EECS3342 System Specification and Refinement
Fall 2026

Digest of Lecture 16, November 10, 2026

Repairing Deadlock Freedom and Introducing the First Refinement

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 9

Main Ideas

Lecture 16 completed the repair of the initial Bridge Controller model and then introduced the first refinement. The main movement was:

- revisit the failed deadlock-freedom proof and compare the consequences of `EQ_LR` and `EQ_RL`;
- interpret the useful unprovable endpoint $\vdash d > 0$, identify the permitted case $d = 0$, and show that together with initialization $n = 0$, both ordinary-event guards are false;
- repair M_0 by adding the axiom $d > 0$, regenerate/re-attempt the DLF obligation, and discharge the previously failing branch;
- summarize the three correctness criteria established for M_0: invariant establishment, invariant preservation, and deadlock freedom;
- begin refinement from M_0 to M_1, replacing the single abstract quantity n with more concrete quantities a, b, c, then add a linking/gluing invariant and a one-way-bridge safety invariant.

Written Test 2 is scheduled for **November 18**. Jackie said the Bridge Controller case study should be reviewed from the beginning and that at least this lecture and Thursday's lecture would be included, while the eventual guide would confirm the final coverage.

Roadmap

1. Revisit the DLF proof and try the opposite equality direction

The previous DLF attempt had split into two branches. One branch closed; the other used `EQ_LR` and eventually reached the unsupported goal

$$\vdash d > 0.$$

Lecture 16 deliberately revisited the equality step. Starting from a branch of the form

$$n = d \ \vdash \ n < d \lor n > 0,$$

Jackie instead applied `EQ_RL`: replace every occurrence of the right-hand equality side d by the left-hand side n. This gives

$$n = d \ \vdash \ n < n \lor n > 0.$$

After MON removes the equality hypothesis, `OR_R2` can select the second disjunct, leaving

$$\vdash n > 0.$$

This endpoint is also unprovable in the current model, but Jackie rejected it as a useful repair condition because requiring $n > 0$ would exclude the intended empty initial state $n = 0$.

The proof-strategy lesson was not that one equality direction is always correct. Jackie emphasized that it is reasonable to try one direction, inspect the resulting goal, and backtrack to the other direction if the first result is not useful for repairing the model.

2. Use the failed goal $d > 0$ to diagnose a deadlock

Jackie returned to the original failed endpoint

$$\vdash d > 0.$$

In the current model, the existing typing axiom is $d \in \mathbb{N}$, hence $d \geq 0$. The possibility of violating $d > 0$ gives $d \leq 0$; together, these point to the permitted case

$$d = 0.$$

Initialization gives $n = 0$. Evaluating the two ordinary-event guards in this state gives the DLF condition

$$n < d \vee n > 0 \quad \Longrightarrow \quad 0 < 0 \vee 0 > 0,$$

which is false. Neither event is enabled, so the model can deadlock immediately after initialization when $d = 0$.

3. Repair M_0 by constraining the constant d

Because d is a constant, Jackie chose to add a new axiom

$$\texttt{axm0_2}: \qquad d > 0.$$

This excludes the deadlocking value $d = 0$. Jackie also noted that constants are not initialized like variables: initialization assigns variables, whereas a constraint on d belongs in the static part of the model.

After changing the model, the relevant proof obligation must be generated or attempted again. The new axiom contributes $d > 0$ as an additional hypothesis to the DLF sequent.

4. Discharge the revised DLF obligation

The repaired proof begins with the added hypothesis $d > 0$. The branch that was already provable remains provable. In the previously failing branch, the same rewrite path can again reach the goal $d > 0$, but now that predicate is an available hypothesis, so `HYP` closes the branch.

Jackie identified this revised proof as an important practice target: first try it with the relevant inference rules visible, then retry it while choosing the rules yourself from the rule sets.

5. Close the initial model M_0

The summary of M_0 separates its static and dynamic parts. The two event guards $n < d$ and $n > 0$ were introduced while repairing invariant preservation, while the new axiom $d > 0$ repairs deadlock freedom. Invariant establishment did not require a model change.

At this point, the three correctness criteria developed for the initial model are:

invariant establishment, invariant preservation, deadlock freedom.

6. Introduce the first refinement M_1

The model then moved from the initial, most abstract view M_0 to the more concrete refinement M_1. In M_0, n represents the total number of cars in the island-and-bridge compound. In M_1, the system is viewed in more detail using three variables:

$$a,\ b,\ c.$$

Jackie's diagram assigns b to cars on the island and uses a and c for cars on the bridge travelling in opposite directions. The refinement is intended to represent the one-way-bridge requirement more explicitly.

The old events `ML_out` and `ML_in` have abstract versions in M_0 and concrete versions in M_1. The refinement also introduces new events `IL_in` and `IL_out`; because they do not exist in M_0, they do not have abstract M_0 versions.

7. Build the M_1 state space

The first three M_1 invariants are typing constraints:

$$a \in \mathbb{N}, \qquad b \in \mathbb{N}, \qquad c \in \mathbb{N}.$$

The fourth invariant links the abstract and concrete states:

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

Jackie called this a **linking invariant** or **gluing invariant**. It allows a concrete M_1 state to be related back to the abstract total in M_0.

The fifth invariant expresses the one-way-bridge safety condition:

$$a = 0 \lor c = 0.$$

At least one traffic direction must contain no cars; both may be zero.

8. Compare alternative safety invariants

Jackie compared the compact invariant $a = 0 \lor c = 0$ with three alternatives:

$$a \cdot c = 0,$$

$$(a > 0 \land c = 0) \lor (c > 0 \land a = 0),$$

and

$$a + c = a \lor a + c = c.$$

The product form is mathematically equivalent but can complicate proof work. The second form prevents opposing traffic, but it is too restrictive because it forbids the acceptable state $a = 0, c = 0$. The third form is equivalent to the compact version after cancellation, but is longer than necessary.

The important modeling distinction is between **safety** and **permissiveness**: an invariant can prevent a crash and still be wrong for the model if it excludes states that should be allowed.

9. Relative meanings of abstract/concrete and old/new

Jackie ended with a terminology map. M_0 is abstract relative to M_1, while M_1 is concrete relative to M_0. Later, when M_2 refines M_1, M_1 will itself be abstract relative to M_2. Therefore "abstract" and "concrete" are relative descriptions, not permanent labels.

Similarly, an **old event** is one already introduced in the abstract model; a **new event** is introduced only in the refinement. Jackie deferred the detailed guards and actions of the M_1 events to the next lecture.

Key Terms, Notation, and Structures

DLF repair
A model correction motivated by an unprovable deadlock-freedom obligation. In this lecture, the relevant repair is the additional axiom $d > 0$.

`EQ_LR` / `EQ_RL`
Equality-rewrite directions. In the worked comparison, `EQ_RL` replaces every occurrence of the right equality side d by the left side n; the opposite direction had been used in the previous lecture.

`axm0_2`
The new M_0 axiom $d > 0$, introduced to exclude the deadlocking value $d = 0$.

Refinement
Moving to a model with greater concreteness while retaining the relationship to the earlier abstract model. Here M_1 refines M_0.

Abstract variable / concrete variables
n is the abstract M_0 total; a, b, c are the more concrete quantities introduced in M_1.

Linking or gluing invariant
An invariant relating abstract and concrete states. Here

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

Safety invariant
A condition excluding unsafe states. The one-way-bridge invariant is

$$a = 0 \lor c = 0.$$

Old event / new event
An old event already exists in the abstract model and receives a concrete refined version. A new event is first introduced in the refinement and has no abstract version in the earlier model.

Key Examples and Demonstrations

Example 1: The alternative equality-rewrite path

Setup.

$$n = d \vdash n < d \vee n > 0.$$

Main point. Using `EQ_RL` replaces d by n, producing $n < n \vee n > 0$. After removing the equality hypothesis and selecting the right disjunct, the endpoint is $\vdash n > 0$. Jackie rejected this as the repair direction because the model must allow $n = 0$.

What to practise. Compare this endpoint with the earlier `EQ_LR` endpoint $\vdash d > 0$, and explain why the latter gives the more useful modeling clue.

Example 2: Locate the deadlocking initial state

Setup. The model allows $d \in \mathbb{N}$, initialization sets $n = 0$, and the ordinary-event guards are $n < d$ and $n > 0$.

Main point. The failed goal $d > 0$ points to the permitted boundary case $d = 0$. Then

$$n < d \vee n > 0 \quad \mapsto \quad 0 < 0 \vee 0 > 0,$$

so both guards are false and the system deadlocks immediately after initialization.

What to practise. Reconstruct the reasoning on p. 4 of Jackie's annotated iPad notes from the failed sequent to the concrete $d = 0, n = 0$ state.

Example 3: Re-prove DLF after adding $d > 0$

Setup. Add $d > 0$ as a new axiom and therefore as a new hypothesis in the DLF proof.

Main point. The earlier proof structure can be reused. The branch that previously ended in $d > 0$ now closes by `HYP` because that predicate is available as a hypothesis.

What to practise. Follow Jackie's two-pass recommendation: first reproduce the proof with the inference-rule schemata visible; then retry it while choosing the rules yourself.

Example 4: Relate a concrete M_1 state to M_0

Setup. Jackie's example has two cars on the island, three on the bridge travelling toward the island, and zero in the opposite direction.

Main point. The abstract model records only the total:

$$n = a + b + c = 5.$$

The linking invariant is what connects the two abstraction levels.

What to practise. Given a concrete assignment to a, b, c, compute the corresponding abstract n and explain which information is lost when moving back to M_0.

Example 5: Judge alternative one-way-bridge invariants

Setup. Compare the chosen condition $a = 0 \vee c = 0$ with the three alternatives from Jackie's annotated iPad notes pp. 9–10.

Main point. Mathematical equivalence, proof convenience, safety, and permissiveness are different questions. In particular, the second alternative prevents a crash but wrongly excludes the both-zero state.

What to practise. Test each candidate against the boundary case $a = 0, c = 0$ and explain whether it is correct, equivalent, unnecessarily complicated, or too restrictive.

Common Pitfalls and Misconceptions

- **Assuming one equality-rewrite direction must always be the right choice.** `EQ_LR` and `EQ_RL` can lead to different goals. Jackie explicitly encouraged backtracking when a chosen direction produces an unhelpful endpoint.
- **Using $n > 0$ as the repair condition.** That would exclude the intended initial state $n = 0$, so it is too restrictive for this model.
- **Treating d like an initialized variable.** d is a constant. Jackie repaired its allowed values through an axiom rather than an initialization assignment.
- **Forgetting to re-attempt proof obligations after changing the model.** A model fix changes the proof context; the relevant obligation must be generated or proved again.
- **Replacing only selected occurrences during the equality rewrite.** Jackie answered a student question by emphasizing that the rule replaces every applicable occurrence specified by the rewrite direction.
- **Thinking M_0 and M_1 describe different systems.** They are different abstraction levels of the same Bridge Controller.
- **Treating "abstract" and "concrete" as permanent labels.** They are relative: M_1 is concrete relative to M_0, but will be abstract relative to M_2.
- **Giving a new event an abstract M_0 version.** An event introduced only in M_1 did not exist in M_0, so there is no such abstract counterpart.
- **Equating safety with a good invariant.** A candidate can prevent collisions but still be too restrictive. The both-zero traffic state is an important boundary case.
- **Assuming equivalent invariants are equally convenient to prove.** Jackie used $a \cdot c = 0$ as an example of a correct form that can introduce unnecessary proof difficulty.

What You Should Be Able to Do

After this session, you should be able to:

- trace how `EQ_LR` and `EQ_RL` produce different endpoints in the DLF proof and justify when backtracking is appropriate;
- reconstruct why the original M_0 permits the deadlocking case $d = 0, n = 0$ and evaluate the two guards in that state;
- explain why adding the axiom $d > 0$ is the selected repair and how it changes the hypotheses of the DLF proof obligation;
- reproduce the essential structure of the second DLF proof attempt and explain why the formerly unprovable branch now closes;
- state the three correctness criteria established for M_0: invariant establishment, invariant preservation, and deadlock freedom;
- explain what it means for M_1 to refine M_0, and distinguish abstract/concrete variables and old/new events;
- use the gluing invariant $n = a + b + c$ to relate an M_1 concrete state to the M_0 abstract state;
- explain why $a = 0 \vee c = 0$ captures the one-way-bridge safety requirement while allowing no traffic in either direction;
- evaluate the three alternative safety-invariant formulations using equivalence, proof convenience, safety, and permissiveness.

Suggested Practice

1. Revisit the clean DLF proof starting point and reproduce the `EQ_RL` alternative from Jackie's annotated iPad notes p. 3. Compare its final $n > 0$ goal with the earlier $d > 0$ endpoint.
2. Reconstruct the p. 4 deadlock diagnosis without copying the annotations: failed $d > 0$ goal → $d = 0$ permitted by the current model → $n = 0$ after initialization → both guards false.
3. Do Jackie's most important proof exercise from pp. 5–6 twice: first with the relevant inference rules visible, then with only the rule sets available for you to choose from.
4. Review the Bridge Controller case study from the beginning in preparation for Written Test 2. Jackie said the final guide will confirm the exact coverage, so use the guide rather than this digest as the final scope authority.

5. Using pp. 9–10, test the chosen one-way invariant and each alternative against $a = 0, c = 0$. Be able to explain why "prevents a crash" is not the same as "appropriate invariant."
6. From p. 11, redraw the old/new and abstract/concrete event table from memory and explain why `IL_in` and `IL_out` have no abstract M_0 versions.

Where to Review

- **Jackie's annotated iPad notes, p. 3:** revisit the first DLF attempt using `EQ_RL` instead of `EQ_LR`, and compare the resulting unprovable endpoint.
- **p. 4:** study the full chain from $\vdash d > 0$ to the $d = 0, n = 0$ deadlock example and the decision to add $d > 0$.
- **pp. 5–6:** reconstruct the second DLF proof attempt after the new axiom is available as a hypothesis.
- **p. 7:** review the final M_0 static/dynamic model and the three correctness criteria.
- **p. 8:** review the visual transition from the abstract compound in M_0 to the more concrete island/bridge representation in M_1.
- **pp. 9–10:** study the M_1 state space, typing invariants, gluing invariant, safety invariant, and the three alternative formulations.
- **p. 11:** review the old/new and abstract/concrete event map and the relative $M_0 \to M_1 \to M_2$ abstraction chain.
- Alongside each logical unit above, study the corresponding presentation-slide material and use the recording where Jackie's spoken explanation or proof transition needs to be reconstructed.

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.