EECS3342 System Specification and Refinement
Fall 2026

Digest of Lecture 12, October 27

Invariant-Preservation Proof Obligations and Inference Rules

Jackie Wang

How to use this Digest

Use this digest as a **roadmap and checklist**. It highlights the main ideas, important examples, common pitfalls, practice targets, and where to return in the course materials. It does **not** replace attending or reviewing the session, using the recording when needed, studying Jackie's annotated iPad notes and other official course materials, completing assigned work, or practising independently. Use it to identify what you understand, what you need to revisit, and what you should be able to do on your own.

If time is limited, start with the sections most relevant to your current review and return to the rest later; selective use is better than skipping the digest because it looks long.

This is an AI-assisted study roadmap reviewed by Jackie and grounded in the session transcript and Jackie's annotated iPad notes. Like any course document, it may still contain an error or omission. If something seems technically doubtful or inconsistent with the original course materials, return to those materials and ask Jackie for clarification rather than treating the digest as final authority.

Table of Contents

Main Ideas 2

Roadmap 2

Key Terms, Notation, and Structures 6

Key Examples and Demonstrations 7

Common Pitfalls and Misconceptions 8

What You Should Be Able to Do 8

Suggested Practice 9

Where to Review 10

Closing Reminder 10

Main Ideas

- **Invariant preservation was moved from an informal sketch to a more systematic proof-obligation framework.** Jackie revisited the pre-state/post-state idea, then showed how a post-state invariant is formed by priming state variables and eliminating those primes through the event's before-after predicate (BAP).
- **The bank-withdraw example showed why function updates require more care than scalar updates.** The balance variable is a partial function, and a withdrawal changes it by relational overriding. The post-state invariant therefore contains the entire overridden function in both the domain and application positions.
- **Only state variables are primed.** Constants and quantified dummy variables do not acquire pre-state/post-state identities. If a state variable is unchanged by an event, its BAP simply records an identity such as $A' = A$.
- **Formal model components were organized into lists and dependencies.** The lecture introduced c, $A(c)$, v, v', $I(c, v)$, $G(c, v)$, $E(c, v)$, and the BAP $v' = E(c, v)$, with labels used to refer to model elements such as `axm0_1`, `inv0_1`, and `inv0_2`.
- **Invariant-preservation POs are generated per event and per invariant.** For M_0, two events and two invariants give four INV proof obligations. Adding an invariant adds one new INV PO for each existing event; adding an event adds one for each existing invariant.
- **Changing a model can require proof maintenance even without creating new POs.** Changing an event guard does not by itself create a new event/invariant combination, but every existing PO involving that event must be revisited.
- **Sequents and inference rules are related but are not the same thing.** A sequent is a proposition to prove. An inference rule has an antecedent that is a set of sequents, a consequent that is one sequent, and a rule label; to prove the consequent, it is sufficient to prove the antecedent sequents.
- **At the October 27 lecture, Written Test 1 logistics were an immediate priority.** Jackie reminded students that the test was October 28 in the enrolled lab session and emphasized the guide/example questions, blank sketch paper, a mobile phone for Duo verification, and working EECS/Passport York access.

Roadmap

1. October 27 assessment and lab reminders

At this lecture, Jackie reminded students that Written Test 1 was scheduled for **October 28** in each student's enrolled lab session. He pointed students back to the guide, example questions, and posted walkthrough video. He also asked students to bring blank sketch paper, bring a mobile phone for Duo verification, and make sure both EECS and Passport

York access worked before the test. These were time-specific reminders from the October 27 class rather than permanent course policies.

Jackie also reminded students to keep Lab 3 on track and said Lab 4 would be the final lab of the semester. The exact Lab 3 deadline is not repeated here because his spoken wording was tentative ("I believe") and the annotated notes did not independently record that date.

2. Rebuilding invariant preservation with the bank-withdraw example

Pages 3–4 of Jackie's annotated iPad notes use a bank model to revisit the mechanics introduced in Lecture 11. The state variable b represents balances as a partial function from accounts to integers, and the selected invariant is

$$I: \quad \forall x \bullet x \in \mathrm{dom}(b) \Rightarrow b(x) \geq -c.$$

The event `withdraw` has parameters a (an account) and $v \in \mathbb{N}$, with the displayed guard

$$a \in \mathrm{dom}(b).$$

Jackie explicitly noted that this guard alone may not be strong enough to preserve the invariant; the purpose of the example was to practise forming the post-state condition, not to finish the bank design.

The action updates the balance function by overriding the entry for a:

$$b := b <+ \{a \mapsto b(a) - v\}.$$

The corresponding BAP is

$$b' = b <+ \{a \mapsto b(a) - v\}.$$

This is the fact used to eliminate b' from the post-state invariant.

3. What gets primed, and what does not

To form the intermediate post-state invariant I', Jackie asked which symbols in I should receive primes. Only the state variable b is primed:

$$\forall x \bullet x \in \mathrm{dom}(b') \Rightarrow b'(x) \geq -c.$$

The quantified x is a dummy variable local to the quantifier, not a state variable. The constant c also does not change. Therefore neither x nor c is primed.

A student asked about variables that an event does not modify. Jackie clarified that such variables still belong to the post-state variable list; their BAP just records that they are unchanged. For a hypothetical additional variable A, for example,

$$A' = A.$$

This is an important mechanical rule when a model later contains several state variables.

4. Expanding the post-state invariant under relational overriding

Substituting the withdraw BAP into I' produces the full post-state condition shown on p. 4 of Jackie's annotated iPad notes:

$$\forall x \bullet x \in \mathrm{dom}(b \mathbin{<\!\!+} \{a \mapsto b(a) - v\}) \Rightarrow (b \mathbin{<\!\!+} \{a \mapsto b(a) - v\})(x) \geq -c.$$

The decisive point is that this is not a scalar replacement such as "$b' = b - v$". The whole relation/function is replaced by its overridden version, so the substitution appears wherever b' appears: inside the domain and in the function application.

Jackie's interpretation was to ask whether every account in the *modified* balance function still satisfies the lower-bound invariant after account a's balance is decremented. This function-update example is more involved than the single numeric variable used in the Bridge Controller, but it demonstrates the same BAP substitution principle.

5. Formal components used in PO/VC rules

Page 5 organizes the notation Jackie will use in proof-obligation rules. The main components are:

- c: the list of constants;
- $A(c)$: the list of axiom labels, depending on the constants;
- v and v': lists of variables in the pre-state and post-state;
- $I(c, v)$: the list of invariant labels;
- $G(c, v)$: the guard of the event currently being considered;
- $E(c, v)$: the effects of that event's actions;
- $v' = E(c, v)$: the event's before-after predicate.

For Bridge Controller M_0, Jackie's annotations instantiate these ideas with $c = \langle d \rangle$, $v = \langle n \rangle$, $v' = \langle n' \rangle$, and invariant labels `inv0_1` and `inv0_2`. The event guards for the original `ML_out` and `ML_in` are `true`; their effects are $n + 1$ and $n - 1$ respectively. The corresponding BAPs are $n' = n + 1$ and $n' = n - 1$.

Jackie also revisited the two-variable swap to show why $E(c, v)$ and v' are lists whose positions must correspond. With variables ordered as $\langle x, y \rangle$, the swap effects are $\langle y, x \rangle$, so the BAP is equivalent to $x' = y \wedge y' = x$.

6. The invariant-preservation PO/VC schema

The formal invariant-preservation condition used in the lecture can be read in horizontal sequent form as

$$A(c),\ I(c, v),\ G(c, v) \ \vdash\ I_i\big(c, E(c, v)\big).$$

The hypotheses use *all* axioms, *all* pre-state invariants, and the guard for the particular event being checked. The goal refers to only one invariant at index i, rewritten in its post-state form by applying the event's effects/BAP.

That distinction between I and I_i is central: invariant preservation is verified one invariant at a time, even though all current invariants may be used as hypotheses.

7. Counting and naming the M_0 invariant-preservation POs

Pages 6–8 turn the schema into a counting exercise. M_0 has two events, `ML_out` and `ML_in`, and two invariants, `inv0_1` and `inv0_2`. Those are two independent choices, so there are

$$2 \times 2 = 4$$

INV proof obligations. The names developed in class are:

`ML_out/inv0_1/INV`,	`ML_out/inv0_2/INV`,
`ML_in/inv0_1/INV`,	`ML_in/inv0_2/INV`.

The final `INV` identifies the proof-obligation family: does this event preserve this particular invariant?

Jackie then extended the model. Adding a new invariant `inv0_3` creates two more INV POs, one for each existing event. If the model then has three invariants and a new event E is added, that event creates three more INV POs, one for each invariant. Rodin automates this generation, but Jackie emphasized that students should understand why the obligations appear.

8. Proof maintenance after a model change

A student asked what happens if an existing event guard changes. Jackie explained that changing the guard of `ML_out` does not add a new event or invariant, so it does not automatically create a new event/invariant combination. However, all existing proof obligations involving `ML_out` must be revisited because their hypotheses have changed. Depending on the modification, an old proof may remain easy, may need a different argument, or may fail.

The broader lesson is to distinguish three kinds of model evolution: adding invariants can create new POs, adding events can create new POs, and modifying an existing component can require rechecking existing POs.

9. Inference rules: syntax and meaning

The final page introduced inference rules and explicitly contrasted them with sequents. An inference rule has:

- an antecedent A, which is a *set of sequents*;
- a consequent C, which is one sequent;
- a label L, which names the rule.

The operational reading is backward: **to prove the consequent C, it is sufficient to prove the antecedent sequents A instead**. Jackie warned not to reverse that direction.

The lecture also discussed the empty-antecedent case. If a rule has no antecedent sequents, proving the antecedent reduces to proving `true`, so the consequent sequent C is treated as an axiom for that rule and does not require further reduction. Detailed use of individual inference rules was deferred to the next lecture.

Key Terms, Notation, and Structures

Invariant preservation
The requirement that an enabled event take a state satisfying the invariants to a post-state that also satisfies the relevant invariant. The lecture formulates this as a PO/VC sequent.

Post-state invariant I'
An intermediate form obtained by priming state variables in the invariant. Constants and quantified dummy variables are not primed. BAP substitution then eliminates the primes.

Before-after predicate (BAP)
A predicate connecting pre-state and post-state variable values. For a modified variable, it states the event effect, such as $n' = n + 1$ or $b' = b <+ \{a \mapsto b(a) - v\}$. For an unchanged variable, the BAP is an identity such as $A' = A$.

Relational overriding
The update mechanism used in the bank example to replace the mapping for one account while retaining the rest of the balance function. The lecture writes the update using Event-B's overriding operator, with ASCII form <+.

c, $A(c)$, v, v', $I(c, v)$
The formal component notation introduced on p. 5: constants; axiom labels; pre-state variables; post-state variables; and invariant labels with their constant/variable dependencies.

$G(c, v)$ and $E(c, v)$
The guard and effect of the particular event being considered. Guards are per-event; $E(c, v)$ supplies the right-hand-side effects used to form the BAP $v' = E(c, v)$.

INV proof obligation
A proof obligation asking whether one event preserves one particular invariant. Its name identifies the event, invariant label, and PO family, for example `ML_out/inv0_1/INV`.

Sequent
A proposition of the form $H \vdash G$ that must be proved. Its hypotheses and goal are predicates. This is distinct from an inference rule.

Inference rule
A labeled rule whose antecedent is a set of *sequents* and whose consequent is one sequent. It permits the proof of the consequent to be reduced to proofs of the antecedent sequents.

Key Examples and Demonstrations

1. Bank withdrawal: from I to the post-state I'

Setup: Use the balance function b, lower-bound invariant $\forall x \bullet x \in \mathrm{dom}(b) \Rightarrow b(x) \geq -c$, and the withdraw event that overrides account a with balance $b(a) - v$.

Main point: Prime only the state variable b, then replace b' by the event's BAP. The final post-state condition contains the overridden function in both $\mathrm{dom}(\cdot)$ and application positions.

What to practise: Reproduce pp. 3–4 of Jackie's annotated iPad notes from the unprimed invariant to the fully substituted I', and explain why x and c are not primed.

2. Component lists for Bridge Controller M_0

Setup: Page 5 instantiates the abstract symbols for M_0: $c = \langle d \rangle$, $v = \langle n \rangle$, two invariant labels, and the two events `ML_out` and `ML_in`.

Main point: The component notation records dependencies and supplies a uniform way to state PO rules. Event effects give the BAPs $n' = n + 1$ and $n' = n - 1$.

What to practise: Given a small model, identify its constant list, variable list, invariant list, the guard/effect for one event, and the resulting BAP without mixing information from another event.

3. Counting and naming INV POs

Setup: Two events and two invariants produce four event/invariant combinations.

Main point: The four base names are the four event–invariant combinations listed in Roadmap §7, each ending in `/INV`. Adding `inv0_3` creates two more; adding a new event when three invariants exist creates three more.

What to practise: Recreate the PO list on pp. 6–8, then explain which POs must be revisited if the guard of `ML_out` changes and why no new name is required merely because the guard predicate changed.

4. Sequent versus inference rule

Setup: Compare the sequent notation reviewed from Lecture 11 with the inference-rule diagram on p. 10.

Main point: In a sequent, hypotheses and goals are predicates. In an inference rule, the antecedent A is a set of *sequents* and the consequent C is a single sequent. The proof direction is from the desired consequent backward to the antecedent obligations.

What to practise: Draw the inference-rule template from memory, label A, C, and L, and explain the empty-antecedent case without confusing it with the empty-hypothesis case inside a sequent.

Common Pitfalls and Misconceptions

- **Priming constants or quantified dummy variables.** Only state variables receive primes when forming a post-state invariant. In the bank example, b becomes b', while x and c remain unprimed.
- **Forgetting unchanged state variables.** A variable that is not modified by the event still belongs to the post-state list; its BAP is an identity such as $A' = A$.
- **Treating a function update as scalar arithmetic.** The bank balance variable is a function. Replacing b' requires the overridden function $b <+ \{a \mapsto b(a) - v\}$ everywhere b' occurs.
- **Assuming the displayed bank guard proves safety.** Jackie explicitly said $a \in \mathrm{dom}(b)$ may be insufficient to preserve the lower-bound invariant; the example was about PO construction and BAP substitution.
- **Mixing guards from different events.** $G(c, v)$ means the guard of the event whose PO is being generated. `ML_out` and `ML_in` are handled separately.
- **Using all post-state invariants as one giant goal.** The PO schema uses all current invariants as pre-state hypotheses but proves one indexed invariant I_i at a time.
- **Thinking model changes always create new PO names.** Adding an event or invariant can create new INV POs; changing an existing guard instead requires revisiting the existing POs for that event.
- **Confusing a sequent with an inference rule.** A sequent is itself a claim to prove. An inference rule operates on sequents: its antecedent is a set of sequents and its consequent is one sequent.
- **Reversing the inference-rule direction.** The lecture's operational reading is: to prove C, it is sufficient to prove the antecedent sequents A.

What You Should Be Able to Do

After this session, you should be able to:

- form a post-state invariant by priming exactly the state variables and leaving constants and quantified dummy variables unprimed;
- derive and use a BAP for both changed and unchanged variables;
- explain the withdraw update $b := b <+ \{a \mapsto b(a) - v\}$ and substitute its BAP into the post-state invariant;
- interpret the fully substituted withdraw condition in terms of the modified balance function's domain and account balances;

- identify the roles of c, $A(c)$, v, v', $I(c, v)$, $G(c, v)$, and $E(c, v)$ in the formal PO notation;
- read the invariant-preservation schema $A(c), I(c, v), G(c, v) \vdash I_i(c, E(c, v))$ and explain why the left side uses all pre-state invariants while the goal selects one invariant at index i;
- construct the BAPs for `ML_out`, `ML_in`, and the multi-variable swap while keeping variable/effect order aligned;
- count and name INV proof obligations from the number of events and invariants in the model;
- predict how adding an invariant, adding an event, or modifying a guard affects the PO set or the need to revisit existing proofs;
- distinguish a sequent from an inference rule, identify the antecedent, consequent, and label of an inference rule, and state the correct proof direction;
- explain why an inference rule with no antecedent treats its consequent sequent as an axiom for that rule.

Suggested Practice

1. **Reproduce the bank-withdraw substitution.** Starting from I, write I' with primes and then eliminate b' using the overriding BAP. Check that neither the constant c nor the quantified variable x is primed.
2. **Test an unchanged-variable case.** Add a hypothetical state variable A that withdraw does not modify and include $A' = A$ in the BAP before doing the post-state substitution.
3. **Review the formal component map on p. 5.** For Bridge Controller M_0, write down c, v, v', the invariant labels, each event's guard, effect, and BAP. Then repeat the swap example with $v = \langle x, y \rangle$ and verify the order of the effects.
4. **Regenerate the four base INV POs.** Without looking at p. 8, list the two `ML_out` obligations and the two `ML_in` obligations, using the event/invariant/`INV` naming convention.
5. **Apply the model-extension counting logic.** Explain why adding one new invariant to the two-event model adds two POs, and why a new event in a three-invariant model adds three.
6. **Do the proof-maintenance thought experiment.** Suppose only the guard of `ML_out` changes. Identify which existing POs must be revisited and explain why the event–invariant pairs themselves have not changed.
7. **Compare sequents and inference rules.** Jackie explicitly asked students to compare the sequent material with the inference-rule material before the next class. Redraw both structures and explain, in your own words, what kinds of objects appear on each side/level and how the proof direction differs.

Where to Review

- **Jackie's annotated iPad notes, p. 2:** October 27 reminders for the October 28 Written Test 1, including the guide and example questions, sketch paper, DuoMobile, and EECS and Passport York access.
- **Jackie's annotated iPad notes, pp. 3–4:** bank-withdraw invariant preservation, priming rules, the relational-override BAP, and the fully expanded post-state invariant.
- **Jackie's annotated iPad notes, p. 5:** formal model-component notation (c, $A(c)$, v, v', I, G, E), Bridge Controller M_0 instantiations, and the swap BAP example.
- **Jackie's annotated iPad notes, pp. 6–9:** the invariant-preservation PO/VC schema, why M_0 yields four base sequents, PO naming, model extensions, and which POs are revisited after a guard change.
- **Jackie's annotated iPad notes, p. 10:** inference-rule syntax and semantics, antecedent versus consequent, rule label, proof direction, and the empty-antecedent case.
- **Lecture recording:** return to the bank-function substitution if the override expression on pp. 3–4 is difficult to interpret, and to the later PO-counting discussion if the distinction between creating and revisiting POs is unclear.
- **Written Test 1 materials referenced on October 27:** the guide, example questions, and posted walkthrough video for the October 28 assessment.

Closing Reminder

Use this digest as a roadmap for active review, not as a substitute for the learning process itself. Attend Jackie's in-person sessions and pay attention to how he develops ideas and works through the important reasoning step by step. If a transition, explanation, example, or reasoning step remains unclear afterwards, return to the recording to fill that gap.

For structured review, start with Jackie's annotated iPad notes because they most closely reflect the actual flow and development of the class. Work through them in manageable logical units. After each logical unit, study the corresponding part of the presentation slides alongside it. Jackie does not normally teach by reading the slides line by line, so the slides may contain relevant supporting material that was not explicitly read aloud.

Finally, choose a suitable selection of the important reasoning from this session and work through it again on your own. The goal is not to memorize Jackie's recorded steps, but to determine whether you can reproduce the reasoning from your own understanding and explain why the important steps work.