EECS4302 Compilers and Interpreters
Fall 2026

Digest of Lecture 11, October 22

DFA Minimization and Partition 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 4

Key Examples and Demonstrations 5

Common Pitfalls and Misconceptions 5

What You Should Be Able to Do 6

Suggested Practice 7

Where to Review 7

Closing Reminder 8

Main Ideas

- Lexical analysis can be viewed as a pipeline from a textual regular expression to an ε-NFA, then to a DFA, and finally to a minimized DFA. These representations recognize the same language; the transformations change representation, not meaning.
- Using lazy evaluation, extended subset construction removes non-determinism and ε-transitions while generating only reachable subset states. Reachability alone does not guarantee that the resulting DFA has no redundant states.
- DFA minimization collapses behaviorally equivalent states while preserving the recognized language. In Jackie's formulation, the output machine M' must satisfy $L(M') = L(M)$ and must not have more states than the input.
- The minimization algorithm works by refining a partition of the DFA states. It starts from accepting versus non-accepting states and repeatedly splits a partition when a maximal proper subset has distinguishable transition behavior to another partition.
- The stopping condition is a fixed point: when the newly refined partition set no longer differs from the previous one, $P = T$, no further split is possible.
- The worked example reduces a seven-state DFA, including its implicit error state S_e, to five states while preserving the language $\{\texttt{fee}, \texttt{fie}\}$. Jackie assigned two additional minimization examples for independent practice.

Roadmap

1. Assessment and review reminders

Jackie reminded the class that the **Programming Test is on November 5** and the **Written Test is on November 19**. He said he would provide programming-test guidance and a practice test by the next Thursday class at the latest, and he may also arrange an EECS-lab practice session. Assignment 2 and the ANTLR4 tutorial remain important preparation. Before the Written Test, students should also complete the self-guided lecture on formulating regular-language tools.

2. From regular expressions to a minimized DFA

Page 3 of Jackie's annotated iPad notes places the whole lexical-analysis transformation in one picture:

$$\text{RE} \longrightarrow \varepsilon\text{-NFA} \longrightarrow \text{DFA} \longrightarrow \text{minimized DFA}.$$

The regular expression is the textual form specified by the user; the later representations can be generated automatically by tools such as ANTLR4. Jackie emphasized the semantic invariant across the pipeline:

$$L(1) = L(2) = L(3) = L(4).$$

The ε-NFA-to-DFA step uses extended subset construction with lazy evaluation: keep only reachable subset states, remove non-determinism, and remove ε-transitions. The final minimization step then removes redundant DFA states that may still remain.

3. Making the DFA complete: the implicit error state

Page 4 uses a simplified DFA diagram for the language $\{\texttt{fee}, \texttt{fie}\}$. Only the interesting edges are drawn, but a true DFA must have a defined transition for every state/input pair. Jackie therefore introduced an implicit sink/error state S_e. For example, from S_0, input `f` follows the shown edge to S_1, while every other input character goes to S_e. Similar set-difference labels complete the other states. Jackie explicitly said that the error state must be included when performing minimization.

4. Partitions of the state set

Page 5 introduces the central mathematical object: a partition of the state set Q. Each partition block is a **non-empty set of states**, and the blocks are disjoint. For the example $Q = \{s_0, s_1, s_2, s_3\}$, the fewest blocks in a partition is one block containing all four states; the greatest possible number of blocks is four singleton blocks. Jackie contrasted this with the power-set idea: partitions are not arbitrary subsets, and an empty block is not allowed.

5. The minimization algorithm and its fixed point

Page 6 gives the algorithmic specification. The input is a DFA M and the output is an equivalent minimized DFA M'. The initial refined partition is

$$T = \{F,\ Q - F\},$$

separating accepting from non-accepting states. The variable P records the previous iteration; T is rebuilt during the current iteration. The loop continues while $P \neq T$ and stops when $P = T$.

For each partition $p \in P$, the algorithm searches for a **maximal proper subset** $S \subset p$ that is splittable. Jackie's condition requires some input $c \in \Sigma$ such that every state in S transitions under c to states in the same different partition p_1, with $p_1 \neq p$. If such an S exists, replace p by S and $p - S$; otherwise keep p unchanged.

6. Worked trace: seven states to five

Pages 7–8 carry out the complete example. The initial partition has two blocks: accepting states $\{S_3, S_5\}$ and all non-accepting states $\{S_0, S_1, S_2, S_4, S_e\}$.

The refinement sequence is:

1. On input `e`, the maximal proper subset $\{S_2, S_4\}$ leaves the non-accepting block and reaches the accepting block, so it is split out.

2. In the remaining block $\{S_0, S_1, S_e\}$, input `i` distinguishes S_1 because $S_1 \xrightarrow{\texttt{i}} S_4$ reaches the new $\{S_2, S_4\}$ block; split out $\{S_1\}$.

3. In $\{S_0, S_e\}$, input `f` distinguishes S_0 because $S_0 \xrightarrow{\texttt{f}} S_1$ reaches the $\{S_1\}$ block; split out $\{S_0\}$.

The accepting block $\{S_3, S_5\}$ is *not* split merely because all its states go to S_e: the whole block has the same behavior, so there is no proper subset to separate. The fixed point therefore contains five blocks:

$$\{S_0\}, \quad \{S_1\}, \quad \{S_2, S_4\}, \quad \{S_3, S_5\}, \quad \{S_e\}.$$

Each block becomes one state in the minimized DFA. The result reduces the machine from seven states to five, which Jackie described as about a 30% reduction.

7. What comes next

Jackie ended by assigning the worked example for retracing plus two additional DFA-minimization exercises. He said he would revisit the algorithm briefly in the next lecture before moving to the final lexical-analysis topic: how to implement a scanner, followed by syntactic analysis.

Key Terms, Notation, and Structures

M, M'	Input DFA and minimized output DFA.
$L(M)$	The language recognized by machine M. Minimization must preserve it: $L(M') = L(M)$.
S_e	Error/sink state used to make the DFA transition function total when a simplified diagram omits uninteresting outgoing edges.
Partition	A collection of disjoint, non-empty blocks whose states together cover the DFA state set under discussion.
F **and** $Q - F$	Accepting states and non-accepting states; these form the algorithm's initial two-way partition in the example developed in class.
P	The partition set from the previous iteration of the refinement loop.
T	The partition set being rebuilt for the current iteration.
Fixed point	The condition $P = T$, meaning another refinement iteration produces no change.
Proper subset	A subset $S \subset p$ that is not the entire partition block p; Jackie's algorithm requires a proper subset before a split is meaningful.
Splittable	A proper subset S of a partition p whose states, for some input $c \in \Sigma$, all transition to states in the same different partition p_1 with $p_1 \neq p$. The minimization algorithm then selects a maximal splittable proper subset when refining p.

Lazy evaluation In extended subset construction, generate only subset states reachable from the initial state; this avoids unreachable DFA states but does not itself guarantee minimization.

Key Examples and Demonstrations

Pipeline from RE to minimized DFA

Setup: Page 3 shows the four representations in sequence.
Main point: The representation changes, but the recognized language does not. Lazy subset construction handles ε-NFA to DFA; minimization is a separate later step.
What to practise: Be able to explain what each transformation removes or changes, and what must stay invariant.

Completing the simplified DFA with S_e

Setup: Page 4 shows the DFA for $\{\texttt{fee}, \texttt{fie}\}$ with only the important edges drawn.
Main point: Every omitted character transition is conceptually routed to the error state so that the DFA remains total.
What to practise: Reconstruct the missing error transitions before attempting minimization.

Partition boundary cases

Setup: For $Q = \{s_0, s_1, s_2, s_3\}$, page 5 compares one block, four singleton blocks, and an intermediate partition.
Main point: Do not confuse "number of blocks in a partition" with the number of subsets in a power set. Partition blocks are non-empty and disjoint.
What to practise: Given a small Q, write valid partitions and explain why an empty block or overlapping blocks would be invalid.

Seven-state minimization trace

Setup: Pages 7–8 start from two blocks and refine them using inputs `e`, `i`, and `f`.
Main point: A split is driven by behavior that crosses from the current partition to a different partition, and the subset being split must be a proper subset.
What to practise: Reproduce the entire color/block sequence from scratch, then reconstruct the five-state minimized DFA and its error transitions.

Common Pitfalls and Misconceptions

- **Confusing partitions with the power set.** For four states, 2^4 counts subsets, not the maximum number of partition blocks. The maximum number of blocks is four singletons.

- **Allowing an empty partition block.** Jackie explicitly required each partition block to be non-empty; do not confuse an empty set of states with the separate notion of an empty language.
- **Ignoring the implicit error state.** The simplified drawing omits many edges, but minimization must reason about a complete DFA, including S_e.
- **Splitting when transitions stay inside the same partition.** If the relevant transition behavior remains entirely within the current block, it does not justify separating a new block.
- **Splitting an entire block.** The block $\{S_3, S_5\}$ goes to the error-state block under any input, but that is the behavior of the whole block. Since there is no proper subset to separate, no split occurs.
- **Splitting one state when several have the same distinguishing behavior.** Jackie emphasized finding the maximal proper subset; in the first iteration, S_2 and S_4 must be grouped together because both take `e` into the accepting partition.
- **Producing more states after "minimization."** Jackie explicitly said a correct minimization cannot increase the number of states. If $|Q_{M'}| > |Q_M|$, something is wrong with the algorithm or its implementation.

What You Should Be Able to Do

After this session, you should be able to:

- explain the RE → ε-NFA → DFA → minimized-DFA pipeline and state what is preserved across all four representations;
- explain what lazy extended subset construction accomplishes and why a reachable DFA can still contain redundant states;
- complete a simplified DFA conceptually by adding the implicit error state and missing transitions;
- distinguish a state partition from a power set and identify the one-block and all-singleton boundary cases;
- state the minimization goals $L(M') = L(M)$ and $|Q_{M'}| \leq |Q_M|$;
- trace the roles of P and T, explain the initial partition $\{F, Q - F\}$, and identify the fixed-point condition $P = T$;
- decide whether a candidate subset is splittable using Jackie's proper-subset and cross-partition transition conditions;
- reproduce the seven-state-to-five-state refinement trace and justify each split and non-split;

- reconstruct the minimized DFA from the final partition blocks, including start, accepting, and error-state behavior.

Suggested Practice

1. Retrace the full seven-state minimization example on pages 7–8 of Jackie's annotated iPad notes without looking at the completed colors or partitions. Check each split by naming the source block, input symbol, destination block, and maximal proper subset.
2. Complete the two additional DFA minimization exercises on pages 10–11 of Jackie's annotated iPad notes, then compare your work with the solutions Jackie said would be posted on the lecture site.
3. Revisit the pipeline on page 3 and explain, in your own words, the role of extended subset construction versus the role of DFA minimization. Do not merge these into one step.
4. For the DFA on page 4, write down the implicit error-state behavior before minimizing. Make sure every state/input combination has a destination.
5. Continue progressing through the ANTLR4 tutorial and Assignment 2 as preparation for the Programming Test, and complete the self-guided regular-language lecture before the Written Test.

Where to Review

- **Jackie's annotated iPad notes, page 3:** the RE → ε-NFA → DFA → minimized-DFA pipeline, language equivalence, and lazy extended subset construction.
- **Page 4:** the simplified DFA drawing, language $\{\texttt{fee}, \texttt{fie}\}$, and the implicit error state S_e.
- **Page 5:** partitions of states, one-block versus singleton extremes, and the distinction from arbitrary subsets.
- **Page 6:** the minimization algorithm, initialization with F and $Q - F$, the P/T fixed-point loop, and Jackie's splittable-subset condition.
- **Pages 7–8:** the complete seven-state-to-five-state worked trace and reconstruction of the minimized DFA.
- **Pages 10–11:** the two additional minimization exercises Jackie assigned for independent practice.
- **Session recording:** return to Jackie's spoken justification for why $\{S_3, S_5\}$ does not split and why $\{S_2, S_4\}$ must be taken together as the maximal subset in the first refinement.
- **Self-guided lecture: Formulating Regular Language Tools:** review the formal-language and automata foundations that Jackie said will be part of Written Test preparation.

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.