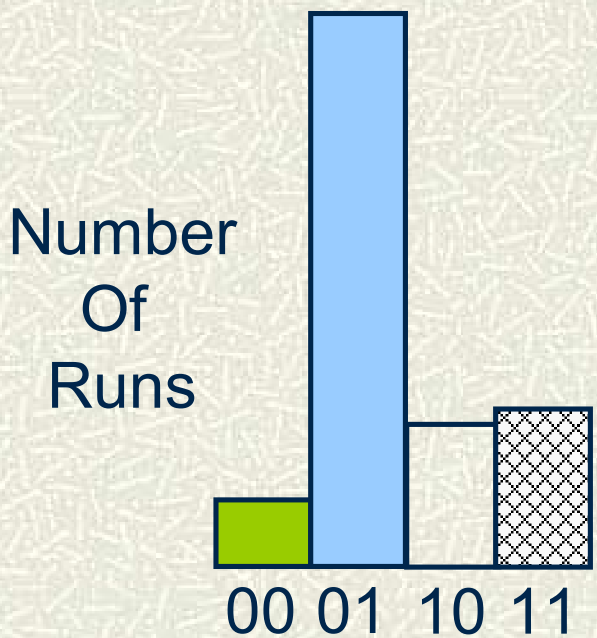
Weighted Control Scheduling

Aravind Vijayakumar and Forrest Brewer
UC Santa Barbara

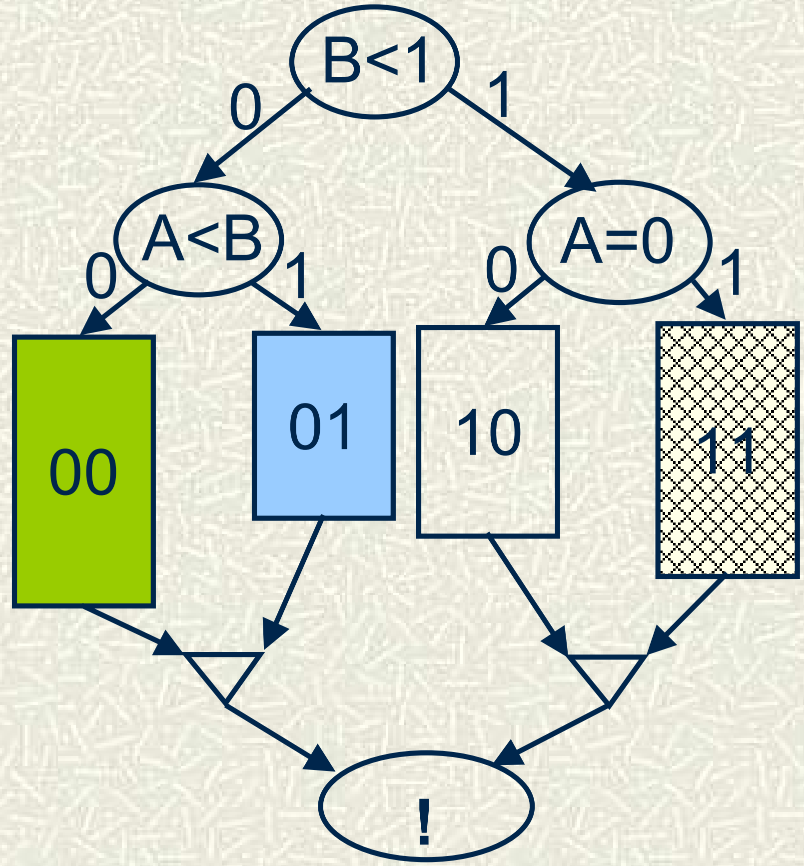
Overview

- Optimal scheduling of forward branching CDFGs under weighted average latency
- BDD-based automata exploration
- Accommodates correlated branches
- Implicitly allows speculation, dynamic branch re-ordering, complex binding constraints

Motivation



Correlated branch profile

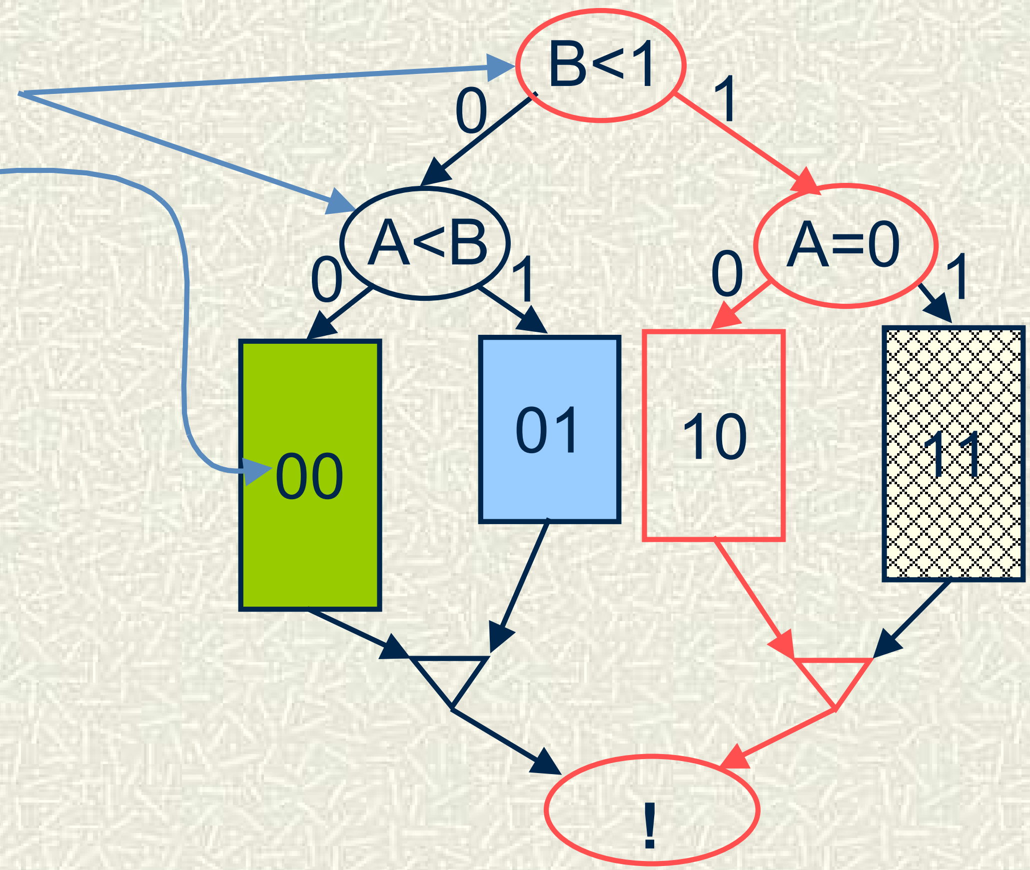


Requirements

- Fixed hardware resources
  - Arbitration
- Optimization across all paths
  - Control alternatives are *not* equal
- Accommodate 1000s to millions of paths
  - K independent branches $\Rightarrow 2^K$ control paths
  - Need to find complete set of compatible (causal) control paths

Definition of Terms

- Control operation
- Control cube
- Control case (path)
  - Path in Red
  - Trace
  - Cost
- Ensemble schedule
  - Cost



Problem Formulation

- Weighted average latency metric
- Ensures **every** trace is part of **some** ensemble of **minimal cost**
- Entails new labeling and pruning mechanism
- Maintenance of potential trace latency
- Schedule termination criterion
- Added complexity allows solution to be performance driven

Weight Model, Restrictions

- Weights – positive, identified with disjoint control path sets
- Weight = Execution probability, in practice
- Forward branching structure
- Control operations binary

Bit Resolution

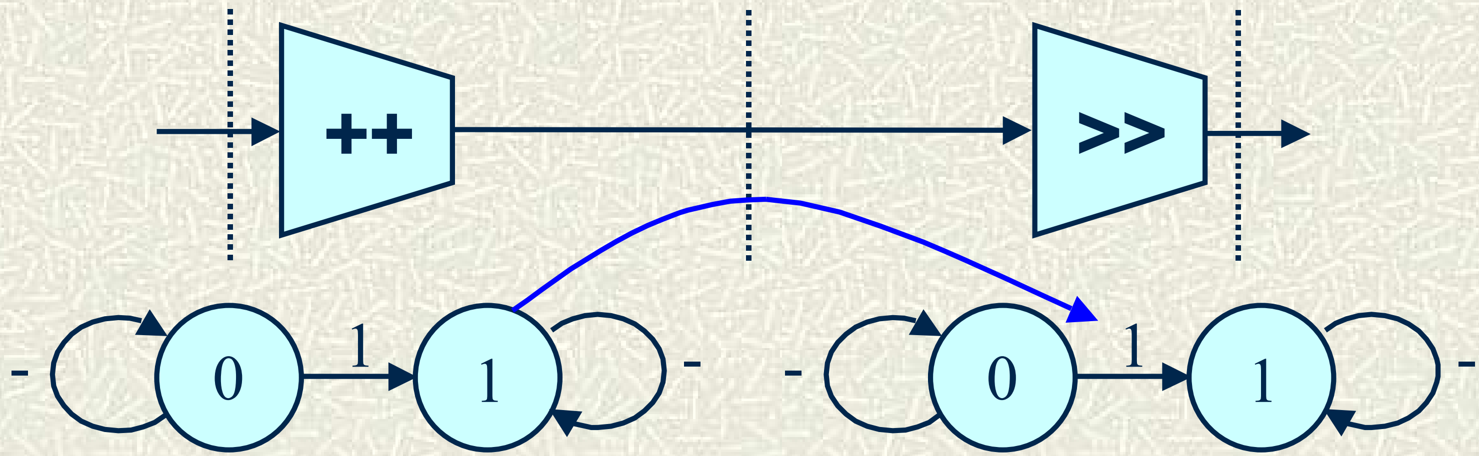
- Expected source of weights: simulation
- Finite precision of measurement
  - Measurement error
- Finite bit-width for cost sufficient
- True even with round-off
  - Absolute vs. Relative

Symbolic Scheduling - Overview

- Non-deterministic Finite Automata (NFA) based modeling of process
- Operation behavior modeled as small NFA
- BDD representation of composite transitions
- CDFG modeled by guarding operation transitions
- Solution consists of a (set of) trace(s) through the model
- Radivojevic ’94, ’96, Haynal ’98, ’00

Operator Models

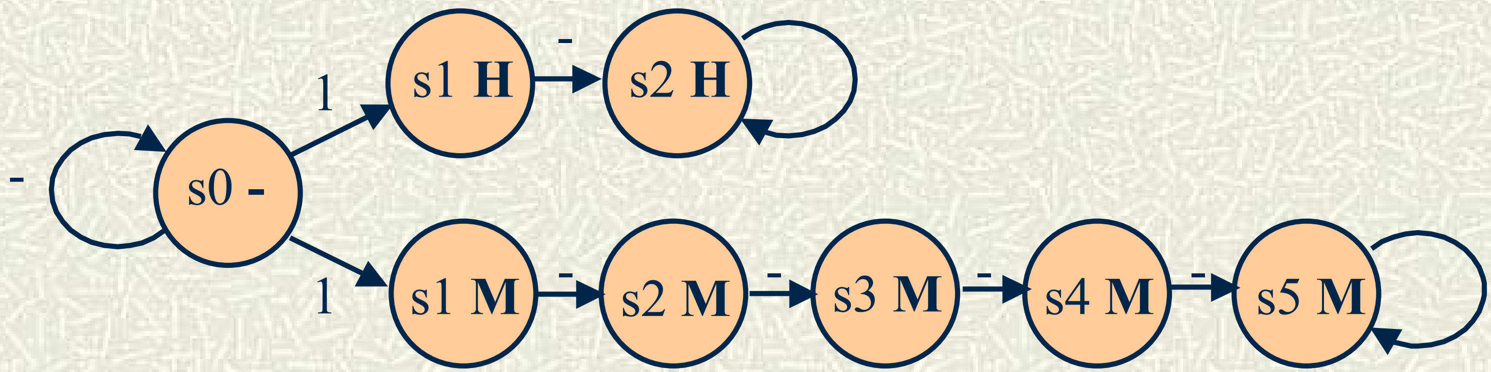
- NFA models encode operation I/O behavior
- Example:
  Simple 1 input, 1 cycle operation model



- State of operator enables transition of dependent operand.

Example Cache Model

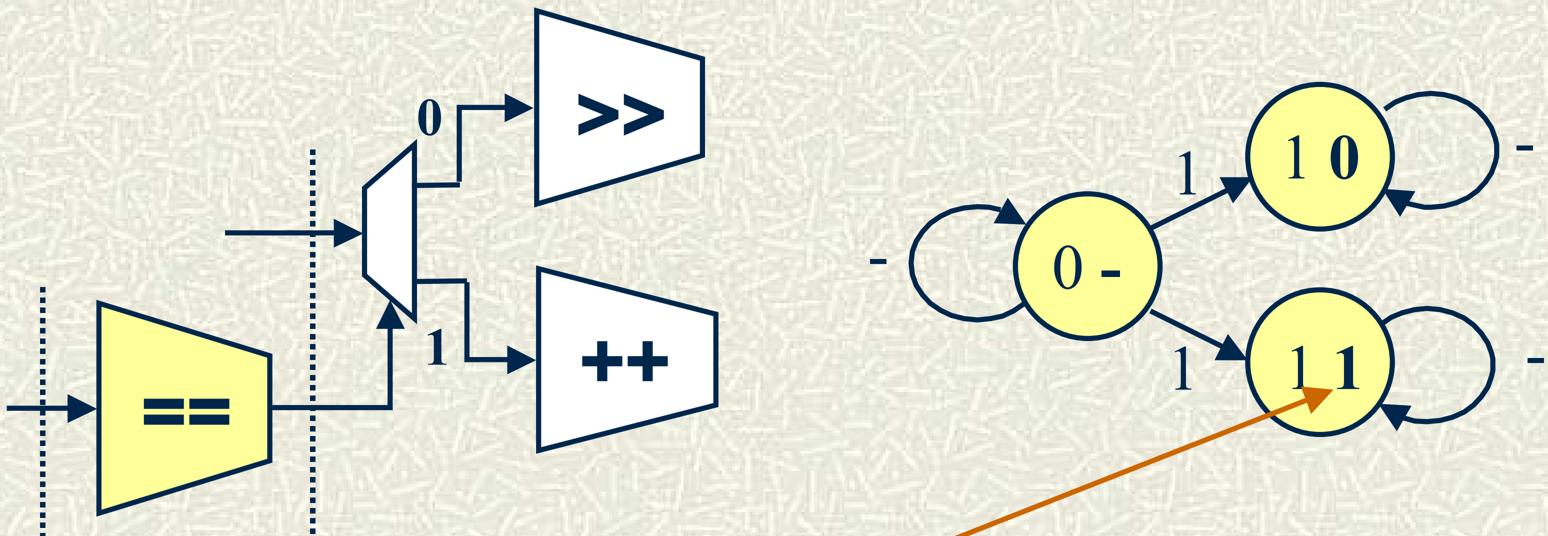
- Simple to construct abstraction of signaling
- E.g. 2 (hit), 5 (miss) cycle latency cached memory access:



Controls

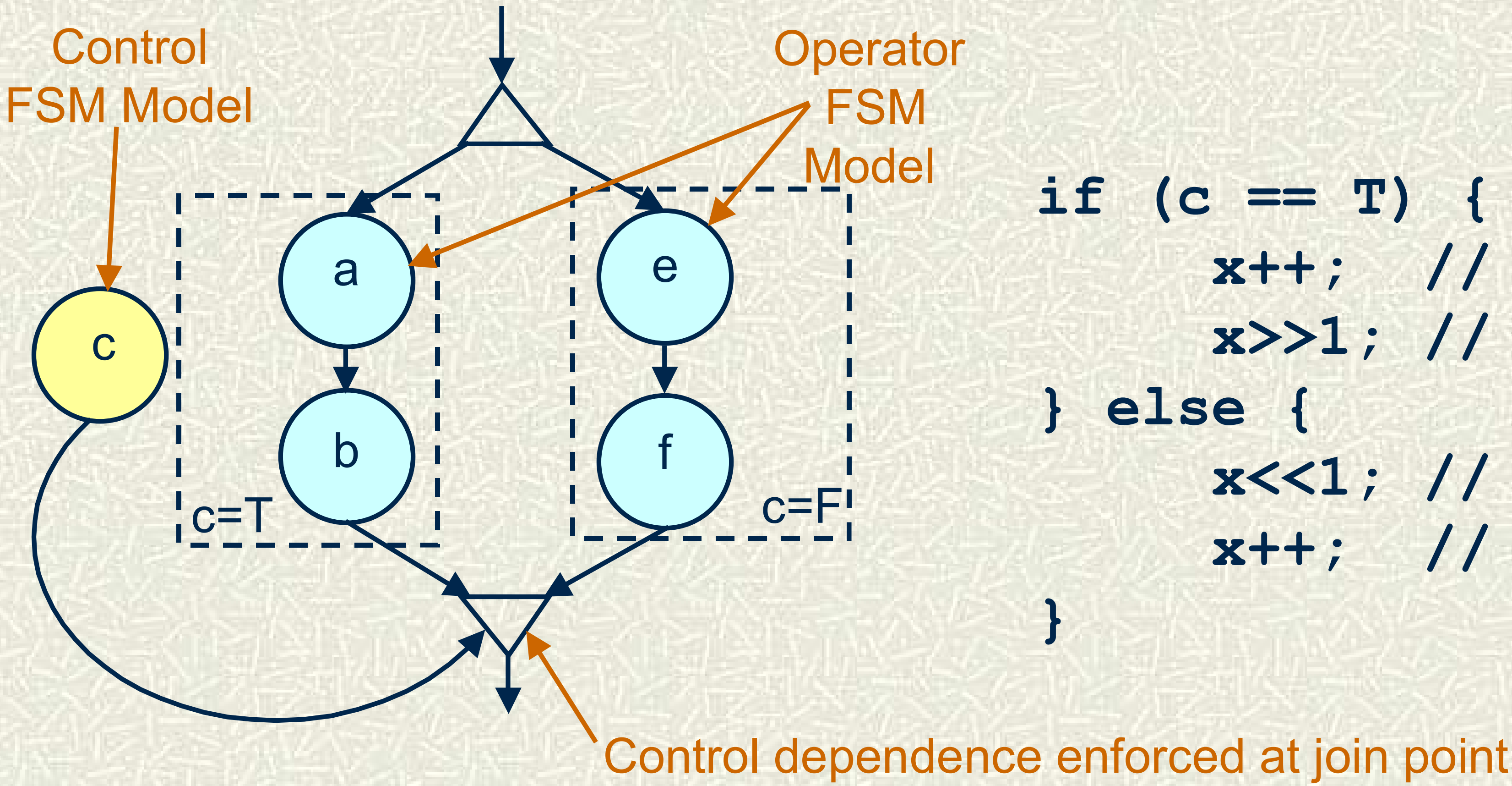
- Control operations need to bifurcate model state space
- Example:

  1 cycle, 1 input, 1 bit control op:



- **Bold** bit distinguishes traces

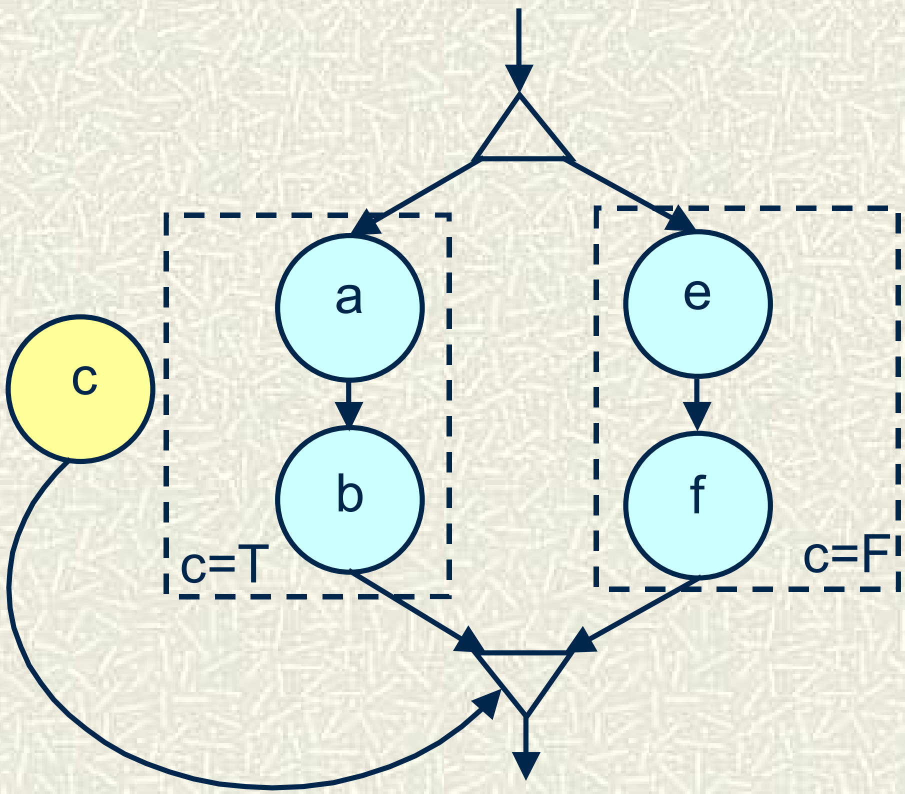
CDFG Model




```
if (c == T) {
      x++;   // a
      x>>1; // b
} else {
      x<<1; // e
      x++;   // f
}
```

Composite NFA Model of CDFG

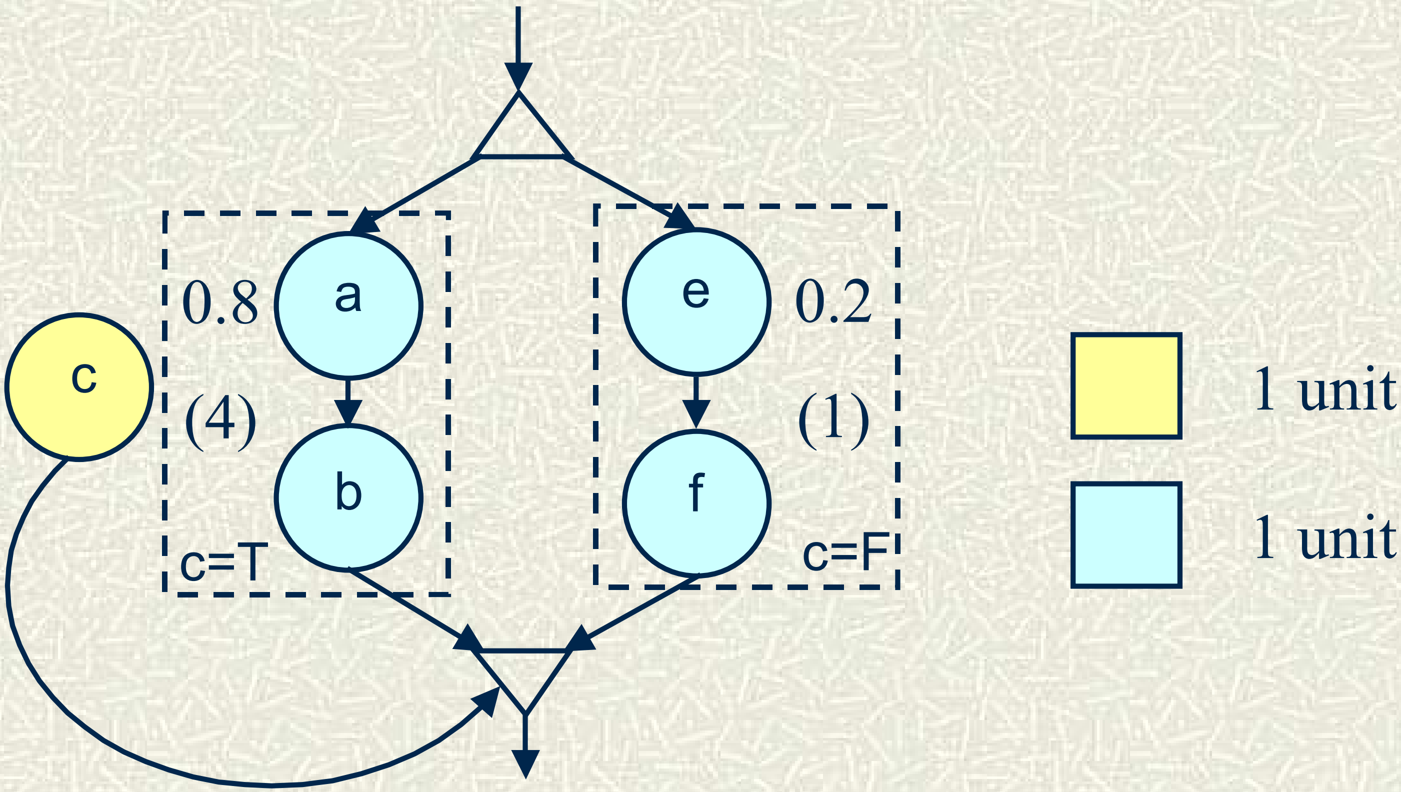
- Boolean encoding of composite FSM



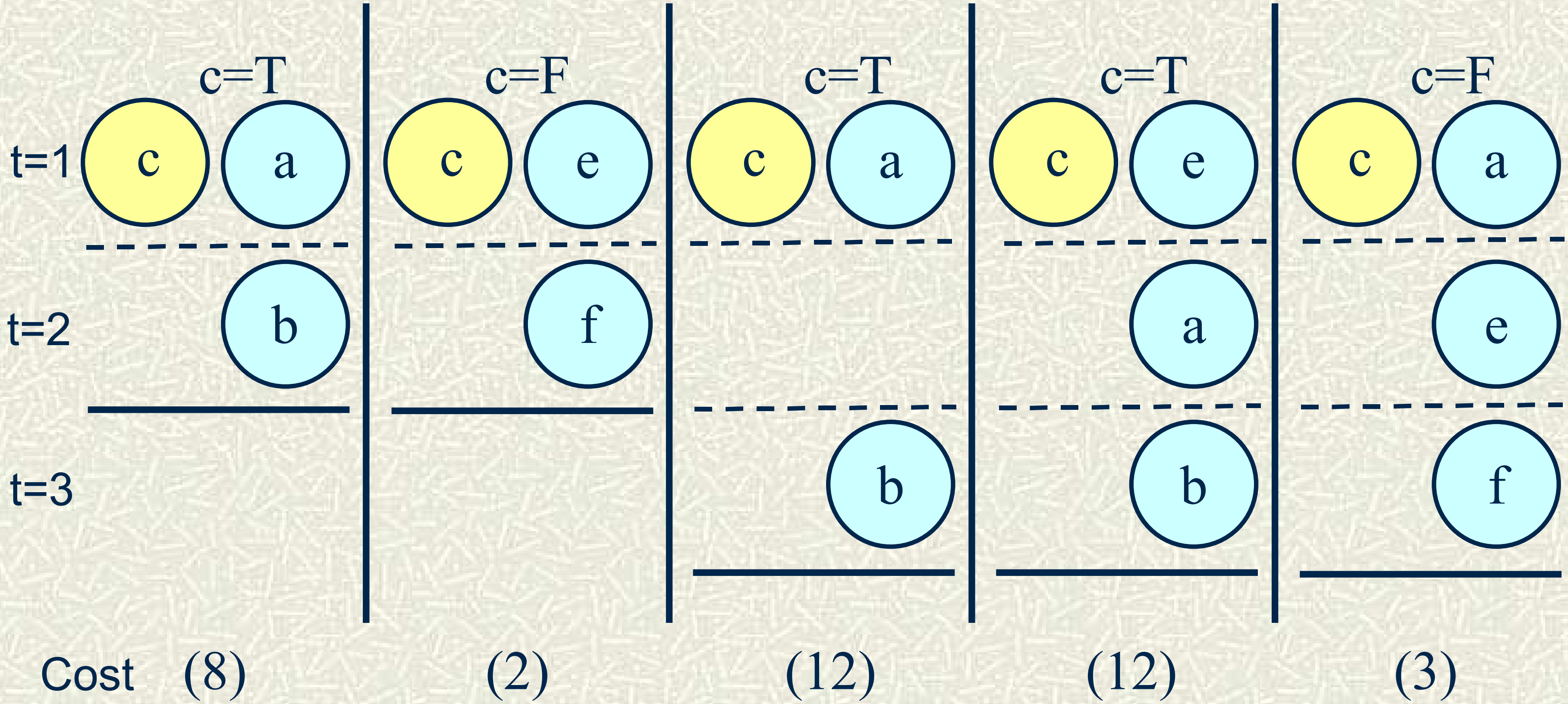
Composite State Encoding

c	a	b	e	f	
10	1	0	0	0	(c=F, a)
0-	1				Triggering for 'b'
11	1				
10	-	-	-	1	Join point
11	-	1	-	-	

Weighted example

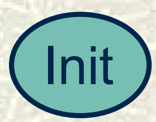


Representative traces



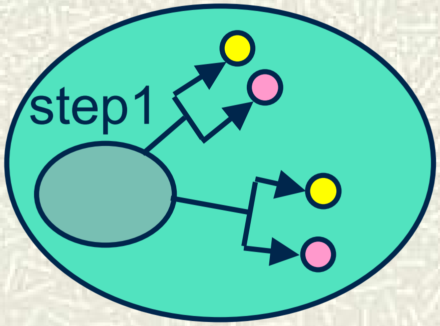
Scheduling - I

- BFS of state space
- Start with initial operands known
- Forward image computation



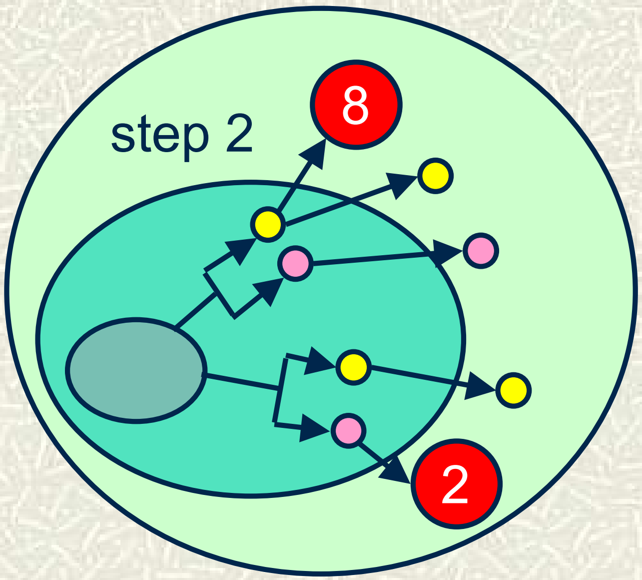
Scheduling - II

- Bifurcating traces
- Speculation



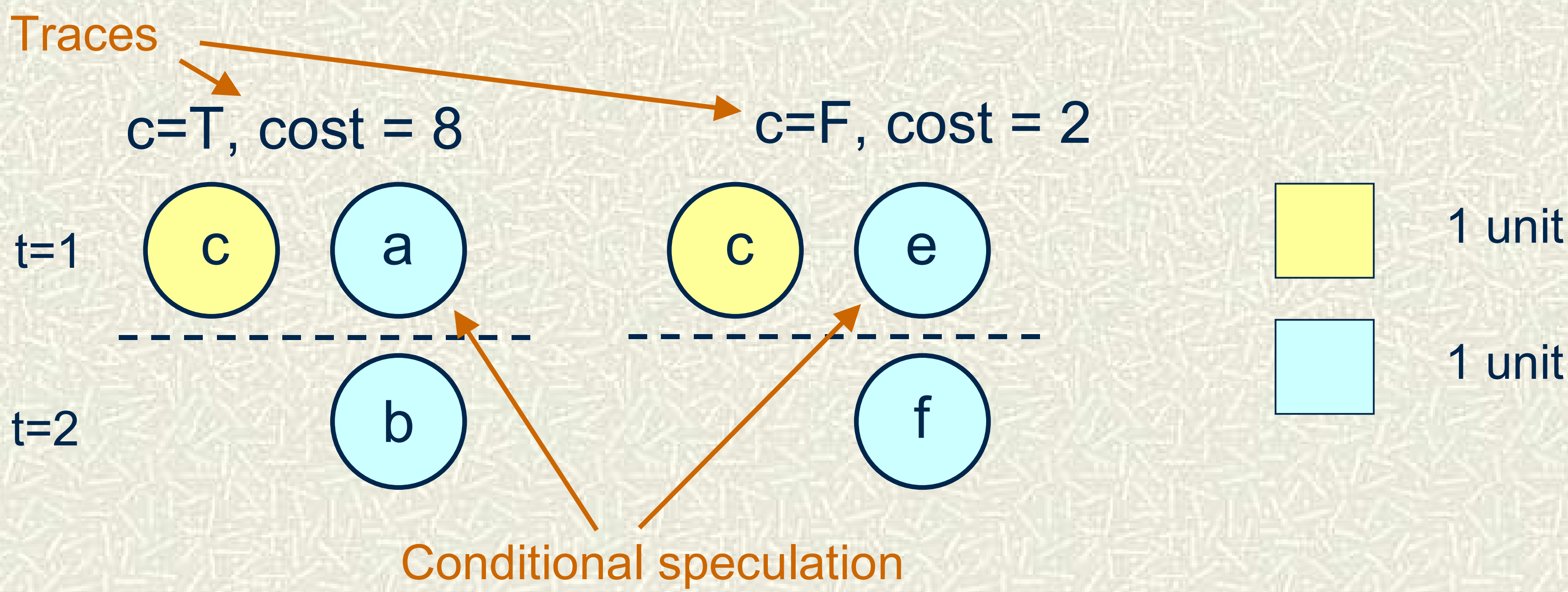
Scheduling - III

- First terminals seen
- Labeling of costs
- Causality



Validation

- Complete but incompatible pair of traces



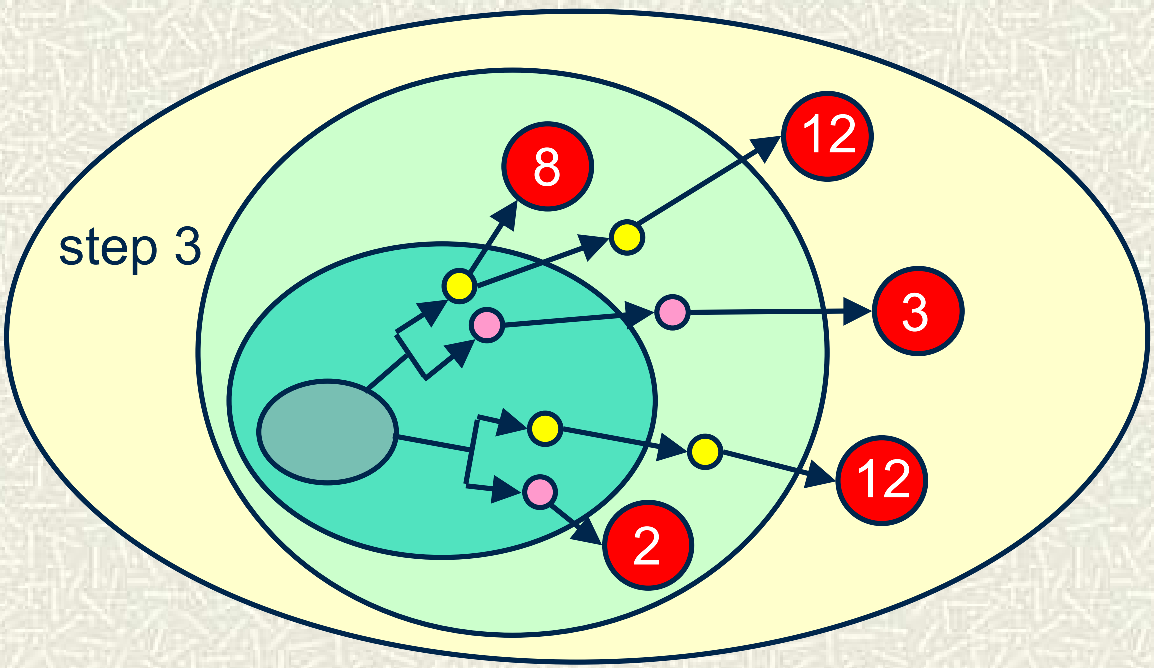
Cost representation

- Cost – normalized weight x latency
- Cost represented as binary-encoded integer
- BDD encoding attached to state BDD
- Example: (after 2 cycles)

```
c  a b e f  <-cost->
11 1 1 0 0   01000  (4 x 2 = 8)
10 0 0 1 1   00010  (1 x 2 = 2)
```

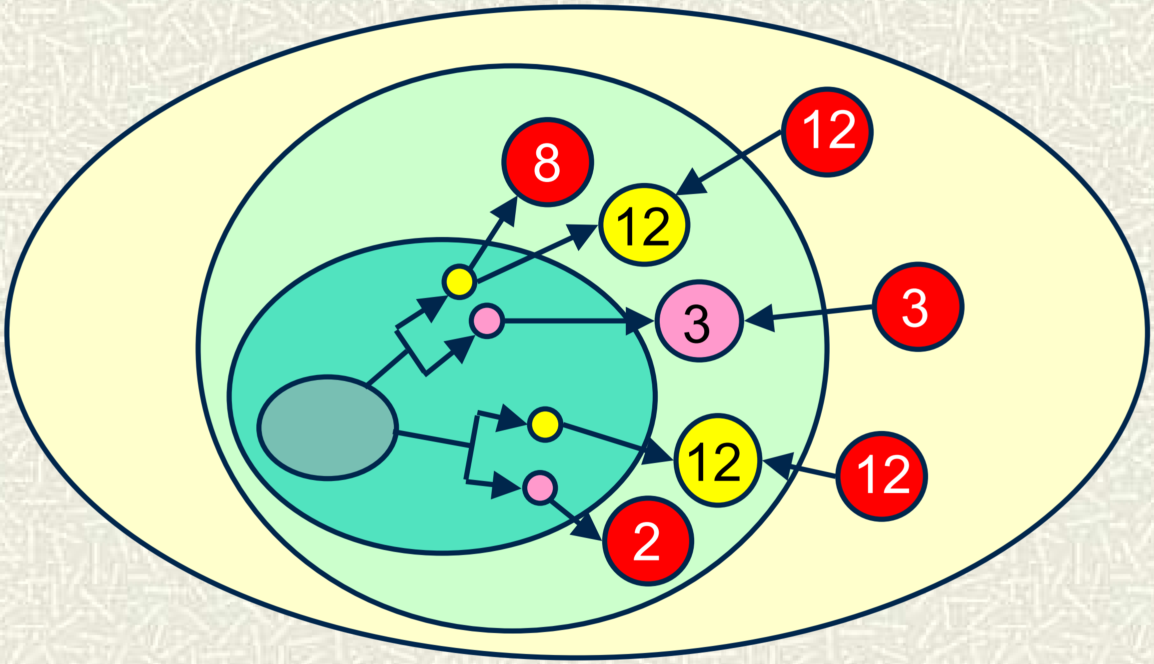
Scheduling - IV

- Complete solutions present!



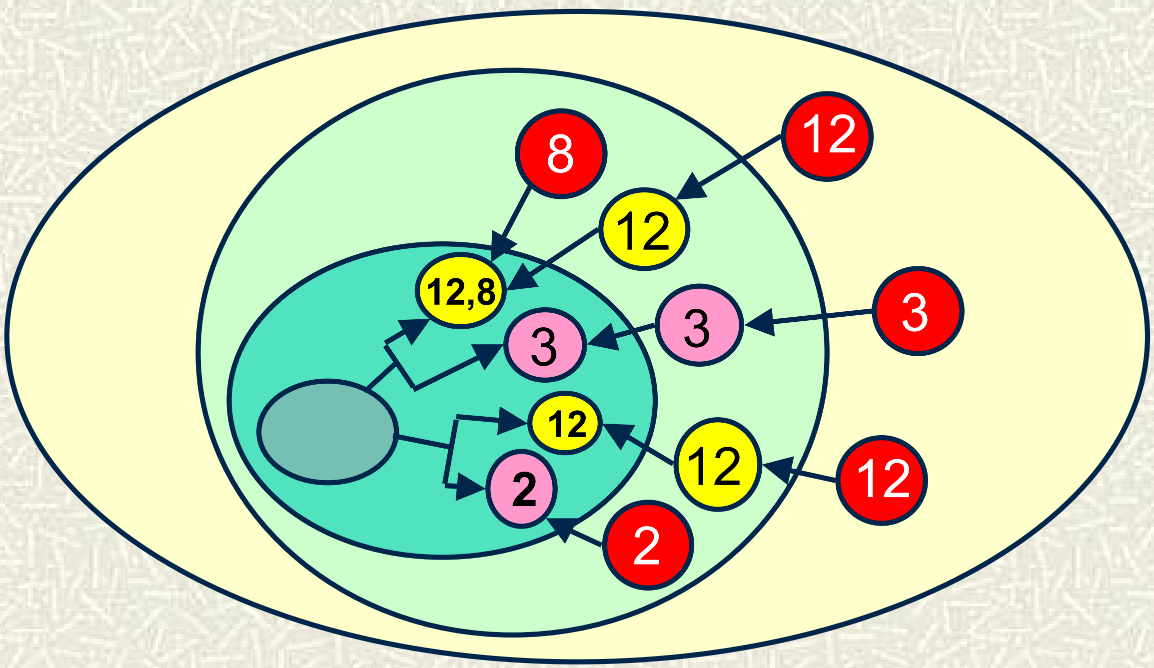
Scheduling - V

- Reverse propagate labels



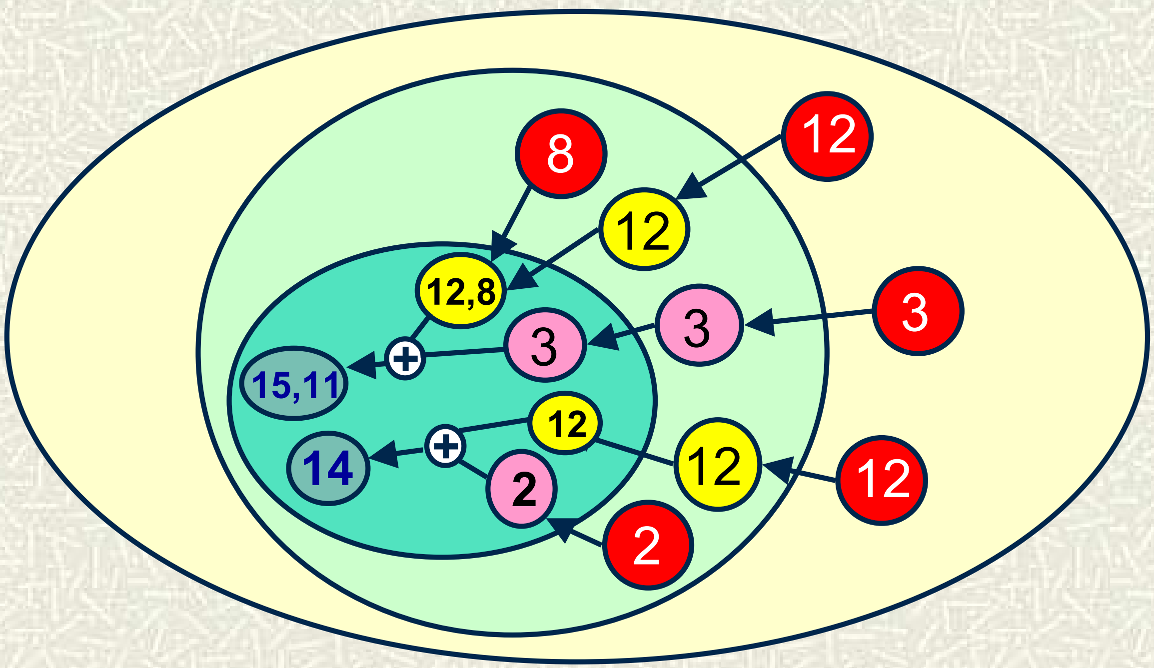
Scheduling - VI

- States may have multiple cost labels

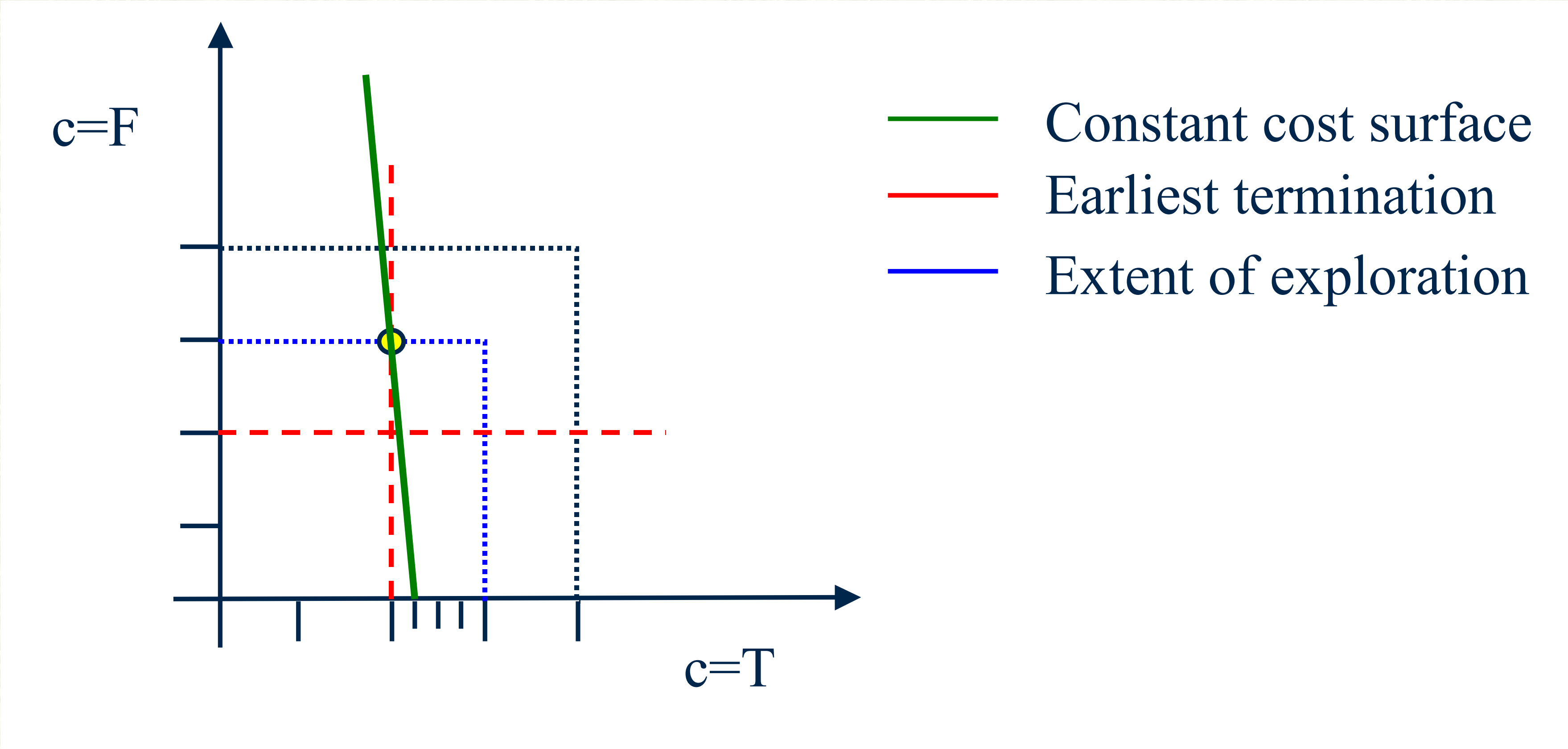


Scheduling - VII

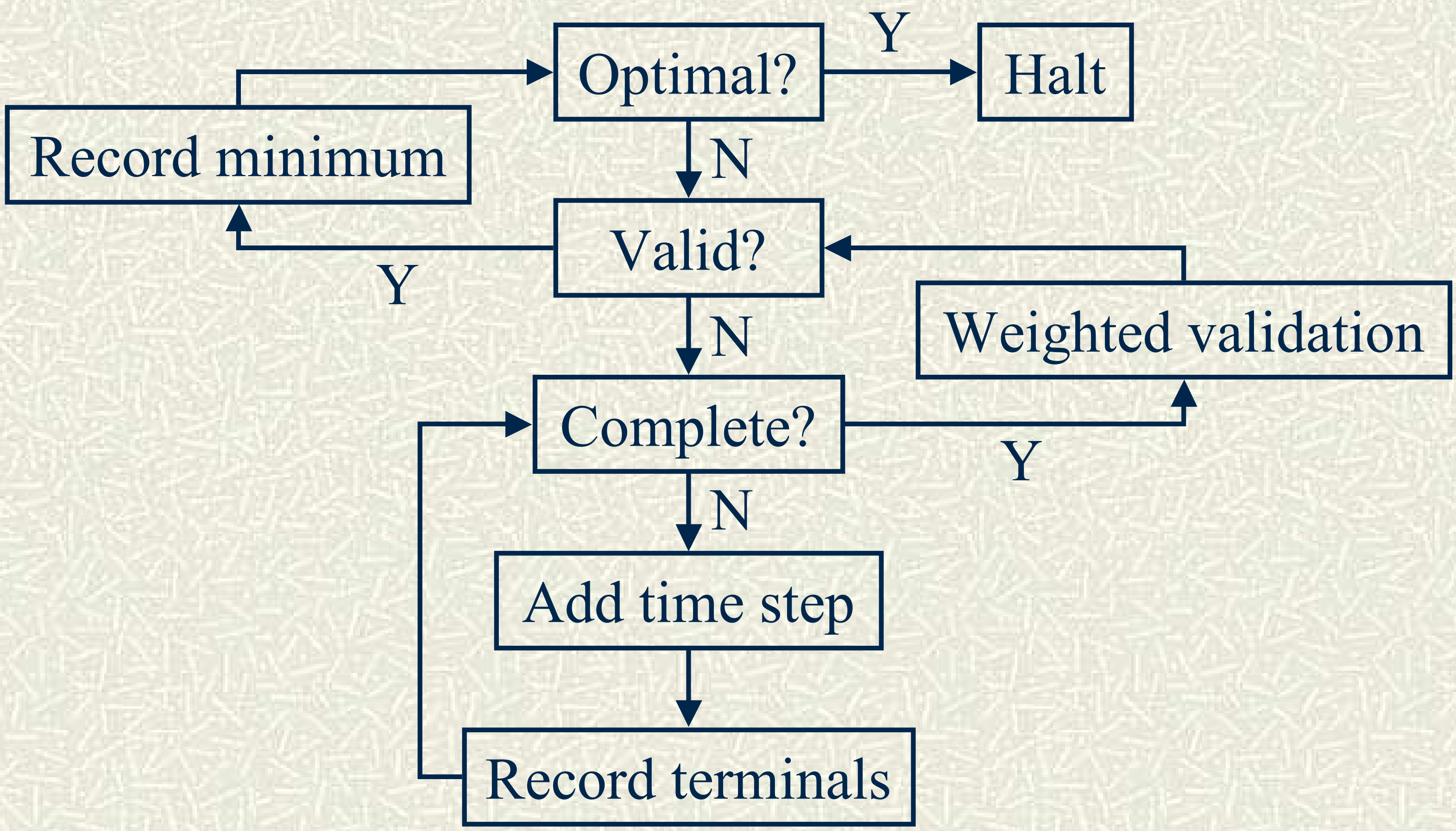
- Parallel addition across resolving transitions



Solution Space and Optimality



Algorithm



Optimal?
Y
Halt
Record minimum
N
Valid?
Y
N
Weighted validation
Complete?
Y
N
Add time step
Record terminals

Note on Cost Calculation

- Addition of trace costs commutes over controls
- Not restricted to numbers, addition!
- Any set of labels
- Any binary operator that commutes over the controls
- Cheap BDD representation of operation needed
- For addition → linear with interleaved variables

Note on Termination Criterion

- Exploration cannot stop at state saturation!
  - Only true for restricted class of models
- Optimal cost
- Path saturation
- Some sequential constraints may render solution impossible

Some experiments

- ROTOR – 28 ops, 3 controls, longest path 7
- ADPCM encoder – 45 ops, 11 controls, longest path 15
- Kim54 – 54 ops, 5 controls, longest path 9 (1 cycle multiply)
- Synthetic CDFG – 6 ops, 1 cached access, 1 black box operator with dynamic abort
- BDD package – CUDD
- P4 3GHz/2GB Linux PC

Experimental results - ROTOR

Set	Q1	Q2	Q3	Q4
W0	0.25	0.25	0.25	0.25
W1	0.97	0.01	0.01	0.01
W2	0.6	0.15	0.15	0.1

Res.*	None	W0	W1	W2
1±,1×,1==	11	10.25(11)	9.05(12)	9.6(12)
2±,2×,1==	8	7.75(8)	7.03(8)	7.35(9)
2±,1×,2==	9	9(9)	8.04(10)	8.5(10)

*1 table look-up

Experimental results – Kim54,ADPCM

- Kim54 : 2+,2-,2×,1==
  Wavesched – 10/14/12.6
  Optimal – 7/9/7.875
  (shortest/longest/average)
- ADPCM encoder : 1±,2==, 2[], 1 <<
- Separately scheduled pre-loop,body,post-loop
- For 10 iterations :
  Spark – 192
  Optimal – 157.75(164)

Experimental Results – Synthetic CDFG

- Black box subsystem imposes sequential constraint
- Cache may fail to meet timing

Metric	Shortest	Longest	Expectation
Worst case	6	8	6.16
Weighted, Independent	5	11	5.48
Weighted, Correlated	5	11	5.45

Conclusions, Future Work

- Technique for optimizing weighted average latency
- No assumption of branch independence
- May be used to develop heuristics
- Explore other cost functions
- Basis for abstraction