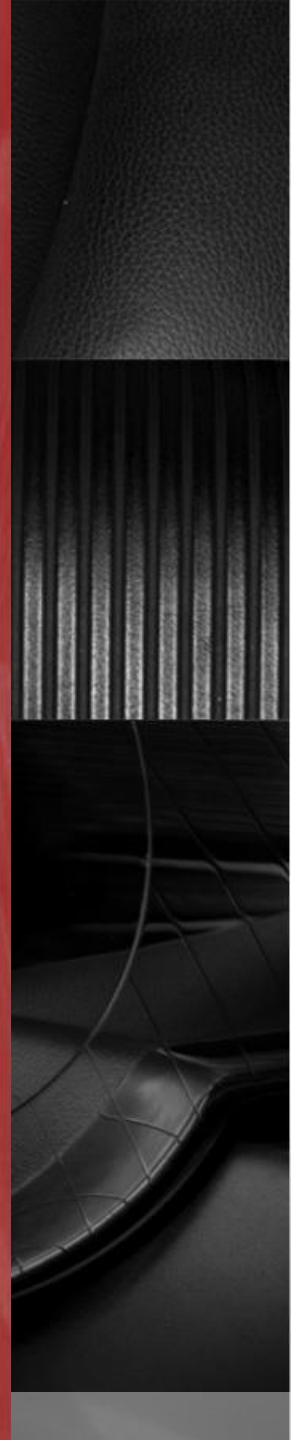
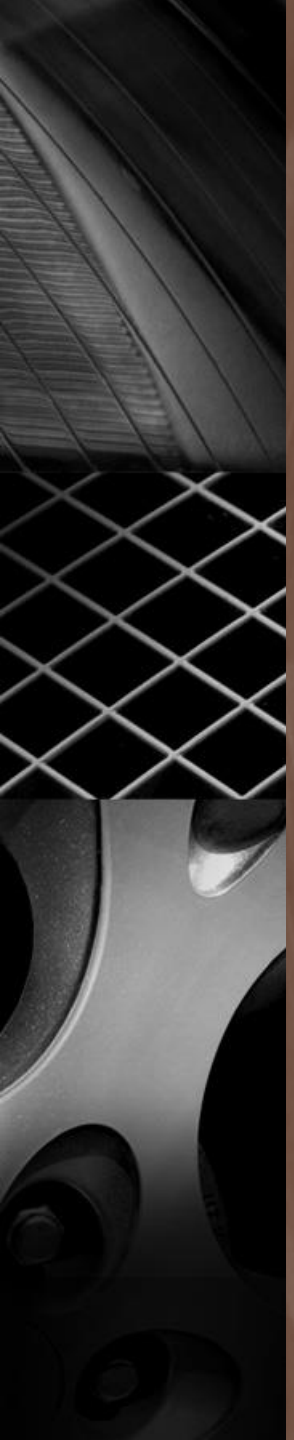
Lecture 4b

Scheduling





Scheduling

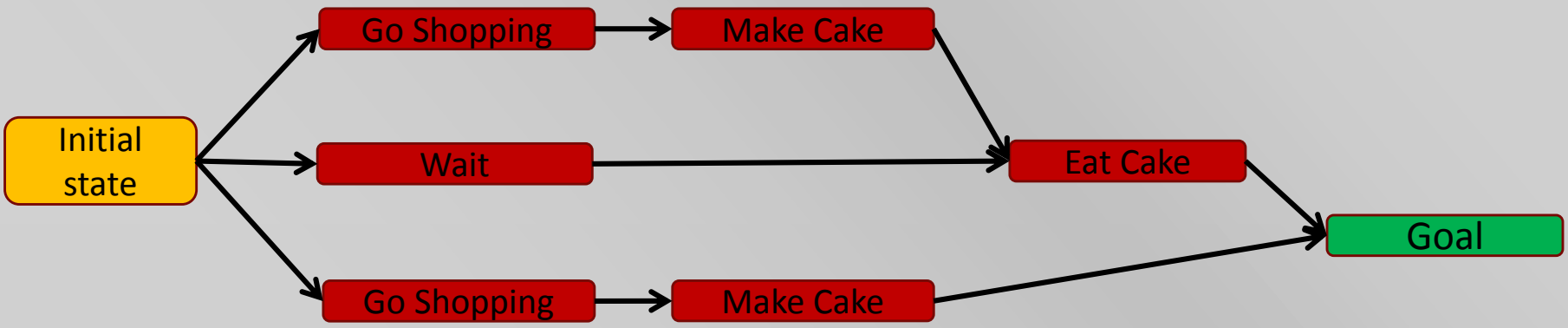
Scheduling

Planning: What to do and in what order

Scheduling: When to do required actions (given required resources)

- Assume we have a partial ordering of actions required to achieve our goals.
  - Total orders need to be relaxed. (We will not cover this.)
- Actions have:
  - Duration
  - Required consumable resources
  - Required durable/reusable resources
- Initial state has a specification of available resources.
- The task of scheduling is to provide an optimal schedule for the performance of the required actions. Optimal may be in terms of time (generally shortest) or resources (generally minimal use) or both.

Our Example Continued...

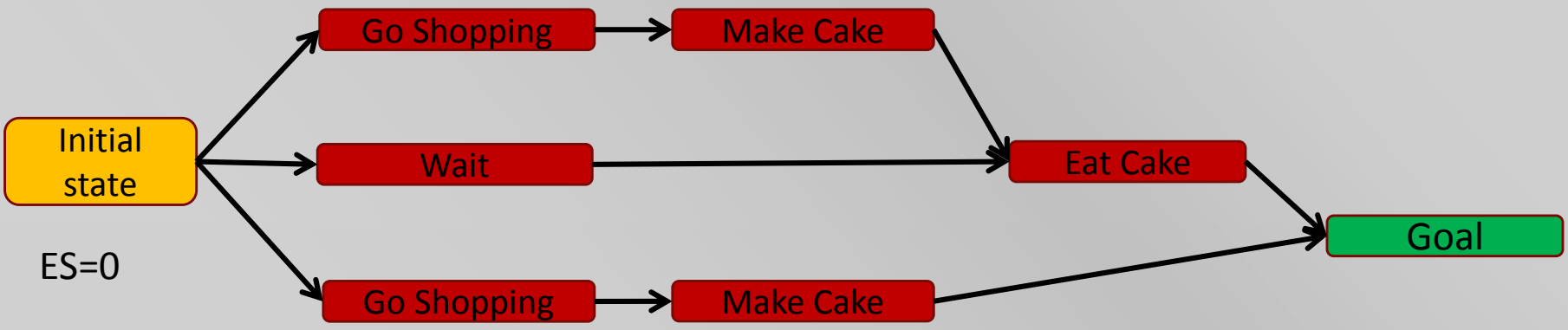


Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Resource-Free Scheduling

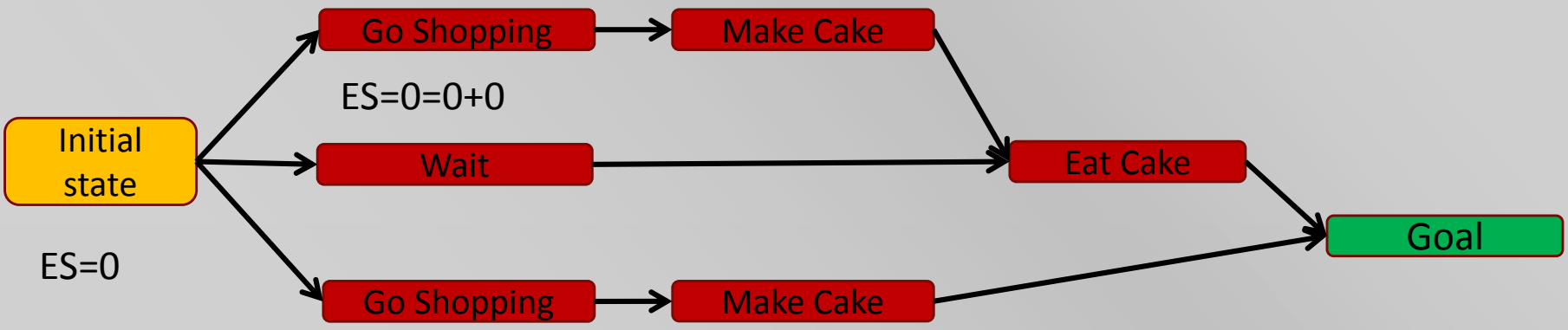
1. Dynamic Programming Algorithm for Resource-Free Scheduling
2. Set ES (Start)=0.
3. Iteratively set ES times for actions once their predecessors have been set ES times:
   - $ES(B) = \max_{A \prec B} ES(A)+Duration(A)$
4. Set LS (Finish)=ES(Finish)
5. Iteratively set LS times for actions once their descendants have been set LS times:
   - $LS(A) = \min_{B \succ A} LS(B)-Duration(A)$

Our Example Continued...



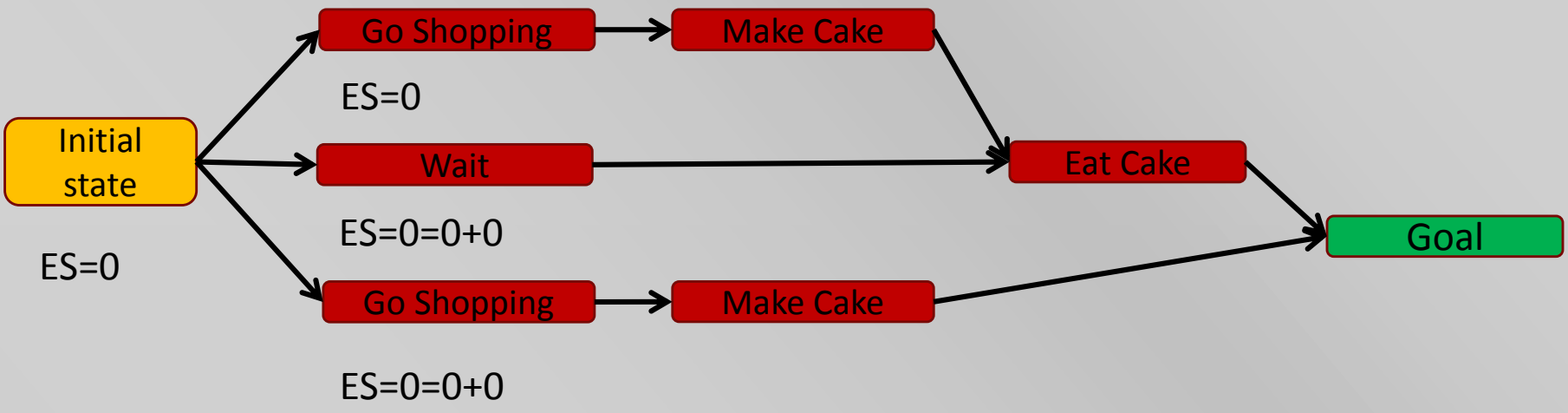
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...



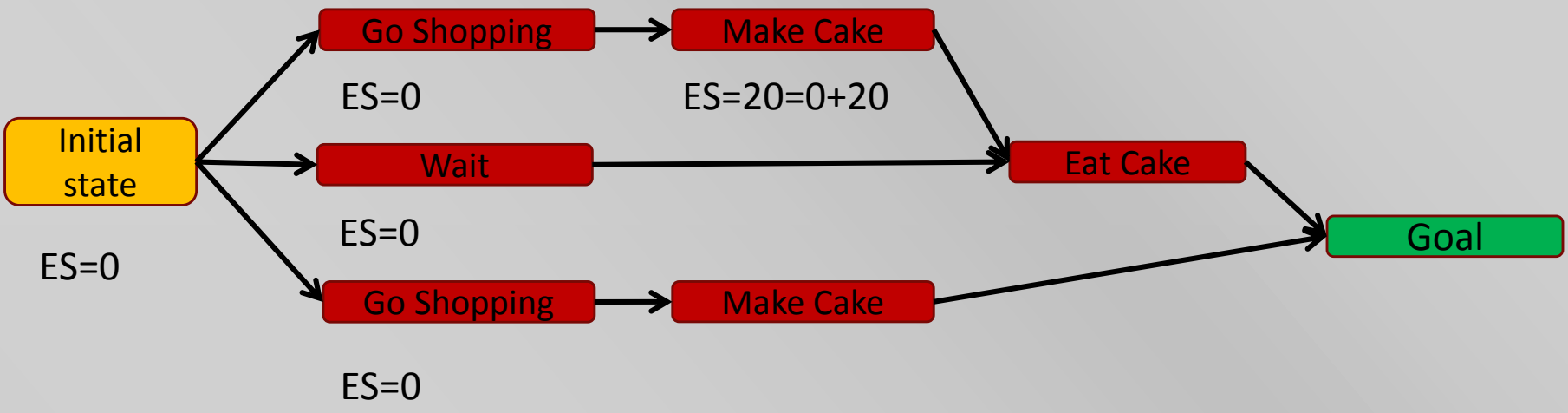
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...



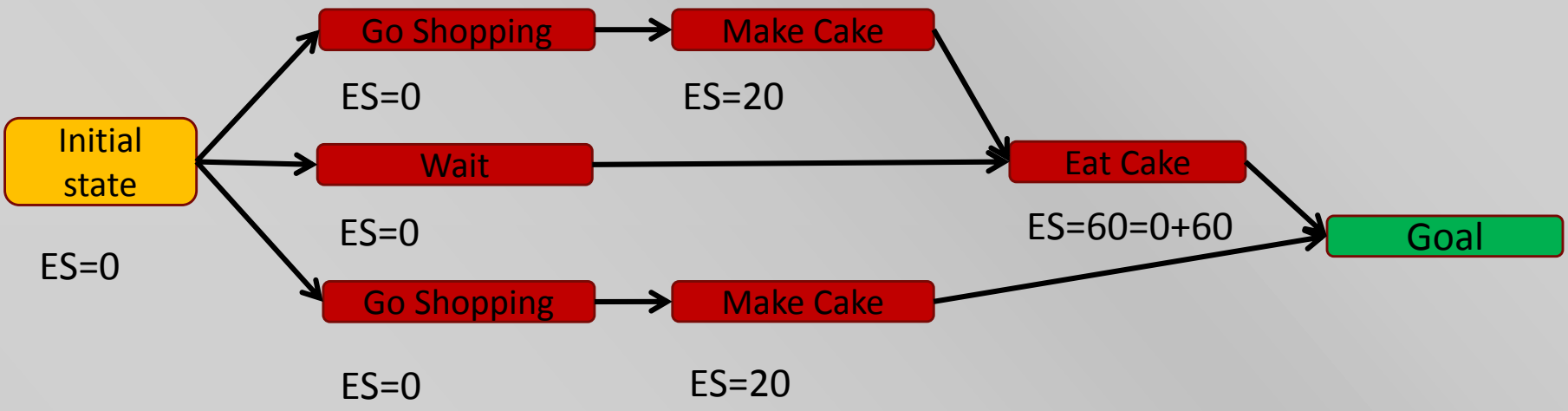
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...



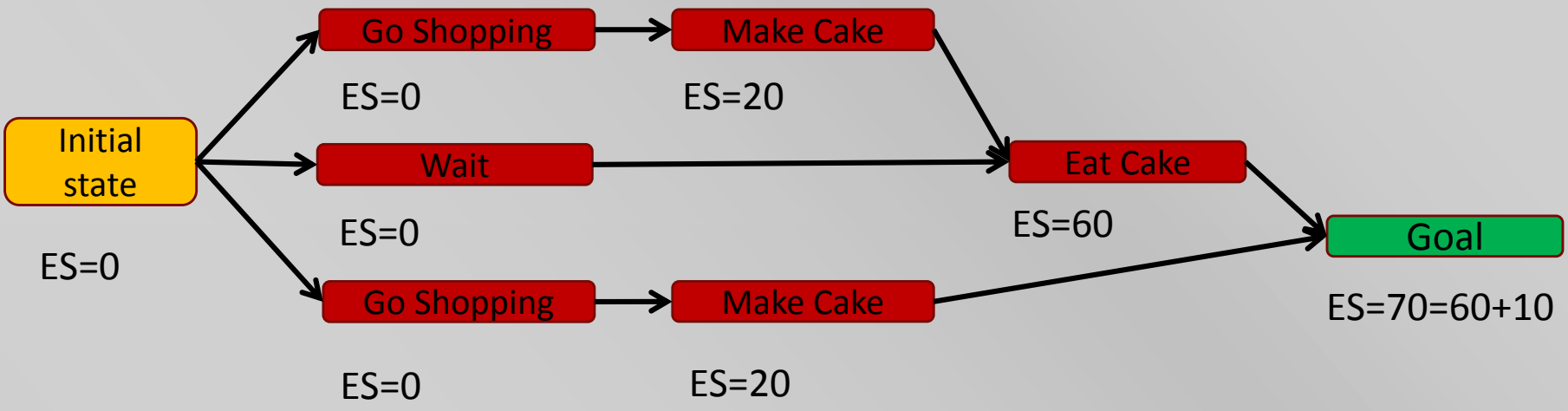
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...



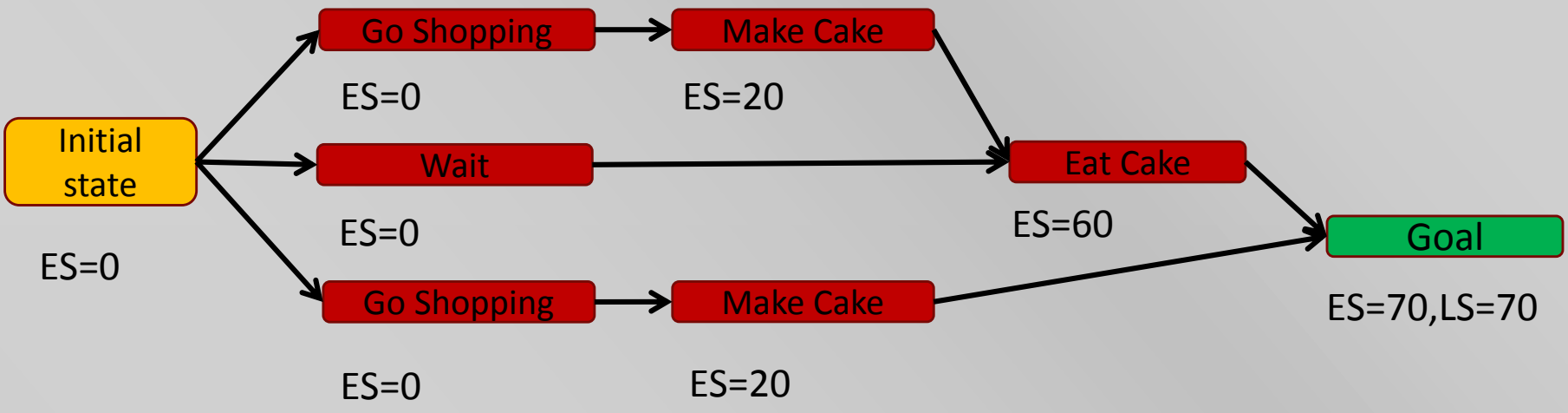
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...



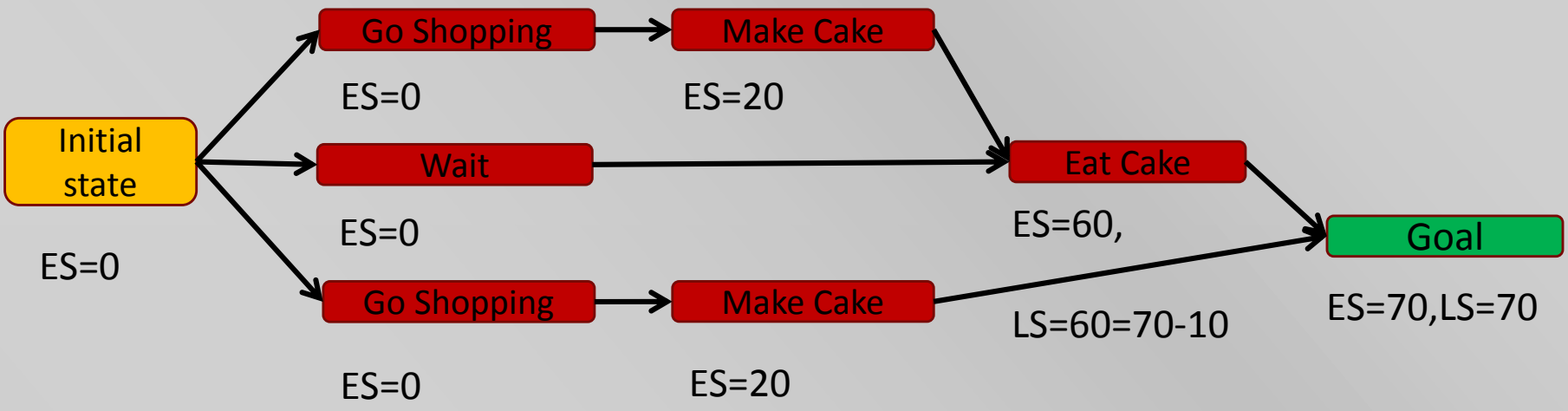
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...



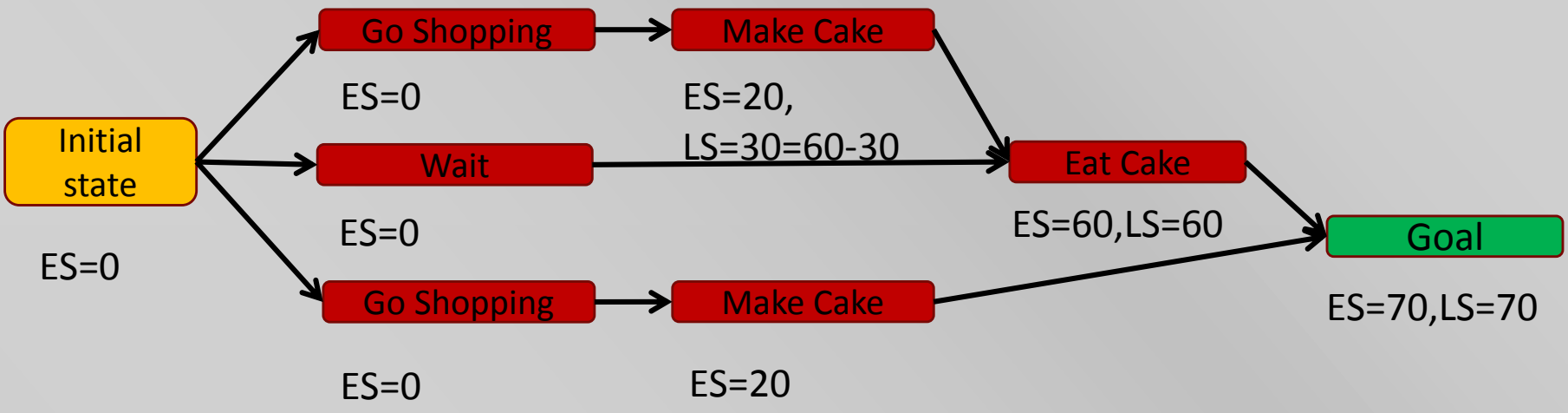
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...



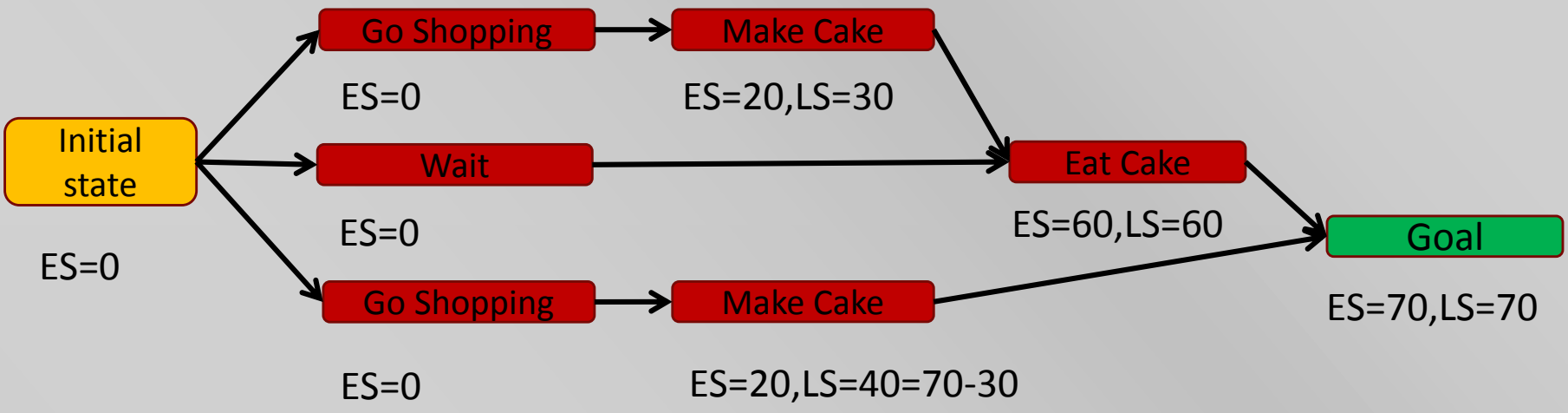
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...



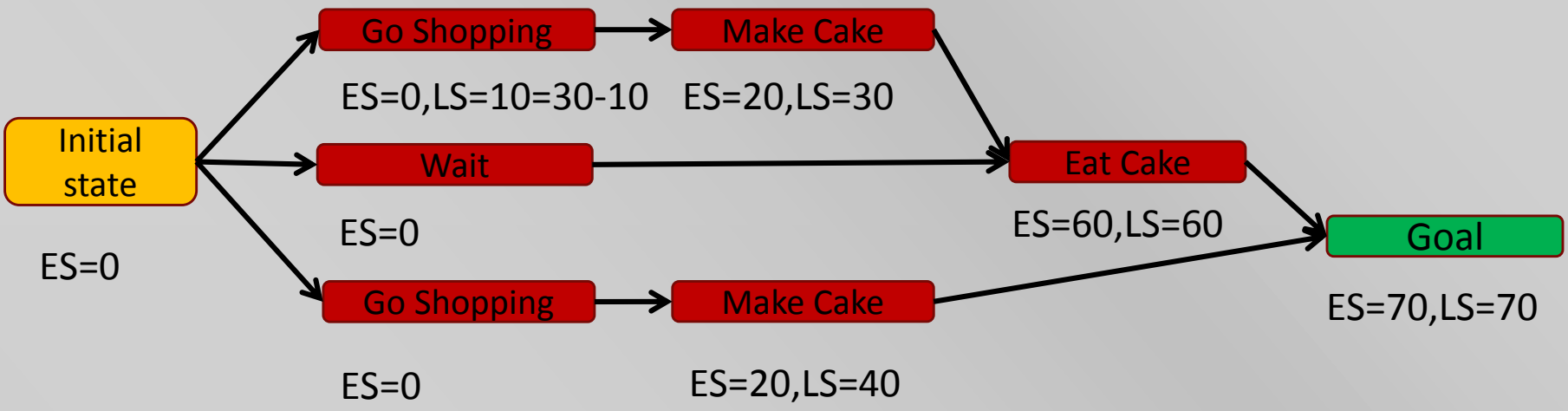
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...



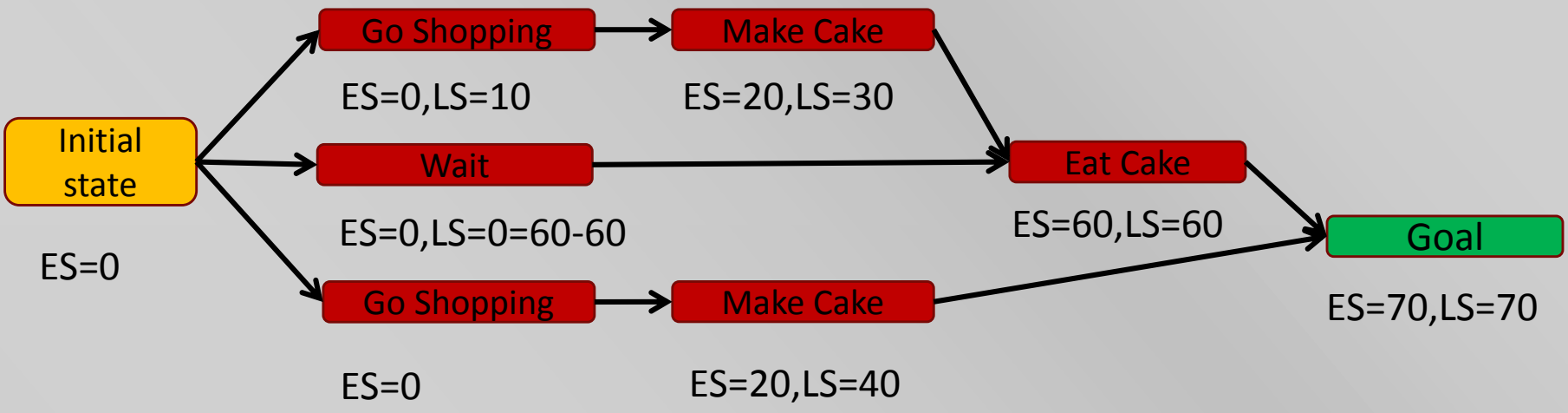
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...



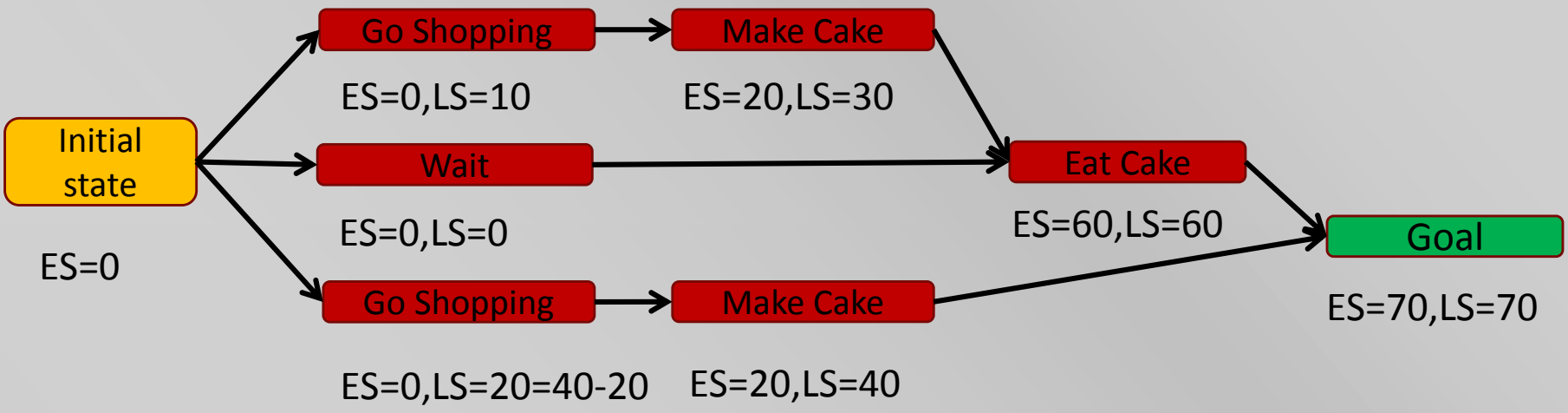
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...



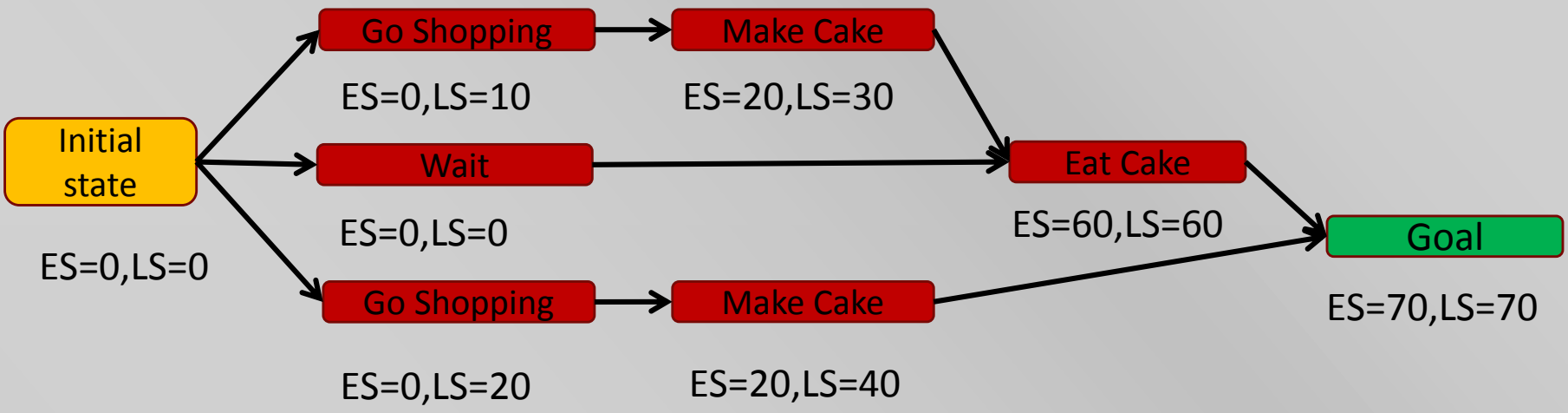
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...



Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...

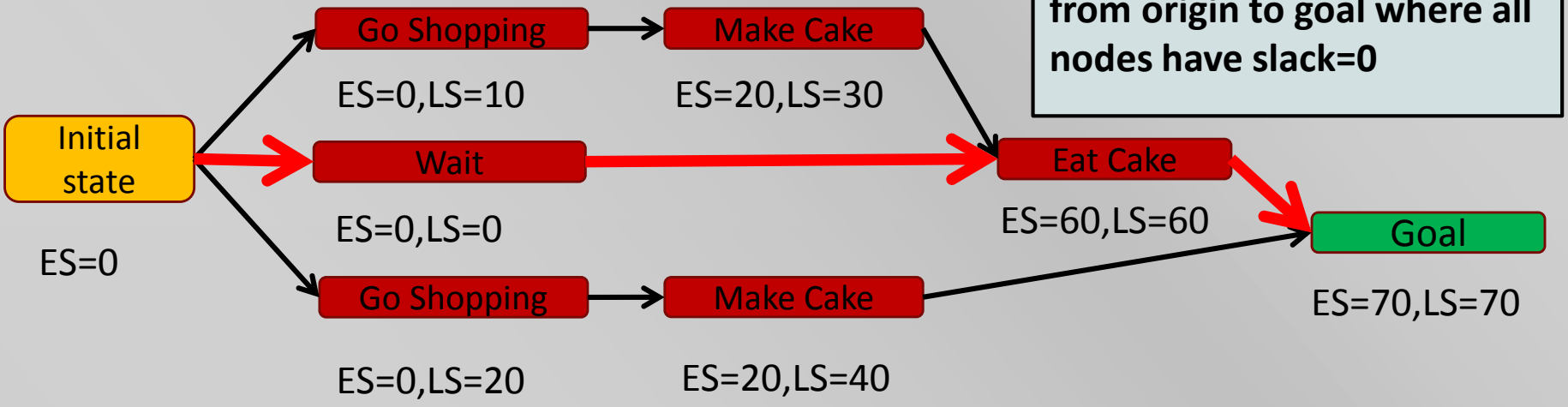


Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Our Example Continued...

Slack= LS-ES

The critical path is the path from origin to goal where all nodes have slack=0



Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Priority based scheduling with resource constraints

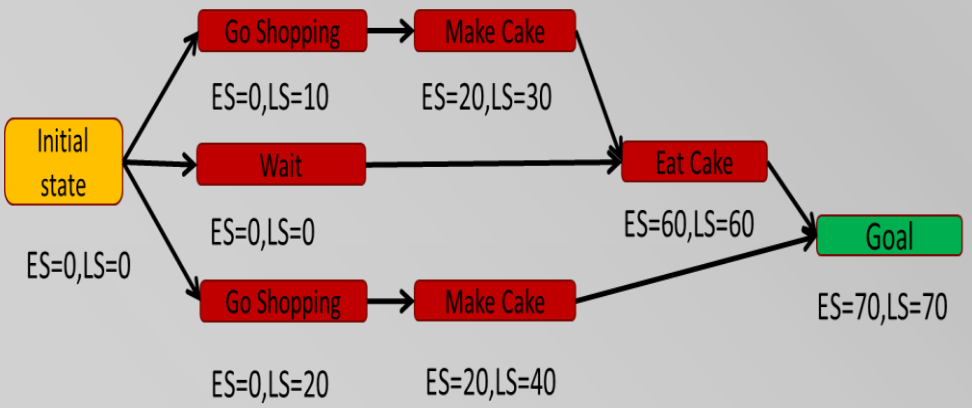
1. Set T (Start)=0.
2. Select highest priority action from those whose predecessors have had their ES scheduled, set its T as early as possible given resource constraints. When doing so, record which resources used and effects on consumed resources.
3. Repeat step 2 until all actions T scheduled.

What should our priority function be?

One idea: Minimum slack

- We prioritize actions based on their slack (LS-ES) in the resource free schedule.

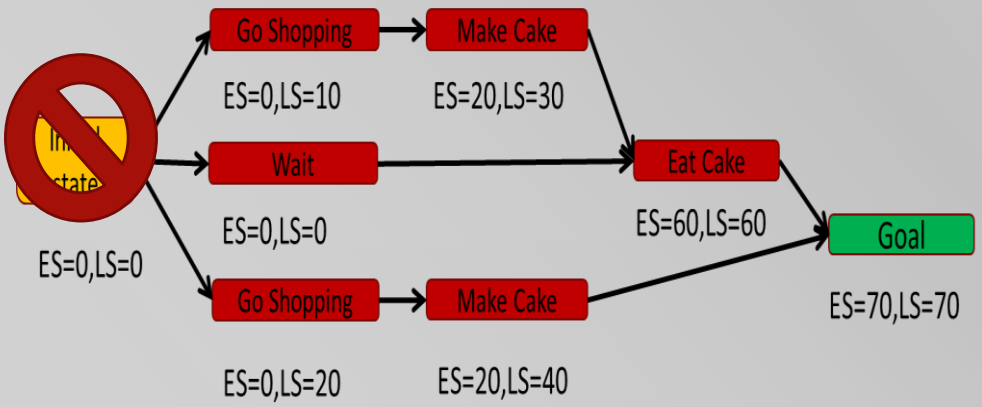
We can redo the resource free algorithm, given (resource constrained) scheduled actions at each iteration of this algorithm.



Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

Initial state

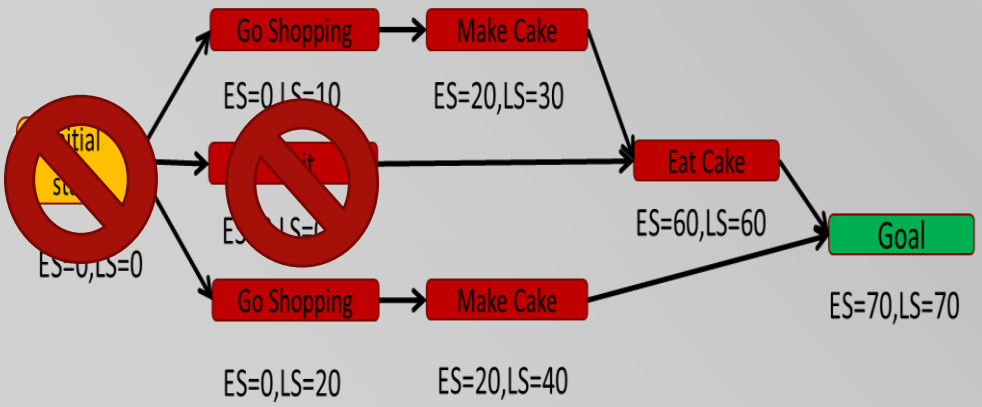
TIME:	0
Mike	
Oven	
Cakes	0
Mixes	0



Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

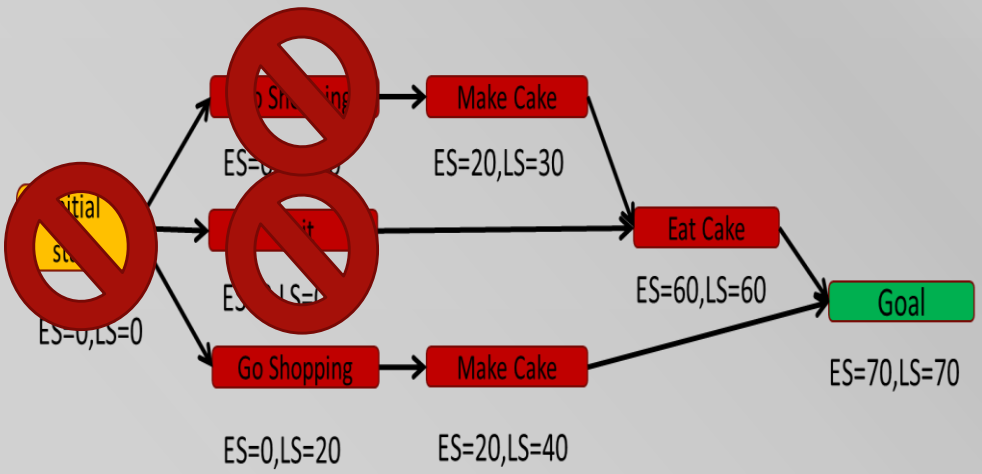
Initial state

TIME:	0
Mike	
Oven	
Cakes	0
Mixes	0



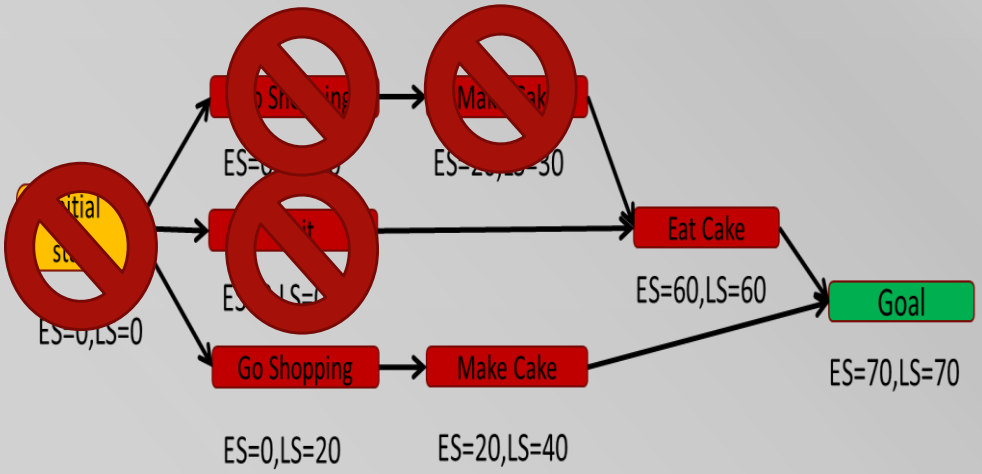
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

TIME:	0	60
Mike		
Oven		
Cakes	0	0
Mixes	0	0



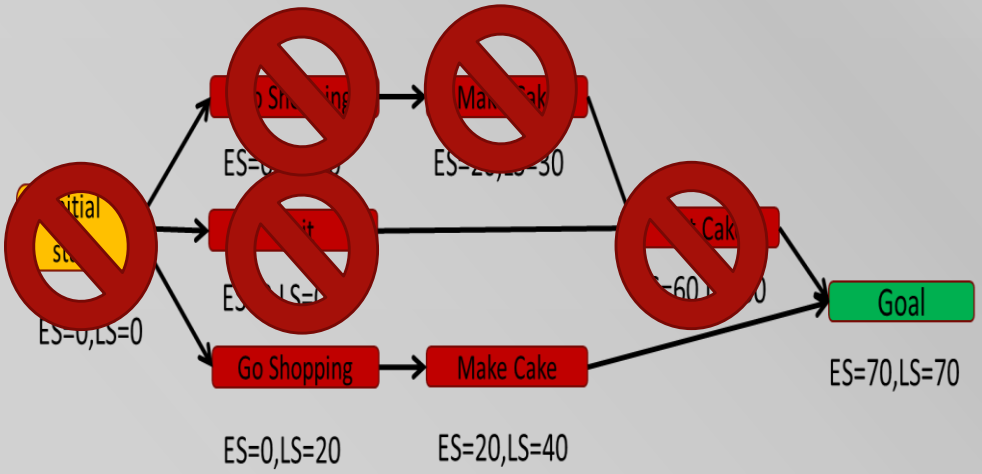
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

TIME:	0	20	60
Mike			
Oven			
Cakes	0	0	0
Mixes	0	1	1

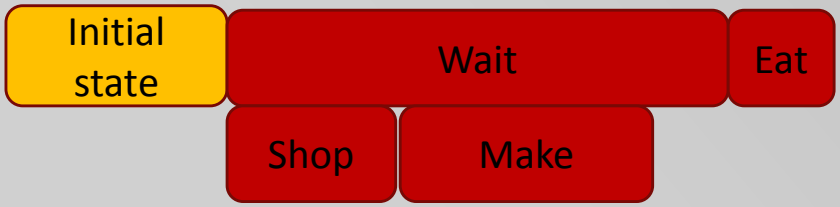


Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min

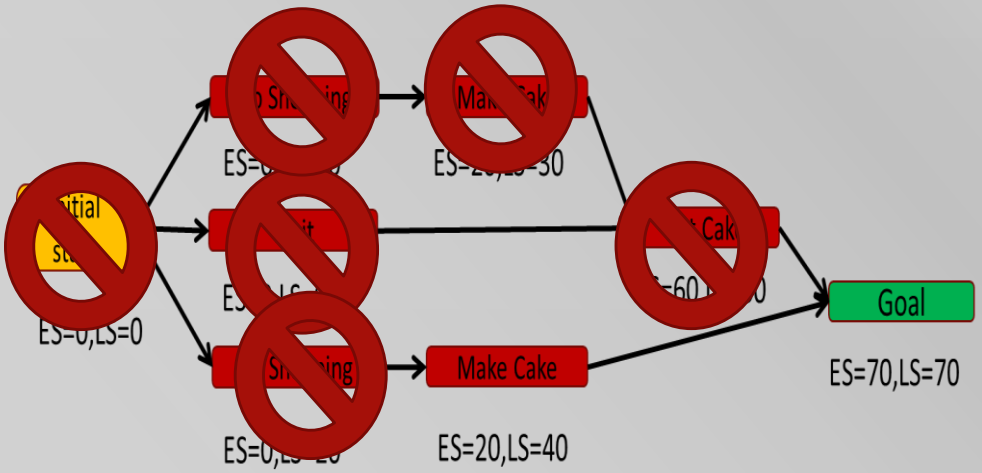
TIME:	0	20	50	60
Mike				
Oven				
Cakes	0	0	1	1
Mixes	0	(1) 0	0	0



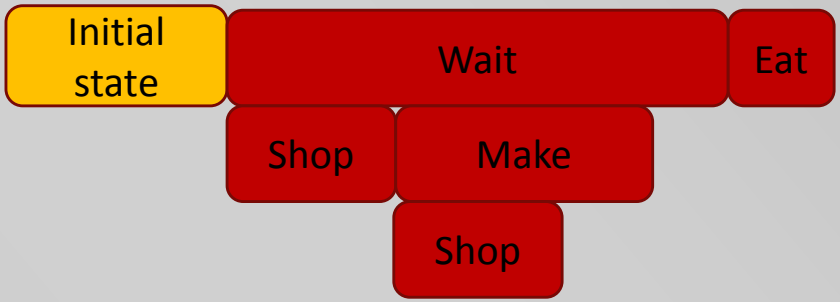
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min



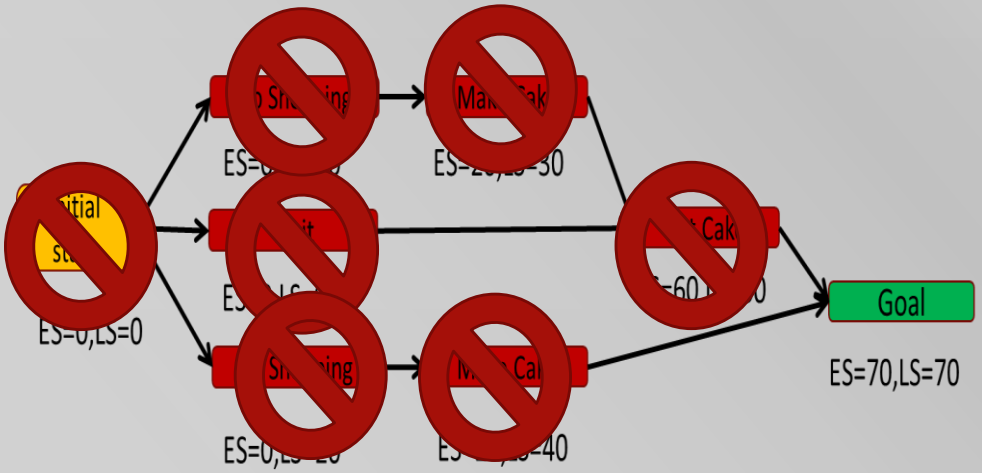
TIME:	0	20	50	60	70
Mike					
Oven					
Cakes	0	0	1	(1) 0	0
Mixes	0	(1) 0	0	0	0



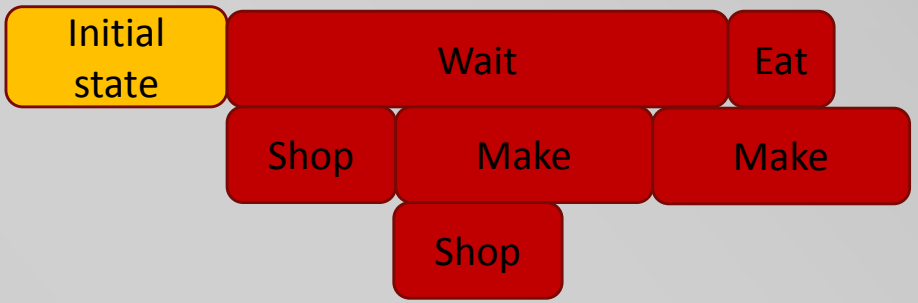
Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min



TIME:	0	20	40	50	60	70
Mike						
Oven						
Cakes	0	0	0	1	(1) 0	0
Mixes	0	(1) 0	1	1	1	1



Action	Consumes	Uses	Duration
Go Shopping	-1 Mix	1 Mike	20 min
Wait	-	-	60 min
Make Cake	1 Mix, -1 Cake	1 Oven	30 min
Eat Cake	1 Cake	1 Mike	10 min



TIME:	0	20	40	50	60	70	80
Mike							
Oven							
Cakes	0	0	0	1	(1) 0	0	1
Mixes	0	(1) 0	1	0	0	0	0

Scheduling with Resources

- Resource Free times are lowest bounds only.
- Minimum slack algorithm is not always optimal.
- No scalable optimal algorithm exists. (As far as I know.)