



# Concurrent Models of Computation for Embedded Software

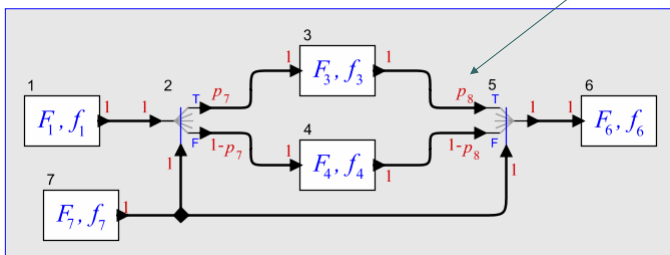
Edward A. Lee

Professor, UC Berkeley  
 EECS 290n – Advanced Topics in Systems Theory  
 Fall, 2004

Copyright © 2004, Edward A. Lee, All rights reserved

Lecture 19: Scheduling Boolean Dataflow

## Recall If-Then-Else Pattern



Symbolic consumption rate.

Solution to the symbolic balance equations:

$$q(\vec{p}) = \begin{bmatrix} 1 \\ 1 \\ p_7 \\ 1 - p_7 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

The if-then-else model is strongly consistent and we can give a *quasi-static* schedule for it:  
 (1, 7, 2, 3, 4, 5, 6)

guard

$$\vec{p} = \begin{bmatrix} p_7 \\ p_8 \end{bmatrix}$$

## Quasi-Static Schedules & Traces

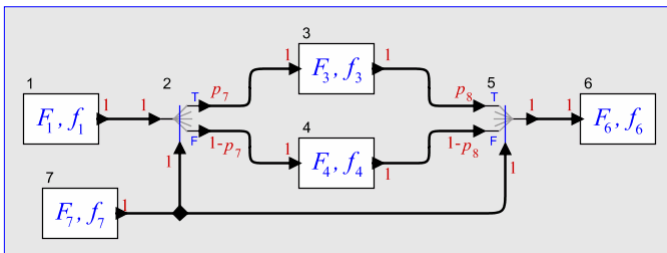
A *quasi-static schedule* is a finite list of guarded firings where:

- The number of tokens on each arc after executing the schedule is the same as before, regardless of the outcome of the Booleans.
- If any arc has a Boolean token prior to the execution of the schedule, then it will have a Boolean token with the same value after execution of the schedule.
- Firing rules are satisfied at every point in the schedule.

A *trace* is a particular execution sequence.

Lee 19: 3

## Quasi-Static Schedules & Traces



Solution to the symbolic balance equations:

$$q(\vec{p}) = [1 \quad 1 \quad p_7 \quad 1-p_7 \quad 1 \quad 1 \quad 1]^T$$

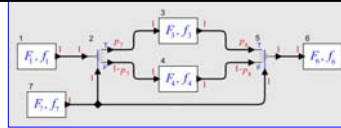
*Quasi-static* schedule: (1, 7, 2, 3, 4, 5, 6)

Possible trace: (1, 7, 2, 3, 5, 6)

Another possible trace: (1, 7, 2, 4, 5, 6)

Lee 19: 4

## Proportion Vectors



- Let  $S$  be a trace. E.g. (1, 7, 2, 3, 5, 6)

- Let  $q_S$  be a repetitions vector for  $S$ . E.g.

$$q_S = [1 \ 1 \ 1 \ 0 \ 1 \ 1]^T$$

- Let  $t_{i,S}$  be the number of TRUEs consumed from Boolean stream  $b_i$  in  $S$ . E.g.  $t_{7,S} = 1, t_{8,S} = 1$ .

- Let  $n_{i,S}$  be the number of tokens consumed from Boolean stream  $b_i$  in  $S$ . E.g.  $n_{7,S} = 1, n_{8,S} = 1$ .

- Let

$$\vec{p}_S = \begin{bmatrix} t_{7,S} / n_{7,S} \\ t_{8,S} / n_{8,S} \end{bmatrix} \longleftarrow \text{proportion vector}$$

- We want a quasi-static schedule s.t. for every trace  $S$  we have  $\Gamma(\vec{p}_S)q_S = \vec{0}$ .

Lee 19: 5

## Proportion Interpretation

Recall the balance equations depend on  $\vec{p}$ , a vector with one symbolic variable for each Boolean stream that affects consumption production rates:

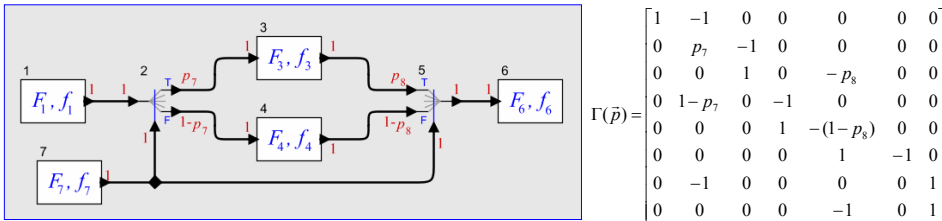
$$\Gamma(\vec{p})q(\vec{p}) = \vec{0}$$

Under a proportion interpretation, for a trace  $S$ ,  $\vec{p}_S$  represents the *proportion* of TRUEs in  $S$ . We seek a schedule that always yields traces that satisfy

$$\Gamma(\vec{p}_S)q_S = \vec{0}$$

Lee 19: 6

## Proportion Interpretation for If-Then-Else



Quasi-static schedule: (1, 7, 2, 3, 5, 6)

Possible trace:  $S = (1, 7, 2, 3, 5, 6)$

$$\vec{p} = [1 \quad 1]^T \quad q_S = [1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1]^T$$

Another possible trace: (1, 7, 2, 4, 5, 6)

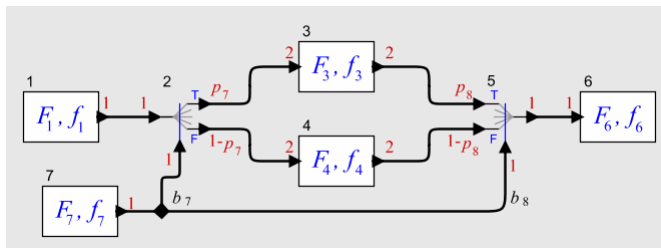
$$\vec{p} = [0 \quad 0]^T \quad q_S = [1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1]^T$$

Both satisfy the balance equations.

Lee 19: 7

## Limitations of Consistency

Consistency is necessary but not sufficient for a dataflow graph to have a bounded-memory schedule. Consider:

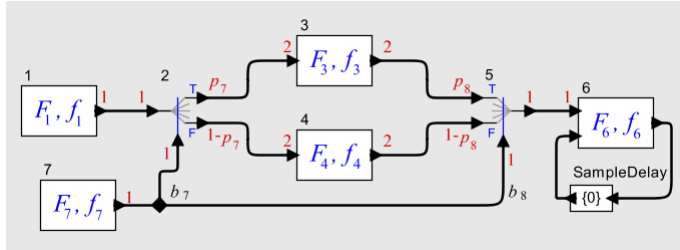


[Gao et al. '92]. This model is strongly consistent. But there is no bounded schedule (e.g., suppose  $b_7 = (F, T, T, T, \dots)$ ).

Lee 19: 8

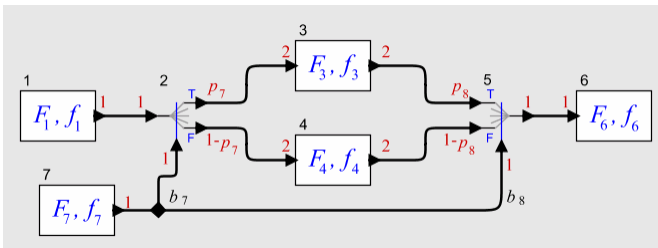
## Limitations of Consistency

Even out-of-order execution (as supported by tagged-token scheduling [Arvind et al.]) doesn't solve the problem:



Lee 19: 9

## Gao's Example has no Quasi-Static Schedule



Solution to the symbolic balance equations is

$$q(\vec{p}) = [2 \quad 2 \quad p_7 \quad 1-p_7 \quad 2 \quad 2 \quad 2]^T$$

A trace  $S$  with  $N$  firings ( $N$  even) of actor 1 must have

$$q_S = [N \quad N \quad t_{7,S}/2 \quad (N-t_{7,S})/2 \quad N \quad N \quad N]^T$$

But this cannot be unless  $t_{7,S}$  is even. There is no assurance of this.

Lee 19: 10

## Another Example

The model is strongly consistent.

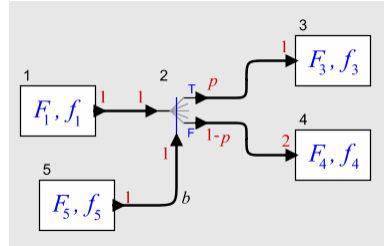
Solution to symbolic equations:

$$q(\vec{p}) = [2 \quad 2 \quad 2p \quad 1-p \quad 2]^T$$

A trace  $S$  with  $N$  firings ( $N$  even) of actor 1 must have:

$$q_S = [N \quad N \quad t \quad (N-t)/2 \quad N]^T$$

where  $t$  is the number of TRUEs consumed. There is no finite  $N$  where this is assured of being an integer vector.



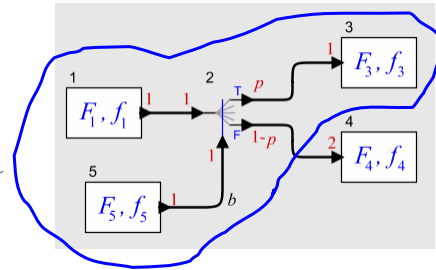
Lee 19: 11

## Clustered Quasi-Static Schedules

Consider the *clustered schedule*:

```

n = 0;
do {
  fire 1;
  fire 5;
  fire 2;
  if (b) {
    fire 3;
  } else {
    n += 1;
  }
} while (n < 2);
fire 4;
    
```

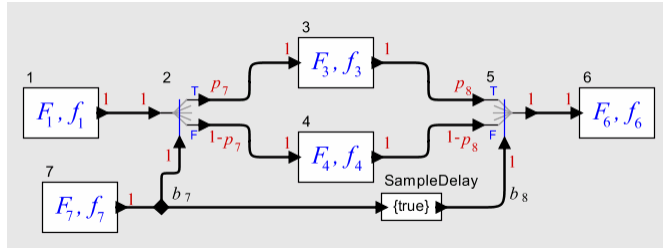


This schedule either fails to terminate or yields an integer vector of the form:

$$q_S = [N \quad N \quad t \quad (N-t)/2 \quad N]^T$$

Lee 19: 12

## Delays Can Also Cause Trouble



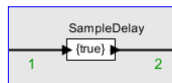
This model is weakly consistent, where the balance equations have a non-trivial solution only if  $p_7 = p_8$ , in which case the solution is:

$$q(\vec{p}) = [1 \quad 1 \quad p_7 \quad 1 - p_7 \quad 1 \quad 1 \quad 1]^T$$

Lee 19: 13

## Relating Symbolic Variables Across Delays

For the sample delay:



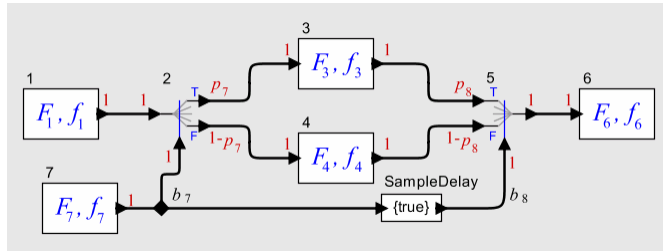
What is the relationship between  $p_1$  and  $p_2$ ?

Since consistency is about behavior in the limit, under the probabilistic of the interpretation for the symbolic variables, it is reasonable to assume  $p_1 = p_2$ .

Is this reasonable under the proportion interpretation?

Lee 19: 14

## Delays Cause Trouble with the Proportion Interpretation



Solution to the symbolic balance equations is

$$q(\vec{p}) = [1 \quad 1 \quad p_7 \quad 1-p_7 \quad 1 \quad 1 \quad 1]^T$$

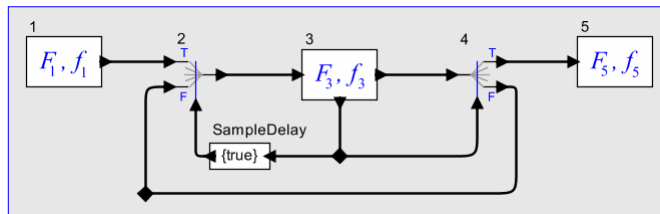
A trace  $S$  with  $N$  firings of actor 1 must have

$$q_S = [N \quad N \quad t_{7,S} \quad (N-t_{7,S}) \quad N \quad N \quad N]^T$$

But for no value of  $N$  is there any assurance of being able to fire actor 5  $N$  times. This schedule won't work.

Lee 19: 15

## Do-While Relies on a Delay



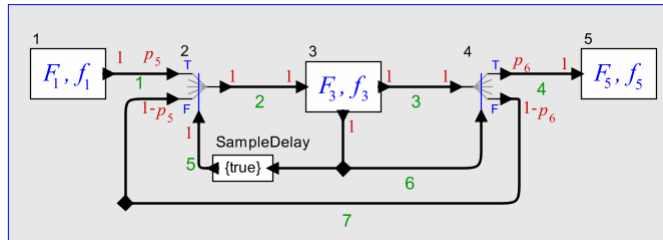
Imperative equivalent:

```
while (true) {
  x = f1();
  b = false;
  while(!b) {
    (x, b) = f3(x);
  }
  f5(x);
}
```

Is this model strongly consistent? Weakly consistent? Inconsistent?

Lee 19: 16

## Checking Consistency of Do-While

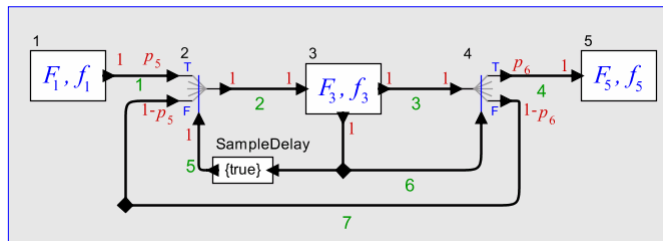


$$\Gamma(\vec{p}) = \begin{bmatrix} 1 & -p_5 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & p_6 & -1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & -(1-p_5) & 0 & 1-p_6 & 0 \end{bmatrix}$$

This model is consistent if and only if  $p_5 = p_6$ , which is true under the probabilistic interpretation, but not under the proportion interpretation.

Lee 19: 17

## Checking Consistency of Do-While



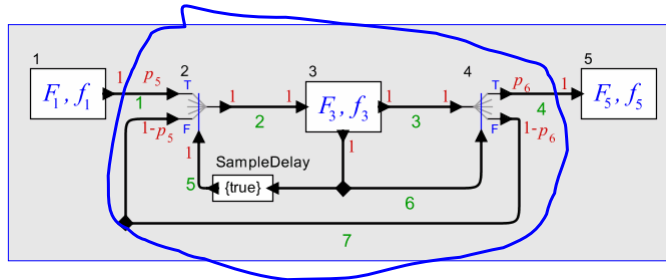
$$\Gamma(\vec{p}) = \begin{bmatrix} 1 & -p_5 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & p_6 & -1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & -(1-p_5) & 0 & 1-p_6 & 0 \end{bmatrix}$$

Let  $p = p_5 = p_6$ , then the solution to the balance equations is:

$$q(\vec{p}) = [1 \quad 1/p \quad 1/p \quad 1/p \quad 1]^T$$

Lee 19: 18

## Clustering Solution for Do-While



Clustered Schedule:

```

fire 1;
do {
  fire 2;
  fire 3;
  fire 4;
} while(!b);
fire 5;
    
```

This schedule yields traces  $S$  for which  $p_5 = p_6 = 1/N$  and

$$q_S = [1 \ N \ N \ N \ 1]^T$$

compare:

$$q(\vec{p}) = [1 \ 1/p \ 1/p \ 1/p \ 1]^T$$

Lee 19: 19

## Extensions

- State enumeration scheduling approach: Seek a finite set of finite guarded schedules that leave the model in a finite set of states (buffer states), and for which there is a schedule starting from each state.
- Integer dataflow (IDF [Buck '94]): Allow symbolic variables to have integer values, not just Boolean values. Extension is straightforward in concept, but reasoning about consistency becomes harder.

Lee 19: 20

## Conclusions and Open Issues

- BDF and IDF generalize the idea of balance equations and introduce *quasi-static scheduling*.
- BDF and IDF are Turing complete, so existence of quasi-static schedules is undecidable.
- Can often construct quasi-static schedules anyway.
- Tricks like clustered schedules make the set of manageable models larger.
- Are Switch and Select like unrestricted GOTO?
- **Fully usable languages have yet to be created.**