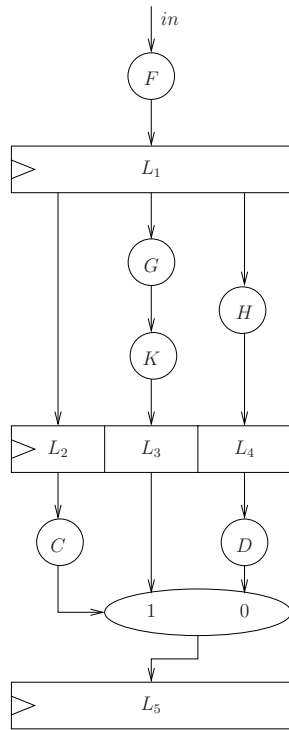


Problem Solving Through SAT/SMT Encodings

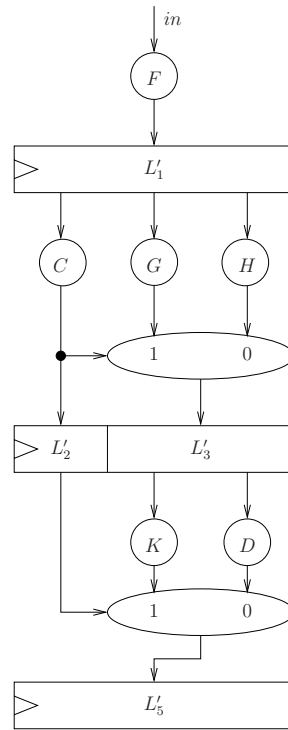
September 29, 2026

1 Warm Up: Proving Equivalence of Two Circuits

Fig 3.4 of the reference book “Decision Procedures” depicts two pipeline circuits. The stages, labelled with capital letters, can be arbitrary Boolean functions. We just assume they have the same delays, and the only advantage of the one on the right (i.e. b) over the one on the left (i.e. a) is that it is a shorter circuit with less overall delay. The goal is to prove that the two circuits compute the same function, and hence we are permitted to use the optimized one on the right in place of the unoptimized one on the left.



(a) Original circuit



(b) After transformation

2 Domino Tiling

A board is a finite set $B \subseteq \mathbb{Z}^2$ of unit squares. In the most general form of the problem, the board does not have to be connected. We want to determine if we can tile the board properly with domino-shaped tiles (i.e. 1×2 rectangles); i.e. every square must be covered, no tiles overlap, and no tiles cover any square outside the board.

Straightforward Solutions

- What is the straightforward algorithmic solution to the problem?
- What is the straightforward SAT encoding of the problem? What is wrong with it?
- How do you think the two compare?

Clever Solutions

- Can you think of a clever algorithmic solution? Hint: Domino tiling is perfect bipartite matching!
- What is the better SAT encoding of the problem? Is this optimal? Why and in what sense?
- How do you think the two compare?

Variations

Think about the variations below and evaluate how easy/hard it is to modify the algorithmic solutions to accommodate them vs the SAT encodings:

- Require or forbid a specific domino placement.
- Forbid two distant placements together.
- Require exactly n vertical dominoes.
- Find an alternative tiling.

Jobshop Scheduling

A workshop has one milling machine and one painting booth. Two customer orders must each be milled and then painted. An operation occupies its machine continuously until it finishes. The workshop can mill one order while painting another, but cannot mill two orders at once or paint two orders at once.

Order	First operation	Second operation
A	Mill for 3 time units (A_1)	Paint for 2 (A_2)
B	Mill for 2 time units (B_1)	Paint for 3 (B_2)

Question. Both orders are available at time 0. Can the workshop finish everything by time 7? What is the earliest possible finishing time?

General Problem

A job is a sequence of operations. General Job-shop scheduling permits different jobs to visit machines in their specified orders. The assumptions are:

- Processing times are fixed positive integers.
- Each operation has a fixed assigned machine.
- Machines have capacity one, and their operations cannot be interrupted.
- Waiting between operations is allowed.
- One operation may start exactly when another finishes.

The goal is to minimize the total length of performing all jobs.

Straightforward Encoding

The problem already reads like a constraint-solving problem.

- If we set the problem (like our small example) as “Can the jobs be finished in time T ?”, then propose an SMT encoding to solve the general problem.
- How do you solve the minimization problem?

Straightforward Encoding

The problem already reads like a constraint-solving problem.

- If we set the problem (like our small example) as “Can the jobs be finished in time T ?”, then propose an SMT encoding to solve the general problem.
- How do you solve the minimization problem?
- Can we still rely only on SAT for an encoding?

Variations

Can you modify the encoding to accommodate the following variations:

1. Constrain the start time of a specific job.
2. Make a certain machine unavailable (due to maintenance for example) during a particular time interval.
3. Add a dependency between two jobs: B may start milling only after A is painted.
4. Add a minimum/maximum delay time between two jobs.