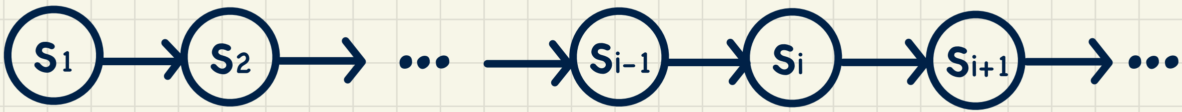


Formulation (over a path)

Q. What is $\pi 3 \vDash X\ p$ checking?

Path Satisfaction: Temporal Operations (2)

A path satisfies $G\phi$
if the every state satisfies it.



Formulation (over a path)

Path Satisfaction: Temporal Operations (3)

A path satisfies $F\phi$

if some future state satisfies it.

$$s_1 \to s_2 \to \cdots \to s_{i-1} \to s_i \to s_{i+1} \to \cdots$$

Formulation (over a path)

Model Satisfaction

Given:

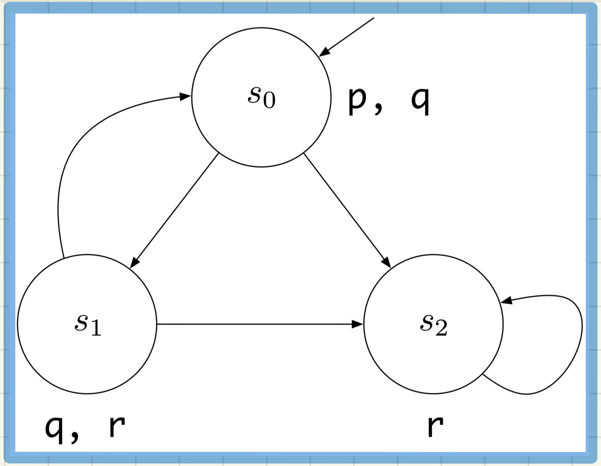
- Model $M = (S, \longrightarrow, L)$
- State $s \in S$
- LTL Formula ϕ

$M, s \vDash \phi$ iff for every path π of M starting at s, $\pi \vDash \phi$.

Formulation (over all paths)

How to prove vs. disprove $M, s \vDash \phi$?

Model vs. Path Satisfaction: Exercises (1.1)



Recall: $\pi \models p \Leftrightarrow p \in L(s_1)$

Say: $\pi = s_0 \longrightarrow s_1 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow \ldots$

$\pi \models \top$

$\pi \not\models \bot$

$\pi \models p \land q$

$\pi \models p \lor q$

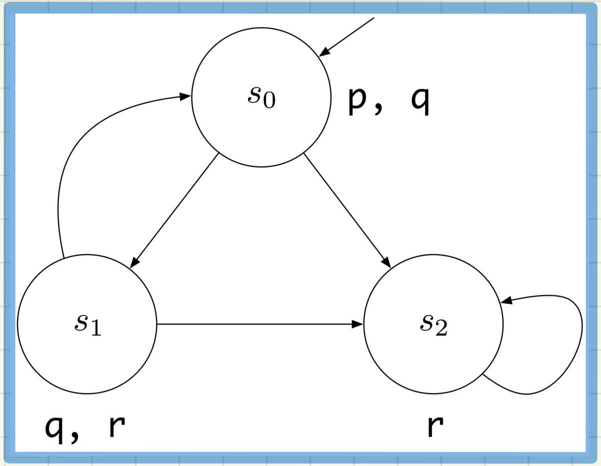
$\pi \models p \Rightarrow q$

$\pi \models r$

$\pi \models r \Rightarrow p \land q \land r$

Exercise: What if we change the LHS to π^2?

Model vs. Path Satisfaction: Exercises (1.2)



$s \models p \Leftrightarrow$ all π starting at s, $\pi \models p$

$s_0 \models \top$

$s_0 \not\models \bot$

$s_0 \models p \land q$

$s_0 \models p \lor q$

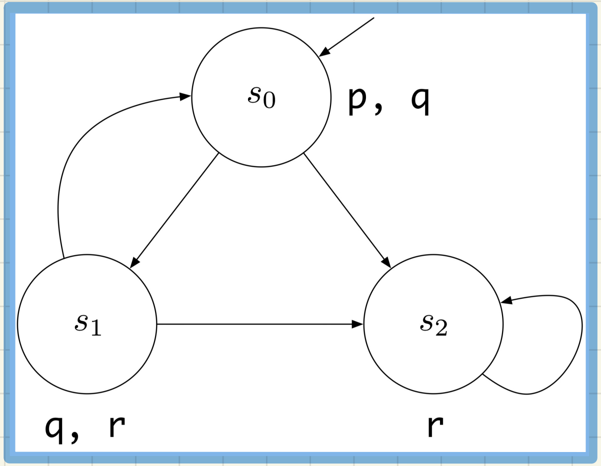
$s_0 \models p \Rightarrow q$

$s_0 \models r$

$s_0 \models r \Rightarrow p \land q \land r$

Exercise: What if we change the LHS to s_1?

Model vs. Path Satisfaction: Exercises (2.1)



Recall: $\pi \models \mathbf{X}\ \phi \Leftrightarrow \pi^2 \models \phi$

Say: $\pi = s_0 \longrightarrow s_1 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow \ldots$

$\pi \models \mathbf{X}\ \top$

$\pi \not\models \mathbf{X}\ \bot$

$\pi \models \mathbf{X}\ (q \land r)$

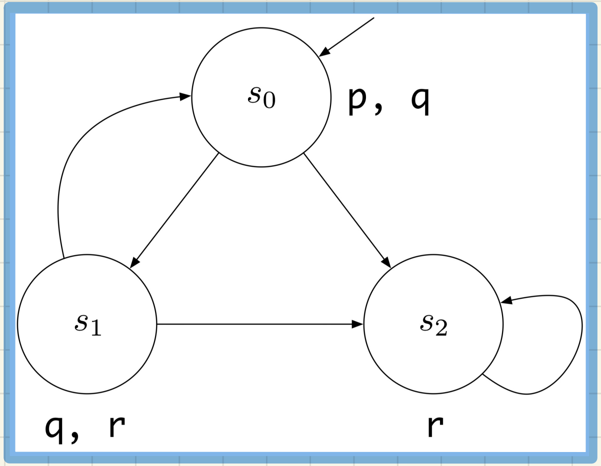
$\pi \models \mathbf{X}\ q \land r$

$\pi \models \mathbf{X}\ (q \Rightarrow r)$

$\pi \models \mathbf{X}\ q \Rightarrow r$

Exercise: What if we change the LHS to π^2?

Model vs. Path Satisfaction: Exercises (2.2)



$s \models \phi \Leftrightarrow$ **all** π starting at s, $\pi \models \phi$

$s_0 \models \mathbf{X}\ \top$

$s_0 \not\models \mathbf{X}\ \bot$

$s_0 \models \mathbf{X}\ (q \land r)$

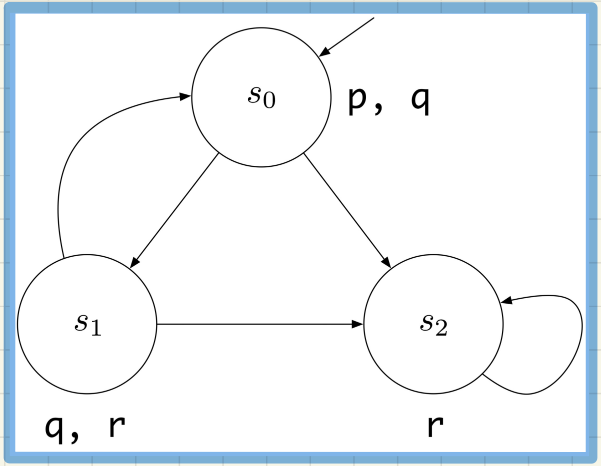
$s_0 \models \mathbf{X}\ q \land r$

$s_0 \models \mathbf{X}\ (q \Rightarrow r)$

$s_0 \models \mathbf{X}\ q \Rightarrow r$

Exercise: What if we change the LHS to s_1?

Model vs. Path Satisfaction: Exercises (3.1)



$\pi \models \mathbf{G}\ \phi \Leftrightarrow \forall i \bullet i \geq 1 \Rightarrow \pi^i \models \phi$

Say: $\pi = s_0 \longrightarrow s_1 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow \ldots$

$\pi \models \mathbf{G}\ \top$

$\pi \not\models \mathbf{G}\ \bot$

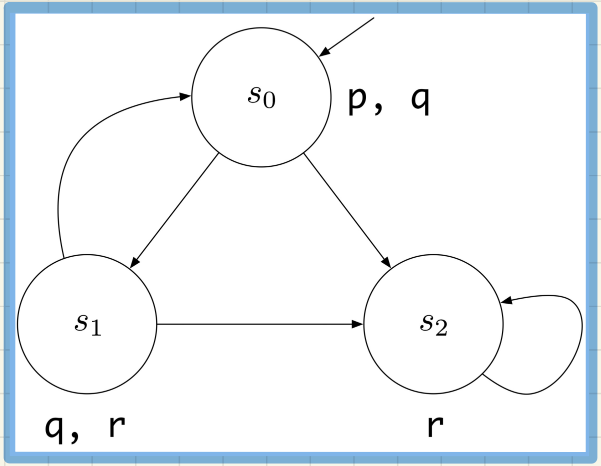
$\pi \models \mathbf{G}\ \neg(p \wedge r)$

$\pi \models \mathbf{G}\ r$

$\pi \models \mathbf{G}\ r$

Exercise: What if we change the LHS to π^2?

Model vs. Path Satisfaction: Exercises (3.2)



$s \models \phi \Leftrightarrow$ all π starting at s, $\pi \models \phi$

$s_0 \models \mathbf{G}\ \top$

$s_0 \not\models \mathbf{G}\ \bot$

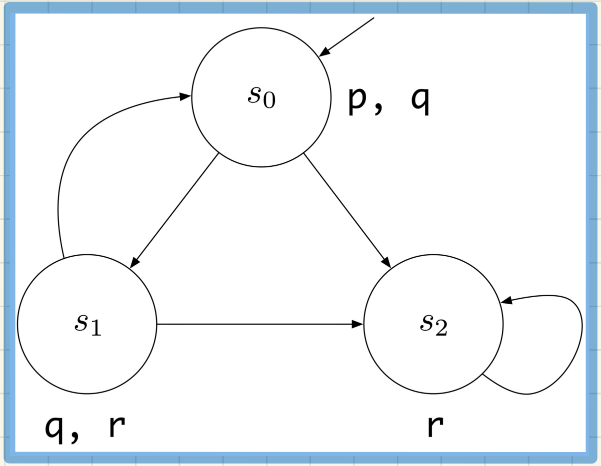
$s_0 \models \mathbf{G}\ \neg(p \land r)$

$s_0 \models \mathbf{G}\ r$

$s_2 \models \mathbf{G}\ r$

Exercise: What if we change the LHS to s_1?

Model vs. Path Satisfaction: Exercises (4.1)



$\pi \models \mathbf{F}\ \phi \Leftrightarrow \exists\, i \bullet i \geq 1 \wedge \pi^i \models \phi$

Say: $\pi = s_0 \longrightarrow s_1 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow \ldots$

$\pi \models \mathbf{F}\ \top$

$\pi \not\models \mathbf{F}\ \bot$

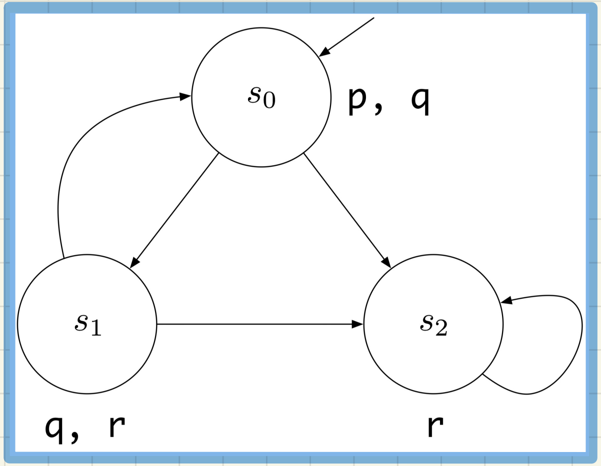
$\pi \models \mathbf{F}\ \neg(p \wedge r)$

$\pi \models \mathbf{F}\ r$

$\pi \models \mathbf{F}\ (q \wedge r)$

Exercise: What if we change the LHS to π^2?

Model vs. Path Satisfaction: Exercises (4.2)



$s \models \phi \Leftrightarrow$ **all** π starting at s, $\pi \models \phi$

$s_0 \models \mathbf{F}\ \top$

$s_0 \not\models \mathbf{F}\ \bot$

$s_0 \models \mathbf{F}\ \neg(p \land r)$

$s_0 \models \mathbf{F}\ r$

$s_0 \models \mathbf{F}\ (q \land r)$

Exercise: What if we change the LHS to s_1?

Nesting "Global" and "Future" in LTL Formulas

$$s \models \mathbf{FG}\ \phi$$

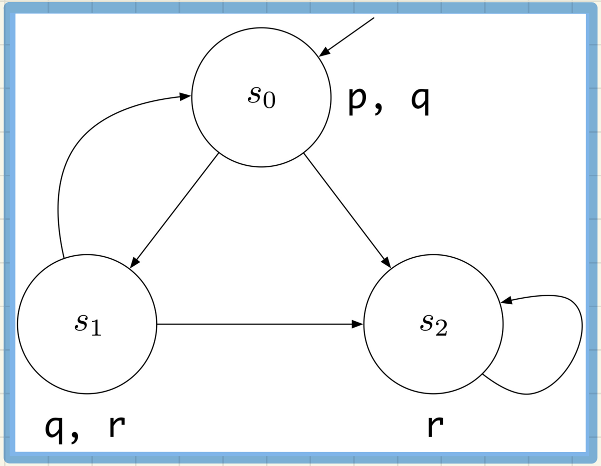
[redacted]

Q. Formulate the above nested pattern of LTL operator.

Q. How to prove the above nested pattern of LTL operators?

Q. How to disprove the above nested pattern of LTL operators?

Model Satisfaction: Exercises (5.1)



$s \models \phi \Leftrightarrow$ all π starting at s, $\pi \models \phi$

$s_0 \models \mathbf{FG}\ r$

$s_0 \models \mathbf{FG}\ (p \lor q)$

$s_0 \models \mathbf{FG}\ (p \lor r)$

Exercise: What if we change the LHS to s_2?

Nesting "Global" and "Future" in LTL Formulas

$$s \models \mathbf{F}\phi 1 \Rightarrow \mathbf{FG}\phi_2$$

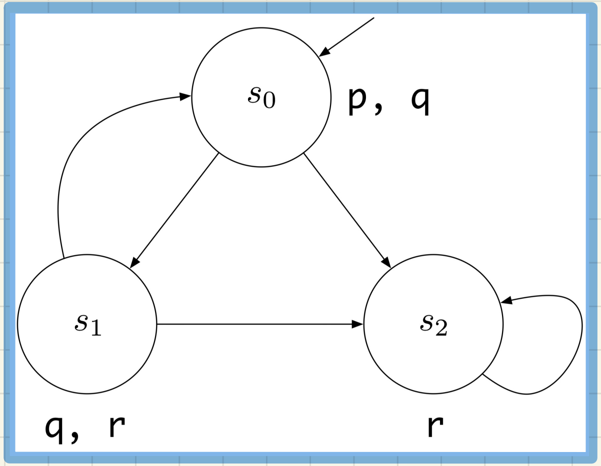
Each path π starting with s is s.t. if eventually $\phi 1$ holds on π, then $\phi 2$ eventually holds on π continuously.

Q. Formulate the above nested pattern of LTL operators.

Q. How to prove the above nested pattern of LTL operators?

Q. How to disprove the above nested pattern of LTL operators?

Model Satisfaction: Exercises (5.2)



$s \vDash \phi \Leftrightarrow$ all π starting at s, $\pi \vDash \phi$

$s_0 \vDash \mathbf{F}\,(\neg q \wedge r) \Rightarrow \mathbf{FG}\ r$

$s_0 \vDash \mathbf{F}\,(\neg q \vee r) \Rightarrow \mathbf{FG}\ r$

Exercise: What if we change the LHS to s_2?

Nesting "Global" and "Future" in LTL Formulas

$$s \models \mathbf{GF}\ \phi$$

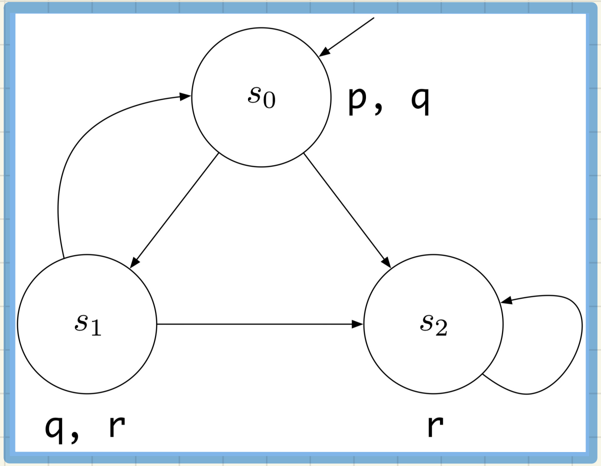
[redacted]

Q. Formulate the above nested pattern of LTL operator.

Q. How to prove the above nested pattern of LTL operators?

Q. How to disprove the above nested pattern of LTL operators?

Model Satisfaction: Exercises (6.1)



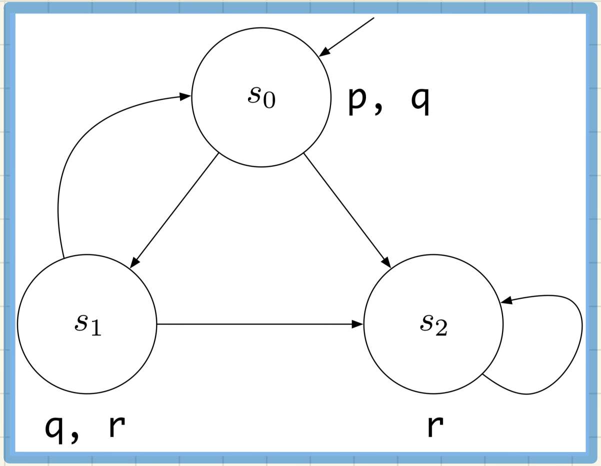
$s \models \phi \Leftrightarrow$ all π starting at s, $\pi \models \phi$

$s_0 \models \mathbf{GF}\ p$

$s_0 \models \mathbf{GF}\ (p \lor q)$

Exercise: What if we change the LHS to s_2?

Model Satisfaction: Exercises (6.2)



$s \models \phi \Leftrightarrow$ all π starting at s, $\pi \models \phi$

$s_0 \models \mathbf{GF}\ p \Rightarrow \mathbf{GF}\ r$

$s0 \models \mathbf{GF}\ r \Rightarrow \mathbf{GF}\ p$

Exercise: What if we change the LHS to s_2?

Path Satisfaction: Temporal Operations (4)

$\pi \models \phi 1 \ \mathbf{U} \ \phi 2$

There is some future state satisfies $\phi 2$, and until then, all states satisfy $\phi 1$.

$$s_1 \to s_2 \to \cdots \to s_{i-1} \to s_i \to s_{i+1} \to \cdots$$

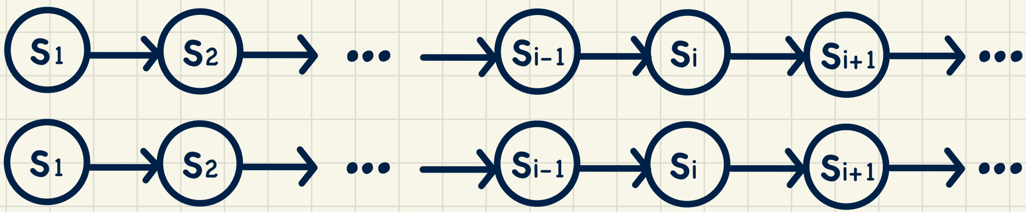
Formulation (over a path)

Path Satisfaction: Temporal Operations (5)

$\pi \models \phi 1 \; \mathbf{W} \; \phi 2$

If there is ever a future state that satisfies $\phi 2$, then until then, all states satisfy $\phi 1$.

Otherwise, $\phi 1$ must always be the case.



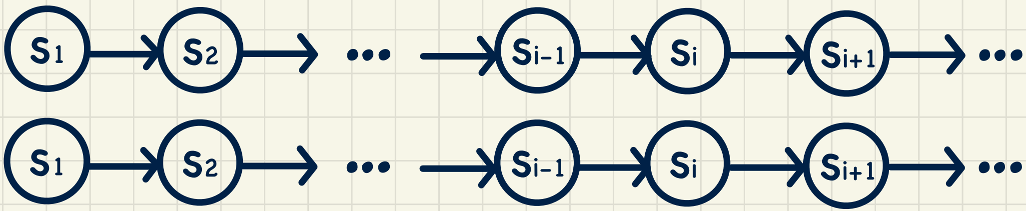
Formulation (over a path)

Path Satisfaction: Temporal Operations (6)

$\pi \models \phi 1 \ \mathbf{R} \ \phi 2$

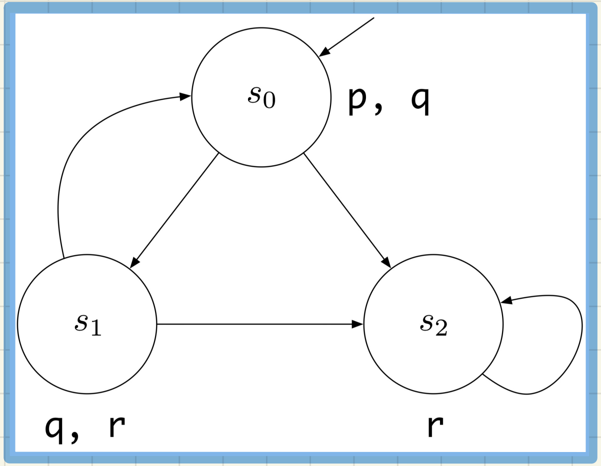
If there is ever a future state that satisfies $\phi 1$, then until then, all states satisfy $\phi 2$.

Otherwise, $\phi 2$ must always hold (i.e., never released).



Formulation (over a path)

Model Satisfaction: Exercises (7.1)



$s \vDash \phi \Leftrightarrow$ all π starting at s, $\pi \vDash \phi$

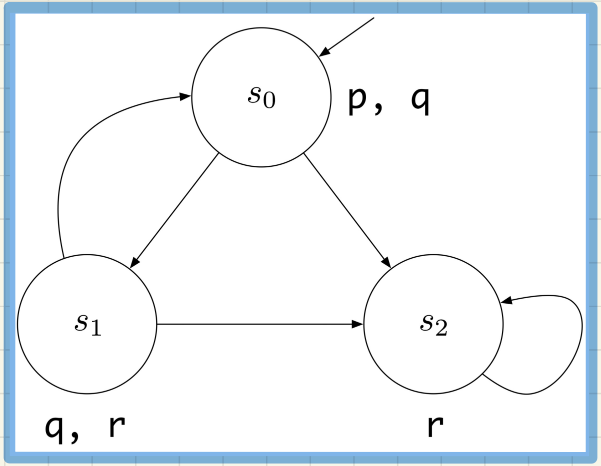
$s_0 \vDash p \; \mathbf{U} \; r$

$s_0 \vDash p \; \mathbf{W} \; r$

$s_0 \vDash r \; \mathbf{R} \; p$

Exercise: What if we change the LHS to s_2?

Model Satisfaction: Exercises (7.2)



$s \vDash \phi \Leftrightarrow$ all π starting at s, $\pi \vDash \phi$

$s_0 \vDash (p \lor r)\ \mathbf{U}\ (p \land r)$

$s_0 \vDash (p \lor r)\ \mathbf{W}\ (p \land r)$

$s_0 \vDash (p \land r)\ \mathbf{R}\ (p \lor r)$

Exercise: What if we change the LHS to s_2?

Formulating Natural Language in LTL (1)

Natural Language:

I had smoked until I was 22.

Atom t: I was 22

Atom s: I smoke

Q. Is s U t an appropriate formulation?

$$\pi \models \phi_1 \, \mathbf{U} \, \phi_2 \iff \left(\exists i \bullet i \geq 1 \land \left(\begin{array}{l} \pi^i \models \phi_2 \\ \land \\ (\forall j \bullet 1 \leq j \leq i - 1 \Rightarrow \pi^j \models \phi_1) \end{array} \right) \right)$$

Formulating Natural Language in LTL (2.1)

Natural Language:

It's impossible to reach a state where the system is started but not ready.

Assumed atoms:

- started
- ready

LTL Formulation

Formulating Natural Language in LTL (2.2)

Natural Language:

Whenever a request is made,
it will be acknowledged eventually.

Assumed atoms:

- requested
- acknowledged

LTL Formulation

Formulating Natural Language in LTL (2.3)

Natural Language:

An elevator traveling upwards at the 2nd floor
does not change its direction
when it has passengers wishing to to to the 5th floor.

Assumed atoms:

- floor2, floor5
- directionUp
- buttonPressed5

LTL Formulation

Lecture

Program Verification

Rules of wp Calculus

Correctness of Programs: Assignment (1)

What is the weakest precondition for a program `x := x + 1` to establish the postcondition $x > x_0$?

$$\{??\}\ \texttt{x := x + 1}\ \{x > x_0\}$$

Correctness of Programs: Assignment (2)

What is the weakest precondition for a program `x := x + 1` to establish the postcondition $x > x_0$?

$$\{??\}\ \texttt{x := x + 1}\ \{x = 23\}$$

Rules of Weakest Precondition: Conditionals

wp(**if** B **then** S1 **else** S2 **end**, R)

Correctness of Programs: Conditionals

Is this program correct?

$\{x > 0 \wedge y > 0\}$

```
if x > y then
  bigger := x ; smaller := y
else
  bigger := y ; smaller := x
end
```

$\{bigger \geq smaller\}$

Correctness of Programs: Sequential Composition

Is { *True* } `tmp := x; x := y; y := tmp` { $x > y$ } correct?

Contracts of Loops: Example

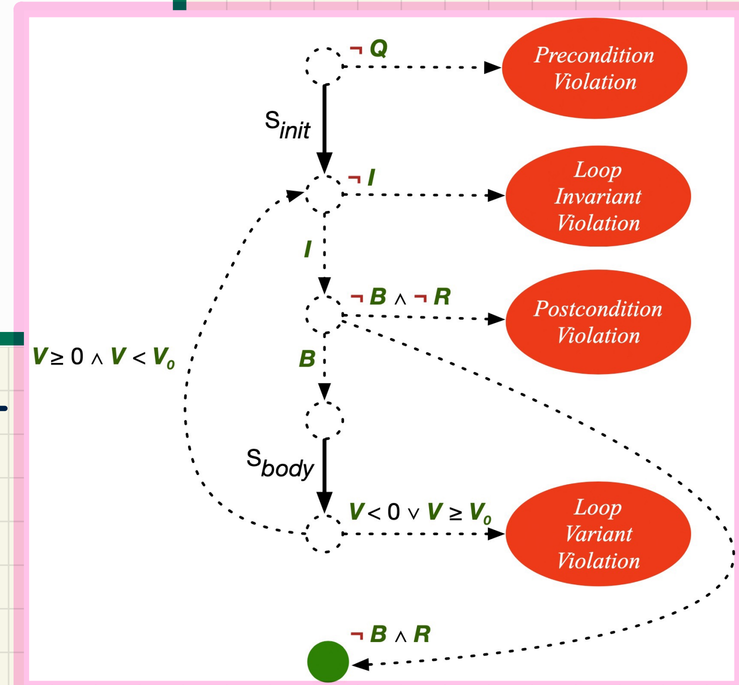
Assume: Q and R are true

Specification

```
I(i) == (1 <= i) /\ (i <= 6)
V(i) == 6 - i
--algorithm loop_invariant_test
 variables i = 1, variant_pre = 0, variant_post = 0;
 {
   assert I(i);
   while (i <= 5) {
     variant_pre := V(i);
     i := i + 1;
     variant_post := V(i);
     assert variant_post >= 0;
     assert variant_post < variant_pre;
     assert I(i);
   } ;
 }
```

end of iteration	i	I	V	B

Runtime Checks



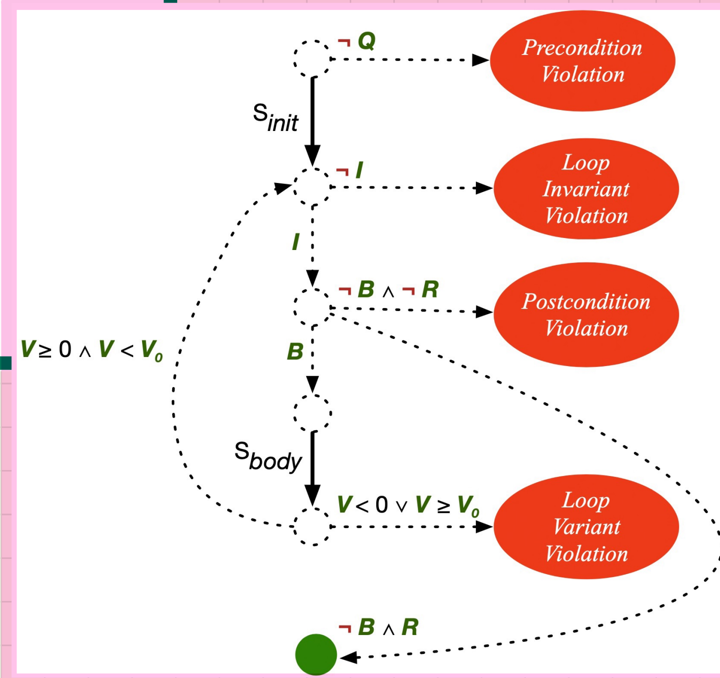
Contracts of Loops: Violations

Assume: Q and R are true

Specification

```
I(i) == (1 <= i) /\ (i <= 6)
V(i) == 6 - i
--algorithm loop_invariant_test
 variables i = 1, variant_pre = 0, variant_post = 0;
 {
   assert I(i);
   while (i <= 5) {
     variant_pre := V(i);
     i := i + 1;
     variant_post := V(i);
     assert variant_post >= 0;
     assert variant_post < variant_pre;
     assert I(i);
   } ;
 }
```

Runtime Checks



invariant: 1 <= i <= 5

variant: 5 - i

Correct Loops: Proof Obligations

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

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

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

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

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

    [redacted]

```
{Q}
S_init
assert I(...);
while( B ) {
  variant_pre := V(...);
  S_body
  variant_post := V(...);
  assert variant_post >= 0;
  assert variant_post < variant_pre;
  assert I(...);
}
{R}
```

Correct Loops: Proof Obligations

Example

Specification

```
I(i) == (1 <= i) /\ (i <= 6)
V(i) == 6 - i
--algorithm loop_invariant_test
  variables i = 1, variant_pre = 0, variant_post = 0;
  {
    assert I(i);
    while (i <= 5) {
      variant_pre := V(i);
      i := i + 1;
      variant_post := V(i);
      assert variant_post >= 0;
      assert variant_post < variant_pre;
      assert I(i);
    } ;
  }
```

- A loop is ***partially** correct* if:
  - Given precondition ***Q***, the initialization step S_{init} establishes ***LI*** *I*.
    [redacted]
  - At the end of S_{body}, if not yet to exit, ***LI*** *I* is maintained.
    [redacted]
  - If ready to exit and ***LI*** *I* maintained, postcondition ***R*** is established.
    [redacted]
- A loop ***terminates*** if:
  - Given ***LI*** *I*, and not yet to exit, S_{body} maintains ***LV*** *V* as non-negative.
    [redacted]
  - Given ***LI*** *I*, and not yet to exit, S_{body} decrements ***LV*** *V*.
    [redacted]