Estimating Contingencies in Complex Construction Projects: A Framework Based on Constraint Driven Temporal Networks

G. Ryan Anderson

M.Sc. Thesis Defense

Committee members:

- Nilufer Onder (CS, co-chair)
- Amlan Mukherjee (CEE, co-chair)
- Steve Seidel (CS)
- David Poplawski (CS)
- Kris Mattila (CEE)

Outline

- **Introduction**
- Case Study
- TONAE Framework & Algorithms
- Experimental Results
- Conclusions

Introduction

- Construction project management
  - As-planned schedules and estimates
  - Fluctuations due to events
  - Contingency funds set aside to help mitigate problematic scenarios

NY Times Office Building

- Problems during construction:
  - Primary steel subcontractor went bankrupt
  - Complicated specifications warranted tremendous amounts of welding
- Problems resulted in the loss of most of the contingency funds

Two Classes of Problems

Aleatory

- Steel contractor going bankrupt
- Unpredictable problems

Epistemic

- Planning problems (e.g., welding)
- Problems inherent to the project design

Thesis Objectives

- To develop a mechanism for making inferences and predictions about construction management projects
- Allow a construction manager to deal with the inherent uncertainties of such a domain

Outline

- Introduction
- **Case Study**
- TONAE Framework & Algorithms
- Experimental Results
- Conclusions

Structural Steel Case Study

- 6-Sequence Steel Framed Building
  - Hoisting
  - Bolting and Connecting
  - Decking

Hoisting

- Lifting the steel members into place
- Securing them with temporary ties

Bolting and Connecting

- Permanently fastening the steel members together at their junction points

Decking

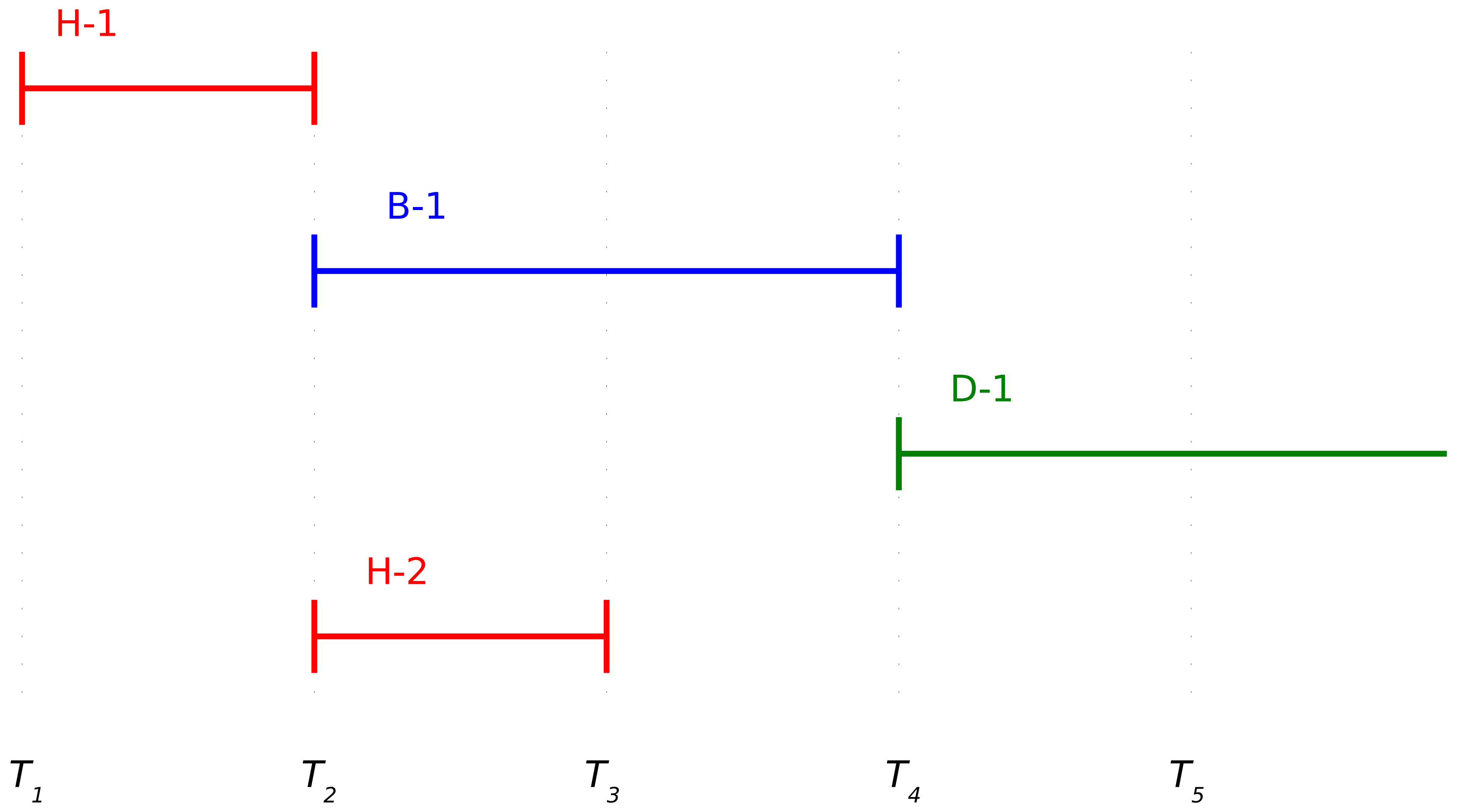
- Fastening the steel decking into place over the beams

After Completion of Sequence 4

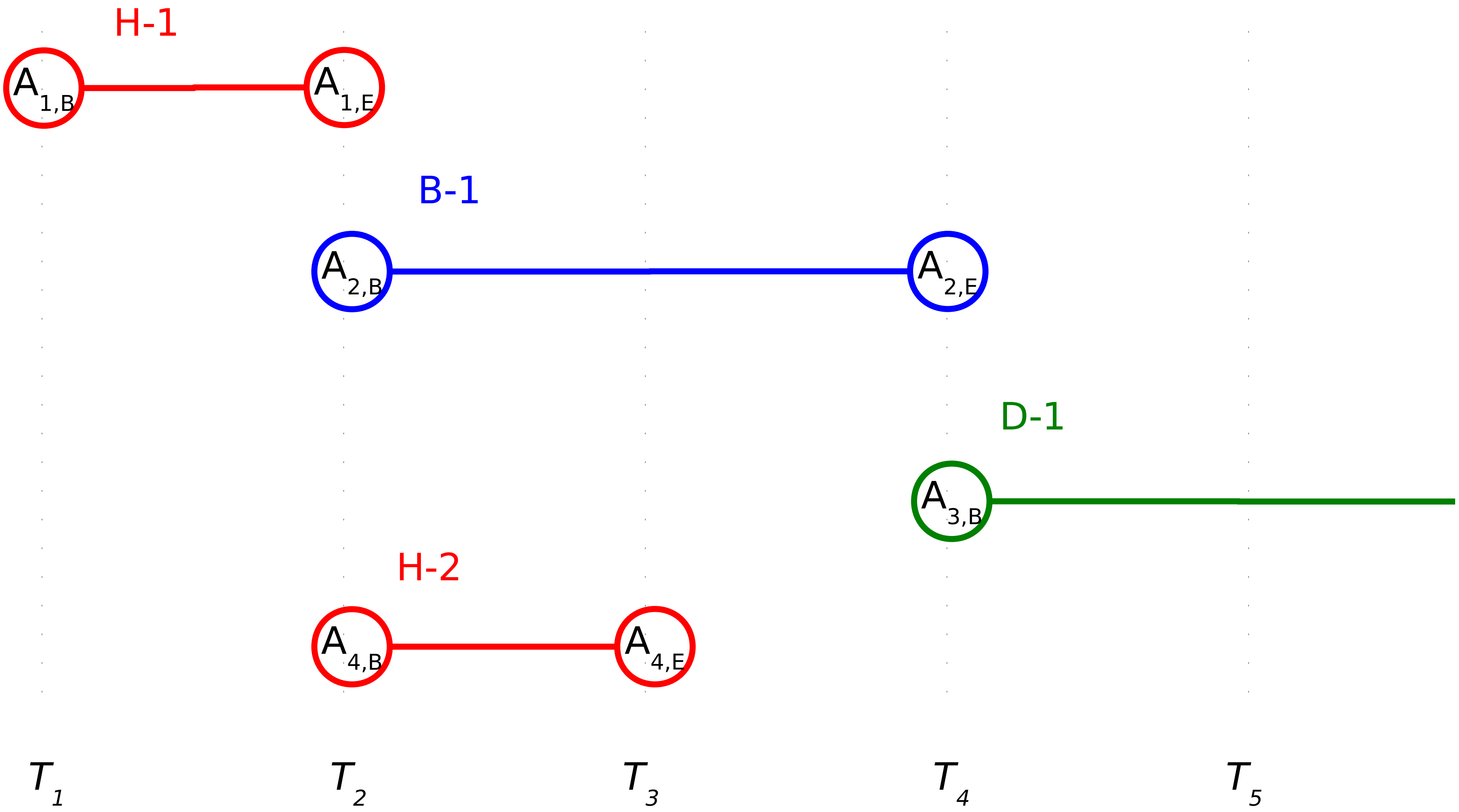
Outline

- Introduction
- Case Study
- **TONAE Framework & Algorithms**
- Experimental Results
- Conclusions

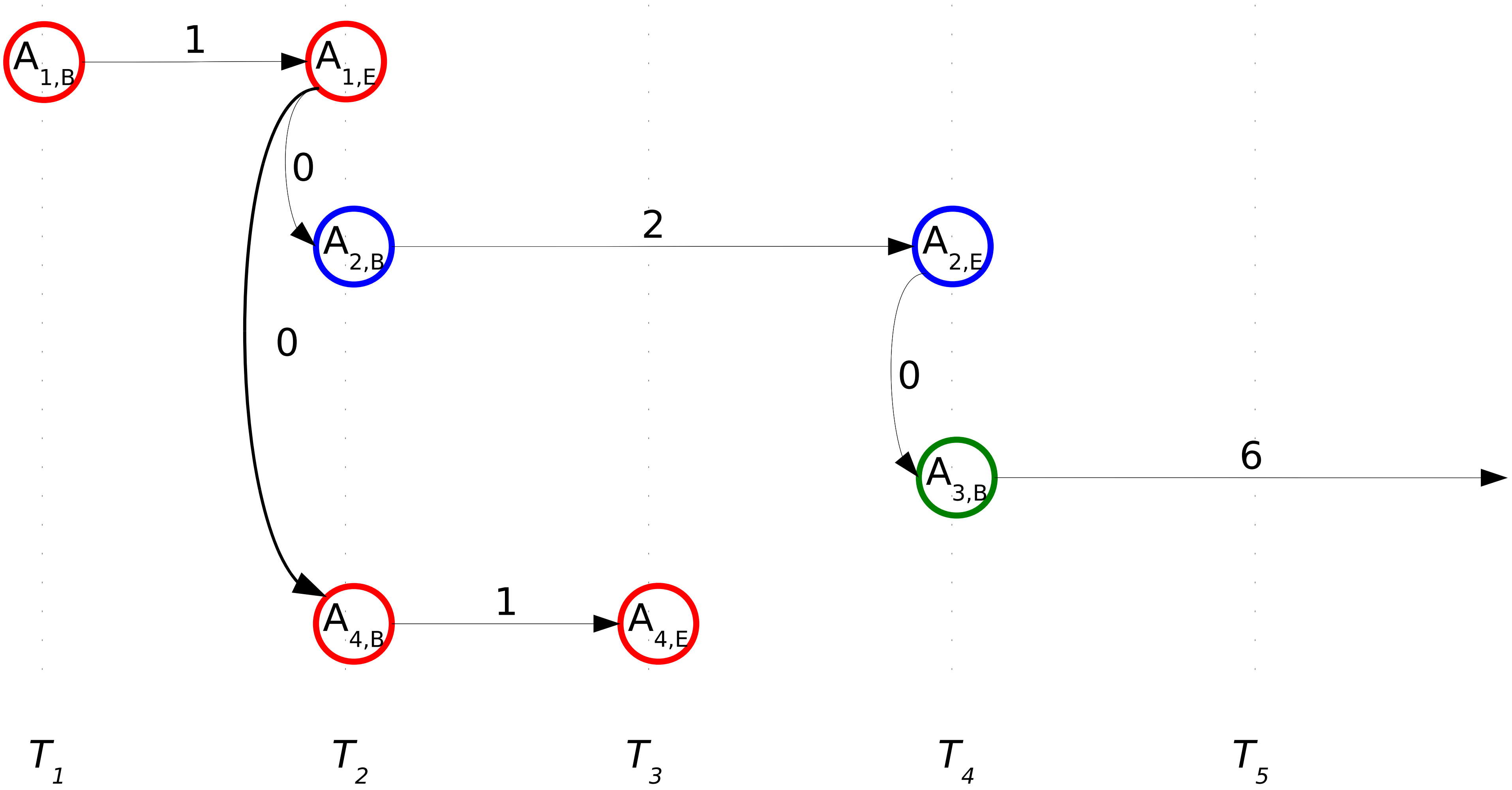
Portion of As-Planned Schedule



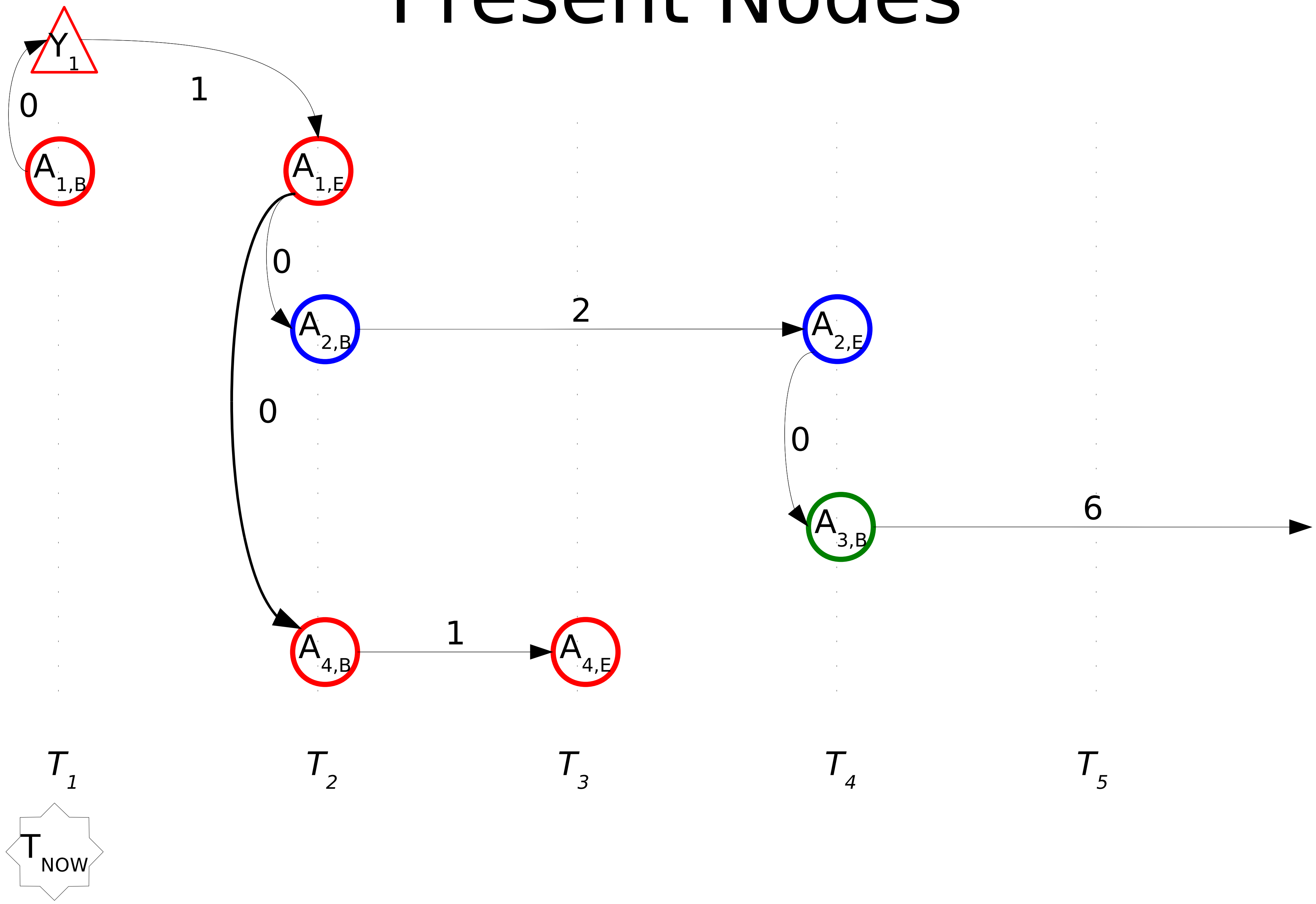
Activity Nodes



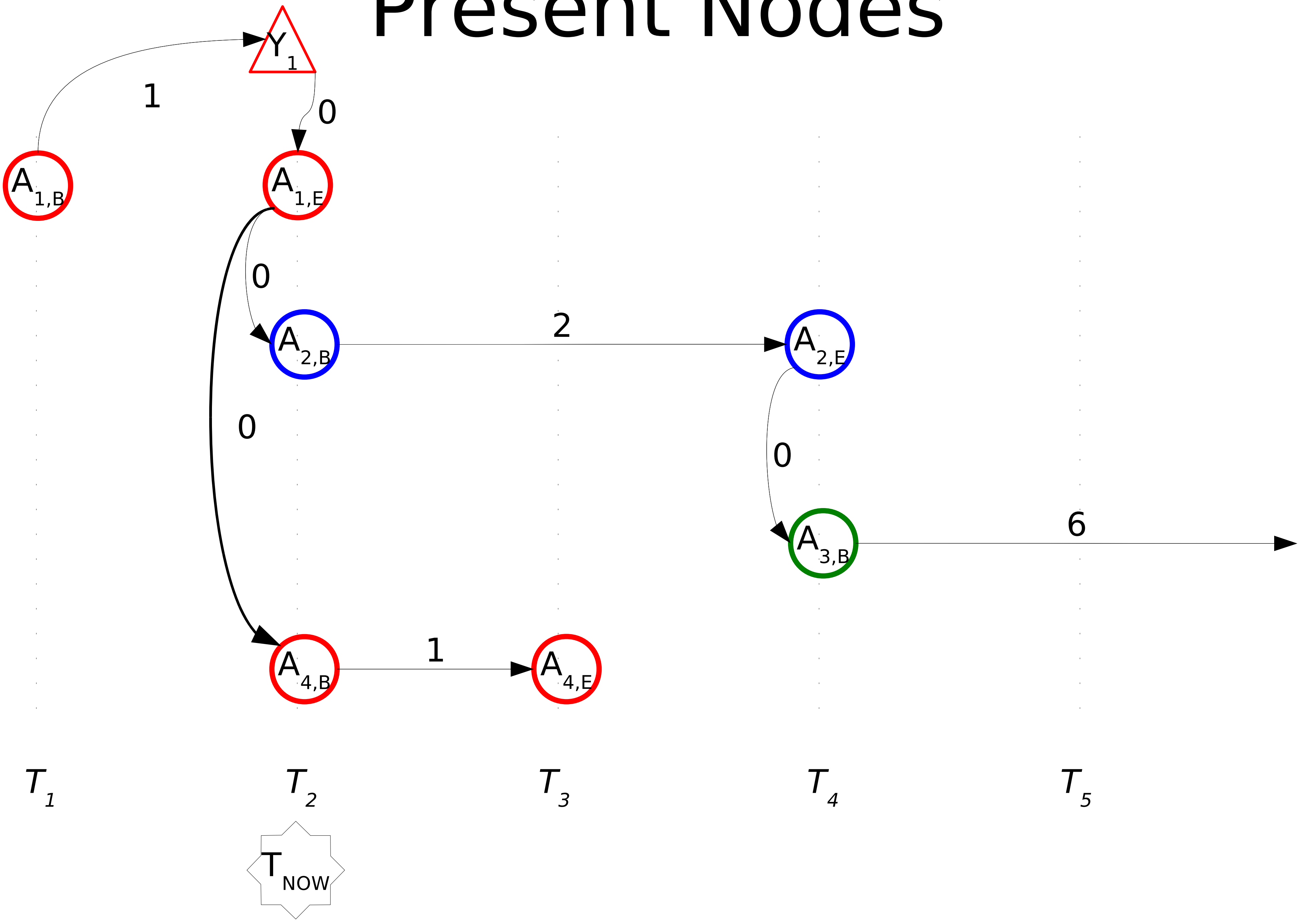
Temporal Constraints



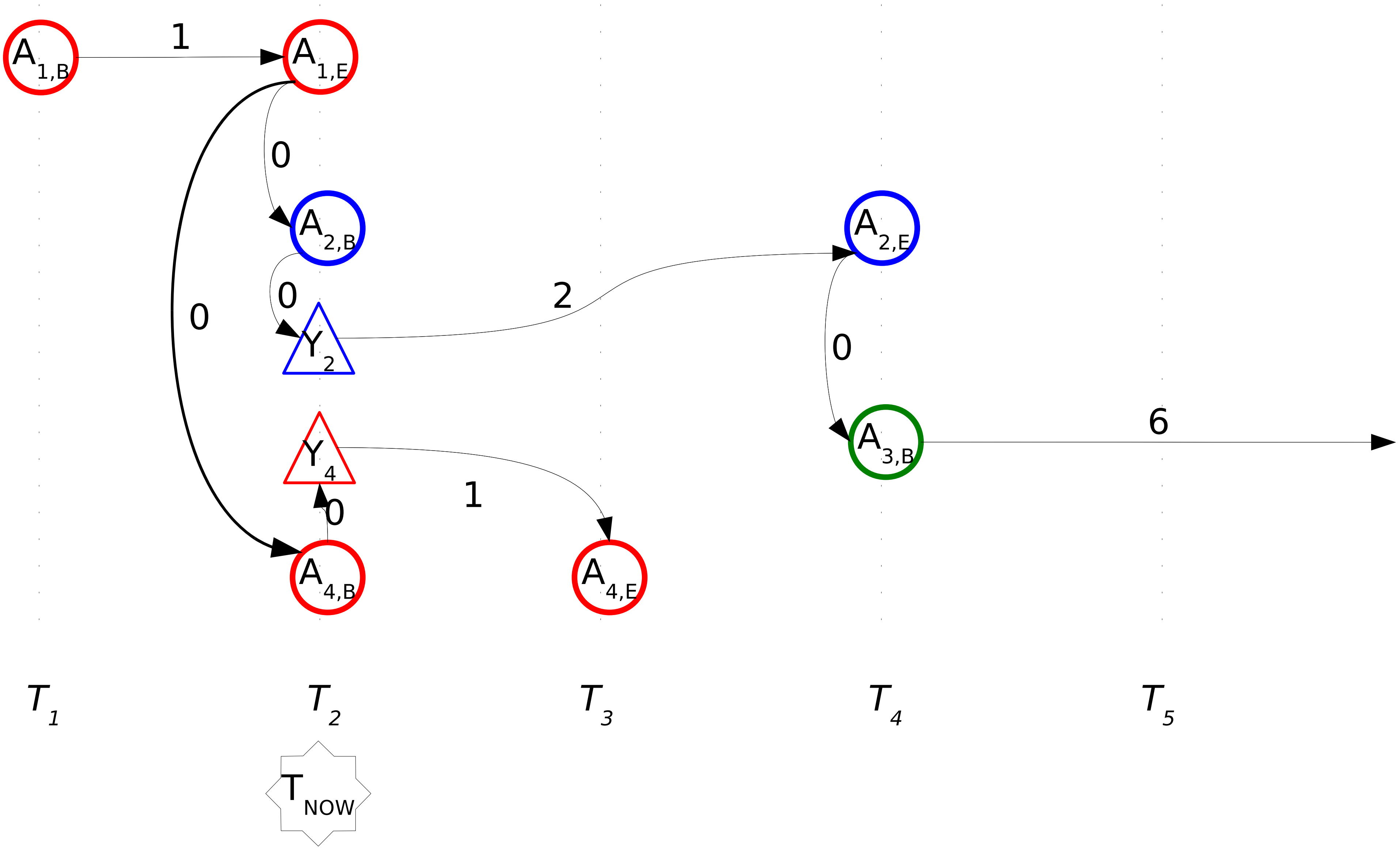
Present Nodes



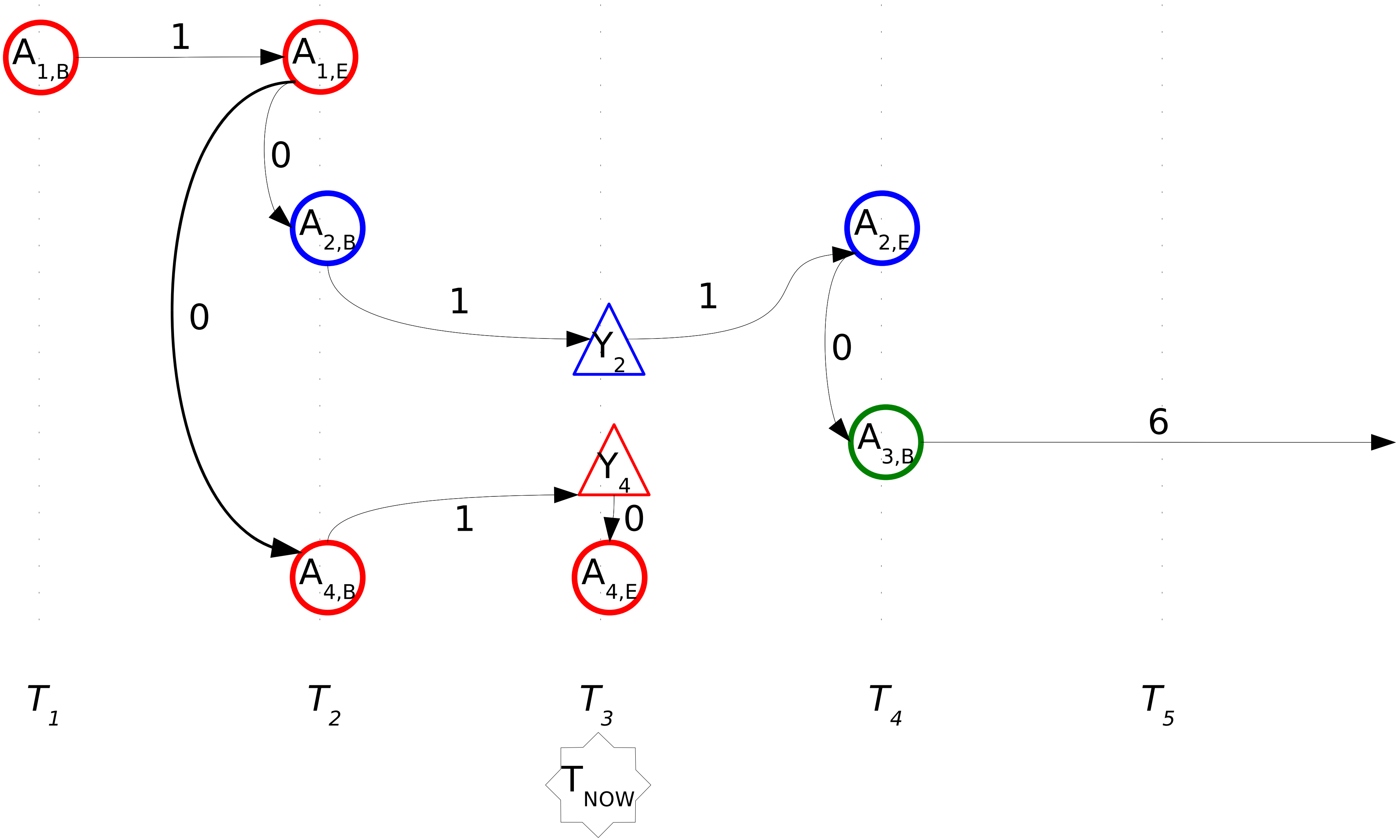
Present Nodes



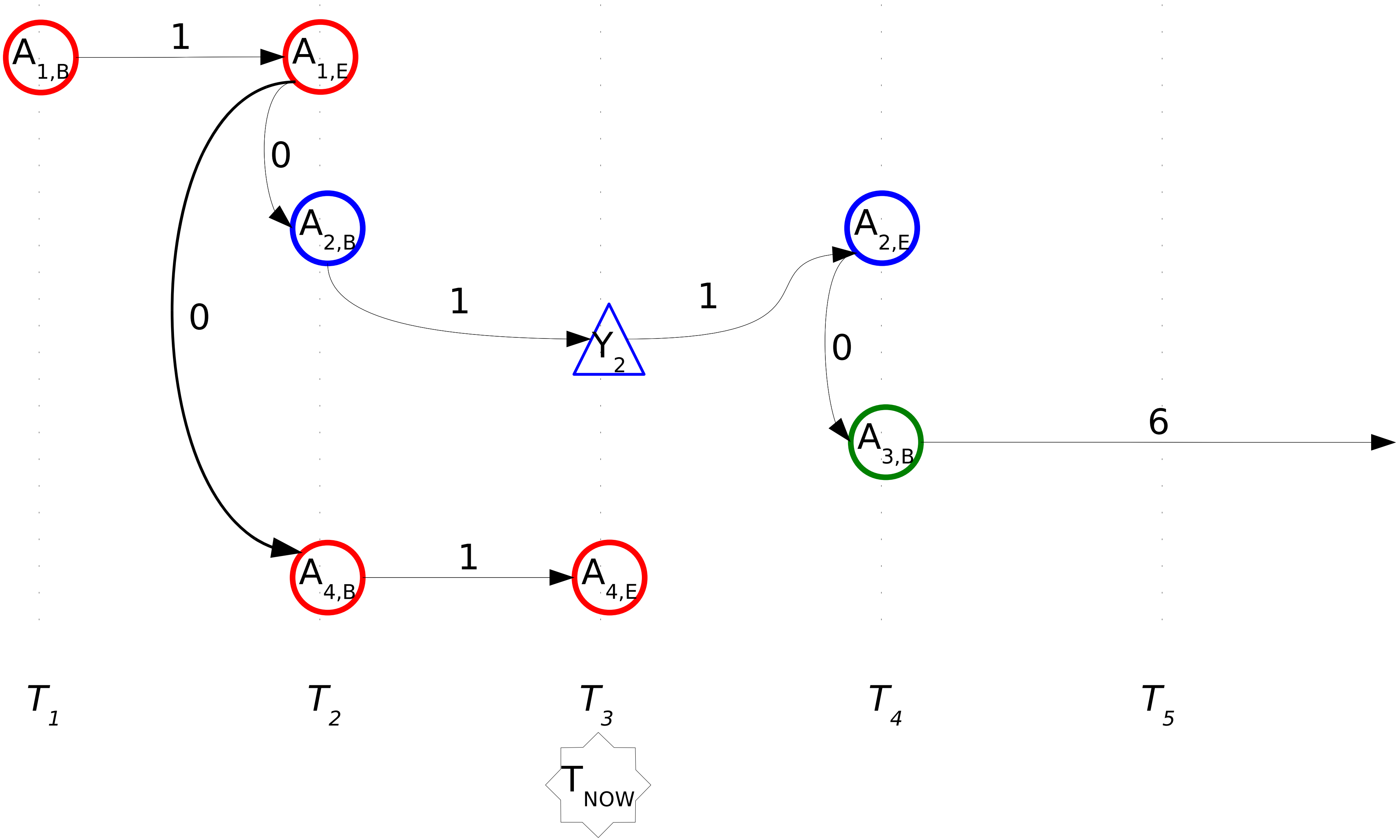
Present Nodes



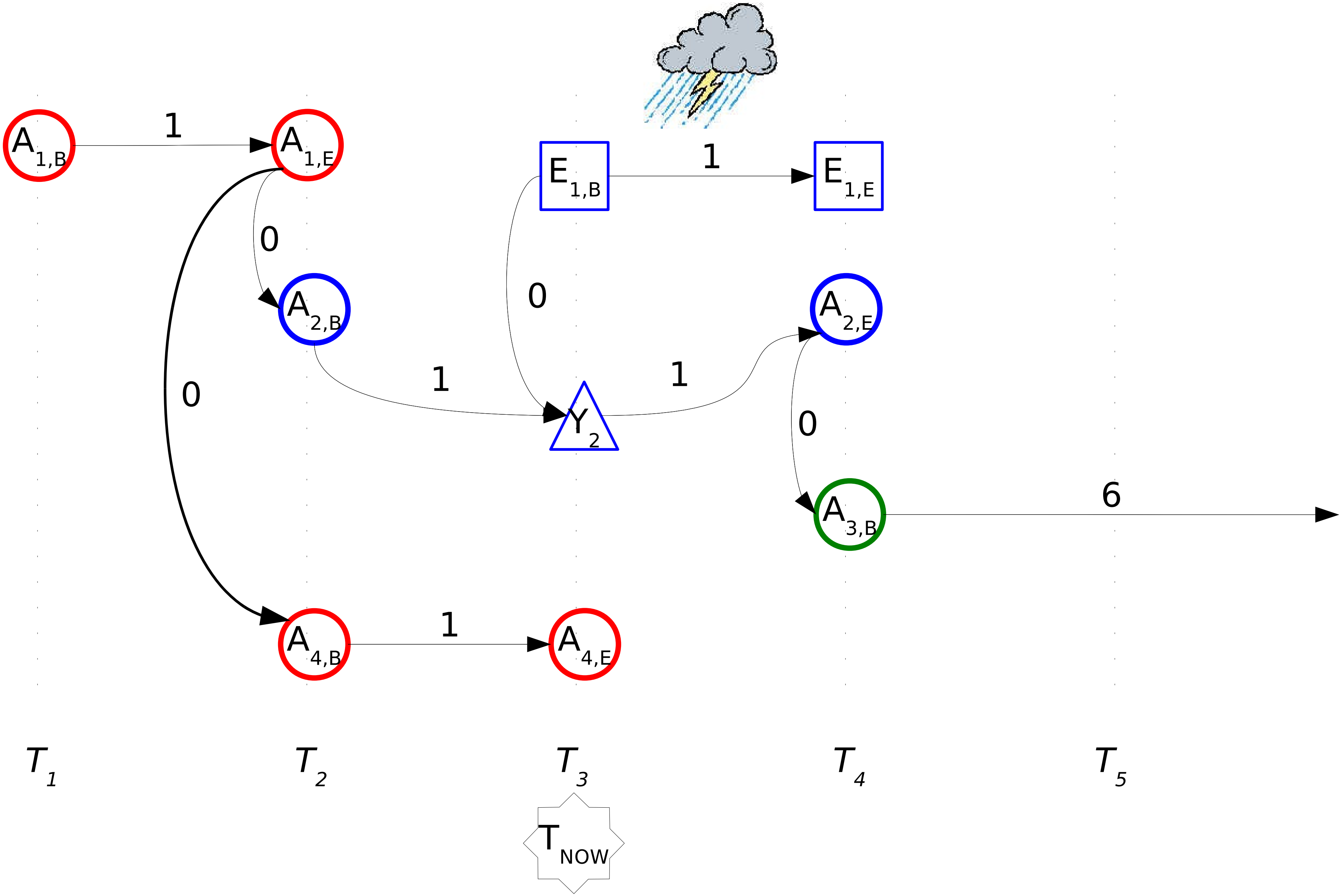
Present Nodes



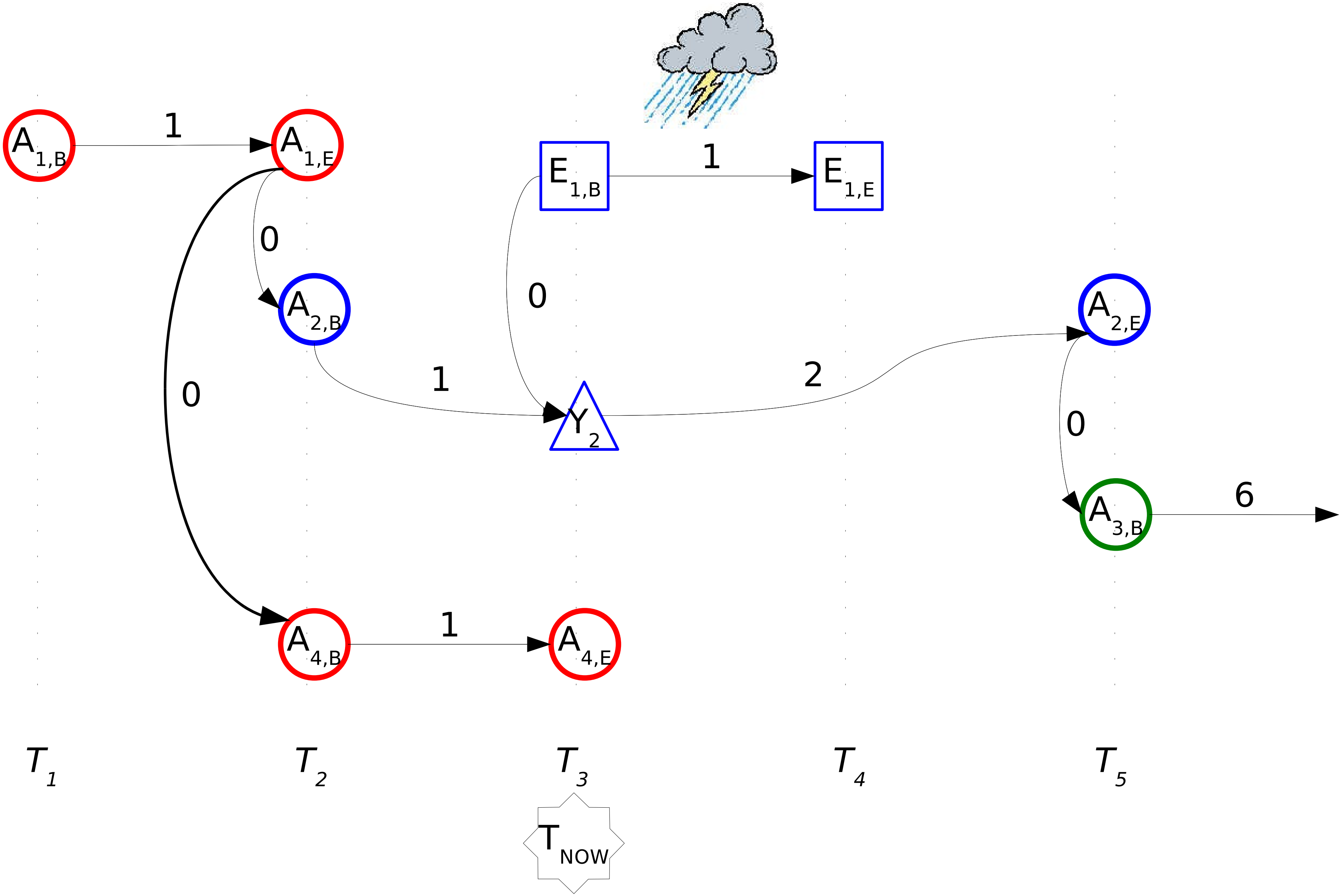
Present Nodes



Event Nodes

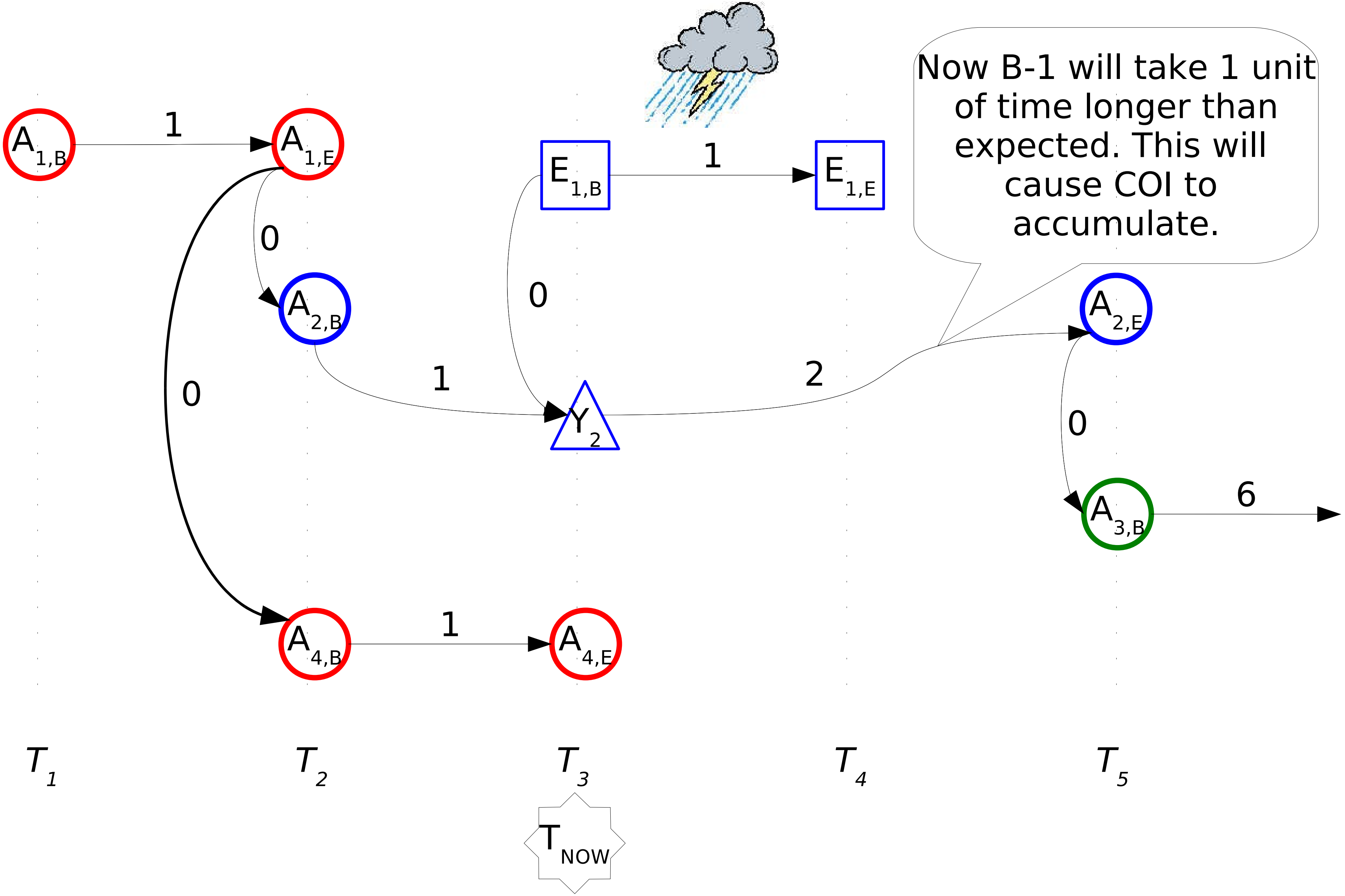

$A_{1,B}$
1
$A_{1,E}$
$E_{1,B}$
1
$E_{1,E}$
0
0
$A_{2,B}$
$A_{2,E}$
0
1
Y_2
1
0
$A_{3,B}$
6
$A_{4,B}$
1
$A_{4,E}$
T_1
T_2
T_3
T_4
T_5
T_{NOW}

Event Nodes



$A_{1,B}$
1
$A_{1,E}$
0
$A_{2,B}$
0
$E_{1,B}$
1
$E_{1,E}$
0
1
Y_2
2
$A_{2,E}$
0
$A_{3,B}$
6
$A_{4,B}$
1
$A_{4,E}$
T_1
T_2
T_3
T_4
T_5
T_{NOW}

Event Nodes



Now B-1 will take 1 unit of time longer than expected. This will cause COI to accumulate.

Cost Overrun Indicator

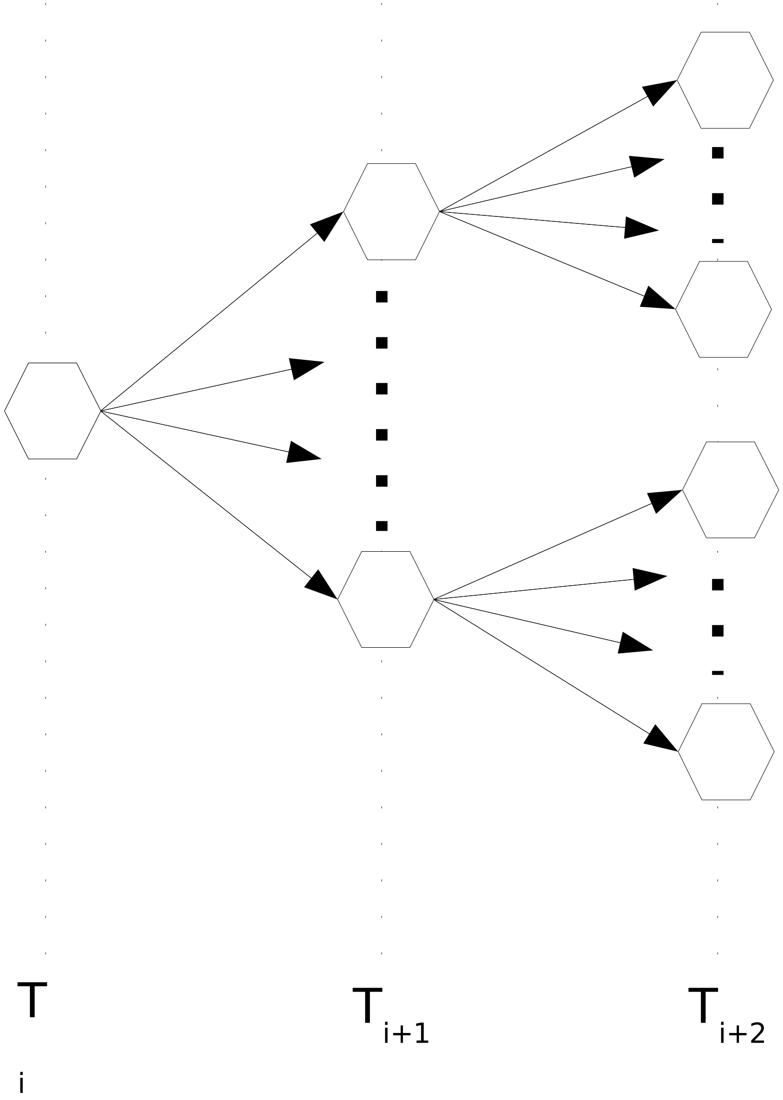
- COI can accumulate as a result of:
  - Delays from events (such as rain)
  - The natural lag in the as-planned schedule
- An *indicator* of budget overruns, not necessarily an exact figure
- Used to show:
  - Cost of delay in different activities
  - Cost of natural lag in the schedule
  - Contrast between various scenarios

Traversal vs. Querying

- *Traversal* is the day-to-day simulation of the project
- *Querying* predicts the most likely futures

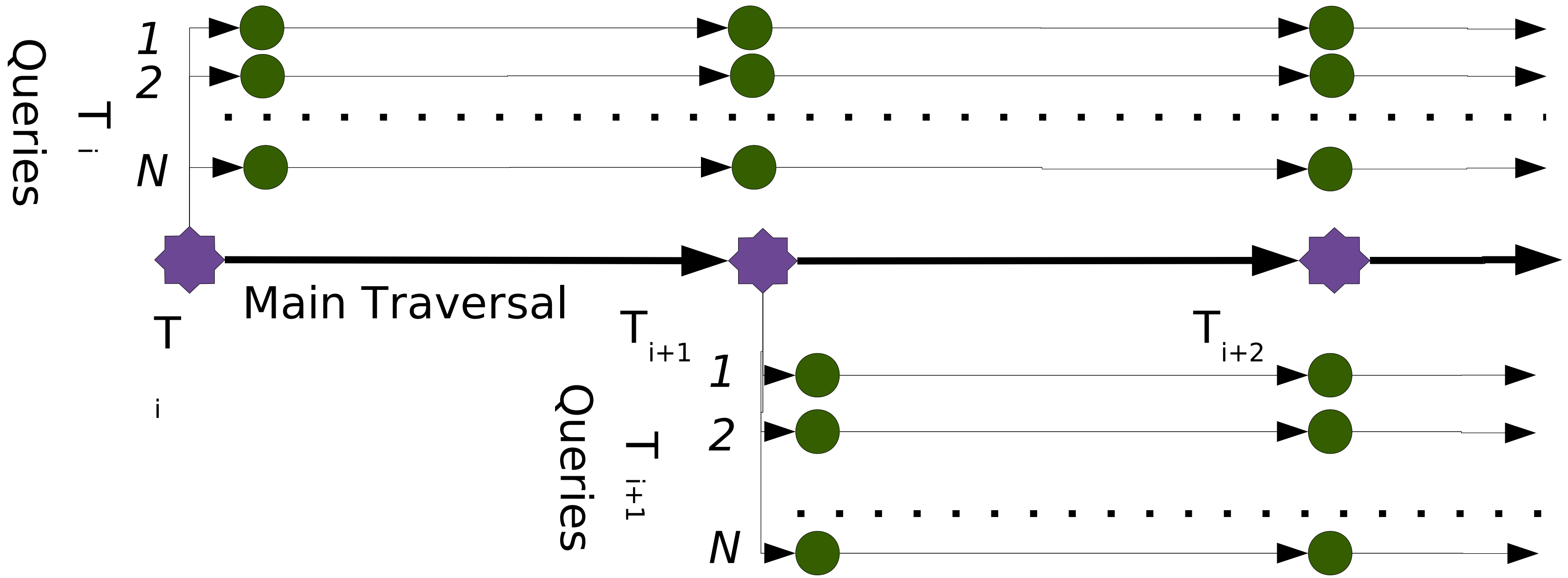
Querying

- From a point in time T_i, a project has numerous futures at time T_{i+1}, each of which has futures at time T_{i+2}, and so on.
- Investigating all futures is intractable



Monte Carlo Solution to Querying

- Probabilistically sample 1 future for each state
- Repeat *N* number of times to get a general picture of what the most probable futures are



What does Querying Provide?

- Given the current state and history of the project:
  - What are the most probable project completion times?
  - What are the most probable COIs?

Outline

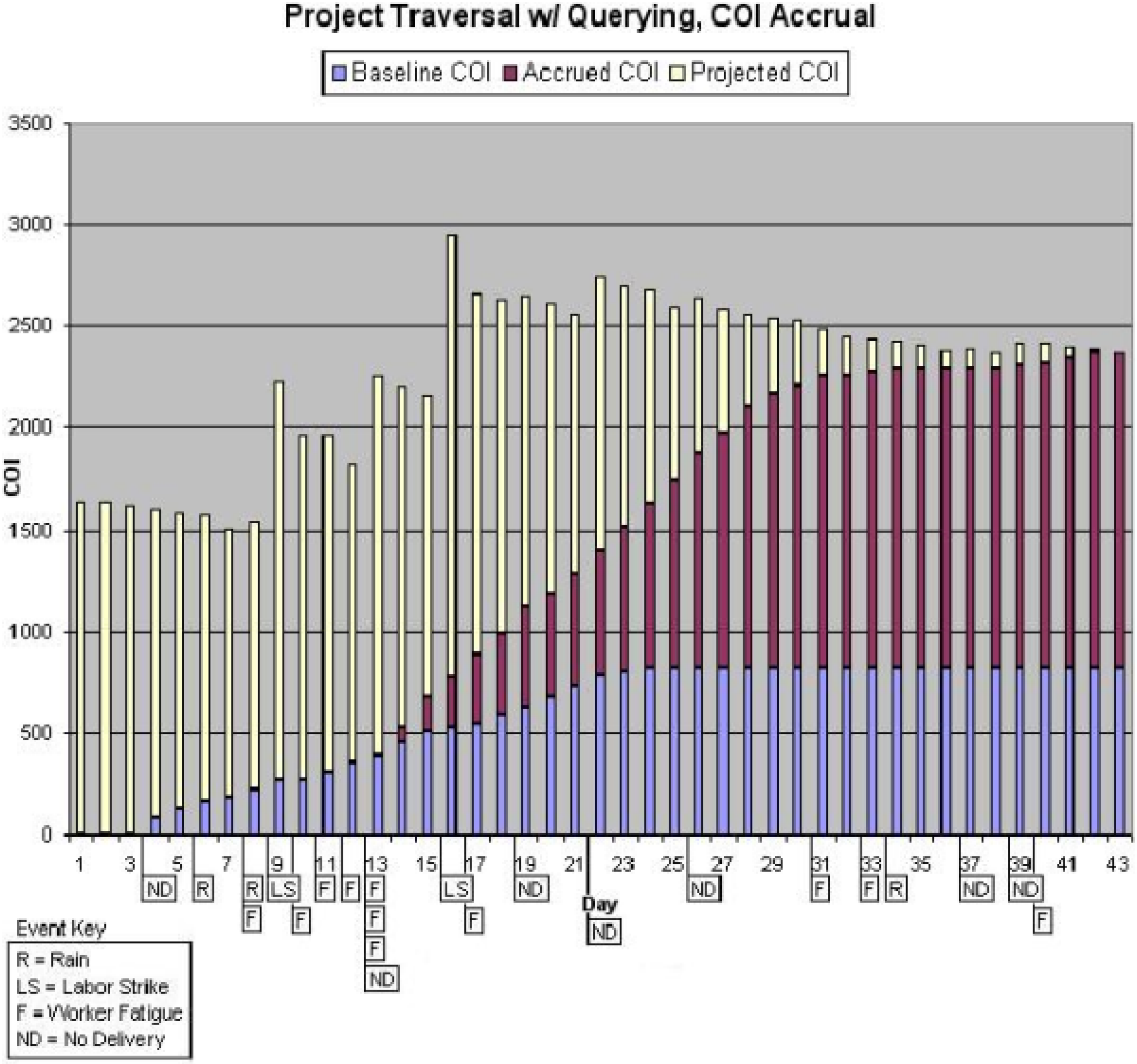
- Introduction
- Case Study
- TONAE Framework & Algorithms
- **Experimental Results**
- Conclusions

Experimental Run

- Single traversal of full, 6-sequence structural steel example
- 1000 query iterations performed per day
- COI (per day) of the three activity types:
  - Hoisting: 41.65
  - Bolting & Connecting: 17.54
  - Decking: 23.58

Independent Events Considered:

- Labor Strike
  - Duration: 3 days
  - Probability: 5%
  - Global
- Rain
  - Duration: 1 day
  - Probability: 10%
  - Global
- No Delivery
  - Duration: 3 days
  - Probability: 5%
  - Local
- Worker Fatigue
  - Duration: 1 day
  - Probability: 10%
  - Local



Project Traversal w/ Querying, COI Accrual

Outline

- Introduction
- Case Study
- TONAE Framework & Algorithms
- Experimental Results
- **Conclusions**

Contributions

- An extension of temporal constraint networks
    - Represents construction management projects
    - Represents uncertain external events, COI
- Means of *traversing* and *querying* these networks to allow the exploration of 'what-if' scenarios by construction managers.

Limitations & Future Work

- PimGenerate
- ComputeEventEffects
- CalculateRemainingDuration
- Integration of the mechanisms into a stronger simulation system to serve as an instructional tool to construction managers

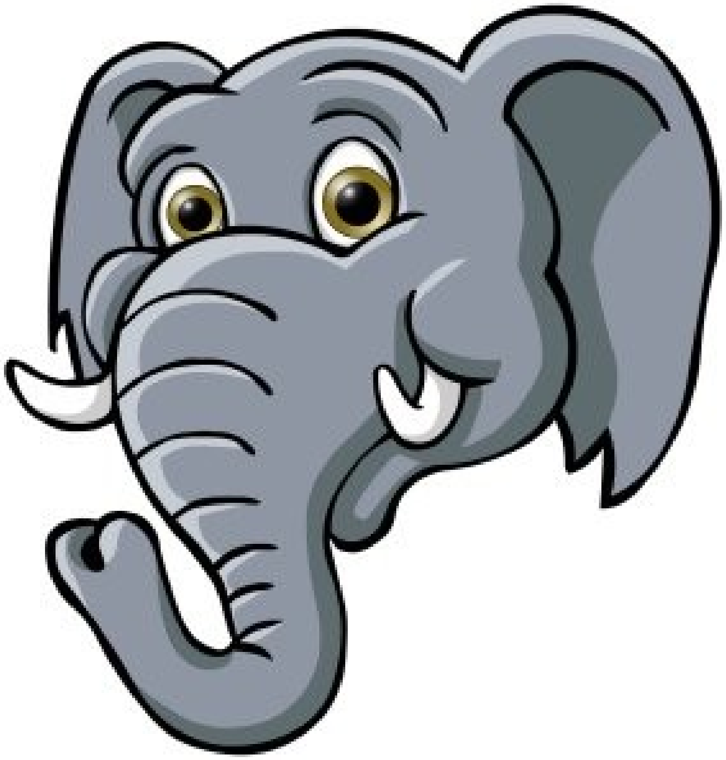
Publications

- Anderson, Onder, Mukherjee. 2007. Expecting the Unexpected: Representing and Reasoning about Construction Process Crisis Scenarios. Winter Simulation Conference. December 9-12, Washington, D.C.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. SES-0624118. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Questions?



Discrete Event Simulations

- General Frameworks (Arena, ProModel, GPSS/H)
- Construction-Based (Simphony, STROBOSCOPE)
- Transaction-flow based model
- Application to construction operations and projects with repetitive sequences of activities

Simple Temporal Networks

- Nodes represent events
- Edges between nodes represent temporal constraints
- Shortest path algorithms are used to check the network for temporal consistency

Temporal Constraints & COI

- Example temporal constraints in the form *Penalty* : *Constraint*

$$0 : 1 \leq A_{1,E} - A_{1,B} \leq 5$$

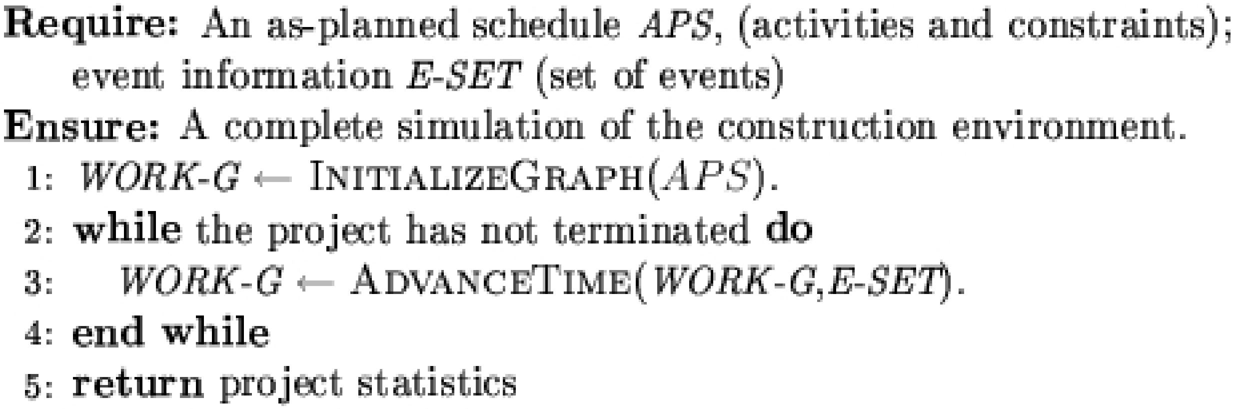
$$1 : 6 \leq A_{1,E} - A_{1,B} \leq 10$$

$$\infty : 11 \leq A_{1,E} - A_{1,B}$$

Formal Definition of TONAE

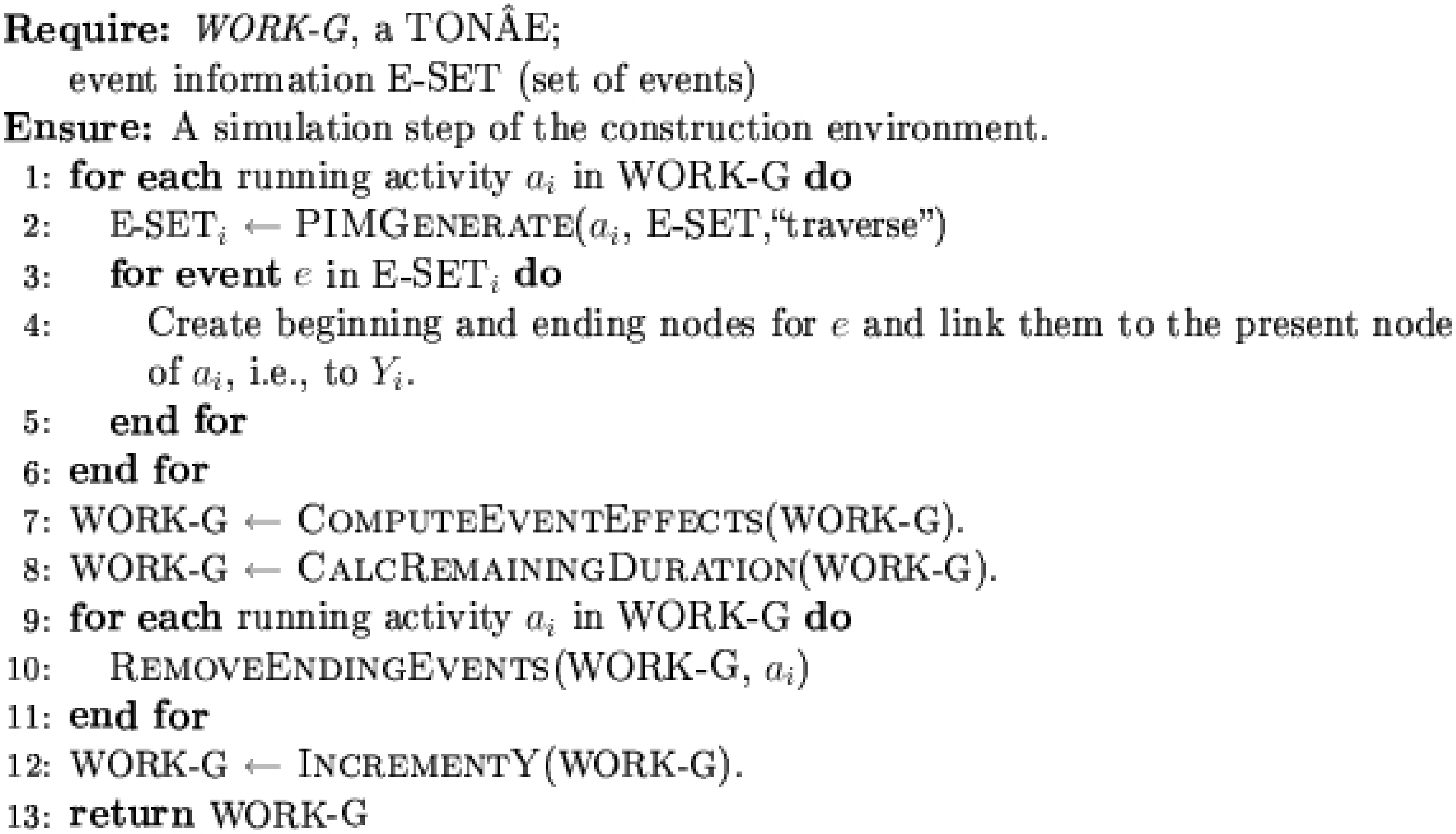
- A TONAE is a quadruple (A, B, C, D), where:
  - A = Set of all Activity Nodes
  - B = Set of all Present Nodes
  - C = Set of all Event Nodes
  - D = Set of all Temporal Constraints

Traversal Algorithm (1)



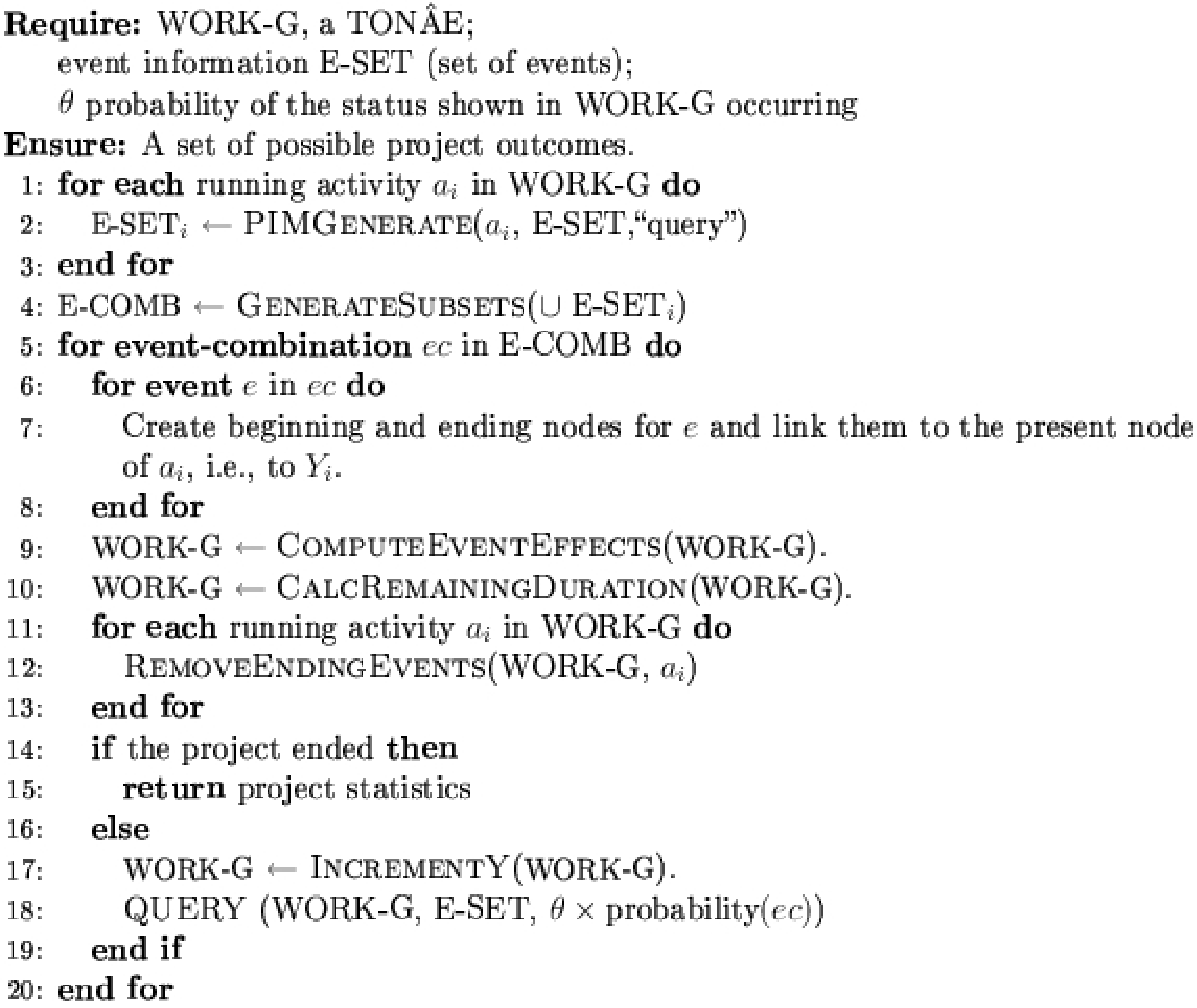
```
Require: An as-planned schedule APS, (activities and constraints);
    event information E-SET (set of events)
Ensure: A complete simulation of the construction environment.
WORK-G ← InitializeGraph(APS).
while the project has not terminated do
    WORK-G ← AdvanceTime(WORK-G,E-SET).
end while
return project statistics
```

Traversal Algorithm (2)



```
Require: WORK-G, a TONÂE;
    event information E-SET (set of events)
Ensure: A simulation step of the construction environment.
for each running activity $a_i$ in WORK-G do
    E-SET$_i$ ← PIMGenerate($a_i$, E-SET,“traverse”)
    for event $e$ in E-SET$_i$ do
        Create beginning and ending nodes for $e$ and link them to the present node
        of $a_i$, i.e., to $Y_i$.
    end for
end for
WORK-G ← ComputeEventEffects(WORK-G).
WORK-G ← CalcRemainingDuration(WORK-G).
for each running activity $a_i$ in WORK-G do
    RemoveEndingEvents(WORK-G, $a_i$)
end for
WORK-G ← IncrementY(WORK-G).
return WORK-G
```

Query Algorithm



```
Require: WORK-G, a TONÂE;
    event information E-SET (set of events);
    $\theta$ probability of the status shown in WORK-G occurring
Ensure: A set of possible project outcomes.
for each running activity $a_i$ in WORK-G do
    E-SET$_i$ $\leftarrow$ PIMGenerate($a_i$, E-SET,“query”)
end for
E-COMB $\leftarrow$ GenerateSubsets($\cup$ E-SET$_i$)
for event-combination $ec$ in E-COMB do
    for event $e$ in $ec$ do
        Create beginning and ending nodes for $e$ and link them to the present node
        of $a_i$, i.e., to $Y_i$.
    end for
    WORK-G $\leftarrow$ ComputeEventEffects(WORK-G).
    WORK-G $\leftarrow$ CalcRemainingDuration(WORK-G).
    for each running activity $a_i$ in WORK-G do
        RemoveEndingEvents(WORK-G, $a_i$)
    end for
    if the project ended then
        return project statistics
    else
        WORK-G $\leftarrow$ IncrementY(WORK-G).
        QUERY (WORK-G, E-SET, $\theta \times$ probability($ec$))
    end if
end for
```