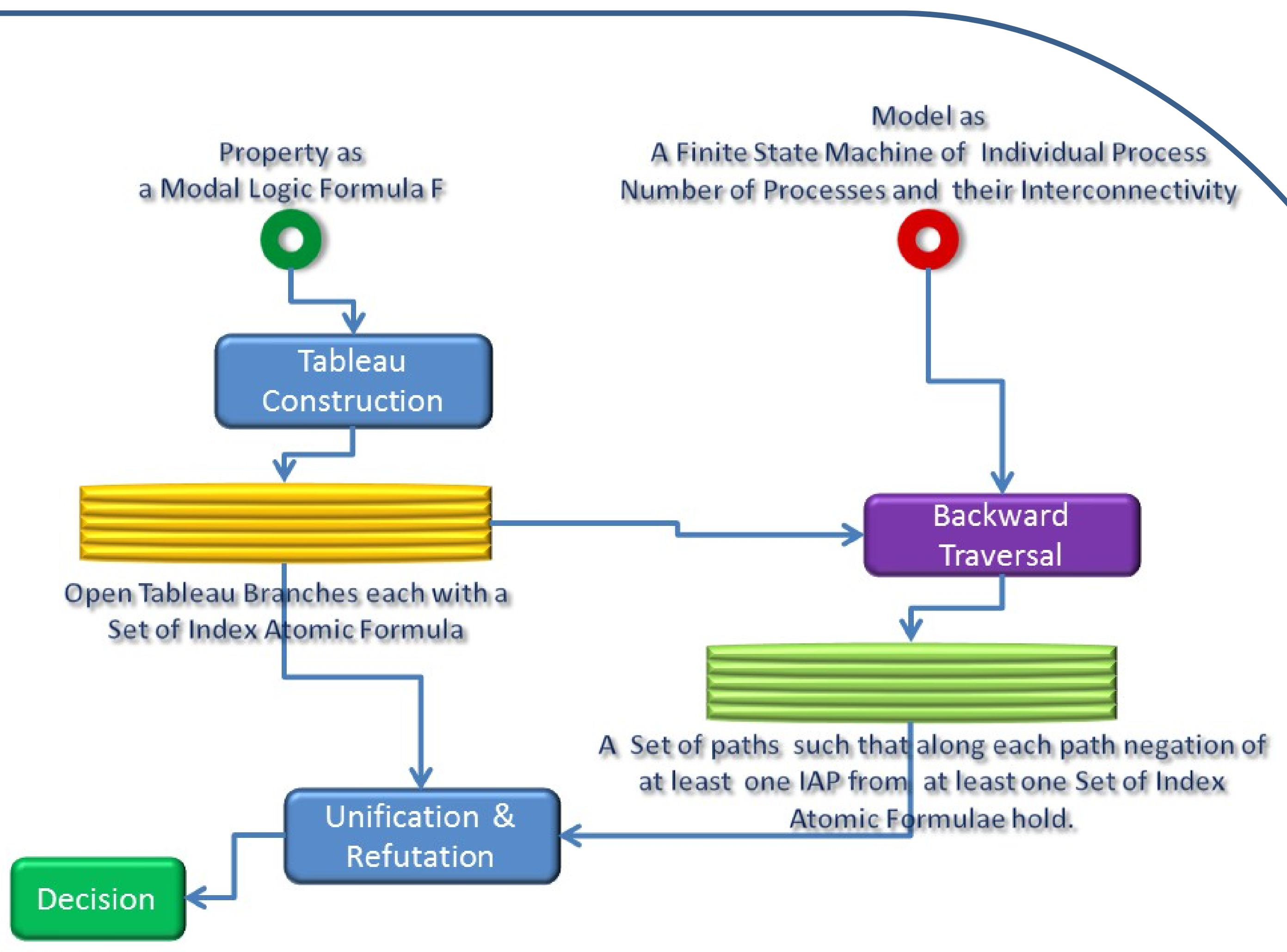
Reachability in a Finite Distributed System Protocol Model by Backward Traversal

Presenter: Tapas Samanta, Variable Energy Cyclotron Centre

Authors: Tapas Samanta (VECC, Kolkata), Dipankar Sarkar (IITKGP, West Bengal), Samarpita Mukherjee (JU, Kolkata)


Property as
a Modal Logic Formula F
Model as
A Finite State Machine of Individual Process
Number of Processes and their Interconnectivity
Tableau Construction
Backward Traversal
Open Tableau Branches each with a Set of Index Atomic Formula
A Set of paths such that along each path negation of at least one IAP from at least one Set of Index Atomic Formulae hold.
Unification & Refutation
Decision

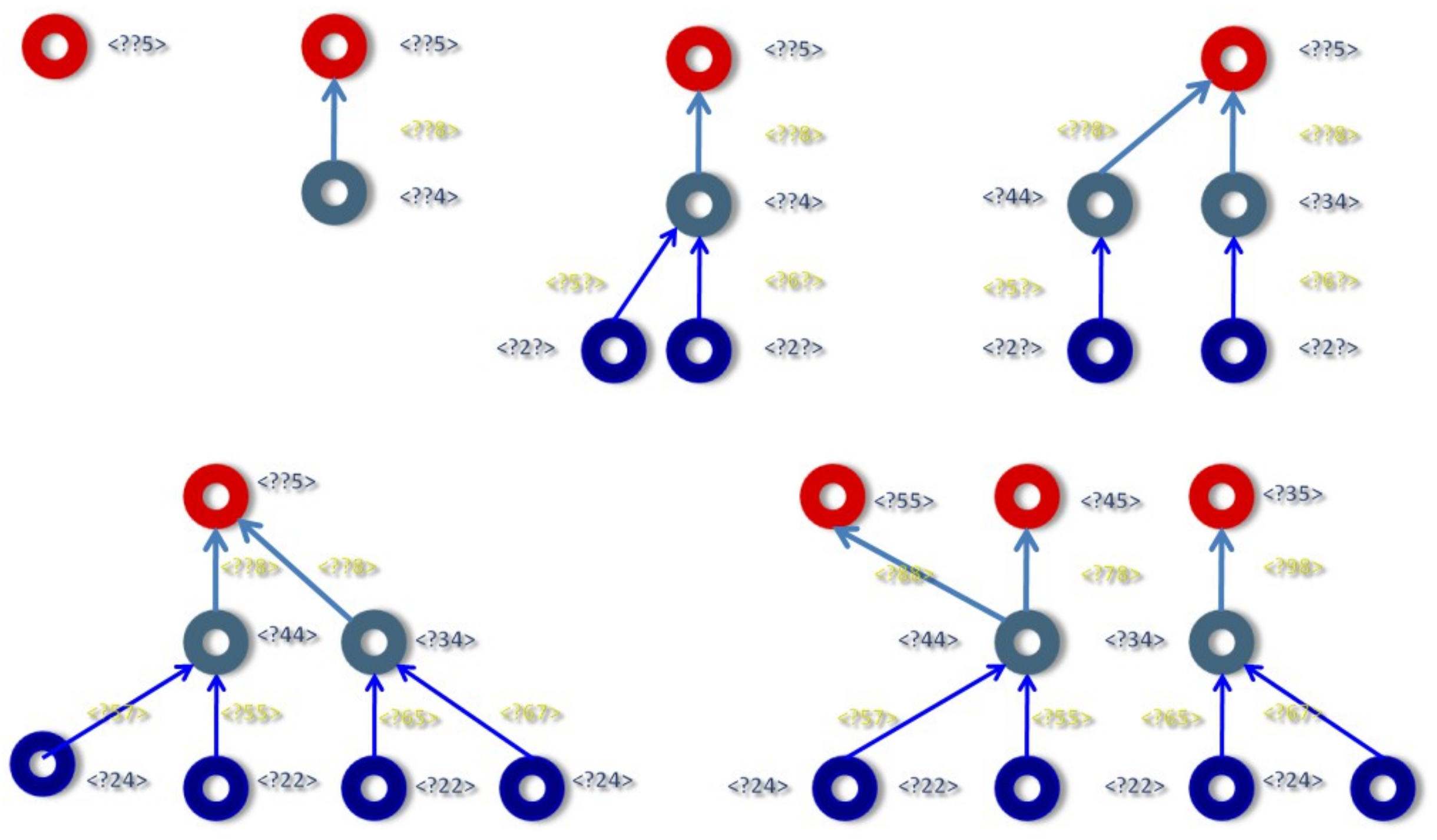
The Mechanized Framework

- Let M be a model and F be a property;
- Suppose it is to be verified whether M⊨F
  Since M⊨F≡⊨(M→F)≡⊨ϕ (suppose). It is to be verified whether ϕ is valid or not i.e. the tableau tree of ¬ϕ is closed or not.
- To construct the tableau of ¬((M→F)) i.e. (M ^ ¬F)
  - Start constructing the Tableau of '¬F"
  - Close all the open branches from the behavior of the model M if possible

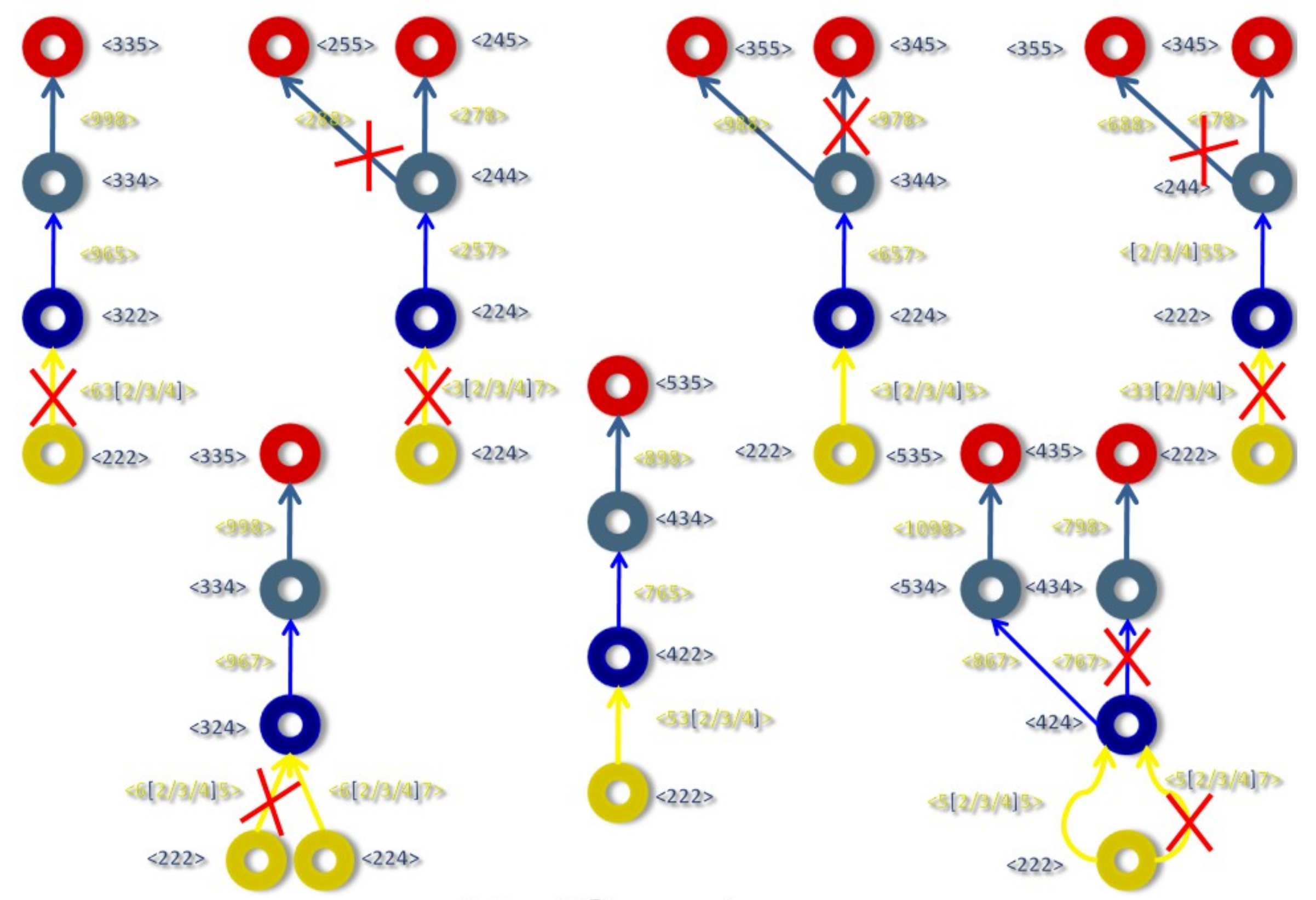
The Steps

- Select an open branch and an IAP (by choice)
  $\neg L(k) : succ(w_1, succ(\_W_0, w_0))$
- Do the Backward Traversal to find a computation path
- Unify the states with the index $succ(w_1, succ(\_W_0, w_0))$
- Extend the unification to close the remaining open branches

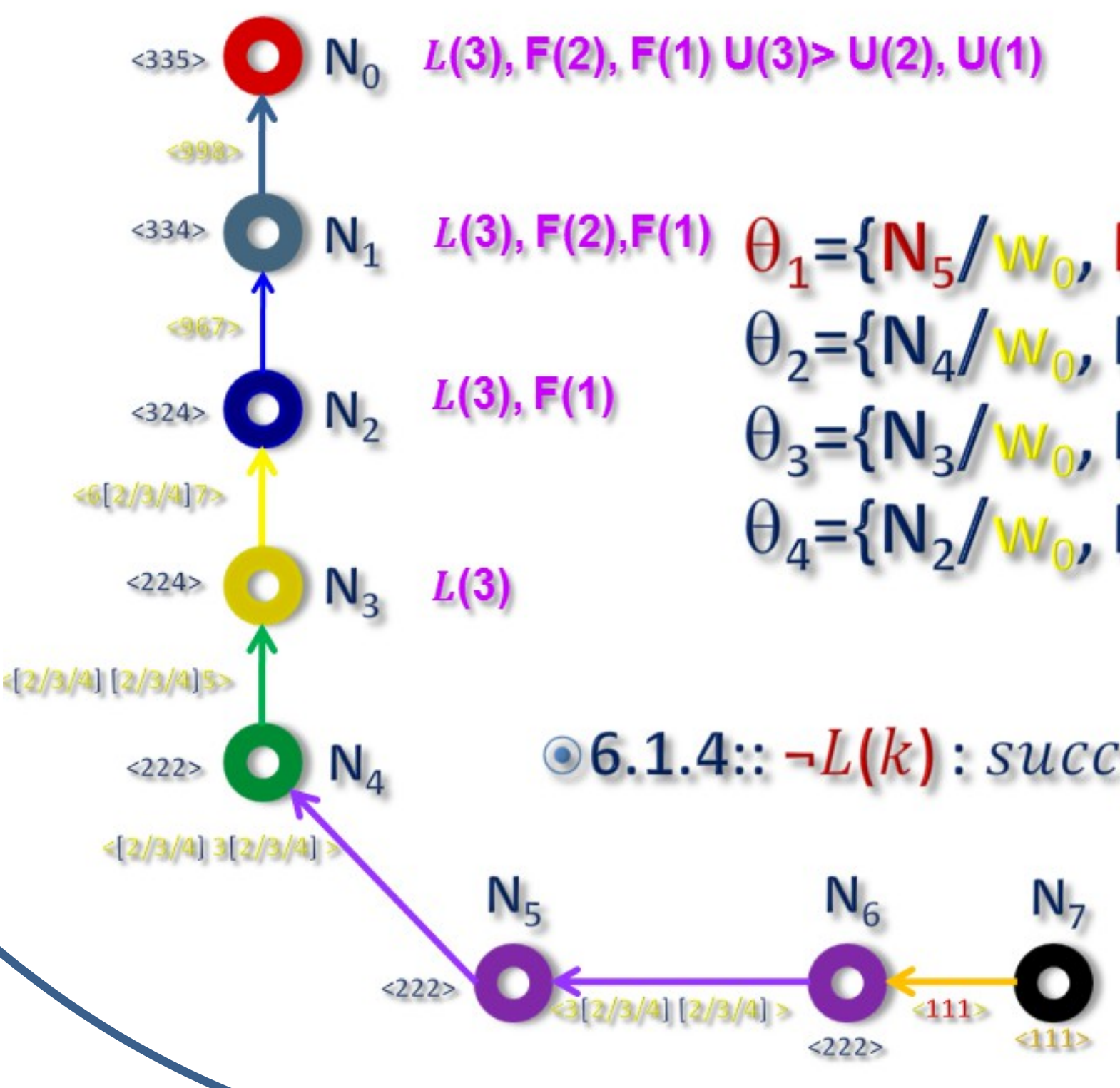
Backward Traversal for LCR-LEP



Backward Traversal



Unification



$\theta_1=\{N_5/w_0, N_4/\_W_0, N_3/w_1\}$

$\theta_2=\{N_4/w_0, N_3/\_W_0, N_2/w_1\}$

$\theta_3=\{N_3/w_0, N_2/\_W_0, N_1/w_1\}$

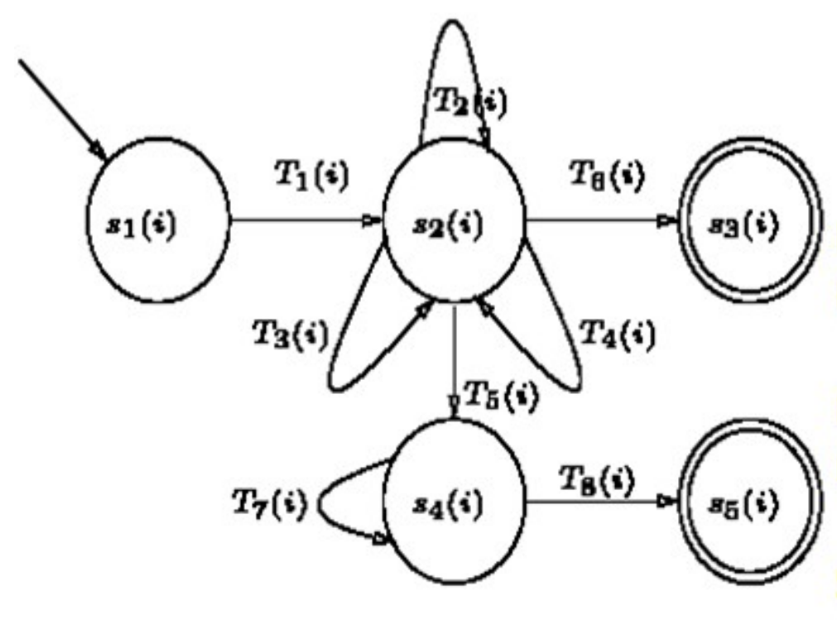
$\theta_4=\{N_2/w_0, N_1/\_W_0, N_0/w_1\}$

- 6.1.4:: $\neg L(k) : succ(w_1, succ(\_W_0, w_0))$

The Backward Traversal

- Let there be N no of processes
- Let AP:idx be an IAP in a leaf of an open branch of the Tableau tree
- Find out from the Model, State Transition Diagram (STD), of the ith (choice) individual process, the set S of states in which ¬AP holds
- 1.Create Root:
  For each s ∈ S, construct an N-tuple $_0N^i$ with ith (by choice) element s and rest as ? (unknown) and create root node r=$_0N^i$
- 2.Create Children of Root:
- 3.Create Children using (action→guard):
- 4.Refine Children:
- 5.Refine Parents: and all the Ancestors
- 6.If any child-node is not in Initial State goto Step-3

The LCR Leader Election Protocol



$T1(i) \equiv \frac{true}{SD(i)=U(i)}$ $T2(i) \equiv \frac{V(i)=null}{SD(i)=null}$

$T3(i) \equiv \frac{type(V(i))=ID, V(i)>U(i)}{SD(i)=V(i)}$ $T5(i) \equiv \frac{V(i)=U(i)}{SD(i)=msg}$

$T4(i) \equiv \frac{type(V(i))=ID, V(i)<U(i)}{SD(i)=null}$ $T6(i) \equiv \frac{V(i)=msg}{SD(i)=msg}$

$T7(i) \equiv \frac{V(i) \neq msg}{SD(i)=null}$ $T8(i) \equiv \frac{V(i)=msg}{SD(i)=null}$

IN(i): i^{th} process is Initialized; holds in state $s_1(i)$

R(i): i^{th} process is Ready to be either Leader or Follower; holds in state $s_2(i)$

F(i): i^{th} process is elected as Follower; holds in state $s_3(i)$

L(i): i^{th} Process is either 'Elected' or 'Elected & Declared'. as the leader; holds in states $s_4(i)$ & $s_5(i)$

SD(i): i^{th} process sends (either msg or $UID(i)$) to its neighbour $(i+1)^{th}$ process)

V(i): The content that the i^{th} process receives from its neighbour $((i-1)^{th}$ process)

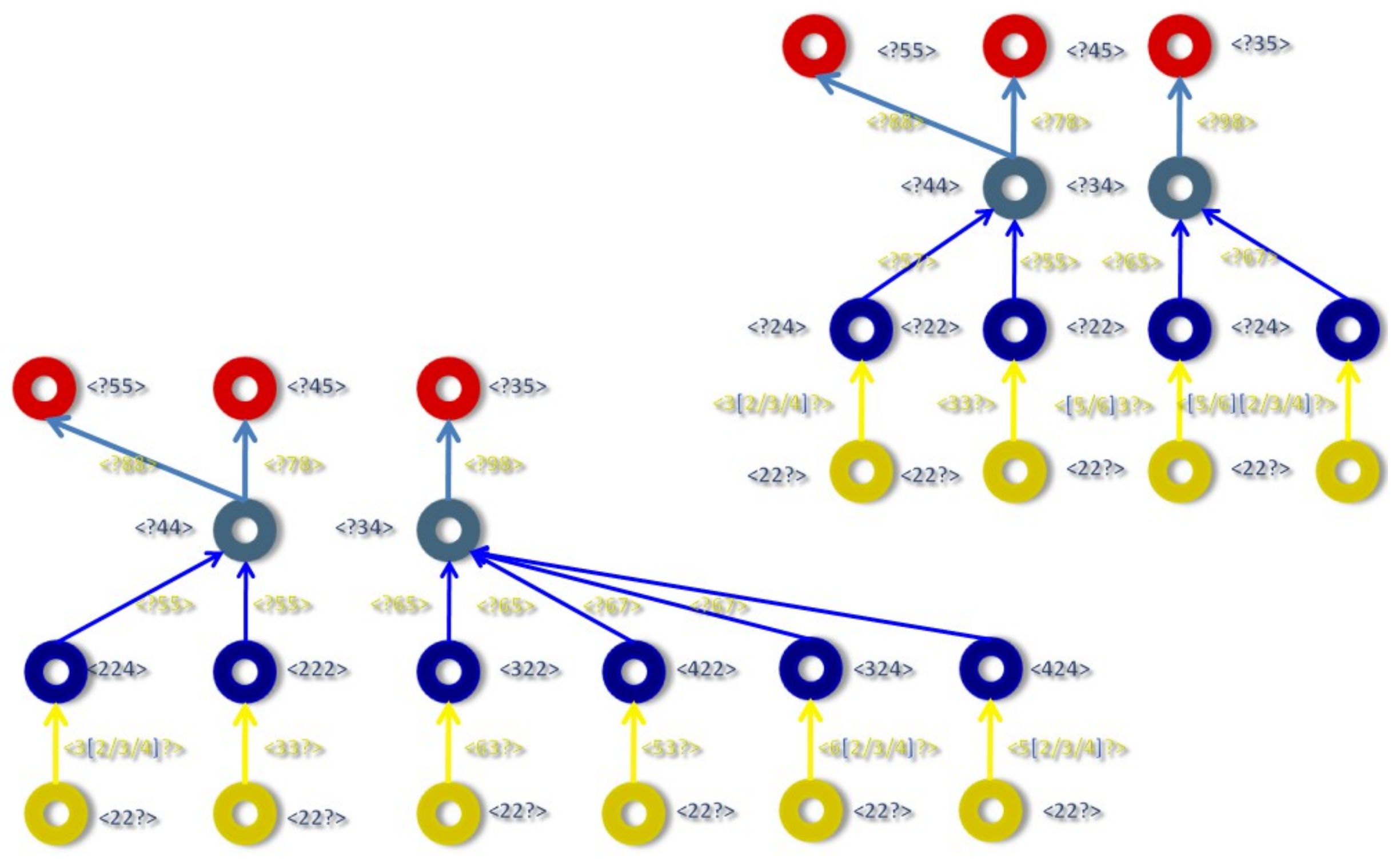
msg: A special message by the leader to inform others that a leader has been elected.

U(i): The Unique Identifier of the i^{th} process.

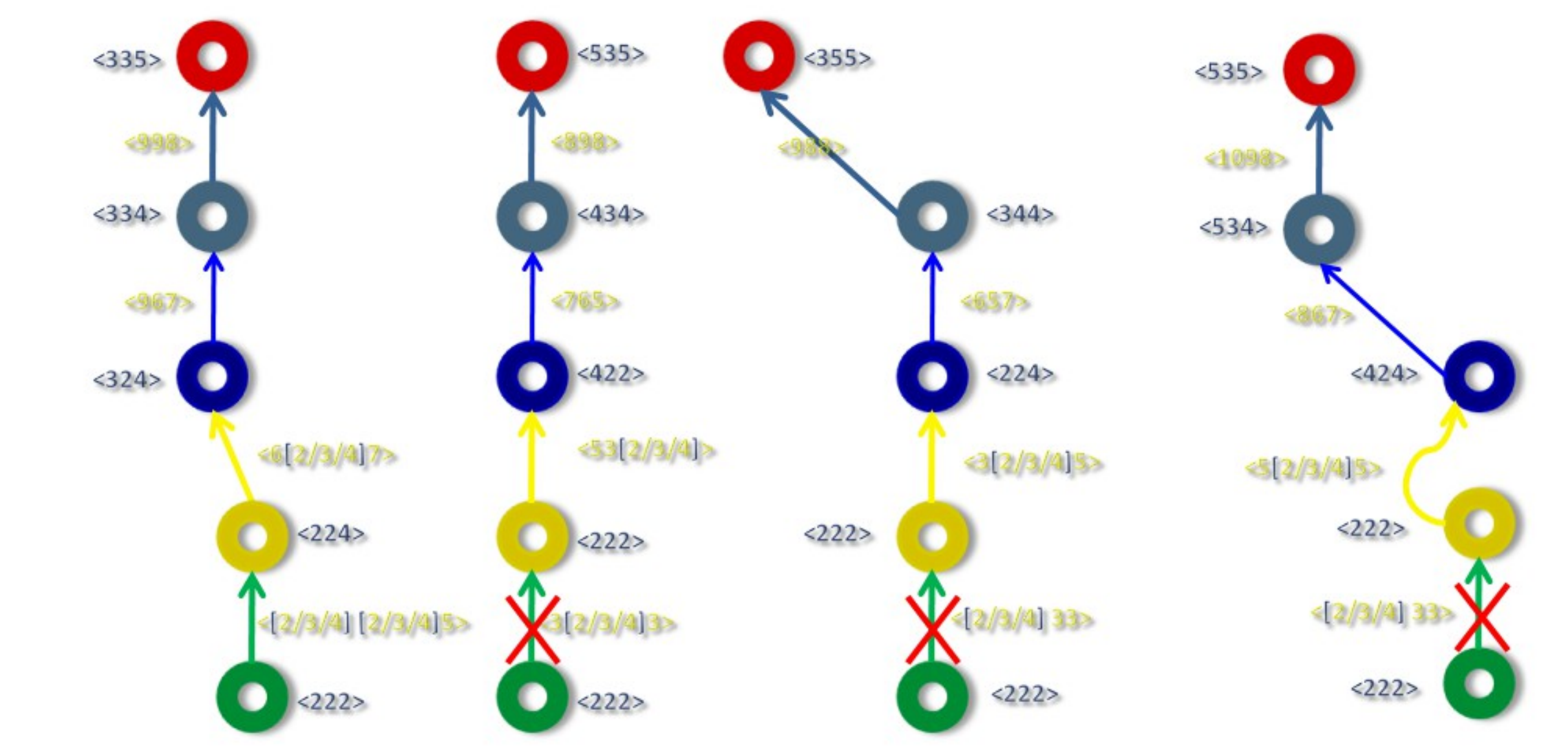
type(x): The type of the variable x. It can be either 'identifier' or special message.

ID: The 'identifier' type.

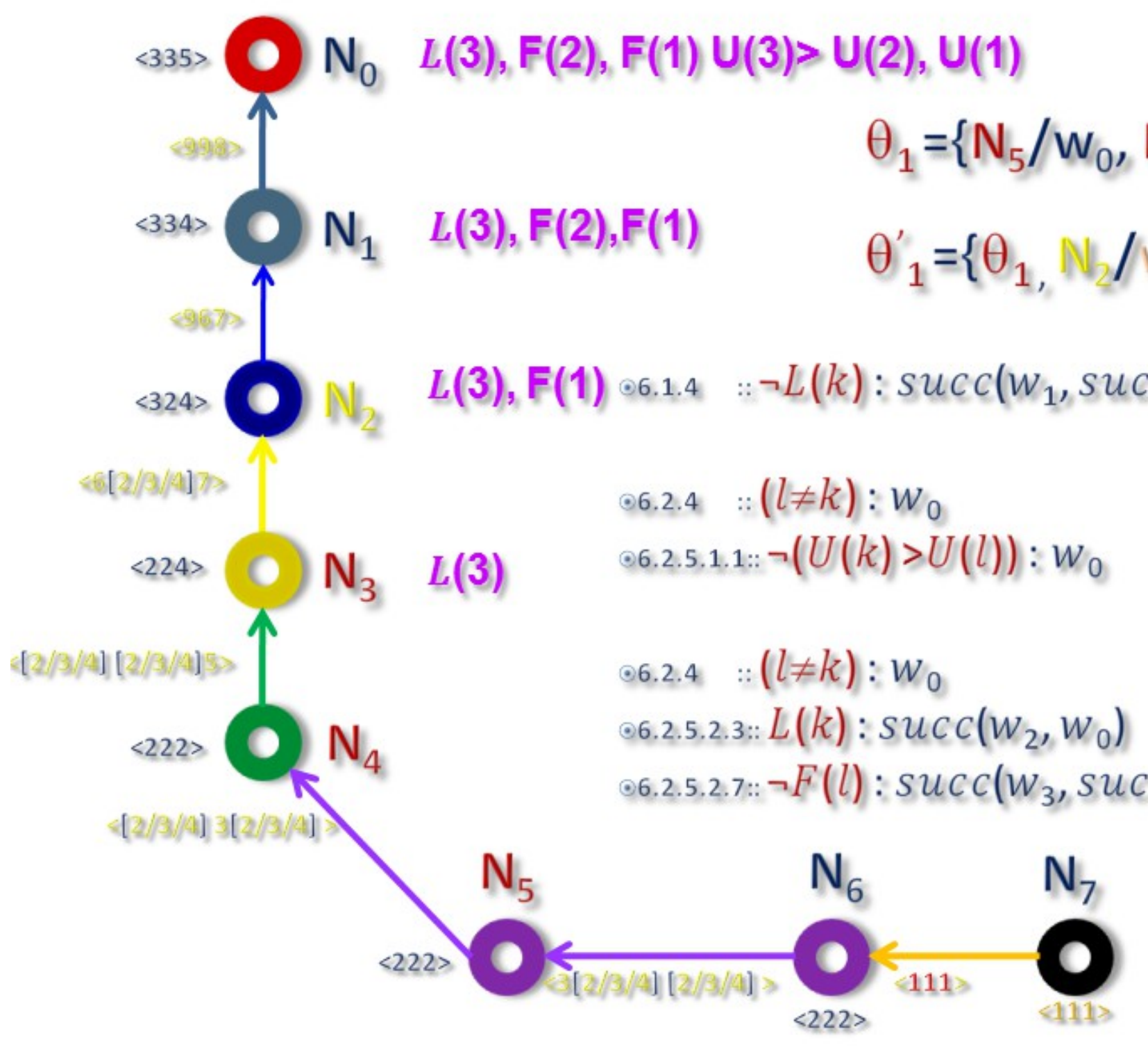
Backward Traversal



Backward Traversal



Unification



$\theta_1=\{N_5/w_0, N_4/\_W_0, N_3/w_1$

$\theta'_1=\{\theta_1, N_2/w_3, \_W_1/w_1, \_W_0/w_2\}$

- 6.1.4 :: $\neg L(k) : succ(w_1, succ(\_W_0, w_0))$ (1)

- 6.2.4 :: $(l \neq k) : w_0$ (2)
- 6.2.5.1.1:: $\neg(U(k) > U(l)) : w_0$ (3)

- 6.2.4 :: $(l \neq k) : w_0$ (4)
- 6.2.5.2.3:: $L(k) : succ(w_2, w_0)$ (5)
- 6.2.5.2.7:: $\neg F(l) : succ(w_3, succ(\_W_1, succ(w_2, w_0)))$ (6)