Deterministic Planning II & Probabilistic Planning I

Nathaniel Osgood

3-15-2004

Announcements

- Posted
  - Primavera tutorials
  - Complicated scheduling case
  - Problem set 4 (Scheduling; due Monday April 5)
- Wednesday guest lecture on behavioral managerial issues

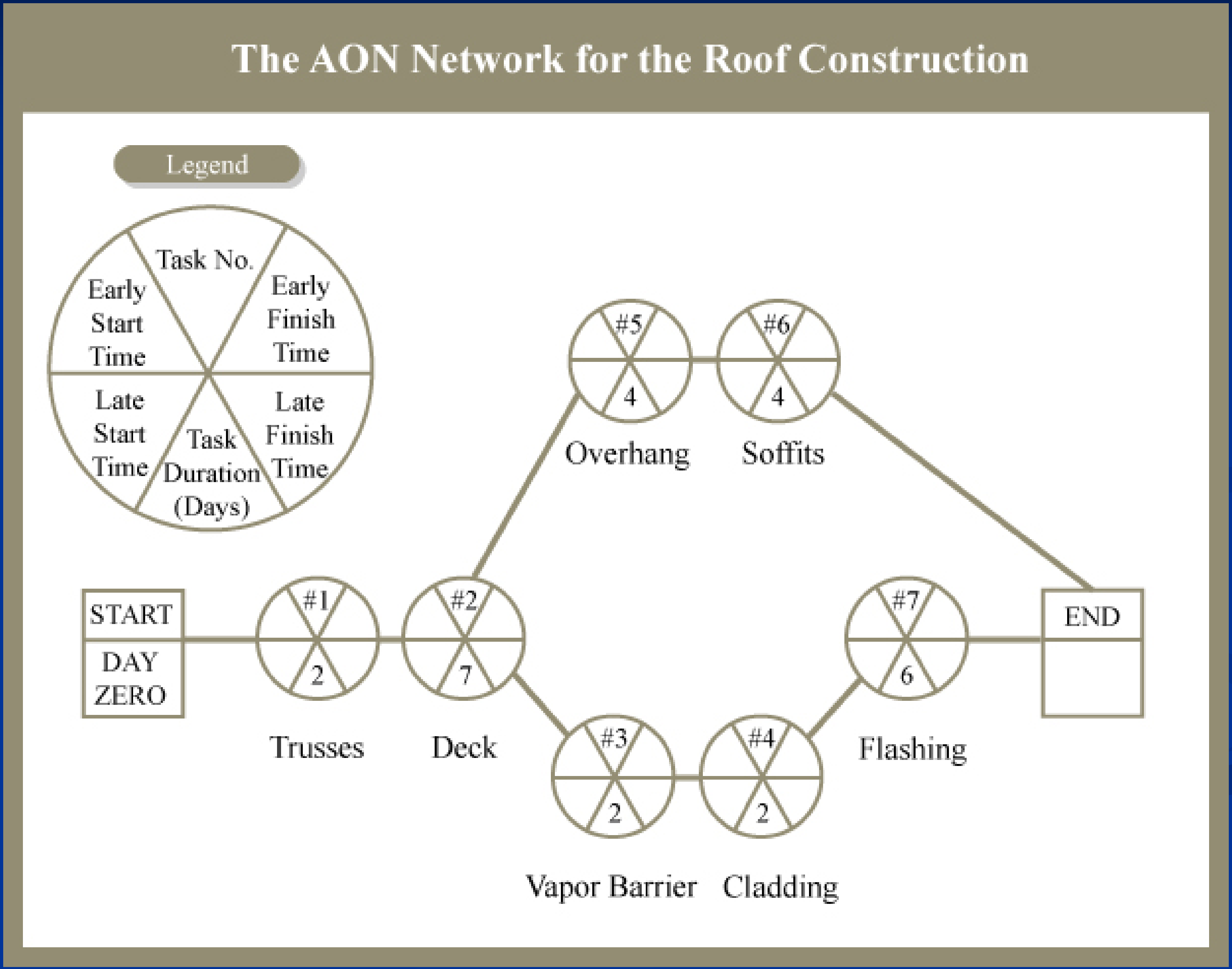
Recall: AON (PDM) Scheduling

- Activities shown on nodes
- O(*n*) Forward/backward pass to determine ES/EF/LS/LF
- Multiple types of relationships
  - FS, SS, FF, SF
- No dummy arrows required

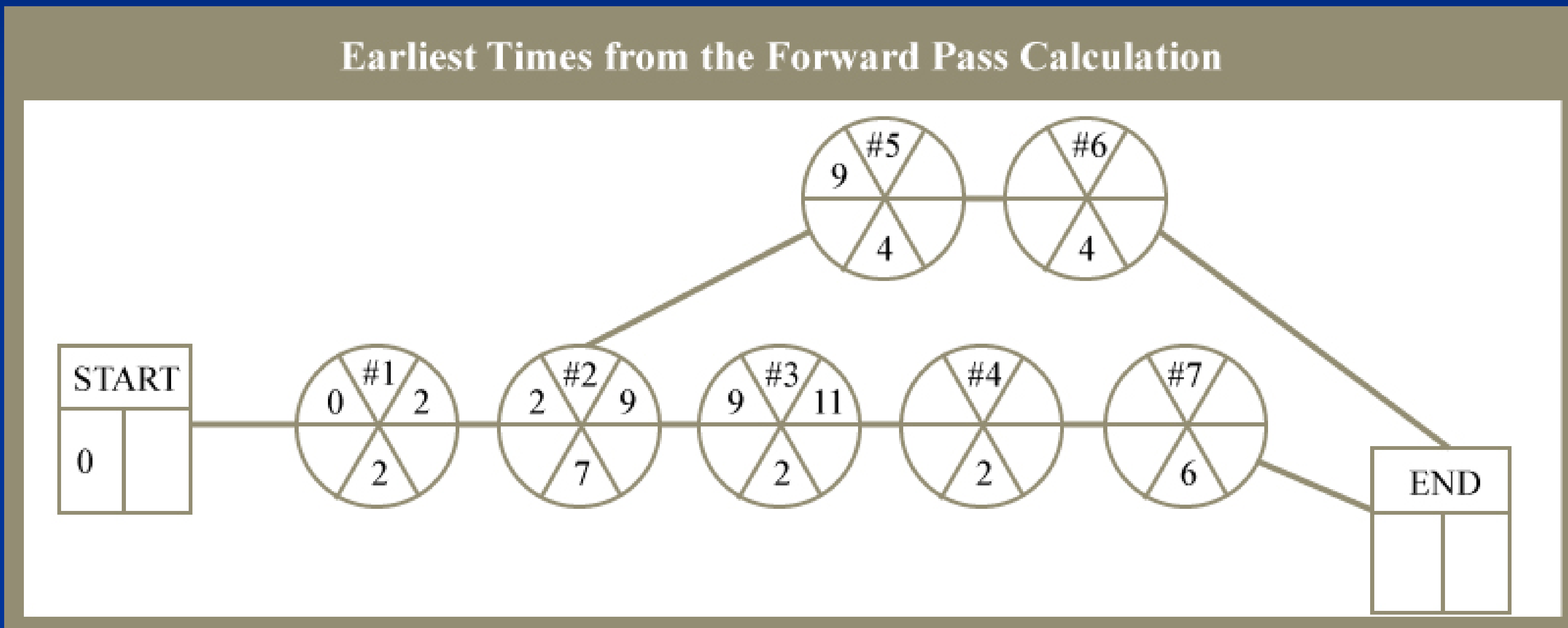
Example Applications

TASK NO.	TASK NAME	DURATION (in days)
1	Place and Secure Trusses	2
2	Install Roof Deck	7
3	Apply Vapor Barrier	2
4	Apply Roof Cladding	2
5	Construct Roof Overhang	4
6	Install Soffits	4
7	Apply Flashing	6

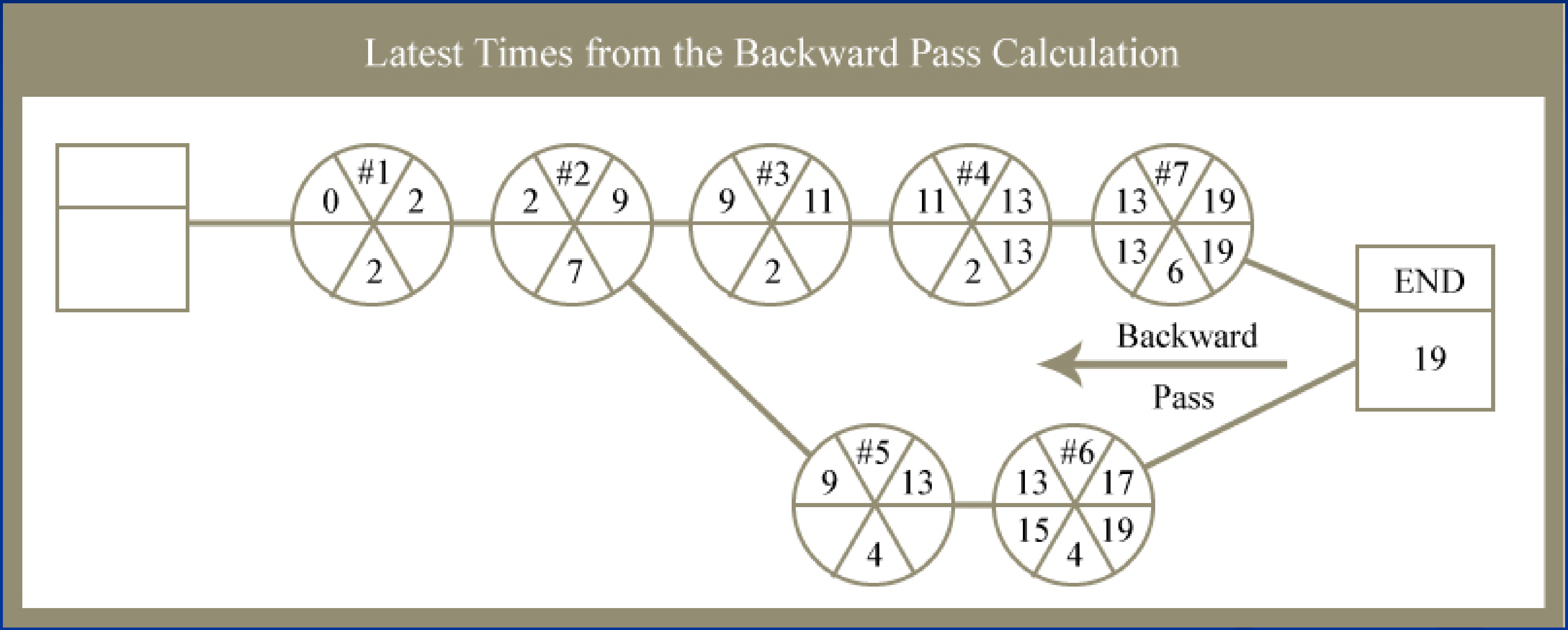
Example Of CPM Algorithm



Forward Pass



Backwards Pass

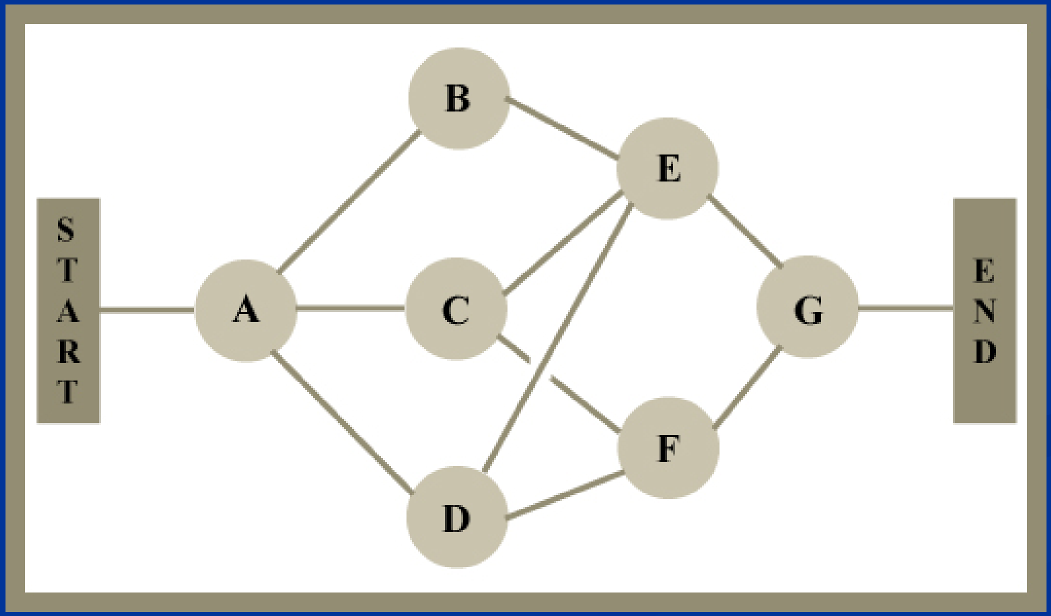


Recall: PDM Relationships

- PDM Extends CPM to include
  - Multiple relationships (SS, SF, FF) beyond FS
  - Lags (negative as "leads")
- Consider relationship XY with lag t between activities A and B
  - $X, Y \in \{ S, F \}, \; t \in \Re$
  - Interpretation is that event Y of activity B can occur *no earlier than* t units after evnt X occurs for activity A
  - Think of relationships as linking *events*
- Special relationships not needed in AOA
  - Can be placed directly between nodes

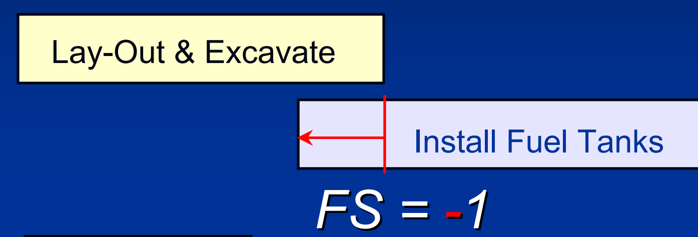
Notation

- Nodes are no longer simply vertices in graph
  - Arrow on left side of node indicates a start relationship
  - Arrow on right side of node indicates finish relationship
- Non-planar networks may require “jumps”

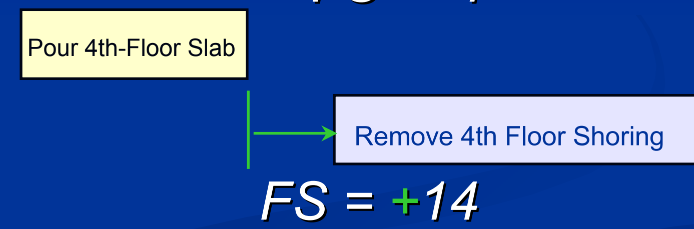


PDM Activity Relationships

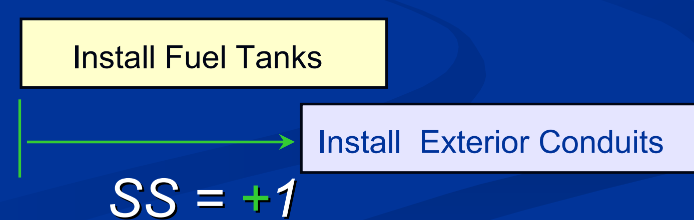
Finish-to-Start Lead



Finish-to-Start Lag



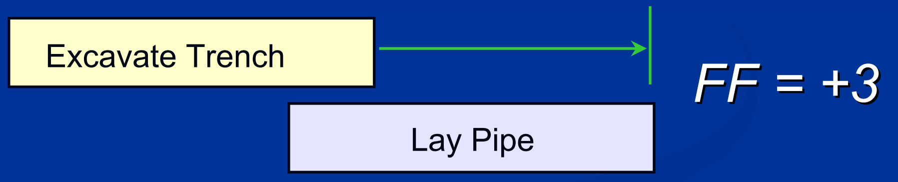
Start-to-Start Lag



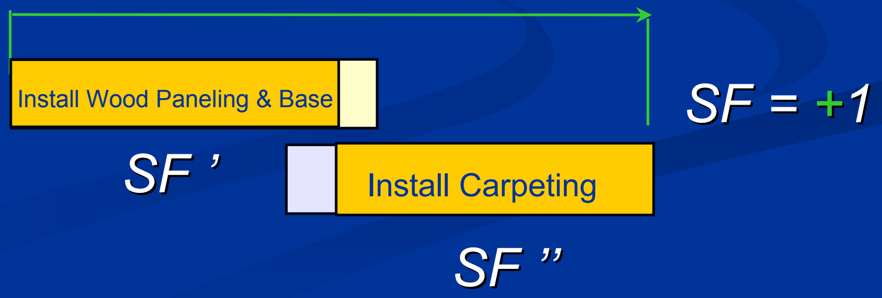
Partially Adapted from Kellegeiros, 2003

PDM Activity Relationships (Cont'd)

Finish-to-Finish Lag



Start-to-Finish Lag



Partially Adapted from Kellegeiros, 2003

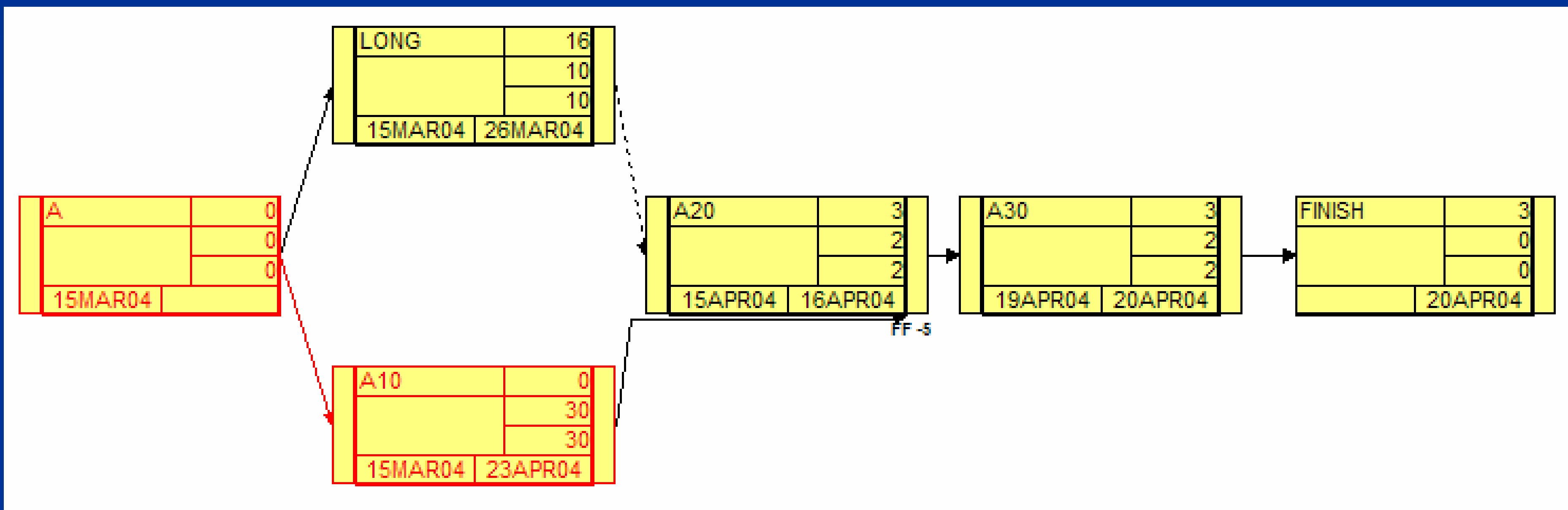
PDM Caveats

- Can have different semantics, but same result
- Asymmetries complicate reasoning
- Make sure you understand the meaning of relationships – for the software you use!
- "Lag" and "Lead" lack standard definition
- May have different floats for same activity
  - Start float (LS-ES)
  - Finish float (LF-EF)
  - Arises from successors for these events

PDM Caveats II– Critical Path

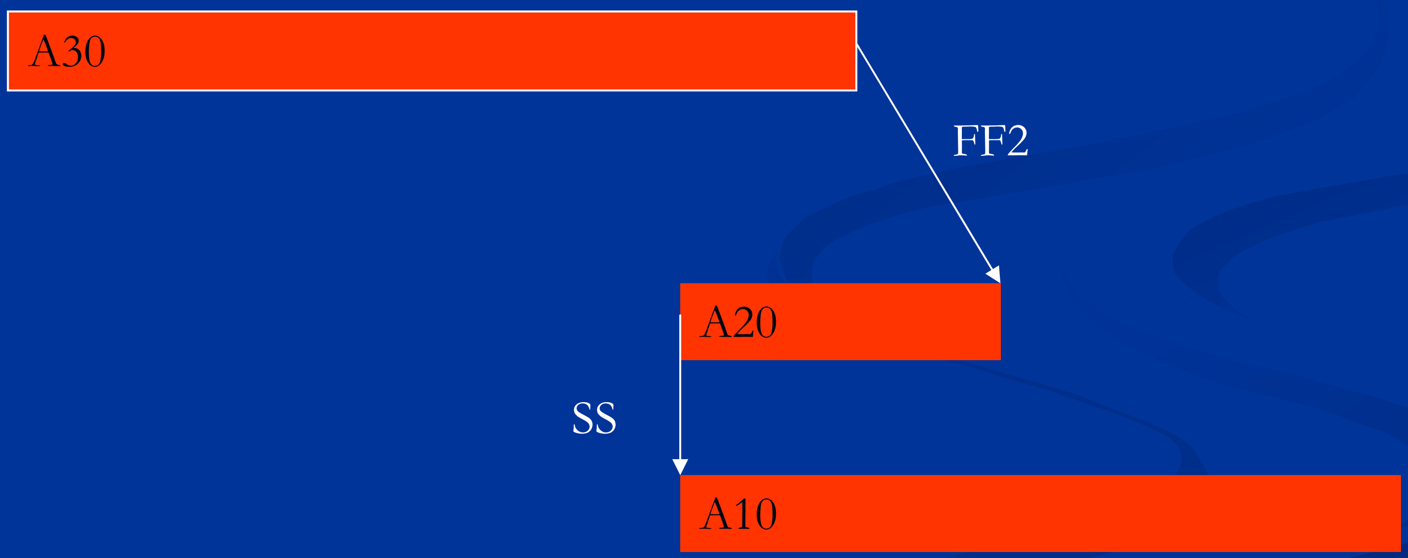
- Choices impact critical path!
  - E.g. Finish-to-start vs. Start-to-start
  - Think of critical path as running through *events*
- Tracing critical path can be difficult
  - Non-critical activity can have critical start/finish
- w/o splitting, can be counter-intuitive (longer duration leads to shorter critical path!)
- Finish-finish constraints with leads can lead to "vanishing" critical path
- How critical path displayed depends on software

"Vanishing Critical Path"

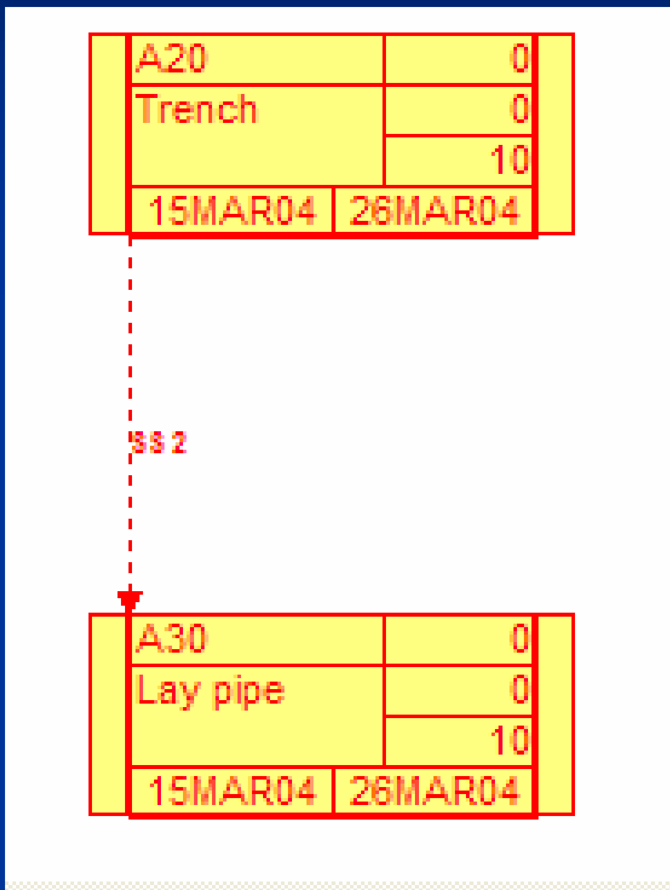


Example of Counter-Intuitive

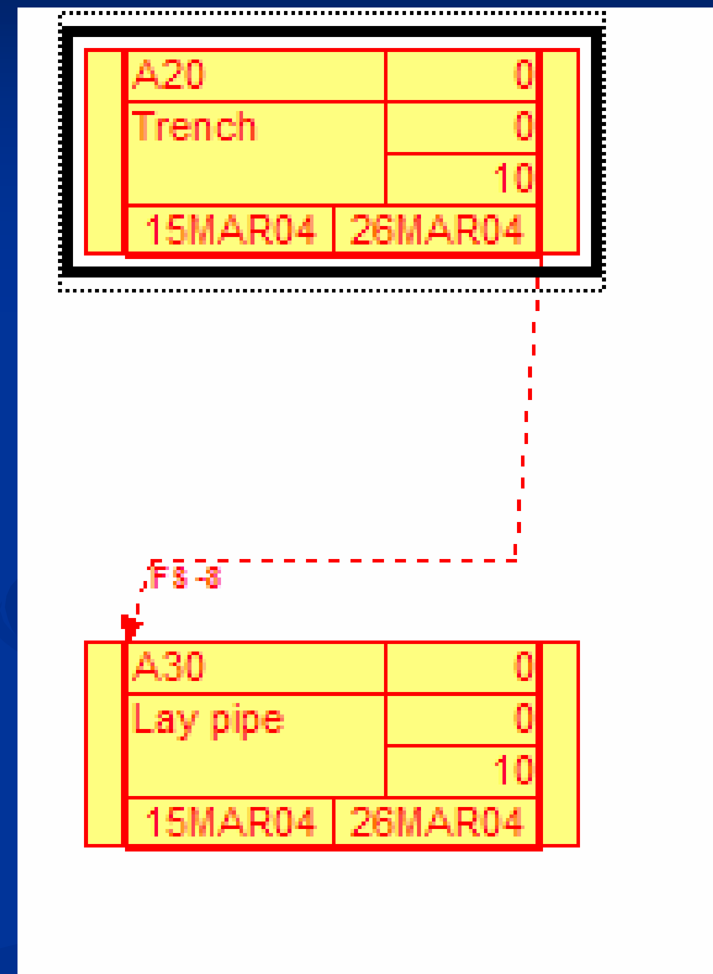
The longer A20 is, the smaller the critical path duration – and quicker can complete!



Equivalent Timing Results



Vs.

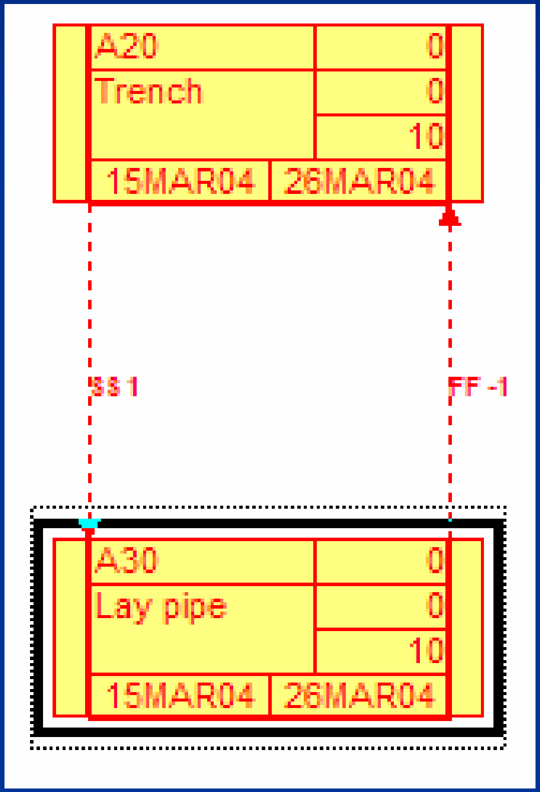


Meaning is different
Critical path may be different

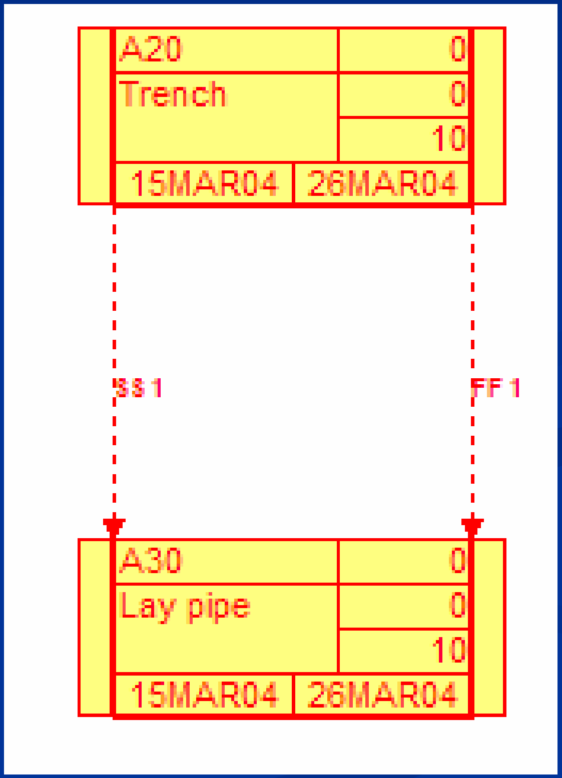
Reasoning about Relationships

- Key Point: PDM relationships often represent relationships between particular parts of an activity. Think about
  - On what *portion* of an activity the other activity depends
  - On how dependency would change if target activity duration changed
- If unclear, think about unbundling activity

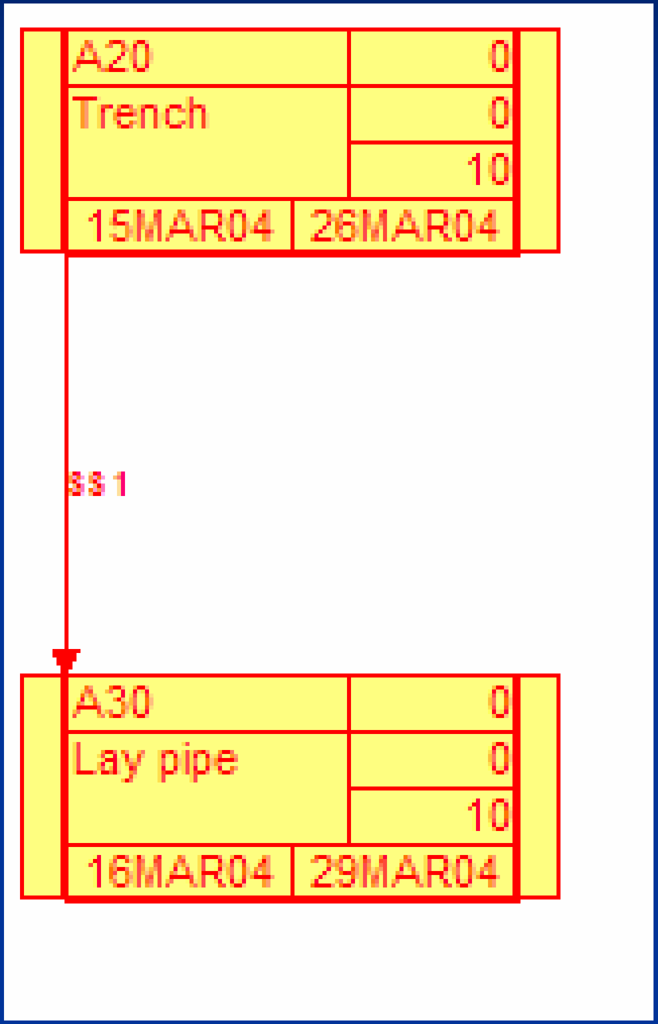
Multiple Relationships



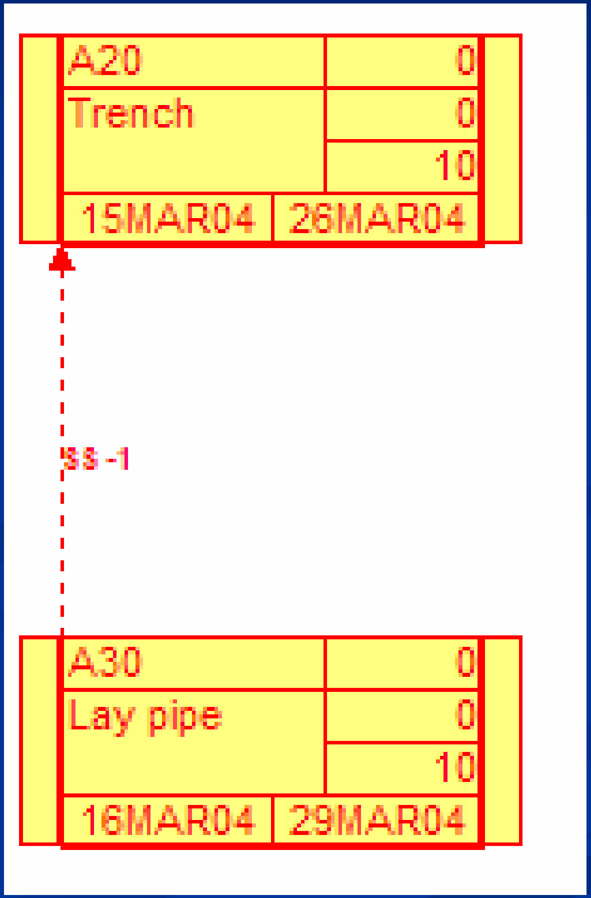
Vs.



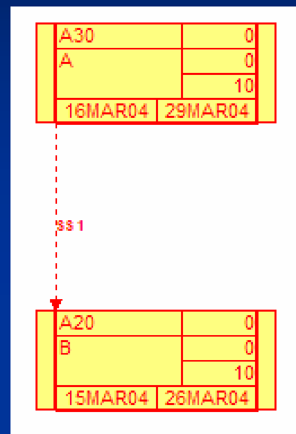
Asymmetries



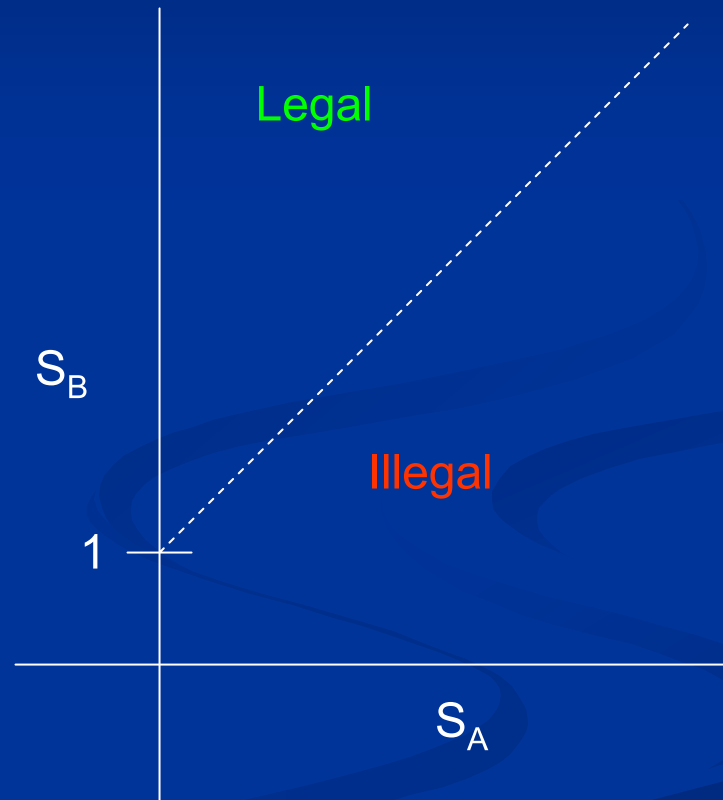
Vs.



Bases for Formal Analysis

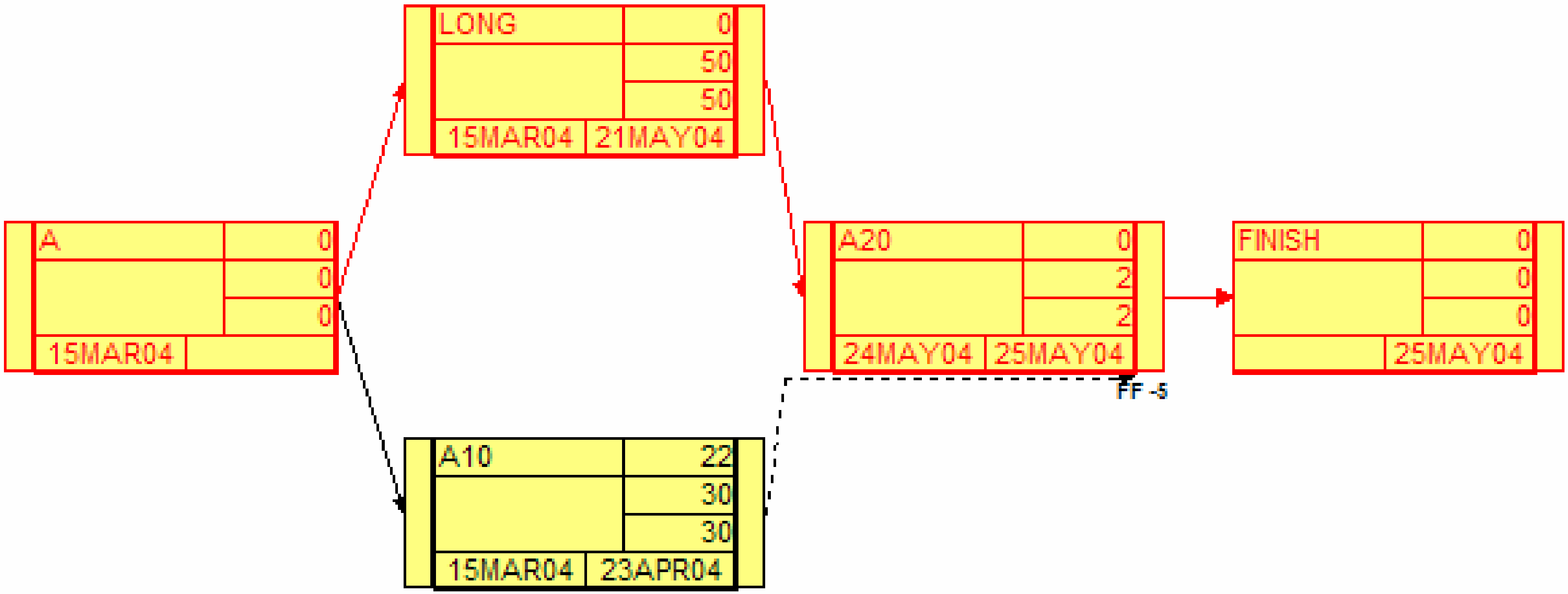


Method 1

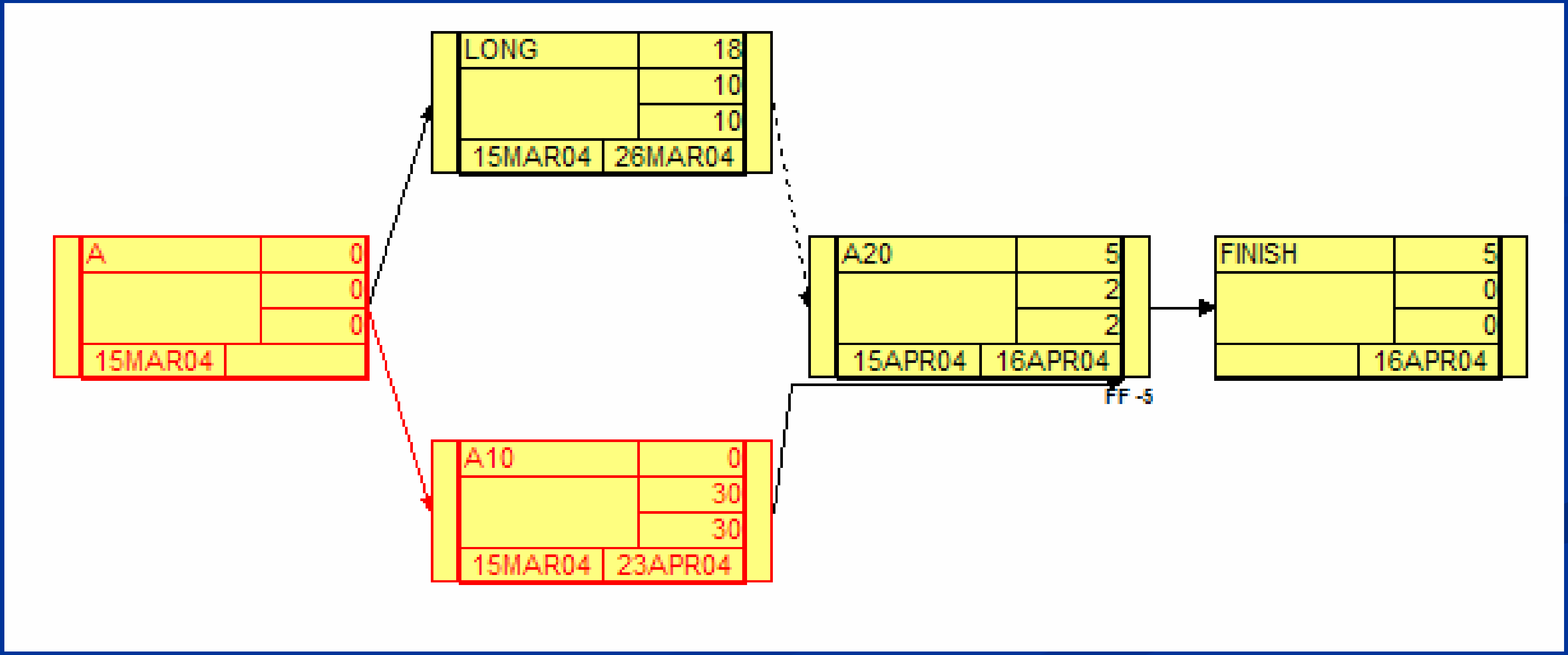


Method 2

Distinguishing F-F Interpretations Non-Binding; A10 Time Unaffected



Distinguishing F-F Interpretations Binding; A20 Waits for A10

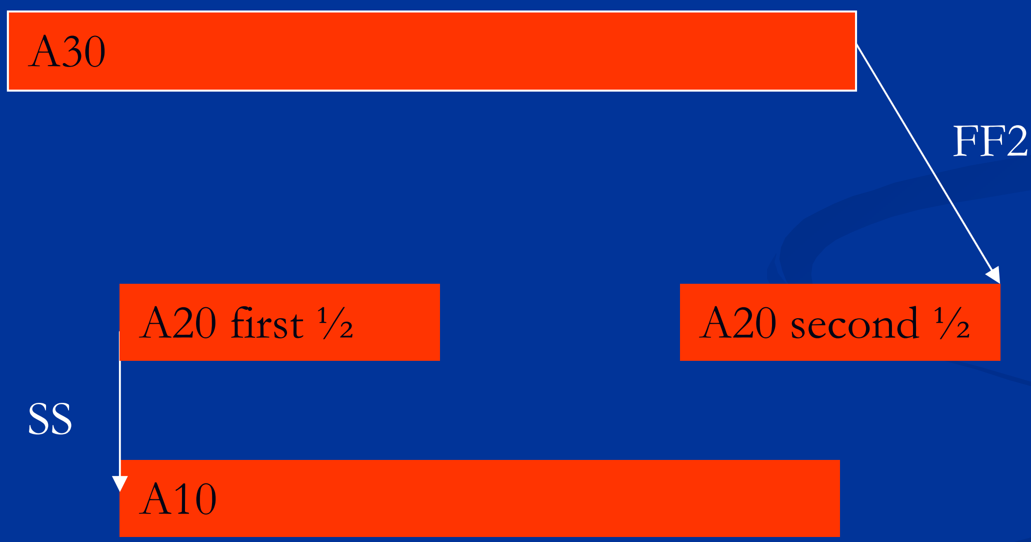


Activity Splitting I: Non-Sequential

- Some algorithms allow division of an activity into two non-sequential pieces
- Advantages: Allows more flexibility in time, resource demands
  - Permits shorter critical paths
  - Eliminates counter-intuitive cases where prefer longer activity
  - Allows predecessor activities connected via SF and SS relationships to begin
  - Allows successor activities connected via SF or FF relationships to begin bulk of work early, and then just wait for event to finish

Example of Counter-Intuitive

The longer A20 is, the smaller the critical path duration – and quicker can complete!

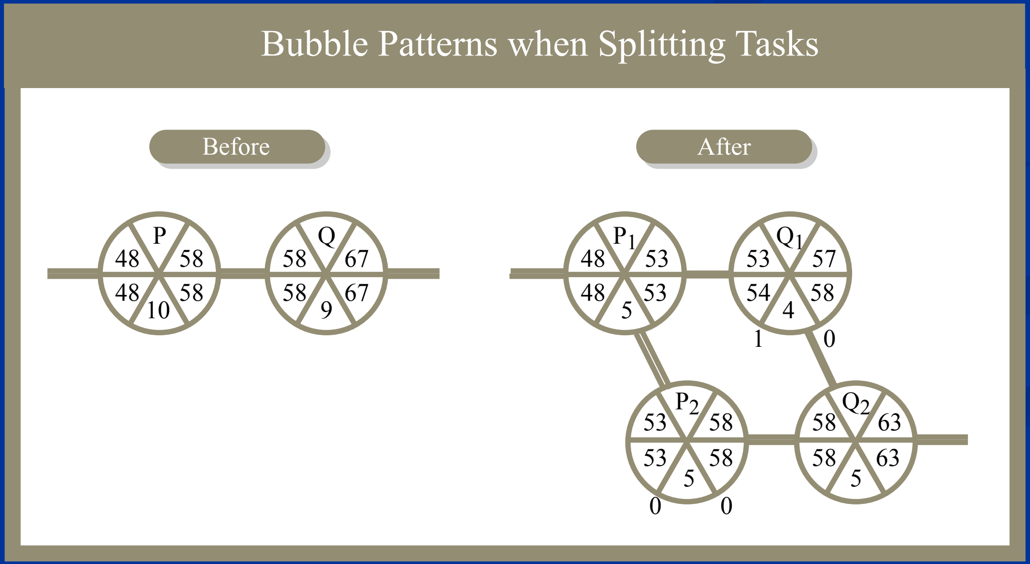


Example

- Because of executive offices, can't *finish* carpeting until wood panelling starts
- Problem: Want carpenters for other work
- Answer: Split wood paneling
  - Do all carpeting except executive offices
  - Allow carpenters to work on executive offices
  - Finish carpeting work for the executive offices
  - Carpenters back to finish job once available

Activity Splitting 2: Pipelining

- Turns monolithic tasks into sub-tasks that operate in parallel
- Typically increases resource demand
- Typically done manually (generally not enough information to permit automation)
- Often represent with S-S constraint



Activity Windows

- Mechanism for imposing time constraints on absolute activity times
- Can impose constraint for any of times
  - ES, EF, LS, LF
  - By set WES=WLS, fix exact timing
- Particularly useful for time-critical milestones

Forward Pass for node k (no splits; no leads)

$$ES_k = \text{Max}_{\text{all } p} \begin{cases} INITIAL\ TIME \\ WES_k \\ WEF_k - D_k \\ EF_p + FS_{pk} \\ ES_p + SS_{pk} \\ EF_p + FF_{pk} - D_k \\ ES_p + SF_{pk} - D_k \end{cases} \qquad EF_k = ES_k + D_k$$

Moder

Key factor: Cannot start until all predecessors ready! Must take maximum of predecessors' values

Backward Pass (node k, no splitting; no leads)

$$LF_k = \text{Min}_{\text{all } s} \begin{cases} TERMINAL\ TIME \\ WLF_k \\ WLS_k + D_k \\ LS_s - FS_{ks} \\ LF_s - FF_{ks} \\ LS_s - SS_{ks} + D_k \\ LF_s - SF_{ks} + D_k \end{cases} , \quad LS_k = LF_k - D_k$$

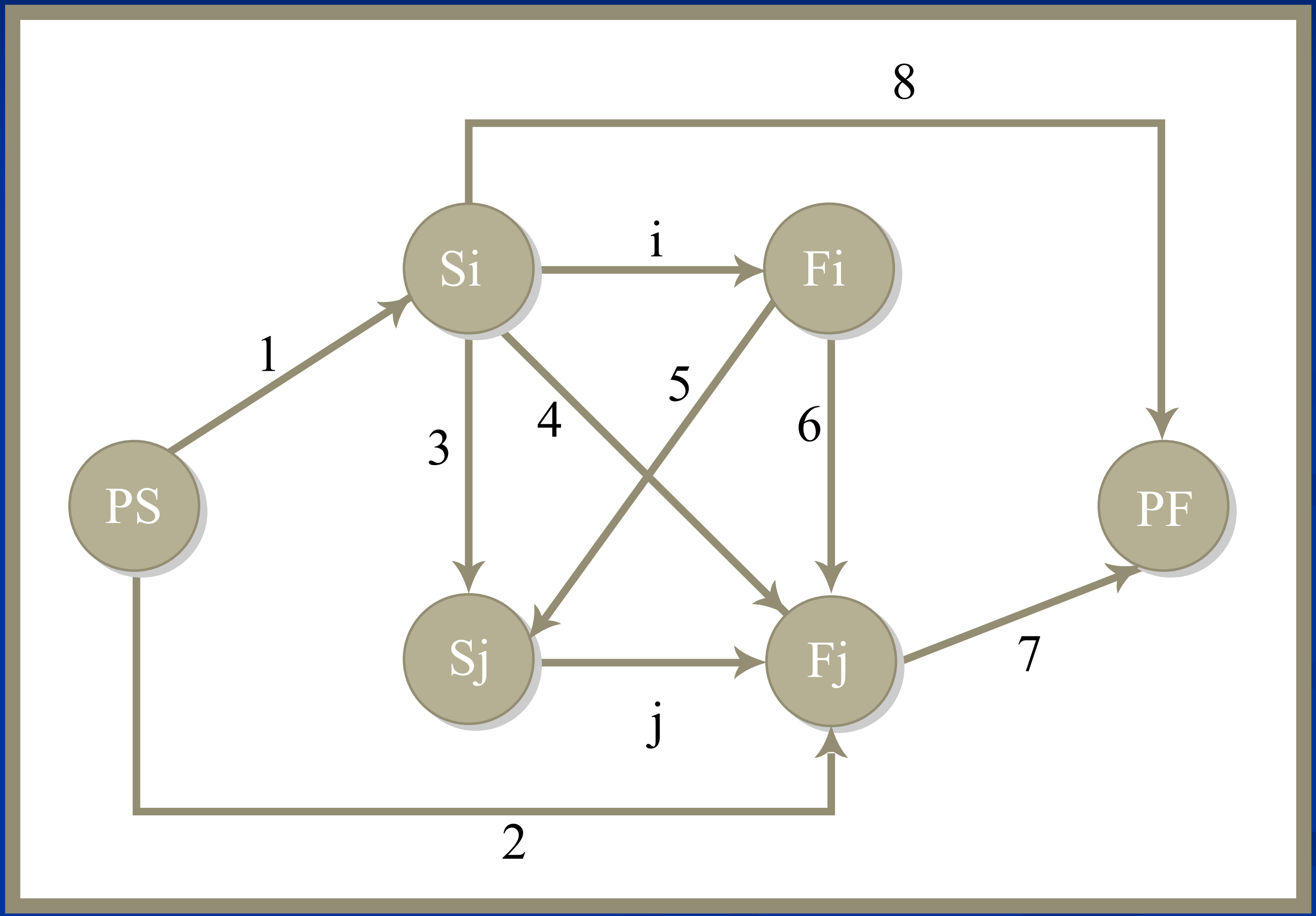
Moder

Key factor: Must finish in time for *all successors* to start in time!
Otherwise will delay project completion time
=> Must take min of successors' values.

Dealing with Leads

- Dealing with *leads* ("negative lags") requires more general algorithm
  - Two O(n) passes may no longer be sufficient
- Be careful about meaning!
- Basic approach: Convert AON into AOA-like form
  - Start/Finish Nodes explicit for every activity
  - Very helpful for thinking through meaning
- Use Dijkstra's algorithm to solve O(VlgV+E)

Example Translated Diagram



Unified Algorithm

Calculations for the Unified Network Model with Negative Link Durations

Forward Pass:

Step 1: Set $PL(i) = -\infty$ and $TL(i) = -\infty$, where ∞ is a number larger than any link duration.

Set $TL(PS) = 0$

where PS = the project start node
$PL(i)$ = the maximim distance from PS to node i
$TL(i)$ = the maximim distance from PS to node i found at intermediate stages

Step 2: Select node i for which $TL(i)$ is the maximum among all nodes.

Set $PL(i) = TL(i)$ and $TL(i) = -\infty$

For each link originating at node i,
If $PL(j) = -\infty$ and $PL(i) + D(i, j) > TL(j)$,
then set $TL(j) = PL(i) + D(i, j)$.

If $PL(j) = -\infty$ and $PL(i) + D(i, j) \leq TL(j)$,
then do not change the labels on j.

If $PL(j) > -\infty$ and $PL(i) + D(i, j) > PL(j)$,
then set $PL(j) = -\infty$ and $TL(j) = PL(i) + D(i, j)$.

If $PL(j) > -\infty$ and $PL(i) + D(i, j) > PL(j)$,
then do not change the label on j.

Step 3: Repeat step 2 until $PL(PF) > -\infty$, where PF is the project finish node.

Step 4: Set the earliest event time for each node, $E(i) = PL(i)$.

Backward Pass:

Repeat application of the algorithm with the following changes:

1. Reverse each link direction.
2. Start with the project finish node PF with $TL(PF) = 0$.
3. At the end of step 3, set the latest event time, $L(i) = E(PF) - PL(i)$ for all nodes i.

Motivations for Dealing with Uncertainty

- Schedules exhibit much uncertainty
  - Weather occurrences, design duration, productivity, delivery times, subcontractor quality, regulatory changes, etc…
- Clients, community may want to know milestones, finish date with confidence
  - Tenant move-in dates
  - Traffic planning
  - Event planning
- Reasoning about schedule constraints such as weather, seasonal traffic
- Extensions may be much worse than early completion

Case 1: Logan International Terminal

- Firm date required by
  - Vendors
  - Airlines
- Sought probabilistic scheduling to quantify uncertainty

Case 2: Philadelphia Children's Hospital

- Described in "Modern Steel Construction" article (posted on STELLAR site)
- Accounting for
  - Emergency helicopter usage
  - Patient area activity
  - Limited time windows for work (2-4am)
  - Contingencies regarding telephone switch relocation

Informal Ways of Handling Uncertainty

- Most common: Ignore!
  - Assume expected duration
  - Hope errors cancel
- Apply contingency factors
- "What if" scenario analysis to examine
  - Optimistic scenario
  - Most likely scenario
  - Pessimistic scenario

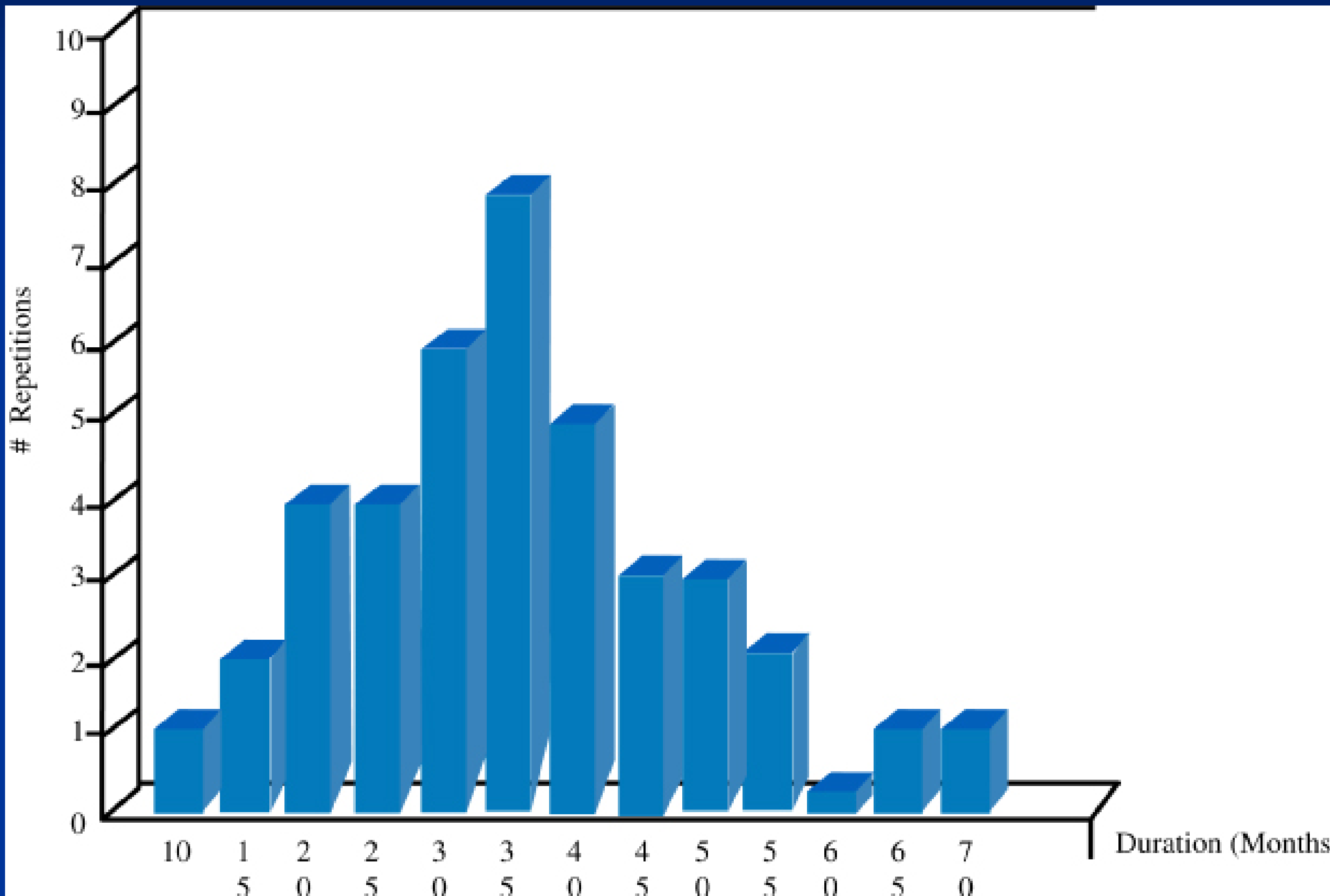
Program Evaluation and Review Technique (PERT)

- Developed by US Navy, Booz-Allen Hamilton and Lockheed Corporation
  - Polaris Missile/Submarine (1958)
- Captures probabilistic activity durations
- Allows analytic solution for
  - Schedule duration
  - Schedule variance

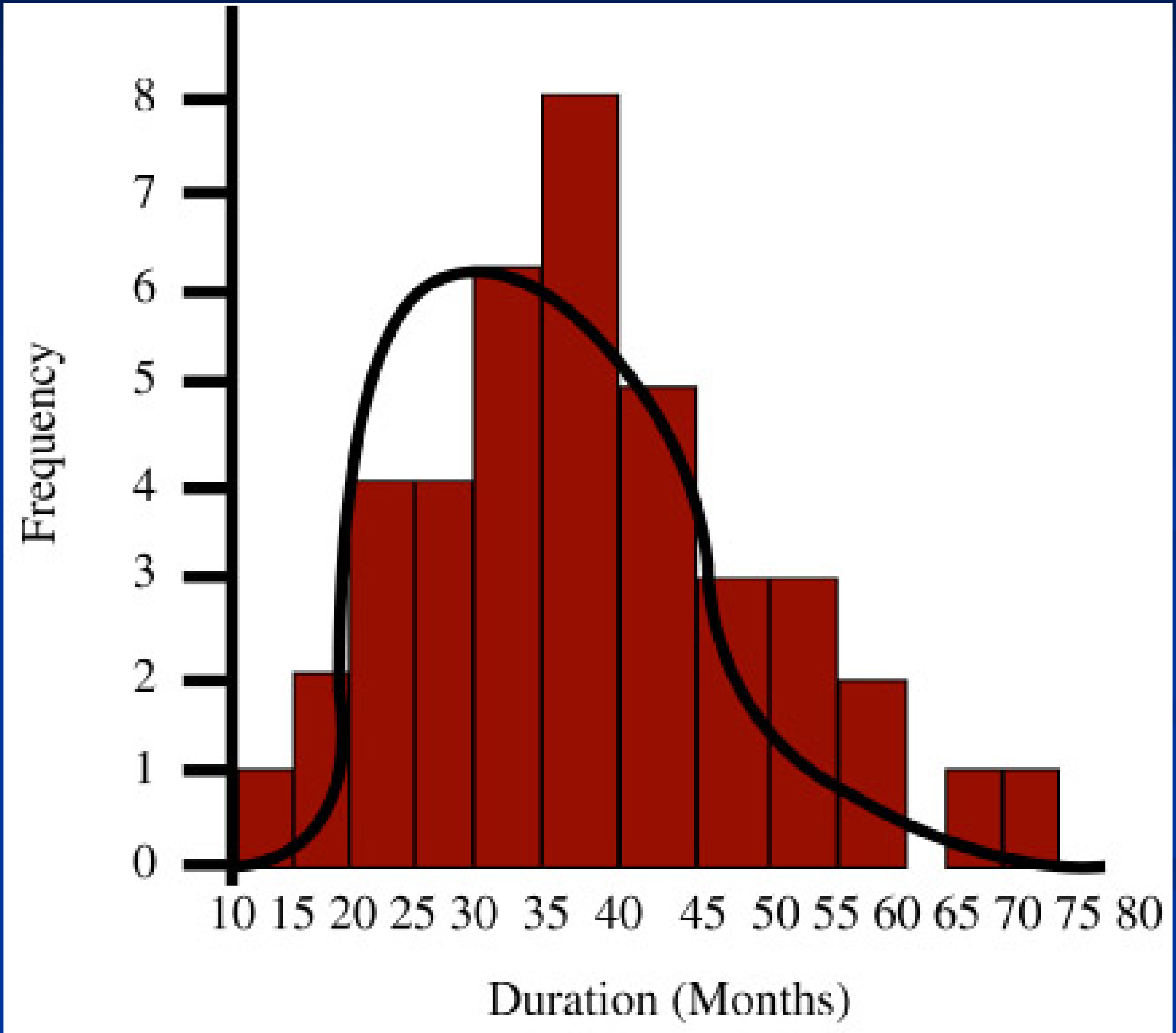
PERT Basics

- Beta Distribution for Activity Duration
- Assume normally distributed project duration
  - Project Duration Tends to be Normally Distributed (approx. sum of random variables)
  - Assumes Independent Activity Durations - Not Always Satisfied

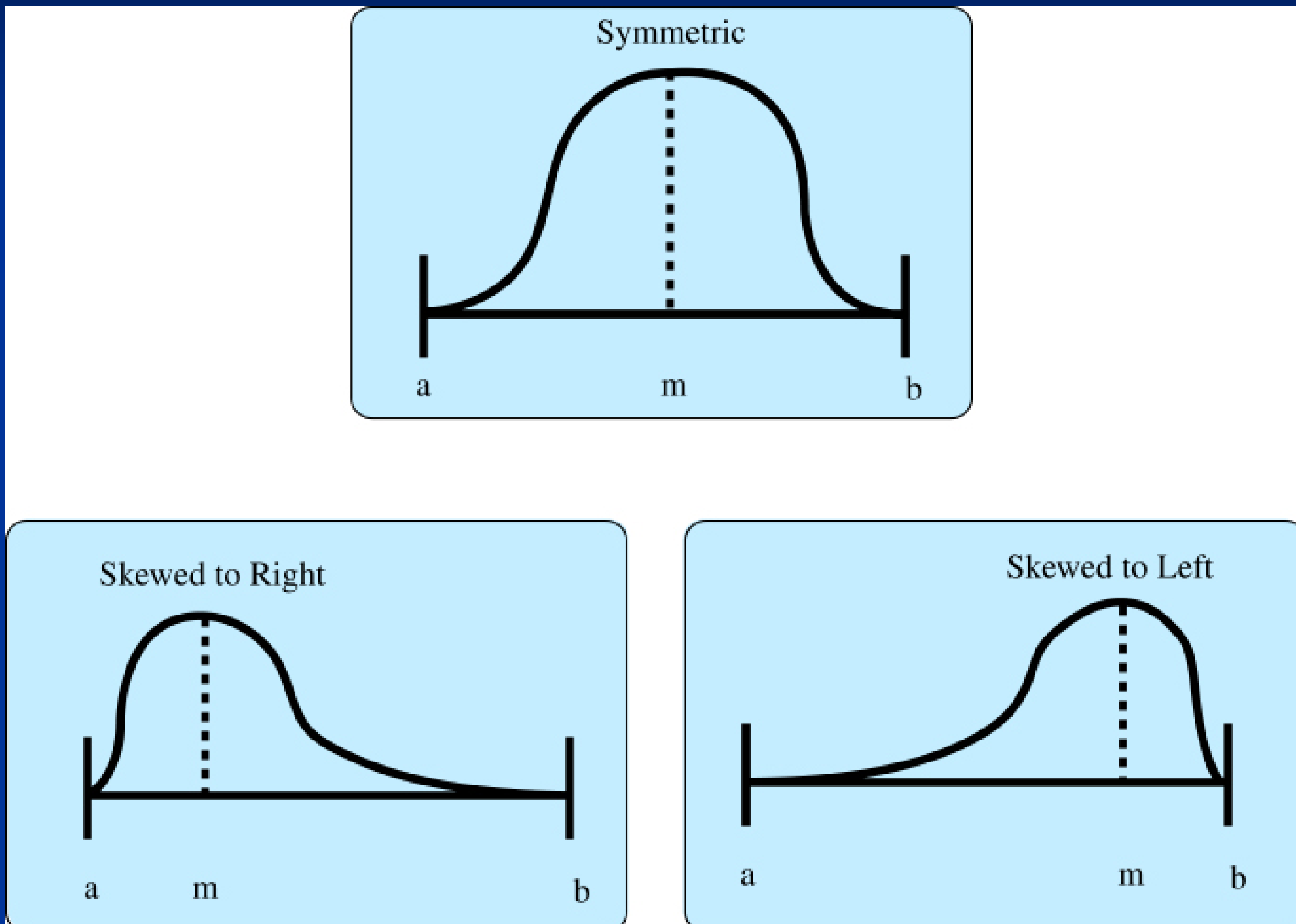
Activity Duration Frequency



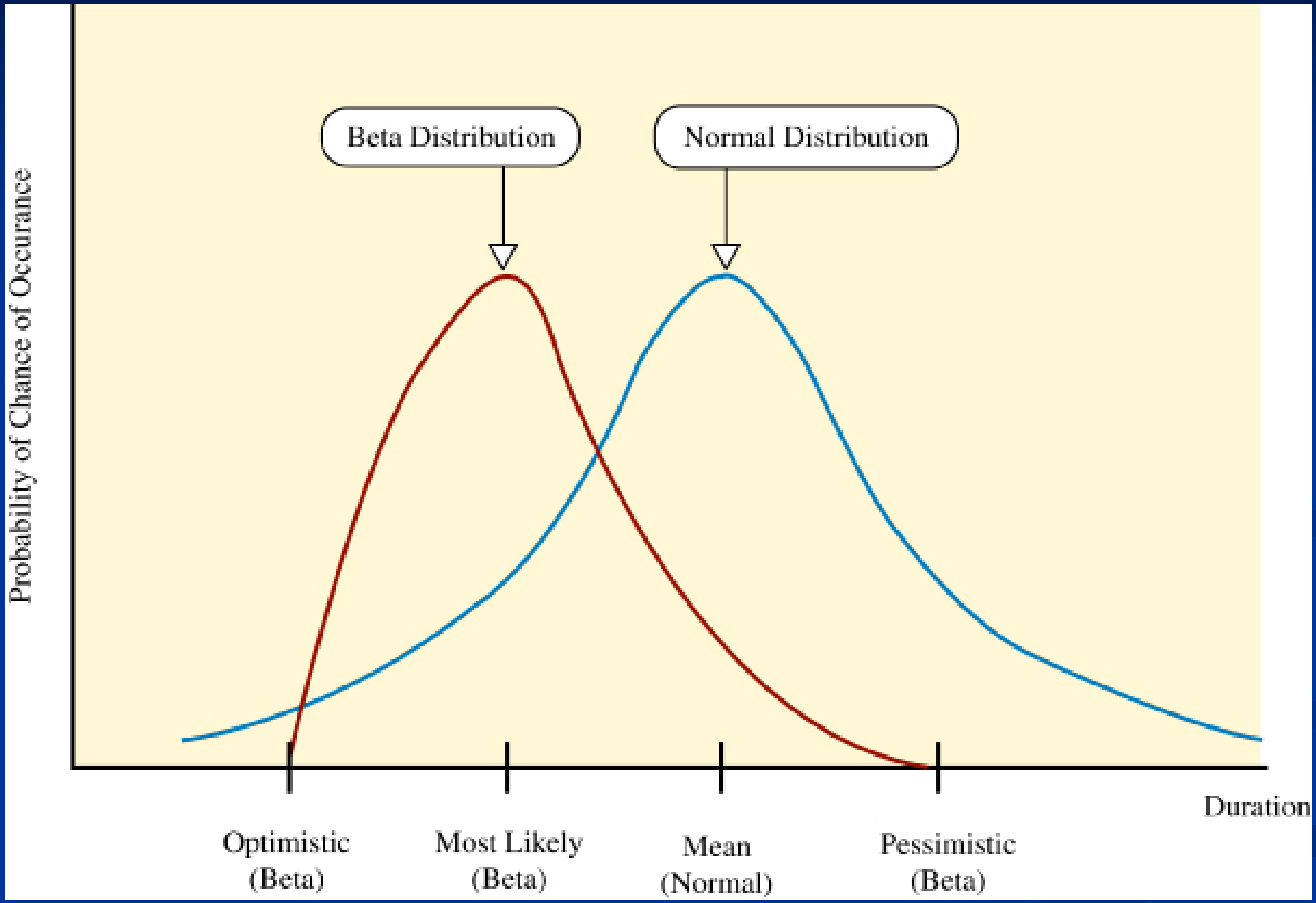
Beta Distribution



Three Cases of Beta Distribution

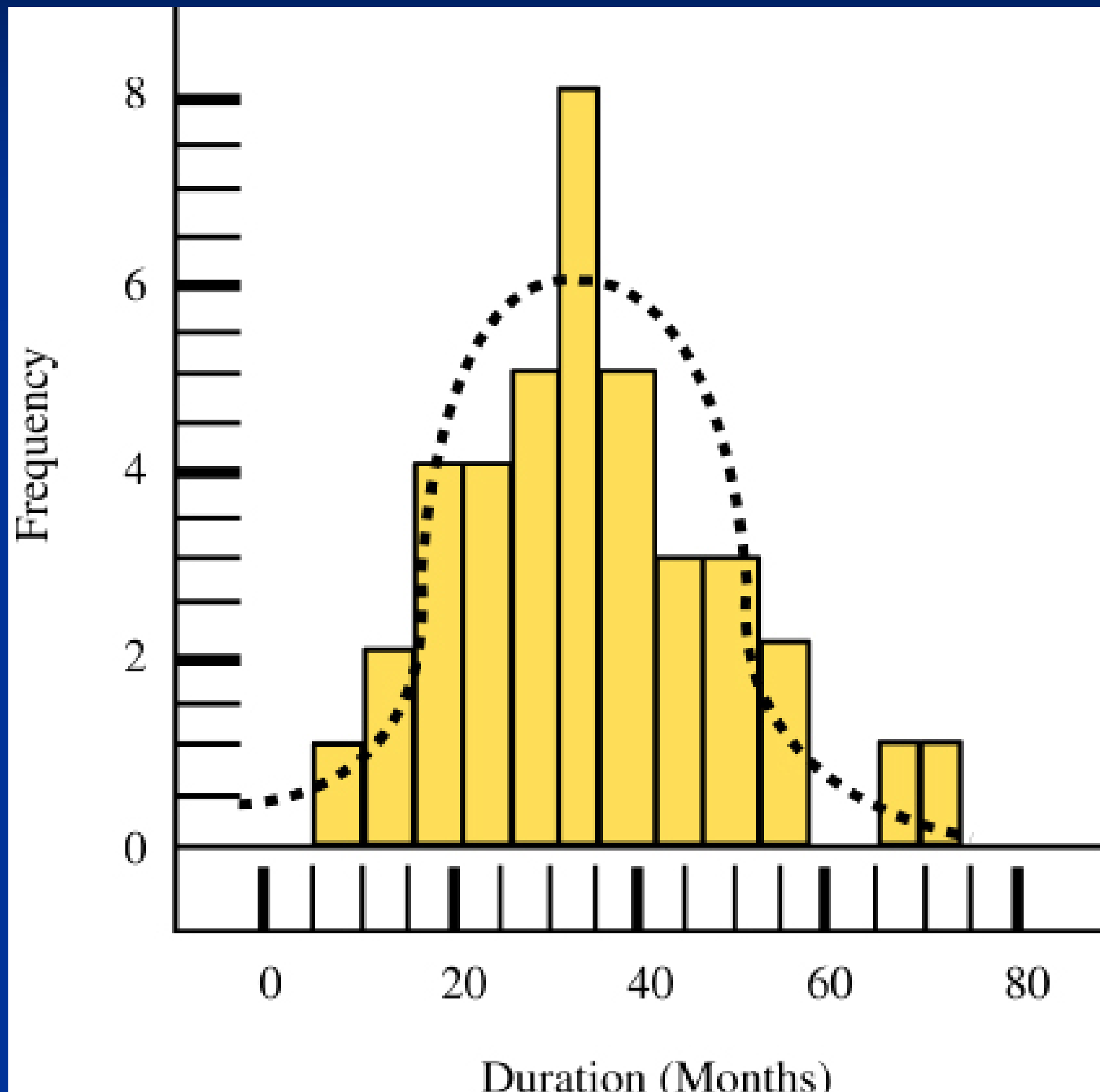


Beta vs. Normal



Can guarantee Beta non-negative

Normal Distribution Assumed for Schedule



Stochastic Approach

- Optimistic — a
- *Most Likely (mode – not mean)* — m
- Pessimistic — b
- Expected Duration — $\bar{d} = \frac{1}{3}\left[2m + \frac{1}{2}(a+b)\right] = \frac{a+4m+b}{6}$
- Variance — $v = s^2$
- Standard Deviation — $s = \frac{b-a}{6}$

Steps in PERT Analysis

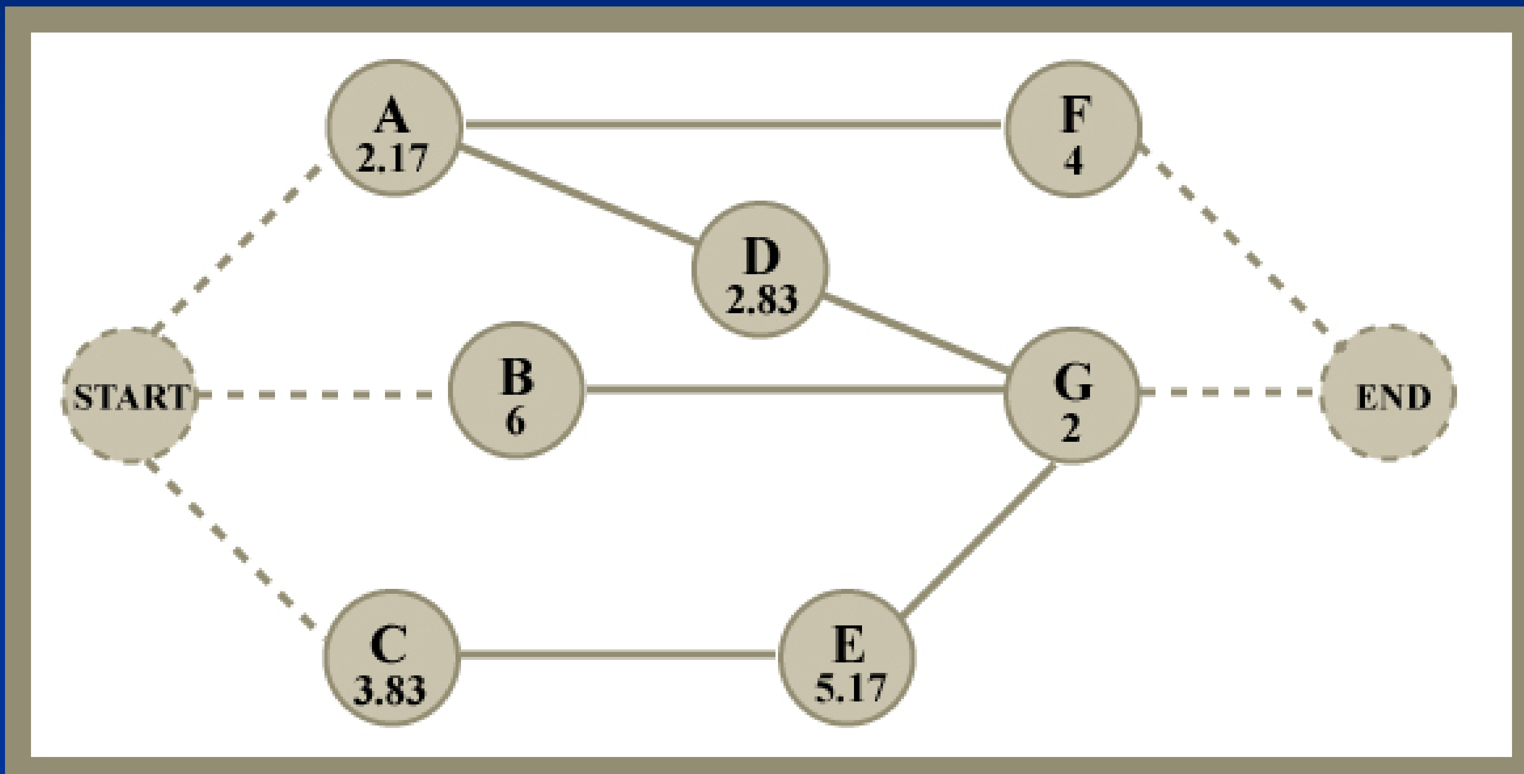
- For each activity k
  - Obtain a_k, m_k (mode) and b_k
  - Compute expected activity duration (mean) $d_k = t_e$
  - Compute activity variance $v_k = s^2$
- Compute expected project duration $D = T_e$ using standard CPM algorithm
- Compute Project Variance $V = S^2$ as sum of critical path activity variance (*this assumes independence!*)
  - In case of multiple critical paths use the one with the largest variance
- Calculate probability of completing the project
  - Assuming project duration normally distributed

PERT Example

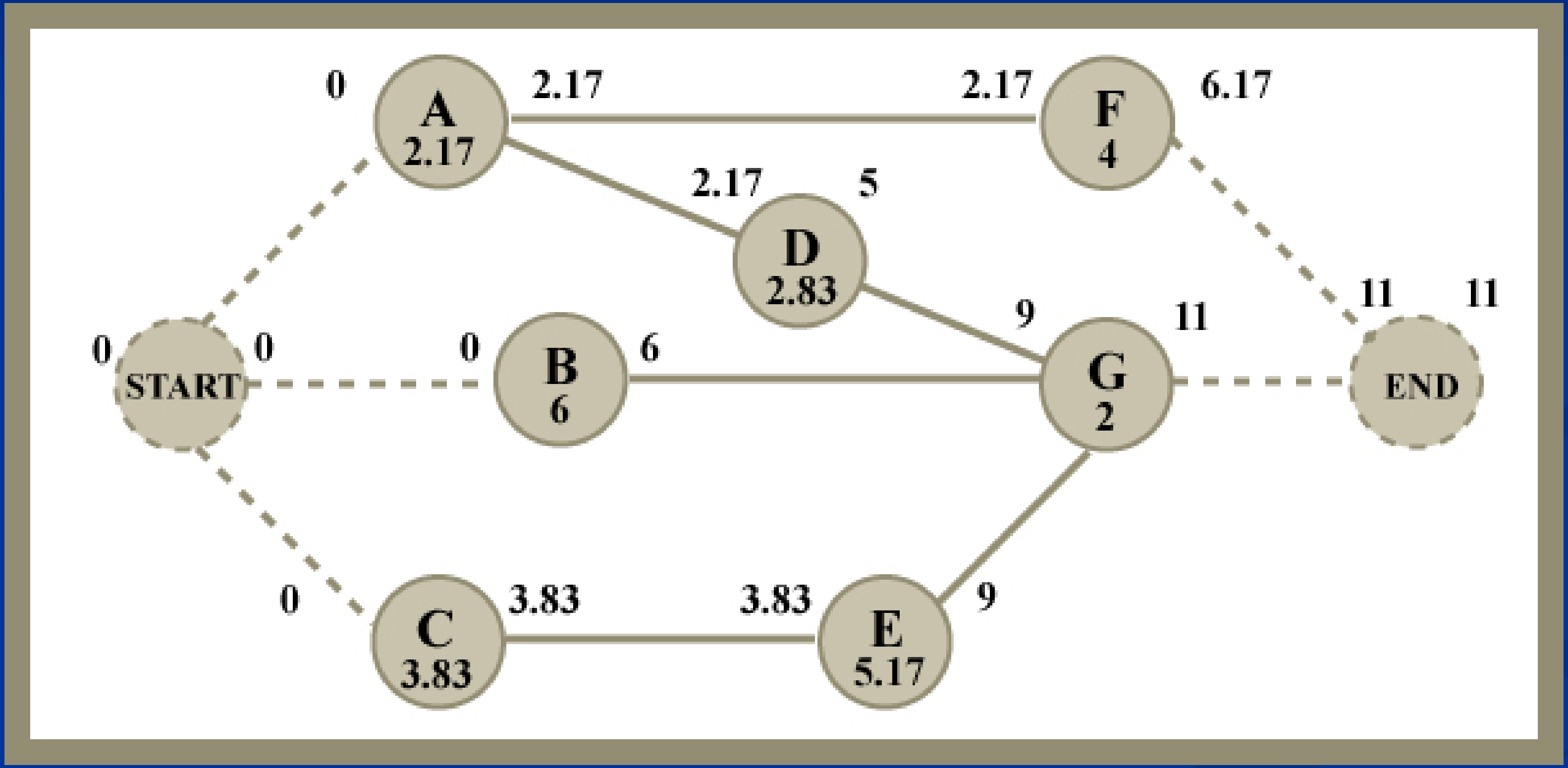
Calculated

Activity	Predecessor	a	m	b	d	v
A	-	1	2	4	2.17	0.25
B	-	5	6	7	6.00	0.11
C	-	2	4	5	3.83	0.25
D	A	1	3	4	2.83	0.25
E	C	4	5	7	5.17	0.25
F	A	3	4	5	4.00	0.11
G	B,D,E	1	2	3	2.00	0.11

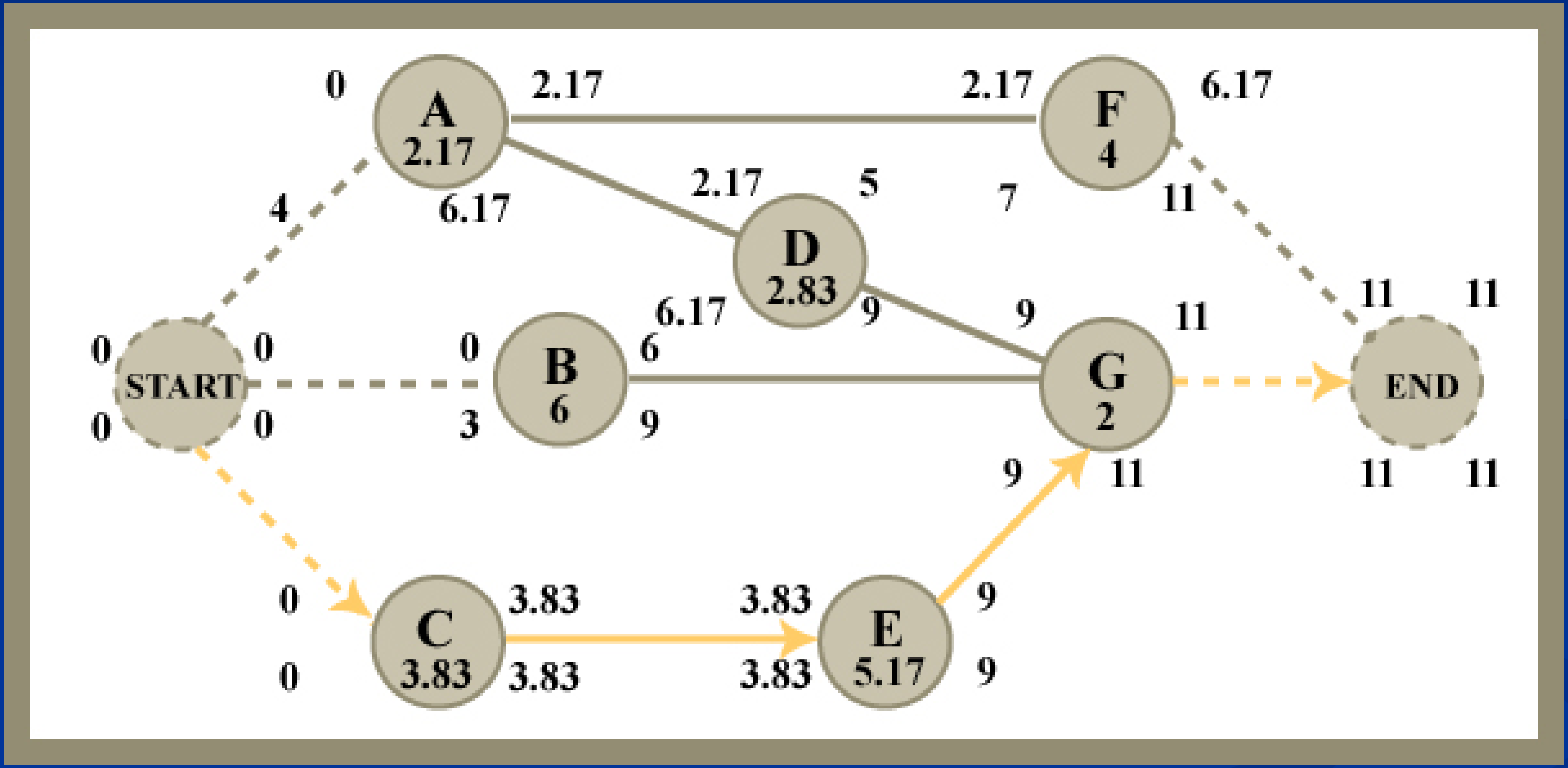
Activity on Node Example



Forward Pass



Backward Pass



PERT Example-Standard Deviation

$$T_e = 11$$

$$S^2 = V[C] + V[E] + V[G]$$

$$= 0.25 + 0.25 + 0.1111$$

$$= 0.6111$$

$$S = \sqrt{0.6111}$$

$$= 0.7817$$

PERT Analysis-Probability of Ending before 10 (Critical Path Only)

$$
\begin{aligned}
P\left(T \le T_d\right) &= P(T \le 10) \\
&= P\left(z \le \frac{10 - T_e}{S}\right) \\
&= P\left(z \le \frac{10 - 11}{0.7817}\right) \\
&= P\left(z \le -1.2793\right) \\
&= 1 - P\left(z \le 1.2793\right) \\
&= 1 - 0.8997 \\
&= 0.1003 \\
&= 10\%
\end{aligned}
$$

PERT Analysis - Probability of Ending before 13 (Critical Path Only)

$$P(T \leq 13) = P\left(z \leq \frac{13-11}{0.7817}\right)$$

$$= P(z \leq 2.5585)$$

$$= 0.9948$$

PERT Analysis - Probability of Ending between 9 and 11.5(CP Only)

$$
\begin{aligned}
P\left(T_L \le T \le T_U\right) &= P(9 \le T \le 15) \\
&= P(T \le 11.5) - P(T \le 9) \\
&= P\left(z \le \frac{11.5-11}{0.7817}\right) - P\left(z \le \frac{9-11}{0.7817}\right) \\
&= P(z \le 0.6396) - P(z \le -2.5585) \\
&= P(z \le 0.6396) - \left[1 - P(z \le 2.5585)\right] \\
&= 0.7389 - [1 - 0.9948] \\
&= 0.7389 - 0.0052 \\
&= 0.7337
\end{aligned}
$$