EECS3342 System Specification and Refinement
Fall 2026

Digest of Lecture 18, November 17, 2026

Invariant Preservation in Refinement and New Events

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 5

Key Examples and Demonstrations 6

Common Pitfalls and Misconceptions 7

What You Should Be Able to Do 8

Suggested Practice 8

Where to Review 9

Closing Reminder 10

Main Ideas

Lecture 18 connected refinement proof obligations to the behavior of the Bridge Controller models and then introduced the first genuinely new events in M1. The main movement was:

- deepen the intuition behind **guard strengthening** $H \Rightarrow G$, including its set picture, contrapositive, and a concrete example of what goes wrong when a proposed concrete guard is weaker than the abstract guard;
- formulate **invariant preservation in refinement**, where the proof context contains both abstract and concrete invariants and the goal accounts for both the abstract and concrete event effects;
- visualize refinement using a **commuting diagram**: every concrete state transition should be simulated by a corresponding abstract state transition, with the pre- and post-states related by the refinement invariants;
- formulate **invariant establishment in refinement**, using the abstract and concrete initialization effects together;
- distinguish old events from genuinely **new events** and introduce the actions of `IL_in` and `IL_out` in M1.

For the November 18 Written Test 2, Jackie confirmed that **slide 58 is the end of the test coverage**. He also said the Event-B Summary and Inference Rule sets 1 and 2 would be available during the test. The later refinement-level invariant-preservation proof exercises were presented as material to practise for the exam rather than Written Test 2.

Roadmap

1. Revisit guard strengthening as a behavioral inclusion condition

For an old event in a refinement, let

$$H \quad \text{be the concrete guard}, \qquad G \quad \text{be the abstract guard}.$$

Guard strengthening requires

$$H \Rightarrow G.$$

Jackie used a set picture to interpret this implication. Every state satisfying H must also satisfy G, so the concrete enabled-state set is contained in (or may equal) the abstract enabled-state set. Equivalence is allowed; a strict reduction of enabled behavior is common but not required.

The same idea can be read through the contrapositive:

$$\neg G \Rightarrow \neg H.$$

If an event is impossible in the abstract model, the corresponding concrete event must also be impossible. Refinement may remove abstract behavior, but it must not introduce a concrete-only enabled scenario.

2. Use `ML_out` to diagnose a failed strengthening

For `ML_out`, the correct guards are

$$G:\ n < d$$

in M0 and

$$H:\ a + b < d, \qquad c = 0$$

in M1, together with the gluing invariant

$$a + b + c = n.$$

When $c = 0$, the concrete condition $a + b < d$ corresponds to $n < d$, so the strengthening direction is consistent.

Jackie then deliberately changed the concrete capacity guard to

$$a + b \le d.$$

Using $c = 0$ and $a + b + c = n$, this becomes $n \le d$. The concrete event can then be enabled at $n = d$, while the abstract guard $n < d$ is false. This is a new concrete-only enabling scenario, so the proposed guard **fails guard strengthening**.

3. Formulate invariant preservation for an old event in a refinement

Lecture 18 then moved beyond the single-model M0 preservation rule. In a refinement, the proof context includes:

$$A(c) \quad \text{axioms},$$

$$I(c, v) \quad \text{abstract invariants}, \qquad J(c, v, w) \quad \text{concrete invariants},$$

$$H(c, w) \quad \text{the concrete event guard}.$$

If $E(c, v)$ is the abstract event effect and $F(c, w)$ is the concrete event effect, the preservation goal for concrete invariant J_i has the form

$$A(c),\ I(c, v),\ J(c, v, w),\ H(c, w)\ \vdash\ J_i\big(c, E(c, v), F(c, w)\big).$$

The important difference from the earlier M0-only rule is that a refinement may need both abstract and concrete state changes when constructing the post-state invariant.

4. Count and instantiate the M1 preservation obligations

For the two old concrete events `ML_out` and `ML_in`, M1 has five concrete invariants. Therefore Jackie counted

$$2 \text{ old events} \times 5 \text{ concrete invariants} = 10$$

invariant-preservation obligations for this part of M1.

Two representative sequents were developed:

- `ML_out/inv1_4/INV`, whose post-state linking-invariant goal becomes

$$(a + 1) + b + c = n + 1;$$

- `ML_in/inv1_5/INV`, whose post-state safety-invariant goal becomes

$$a = 0 \vee (c - 1) = 0.$$

The point is to start from the primed invariant and substitute the correct abstract and concrete event effects. Jackie explicitly corrected an in-class slip while doing this substitution, so use the final forms in the annotated notes rather than an intermediate spoken attempt.

5. Interpret invariant preservation through the commuting diagram

Jackie then built the commuting diagram for an old event such as `ML_out`. The abstract pre-state v and concrete pre-state w are related by the refinement invariants. When the concrete event is enabled, guard strengthening ensures that the abstract event is also enabled. The abstract transition produces v', while the concrete transition produces w'. Invariant preservation then requires the resulting post-states to remain related by the concrete/refinement invariants.

The central interpretation was:

> Each concrete state transition should be simulated by an abstract state transition.

The diagram "commutes" because either path—abstract transition followed by the post-state relation, or concrete transition followed by that relation—must lead to corresponding states.

6. Extend invariant establishment to the refinement

For initialization, Jackie distinguished the two effects

$$K(c) \quad \text{abstract initialization effect}, \qquad L(c) \quad \text{concrete initialization effect}.$$

The establishment rule for concrete invariant J_i is

$$A(c) \ \vdash \ J_i\big(c, K(c), L(c)\big).$$

M1 has one initialization event and five concrete invariants, so there are

$$1 \times 5 = 5$$

establishment obligations. Jackie used the linking and safety invariants as simple examples after initialization:

$$0 + 0 + 0 = 0, \qquad 0 = 0 \vee 0 = 0.$$

7. Introduce new events and their actions

M0 contains the old events `ML_out` and `ML_in`. M1 keeps concrete versions of those old events and additionally introduces two new events that do not exist in M0:

$$\texttt{IL_in}, \qquad \texttt{IL_out}.$$

Here `IL` refers to the island side of the bridge.

Jackie introduced their actions visually:

$$\begin{aligned} \texttt{IL_in}: &\quad a := a - 1, \quad b := b + 1, \\ \texttt{IL_out}: &\quad b := b - 1, \quad c := c + 1. \end{aligned}$$

Thus `IL_in` moves a car from the bridge onto the island, while `IL_out` moves a car from the island onto the bridge in the opposite direction. The guards for these new events were deliberately deferred to the next lecture.

Key Terms, Notation, and Structures

Guard strengthening
The refinement condition

$$H \Rightarrow G,$$

where H is the concrete guard and G is the corresponding abstract guard. The satisfying-state set of H must be contained in or equal to that of G.

Contrapositive view of guard strengthening

$$\neg G \Rightarrow \neg H.$$

A state in which the abstract event is disabled must not enable the concrete counterpart.

Abstract versus concrete invariants
$I(c, v)$ denotes the abstract invariants. $J(c, v, w)$ denotes the concrete/refinement invariants, which may involve both abstract and concrete variables; the gluing invariant is the key example.

Abstract and concrete event effects
$E(c, v)$ denotes the effect of the abstract old event, while $F(c, w)$ denotes the effect of its concrete version.

Invariant preservation in refinement
For each concrete invariant J_i, prove that applying the corresponding abstract and concrete event effects still yields a state satisfying J_i.

Commuting diagram
A diagram relating abstract and concrete pre/post-states. It visualizes the requirement that a concrete transition be simulated by an abstract transition while the refinement relation holds before and after.

Initialization effects $K(c)$ and $L(c)$
$K(c)$ is the effect of the abstract initialization and $L(c)$ is the effect of the concrete initialization.

Old versus new event
An old event already exists in the abstract model and has a refined concrete version. A new event, such as `IL_in` or `IL_out`, is first introduced in the refinement and therefore has no old abstract counterpart in M0.

Key Examples and Demonstrations

Example 1: Why $a + b \leq d$ is not a valid strengthening

Setup. Replace the correct concrete `ML_out` capacity guard $a + b < d$ by $a + b \leq d$, while keeping $c = 0$ and the gluing invariant $a + b + c = n$.

Main point. The candidate guard corresponds to $n \leq d$. At the boundary state $n = d$, the concrete event would be enabled but the abstract guard $n < d$ would be false. The concrete model would therefore introduce a new behavior.

What to practise. Use Jackie's annotated iPad notes p. 2 to explain the set-inclusion picture and identify the forbidden region where H is true but G is false.

Example 2: Construct a post-state linking-invariant goal

Setup. For `ML_out`, start from

$$a' + b' + c' = n'.$$

The concrete action changes a to $a+1$, while b, c stay unchanged; the corresponding abstract event changes n to $n + 1$.

Main point. Substitution gives

$$(a + 1) + b + c = n + 1.$$

Because the linking invariant contains variables from both levels, both the abstract and concrete effects matter.

What to practise. Reconstruct the primed form first, then perform the substitutions shown by the model. Do not copy an intermediate spoken slip from the recording in place of the corrected final form.

Example 3: Construct a post-state safety-invariant goal

Setup. For `ML_in`, start from

$$a' = 0 \vee c' = 0.$$

The concrete event leaves a unchanged and updates $c := c - 1$.

Main point. The instantiated goal becomes

$$a = 0 \vee (c - 1) = 0.$$

Only the concrete effect is needed because this invariant does not contain the abstract variable n.

What to practise. Use Jackie's annotated iPad notes p. 6 as the sequent and inference-rule reference, then try to construct the proof independently.

Example 4: Read the commuting diagram in both directions

Setup. Begin with related pre-states v and w. The abstract event produces v'; the concrete event produces w'.

Main point. Guard strengthening justifies the abstract transition when the concrete event is enabled, and invariant preservation restores the refinement relation between v' and w'.

What to practise. On p. 4 of Jackie's annotated iPad notes, identify which parts represent guard strengthening, the linking/refinement invariants, and the two state transitions. Then explain why the diagram is called "commuting."

Example 5: Initialization establishes the gluing and safety invariants

Setup. Abstract initialization gives $n = 0$; concrete initialization gives $a = b = c = 0$.

Main point. Two representative establishment goals reduce to

$$0 + 0 + 0 = 0$$

and

$$0 = 0 \vee 0 = 0.$$

What to practise. Use pp. 7–8 of Jackie's annotated iPad notes to construct the five initialization obligations and explain why only one init event is counted.

Common Pitfalls and Misconceptions

- **Reversing guard strengthening.** The required direction is $H \Rightarrow G$, from concrete guard to abstract guard. Reversing it can allow a concrete-only enabled state.
- **Assuming strengthening must be strict.** Equality is allowed: the concrete and abstract guards may be logically equivalent. What is forbidden is a state satisfying H but not G.
- **Forgetting the boundary case when comparing $<$ and $\leq$.** The proposed guard $a + b \leq d$ admits the equality case and is therefore weaker than the required capacity condition in this example.
- **Using only concrete artifacts in a linking-invariant goal.** If the invariant contains the abstract variable n, its post-state form must account for the abstract event effect as well as the concrete effect.
- **Using an intermediate mistaken substitution from the spoken development.** Jackie corrected himself while deriving the representative preservation goals. Use the final event definitions and the final sequents in the annotated notes.
- **Counting new events in the same old-event preservation count.** The $2 \times 5 = 10$ count discussed in class is for the two old events in their concrete versions. Jackie explicitly said new events require a different treatment.

- **Giving a new event an abstract counterpart in M0.** `IL_in` and `IL_out` are introduced only in M1.
- **Putting guard conditions into the action.** For the new-event introduction, Jackie separated what changes from when the event is allowed; the actions were introduced first and the guards were deferred.

What You Should Be Able to Do

After this session, you should be able to:

- explain guard strengthening as $H \Rightarrow G$, using both the satisfying-state picture and the contrapositive interpretation;
- diagnose why a candidate concrete guard such as $a + b \leq d$ can violate guard strengthening;
- distinguish the roles of $A(c)$, $I(c, v)$, $J(c, v, w)$, $H(c, w)$, $E(c, v)$, and $F(c, w)$ in a refinement preservation obligation;
- count the ten old-event invariant-preservation obligations for this M1 model and the five invariant-establishment obligations for initialization;
- construct the post-state goal for a concrete invariant by priming it first and substituting the appropriate abstract/concrete event effects;
- explain the commuting diagram and the statement that a concrete state transition should be simulated by an abstract state transition;
- distinguish invariant preservation from invariant establishment in a refinement and identify the roles of $K(c)$ and $L(c)$;
- distinguish old events from new events in M1 and state the actions of `IL_in` and `IL_out` introduced in this lecture.

Suggested Practice

1. **Guard strengthening review.** Recreate the p. 2 set diagram from memory and explain the three regions Jackie discussed: states satisfying both guards, states enabled abstractly but not concretely, and states where neither guard is enabled. Then explain why there must be no concrete-only region.
2. **Boundary counterexample.** Starting from $a + b \leq d$, $c = 0$, and $a + b + c = n$, derive $n \leq d$ and identify the value that witnesses failure of strengthening against $n < d$.
3. **Preservation setup.** Without looking at the completed sequent, reconstruct the `ML_out` preservation obligation for `inv1_4`: list the axioms, abstract invariants, five concrete invariants, concrete guards, primed invariant goal, and final substituted goal.

4. **Second preservation exercise.** Do the same for `ML_in/inv1_5/INV`. Use the inference rules shown with Jackie's annotated iPad notes p. 6 to attempt the proof yourself. Jackie explicitly recommended drilling these proof obligations for the exam, first with the relevant rules visible and later with less prompting.

5. **Commuting diagram.** Redraw the four-state diagram. Mark guard strengthening, both state transitions, and the pre/post refinement relations. Explain the two paths that should lead to corresponding post-states.

6. **Initialization count.** Generate the five M1 initialization sequents from the establishment rule. Check the linking and safety examples against pp. 7–8 of Jackie's annotated iPad notes.

7. **New-event actions.** From the bridge/island picture, derive why `IL_in` uses $a := a-1,\ b := b+1$ and why `IL_out` uses $b := b-1,\ c := c+1$. Leave the guards for the next lecture rather than inventing them.

Where to Review

- **Jackie's annotated iPad notes p. 2:** guard strengthening, $H \Rightarrow G$, the satisfying-state-set visualization, contrapositive, and the failed $a + b \leq d$ candidate.
- **p. 3:** refinement invariant-preservation rule, abstract versus concrete components, the $2 \times 5 = 10$ obligation count, and the two representative instantiated goals.
- **p. 4:** the commuting diagram relating abstract and concrete pre/post-states and the simulation interpretation.
- **pp. 5–6:** the instantiated `ML_out/inv1_4/INV` and `ML_in/inv1_5/INV` sequents and the relevant inference-rule references. Use them as starting points for your own proof construction.
- **pp. 7–8:** invariant establishment in refinement, the roles of $K(c)$ and $L(c)$, the five-obligation count, and the representative linking/safety establishment sequents.
- **pp. 9–10:** old versus new events and the introduction of the `IL_in/IL_out` actions.
- **Written Test 2 boundary:** Jackie stated in the recording that slide 58 is the end of the November 18 test coverage and that the Event-B Summary plus Inference Rule sets 1 and 2 would be available during the test. Use the official guide/materials for the final test arrangement.

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.