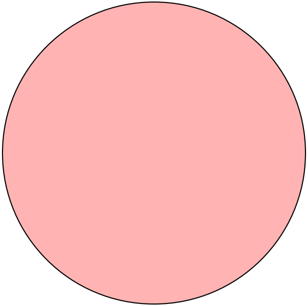
Probabilistic Timed Graph Transformation Systems

Maria Maximova, Holger Giese, Christian Krause

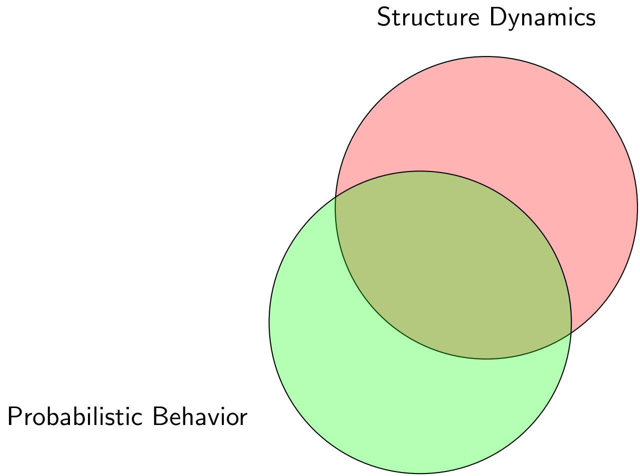
19th July 2017

System's Aspects and Modeling Formalisms

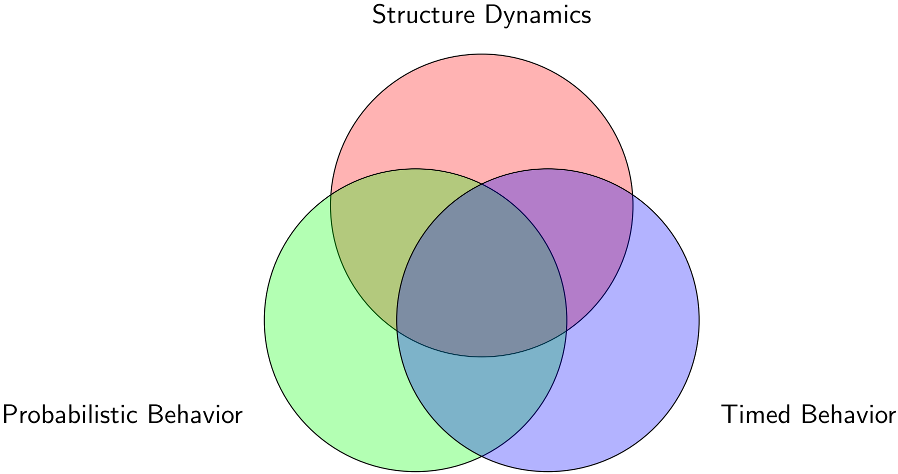
Structure Dynamics



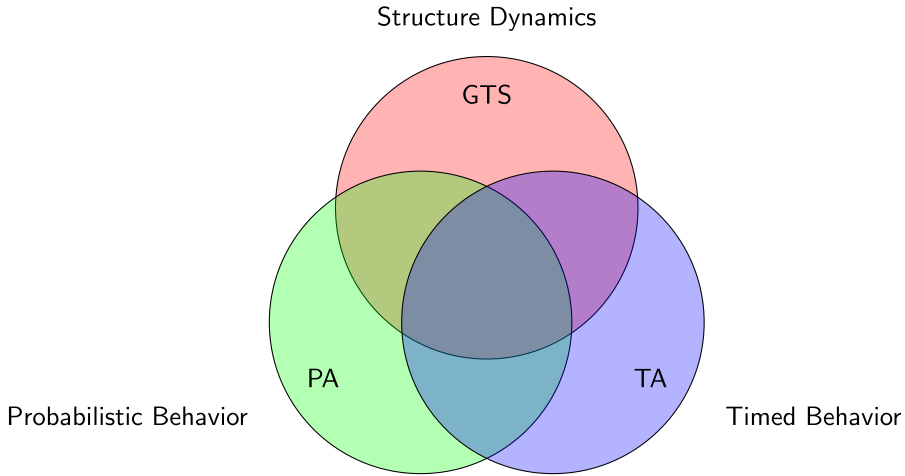
System's Aspects and Modeling Formalisms



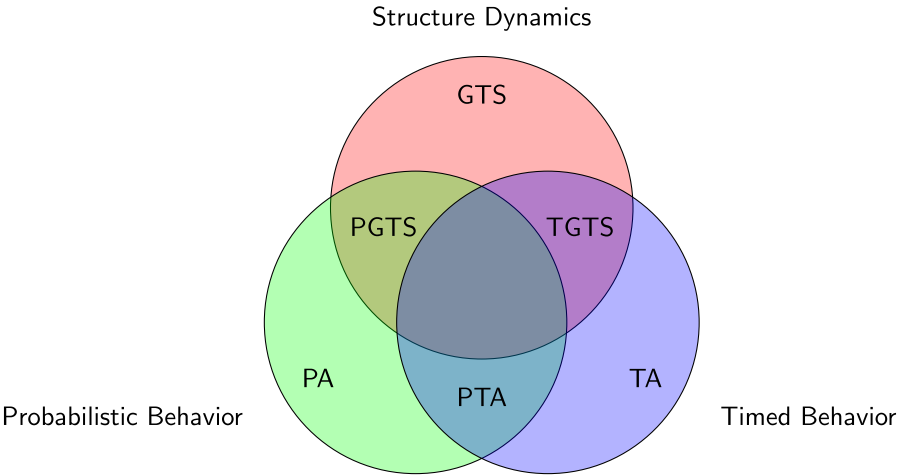
System's Aspects and Modeling Formalisms



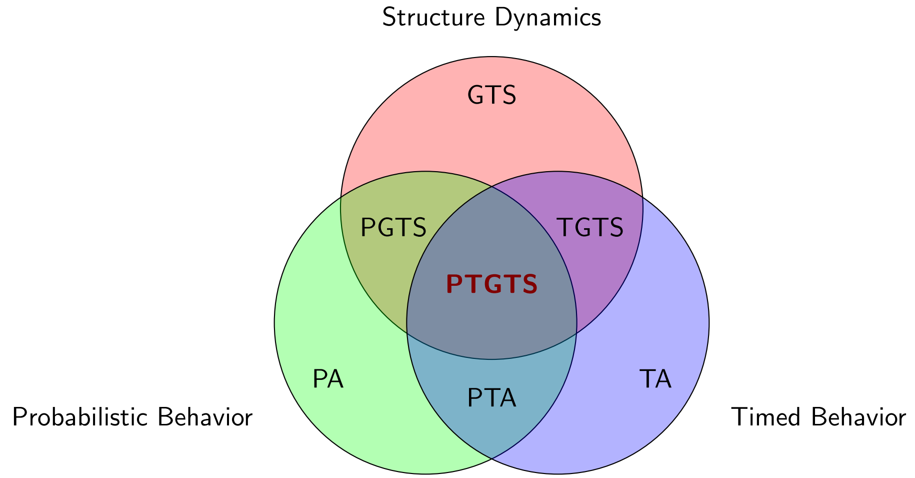
System's Aspects and Modeling Formalisms



System's Aspects and Modeling Formalisms



System's Aspects and Modeling Formalisms



System's Aspects and Modeling Formalisms

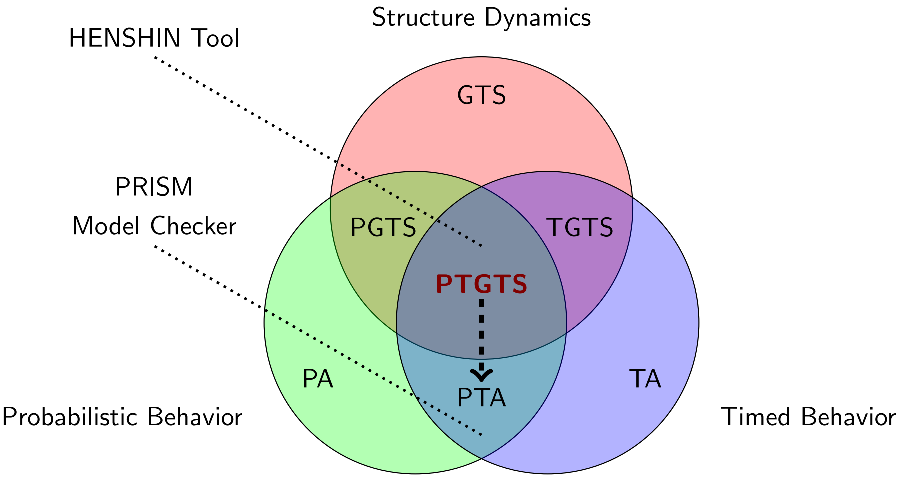
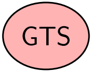


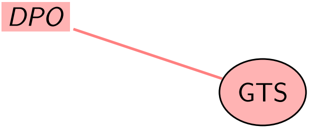
Table of Contents

Table of Contents

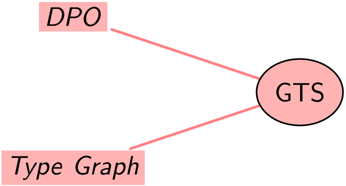
From GTS to PTGTS



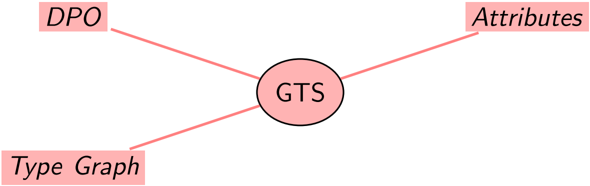
From GTS to PTGTS



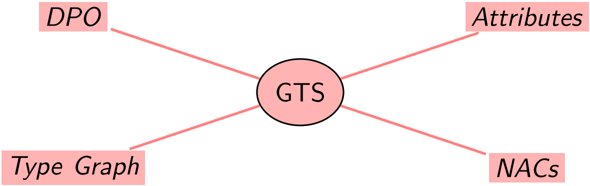
From GTS to PTGTS



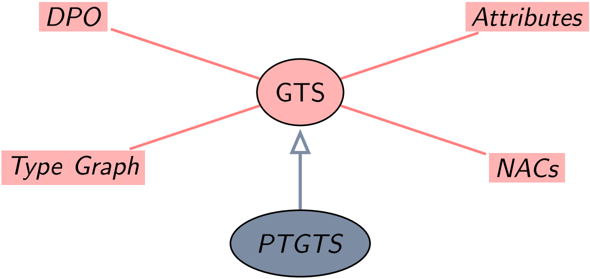
From GTS to PTGTS



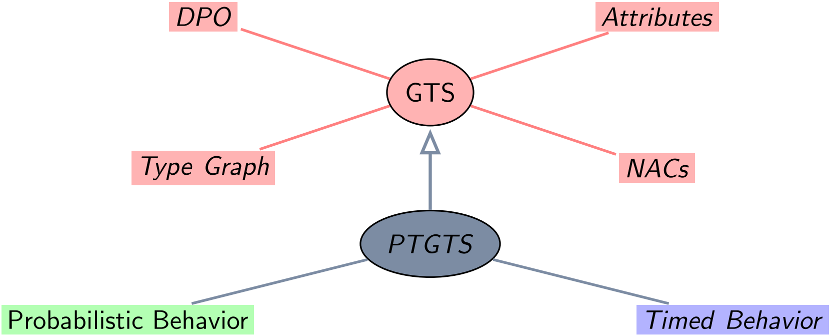
From GTS to PTGTS



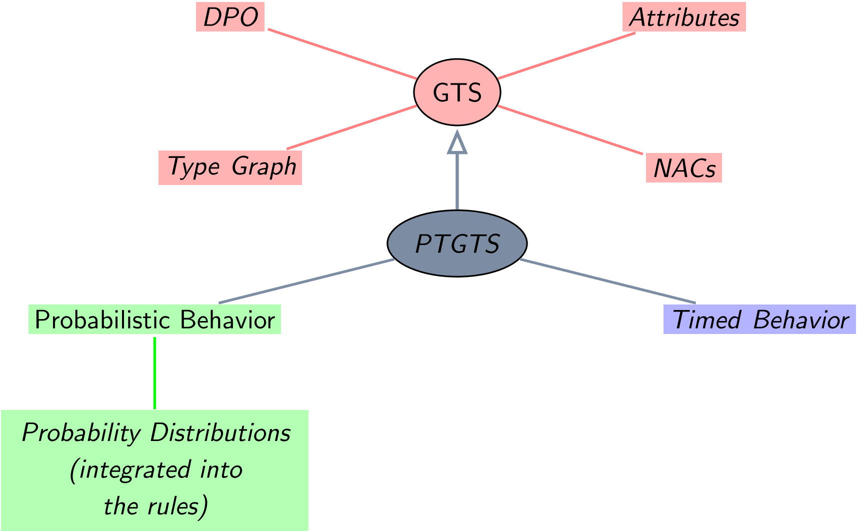
From GTS to PTGTS



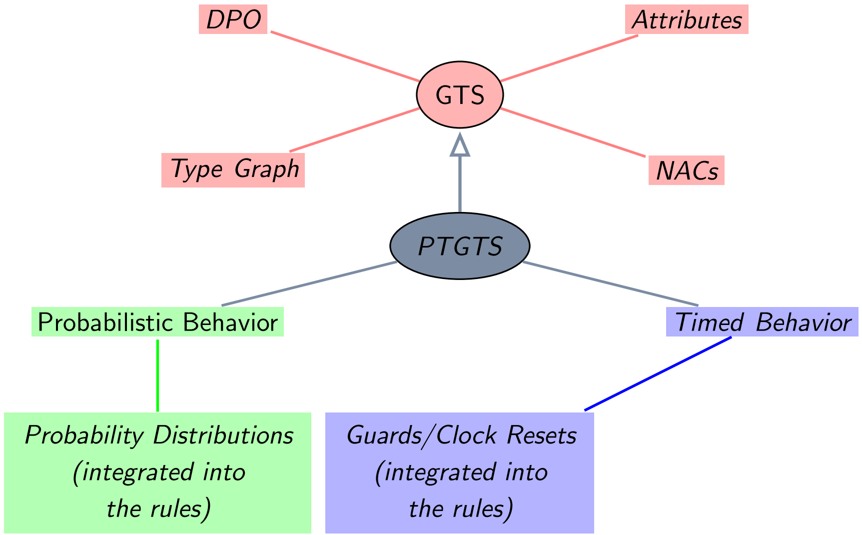
From GTS to PTGTS



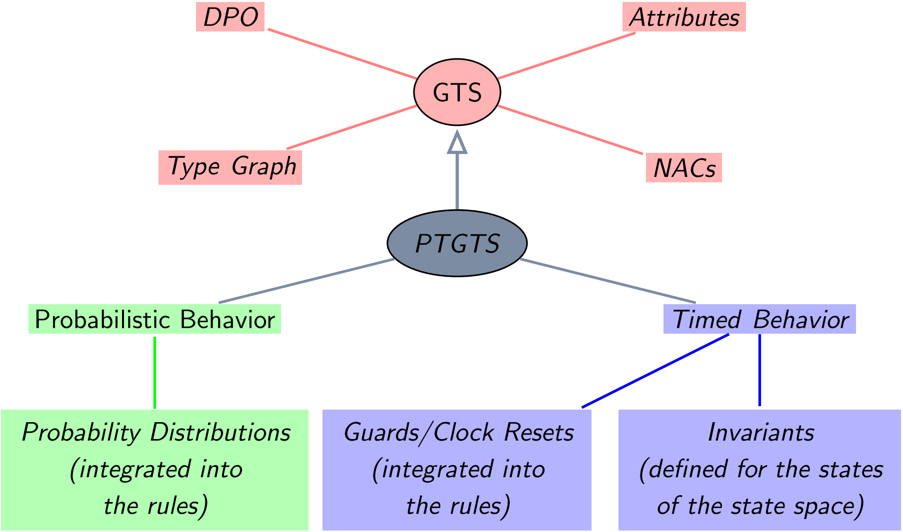
From GTS to PTGTS



From GTS to PTGTS



From GTS to PTGTS



Formal Model

Definition (Probabilistic Timed Graph Transformation System)

$S = (TG, G_0, v_0, \Pi, I, AP, prio)$ is a *PTGTS* if

Formal Model

Definition (Probabilistic Timed Graph Transformation System)

$S = (TG, G_0, v_0, \Pi, I, AP, prio)$ is a *PTGTS* if

- TG is a finite type graph including the type node *Clock*,

Definition (Probabilistic Timed Graph Transformation System)

$S = (TG, G_0, v_0, \Pi, I, AP, prio)$ is a *PTGTS* if

- TG is a finite type graph including the type node *Clock*,
- G_0 is a finite initial graph over TG,

Formal Model

Definition (Probabilistic Timed Graph Transformation System)

$S = (TG, G_0, v_0, \Pi, I, AP, prio)$ is a *PTGTS* if

- TG is a finite type graph including the type node *Clock*,
- G_0 is a finite initial graph over TG,
- $v_0 : ClockNodes(G_0) \rightarrow \mathbb{R}_0^+$ is the initial clock valuation,

Formal Model

Definition (Probabilistic Timed Graph Transformation System)

$S = (TG, G_0, v_0, \Pi, I, AP, prio)$ is a *PTGTS* if

- TG is a finite type graph including the type node *Clock*,
- G_0 is a finite initial graph over TG,
- $v_0 : ClockNodes(G_0) \to \mathbb{R}_0^+$ is the initial clock valuation,
- Π is a finite set of transformation rules,

Formal Model

Definition (Probabilistic Timed Graph Transformation System)

$S = (TG, G_0, v_0, \Pi, I, AP, prio)$ is a *PTGTS* if

- TG is a finite type graph including the type node *Clock*,
- G_0 is a finite initial graph over TG,
- $v_0 : ClockNodes(G_0) \rightarrow \mathbb{R}_0^+$ is the initial clock valuation,
- Π is a finite set of transformation rules,
- I is a finite set of invariants,

Formal Model

Definition (Probabilistic Timed Graph Transformation System)

$S = (TG, G_0, v_0, \Pi, I, AP, prio)$ is a *PTGTS* if

- TG is a finite type graph including the type node *Clock*,
- G_0 is a finite initial graph over TG,
- $v_0 : ClockNodes(G_0) \to \mathbb{R}_0^+$ is the initial clock valuation,
- Π is a finite set of transformation rules,
- I is a finite set of invariants,
- AP is a finite set of atomic propositions,

Formal Model

Definition (Probabilistic Timed Graph Transformation System)

$S = (TG, G_0, v_0, \Pi, I, AP, prio)$ is a *PTGTS* if

- TG is a finite type graph including the type node $Clock$,
- G_0 is a finite initial graph over TG,
- $v_0 : ClockNodes(G_0) \to \mathbb{R}_0^+$ is the initial clock valuation,
- Π is a finite set of transformation rules,
- I is a finite set of invariants,
- AP is a finite set of atomic propositions,
- $prio : \Pi \to \mathbb{N}$ is a priority function.

Formal Model

Definition (Probabilistic Timed Graph Transformation System)

$S = (TG, G_0, v_0, \Pi, I, AP, prio)$ is a *PTGTS* if

TG is a finite type graph including the type node *Clock*,	GTS		
G_0 is a finite initial graph over TG,	GTS		
$v_0 : ClockNodes(G_0) \to \mathbb{R}_0^+$ is the initial clock valuation,		Time	
Π is a finite set of transformation rules,	GTS	Time	Prob
I is a finite set of invariants,		Time	
AP is a finite set of atomic propositions,	GTS		
$prio : \Pi \to \mathbb{N}$ is a priority function.	GTS		

Formal Model

Definition (Probabilistic Timed Graph Transformation Rule)

$R = (L, P, \mu, \phi, r_C)$ is a *PTGT rule* if

Formal Model

Definition (Probabilistic Timed Graph Transformation Rule)

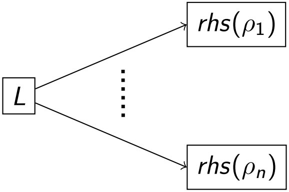
$R = (L, P, \mu, \phi, r_C)$ is a *PTGT rule* if

- L is a common left-hand side graph,

Definition (Probabilistic Timed Graph Transformation Rule)

$R = (L, P, \mu, \phi, r_C)$ is a *PTGT rule* if

- L is a common left-hand side graph,
- P is a finite set of graph transformation rules ρ with $lhs(\rho) = L$,

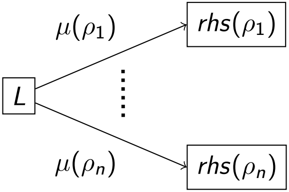


$$P = \{\rho_1, \ldots, \rho_n\}$$

Definition (Probabilistic Timed Graph Transformation Rule)

$R = (L, P, \mu, \phi, r_C)$ is a *PTGT rule* if

- L is a common left-hand side graph,
- P is a finite set of graph transformation rules ρ with $lhs(\rho) = L$,
- $\mu \in Dist(P)$ is a probability distribution,



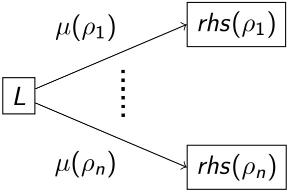
$$P = \{\rho_1, \ldots, \rho_n\}$$

Formal Model

Definition (Probabilistic Timed Graph Transformation Rule)

$R = (L, P, \mu, \phi, r_C)$ is a *PTGT rule* if

- L is a common left-hand side graph,
- P is a finite set of graph transformation rules ρ with $lhs(\rho) = L$,
- $\mu \in Dist(P)$ is a probability distribution,
- $\phi \in \Phi(ClockNodes(L))$ is a guard over nodes of the type *Clock*,



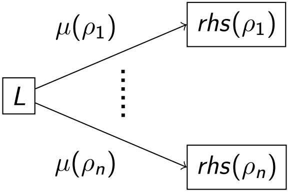
$$P = \{\rho_1, \ldots, \rho_n\}$$

Formal Model

Definition (Probabilistic Timed Graph Transformation Rule)

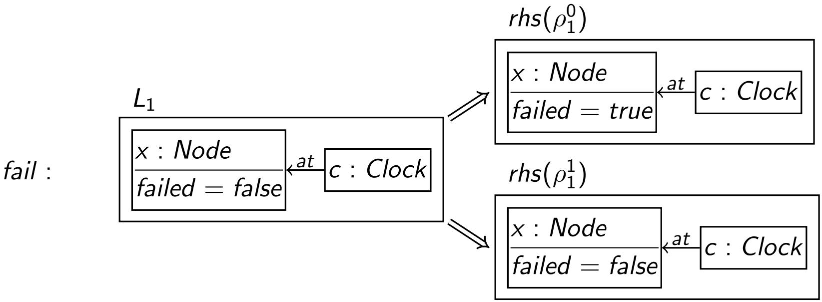
$R = (L, P, \mu, \phi, r_C)$ is a *PTGT rule* if

- L is a common left-hand side graph,
- P is a finite set of graph transformation rules ρ with $lhs(\rho) = L$,
- $\mu \in Dist(P)$ is a probability distribution,
- $\phi \in \Phi(ClockNodes(L))$ is a guard over nodes of the type *Clock*,
- $r_C \subseteq ClockNodes(L)$ is a set of nodes of the type *Clock* to be reset.

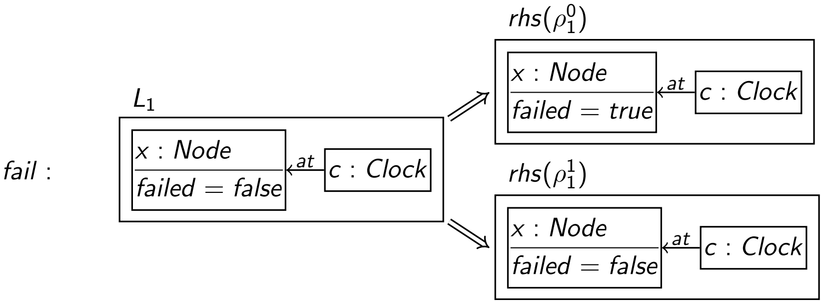


$P = \{\rho_1, \ldots, \rho_n\}$

Example for a PTGT Rule



Example for a PTGT Rule



$$
\begin{aligned}
\mathit{fail} &= (L_1, P_1, \mu_1, \phi_1, r_{C_1}) \text{ with} \\
P_1 &= \{\rho_1^0, \rho_1^1\} \\
\mu_1 &= \{(\rho_1^0, 0.1), (\rho_1^1, 0.9)\} \\
\phi_1 &= (c \geq 2) \\
r_{C_1} &= \{c\}
\end{aligned}
$$

Formal Model

Definition (Probabilistic Timed Graph Transformation Invariant)

$\Theta = (G, \phi)$ is a *PTGT invariant* if

Formal Model

Definition (Probabilistic Timed Graph Transformation Invariant)

$\Theta = (G, \phi)$ is a *PTGT invariant* if

- G is a graph,

Formal Model

Definition (Probabilistic Timed Graph Transformation Invariant)

$\Theta = (G, \phi)$ is a *PTGT invariant* if

- G is a graph,
- $\phi \in \Phi(ClockNodes(G))$ is an invariant formula over nodes of the type *Clock*.

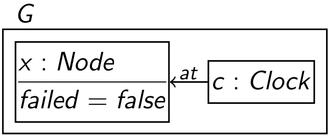
Formal Model

Definition (Probabilistic Timed Graph Transformation Invariant)

$\Theta = (G, \phi)$ is a *PTGT invariant* if

- G is a graph,
- $\phi \in \Phi(ClockNodes(G))$ is an invariant formula over nodes of the type *Clock*.

Θ_{fail} :



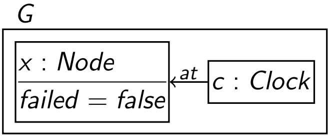
Formal Model

Definition (Probabilistic Timed Graph Transformation Invariant)

$\Theta = (G, \phi)$ is a *PTGT invariant* if

- G is a graph,
- $\phi \in \Phi(ClockNodes(G))$ is an invariant formula over nodes of the type *Clock*.

Θ_{fail} :



$$\Theta_{fail} = (G, \phi) \text{ with } \phi = (c \leq 5)$$

Formal Model

- APs set the appropriate labels to the PTGTS states.

Formal Model

- APs set the appropriate labels to the PTGTS states.
- An AP is of the form $(G, \phi = \mathit{true})$ for a graph G.

Formal Model

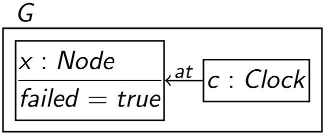
- APs set the appropriate labels to the PTGTS states.
- An AP is of the form $(G, \phi = \mathit{true})$ for a graph G.
- APs do not influence the semantics of PTGTS.

Formal Model

- APs set the appropriate labels to the PTGTS states.
- An AP is of the form $(G, \phi = \mathit{true})$ for a graph G.
- APs do not influence the semantics of PTGTS.
- APs are used to state PTCTL properties.

- APs set the appropriate labels to the PTGTS states.
- An AP is of the form $(G, \phi = true)$ for a graph G.
- APs do not influence the semantics of PTGTS.
- APs are used to state PTCTL properties.

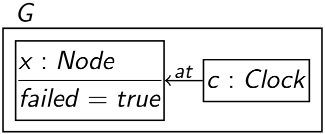
AP_{failed} :



Formal Model

- APs set the appropriate labels to the PTGTS states.
- An AP is of the form $(G, \phi = \mathit{true})$ for a graph G.
- APs do not influence the semantics of PTGTS.
- APs are used to state PTCTL properties.

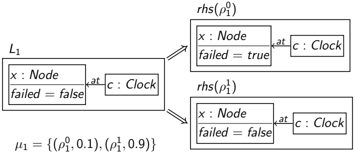
AP_{failed} :



$$AP_{failed} = (G, \phi) \text{ with } \phi = \mathit{true}$$

Semantics of PTGTS

fail :

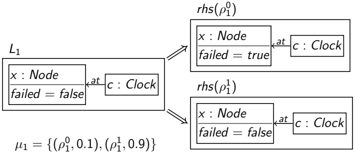


$$\mu_1 = \{(\rho_1^0, 0.1), (\rho_1^1, 0.9)\}$$

$$\phi_1 = (c \geq 2)$$

$$r_{C_1} = \{c\}$$

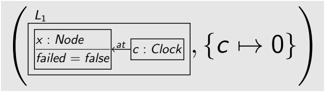
fail :



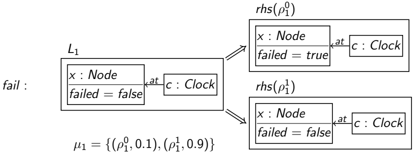
$\mu_1 = \{(\rho_1^0, 0.1), (\rho_1^1, 0.9)\}$

$\phi_1 = (c \geq 2)$

$r_{C_1} = \{c\}$



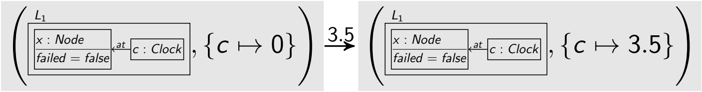
Semantics of PTGTS



$$\mu_1 = \{(\rho_1^0, 0.1), (\rho_1^1, 0.9)\}$$

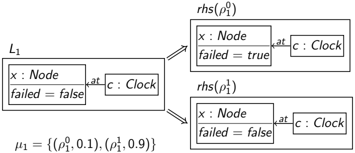
$$\phi_1 = (c \geq 2)$$

$$r_{C_1} = \{c\}$$



Semantics of PTGTS

fail :

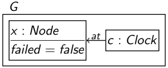


$\mu_1 = \{(\rho_1^0, 0.1), (\rho_1^1, 0.9)\}$

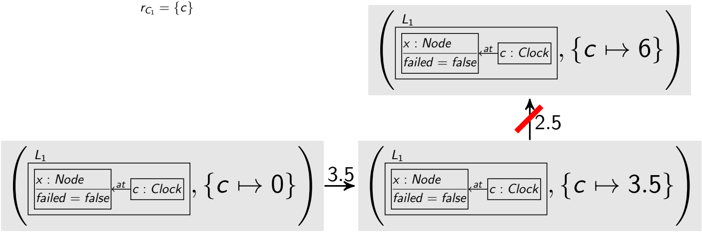
$\phi_1 = (c \geq 2)$

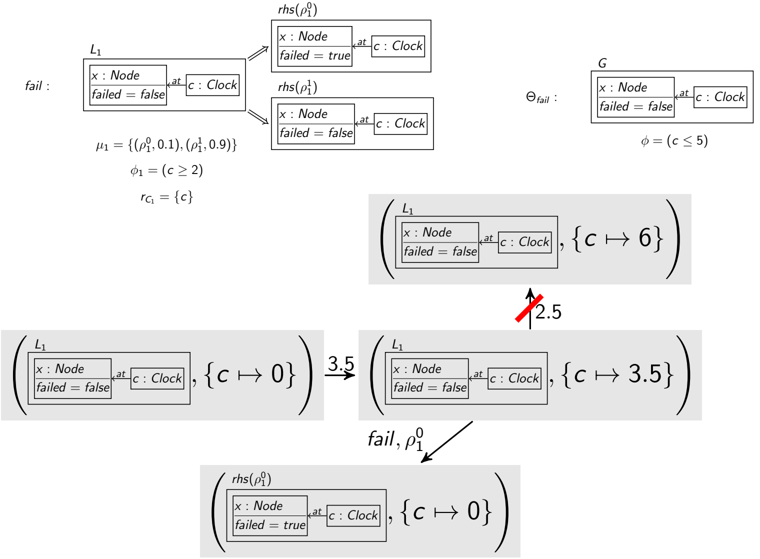
$r_{C_1} = \{c\}$

Θ_{fail} :



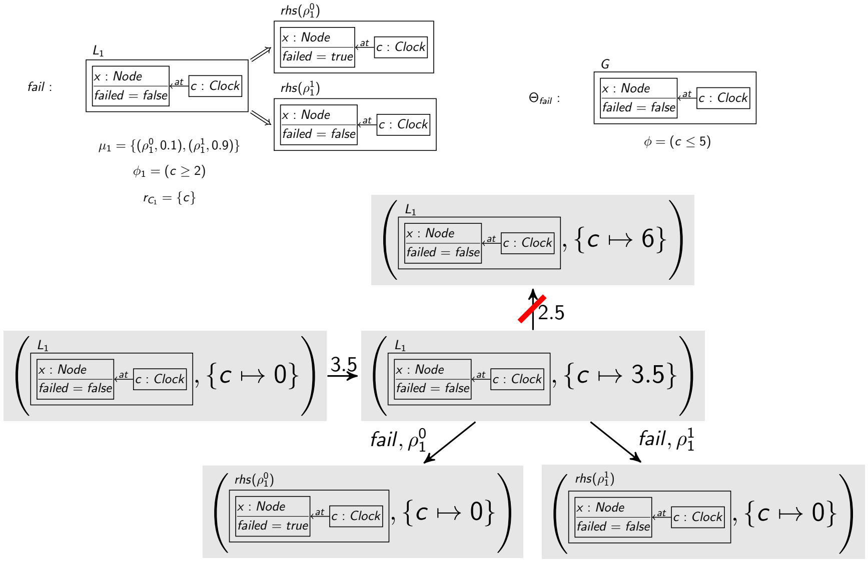
$\phi = (c \leq 5)$





fail :
L_1
x : Node
failed = false
at
c : Clock
$rhs(\rho_1^0)$
x : Node
failed = true
$rhs(\rho_1^1)$
$\mu_1 = \{(\rho_1^0, 0.1), (\rho_1^1, 0.9)\}$
$\phi_1 = (c \geq 2)$
$r_{C_1} = \{c\}$
Θ_{fail} :
G
$\phi = (c \leq 5)$
$\{c \mapsto 0\}$
3.5
$\{c \mapsto 3.5\}$
2.5
$\{c \mapsto 6\}$
$fail, \rho_1^0$

Semantics of PTGTS



Semantics of PTGTS

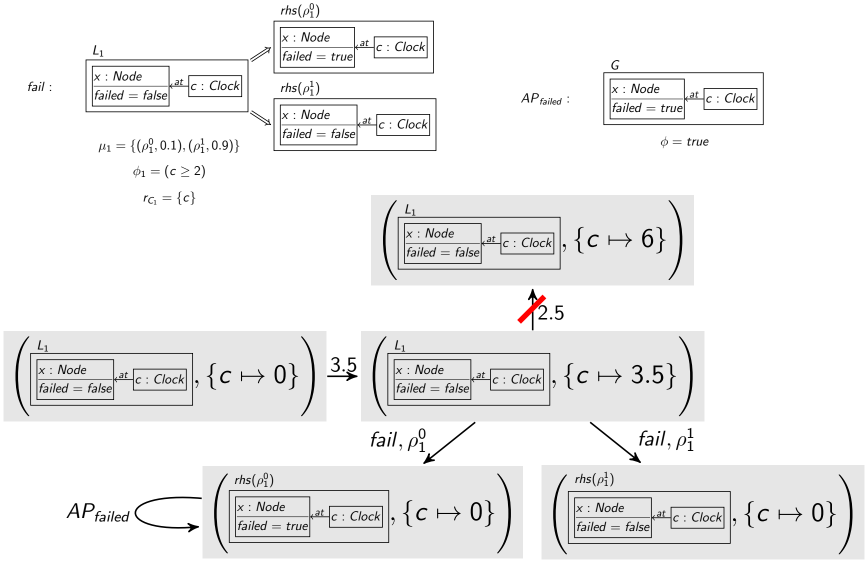
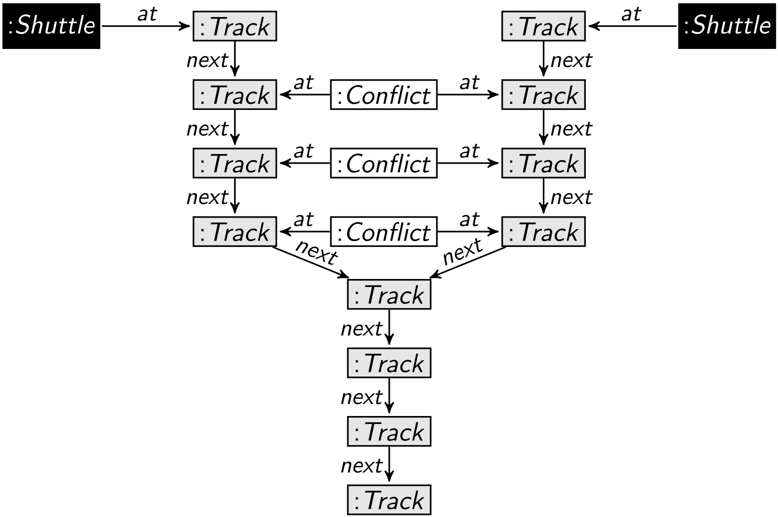
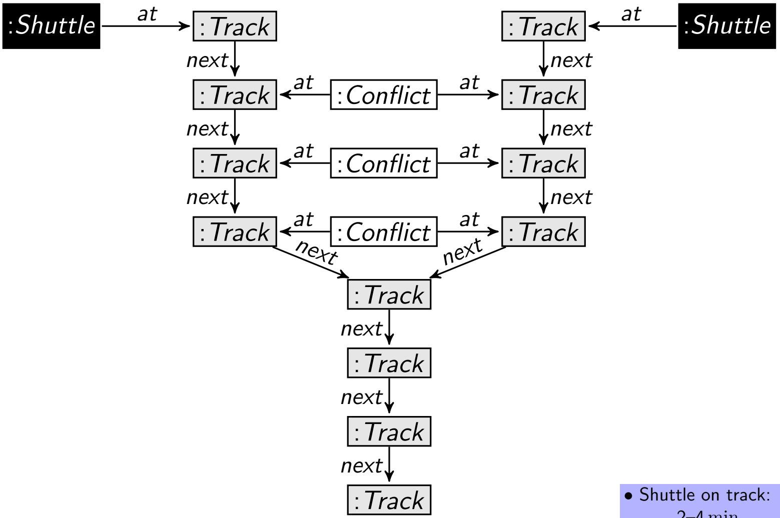


Table of Contents

Shuttle Scenario

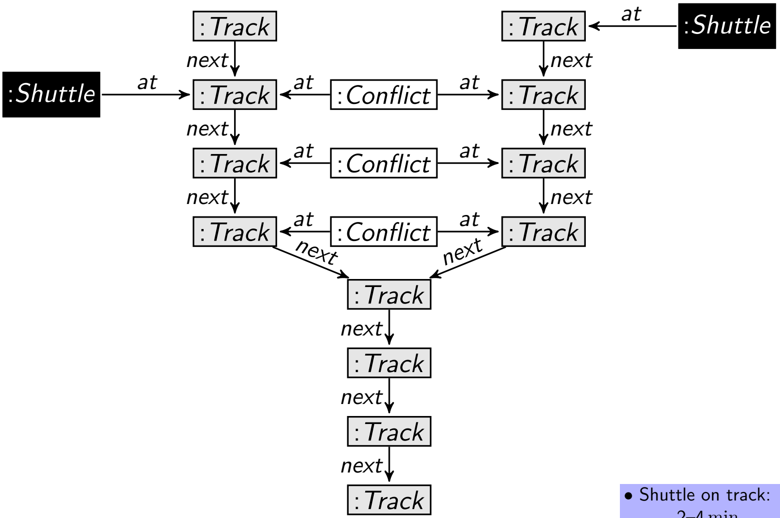


Shuttle Scenario

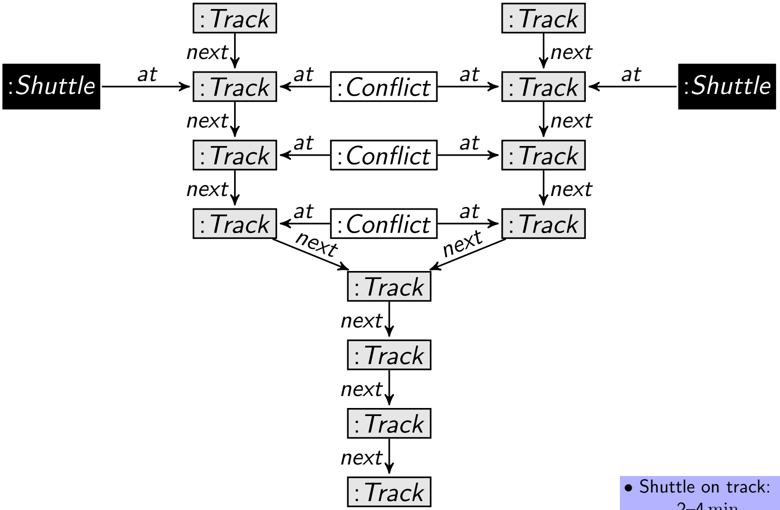


- Shuttle on track:
  2–4 min

Shuttle Scenario

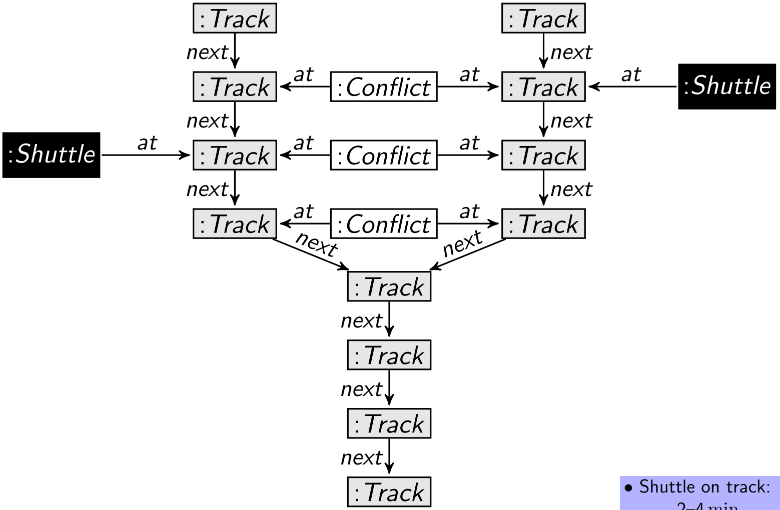


- Shuttle on track: 2–4 min



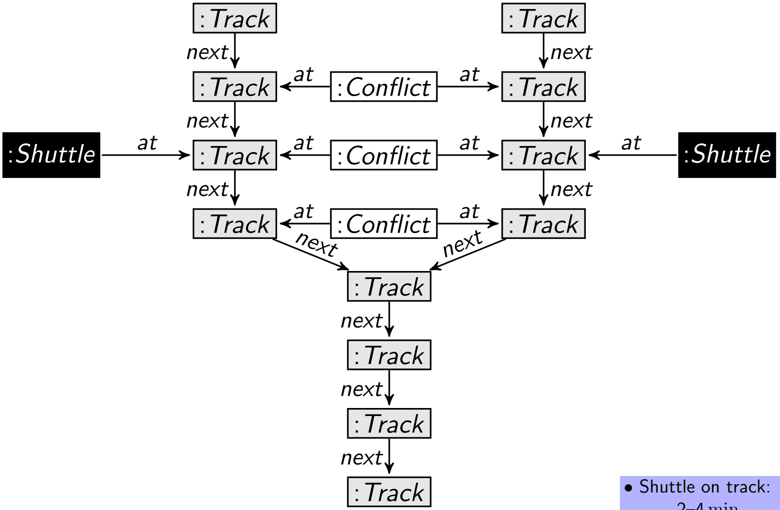
- Shuttle on track: 2–4 min

Shuttle Scenario



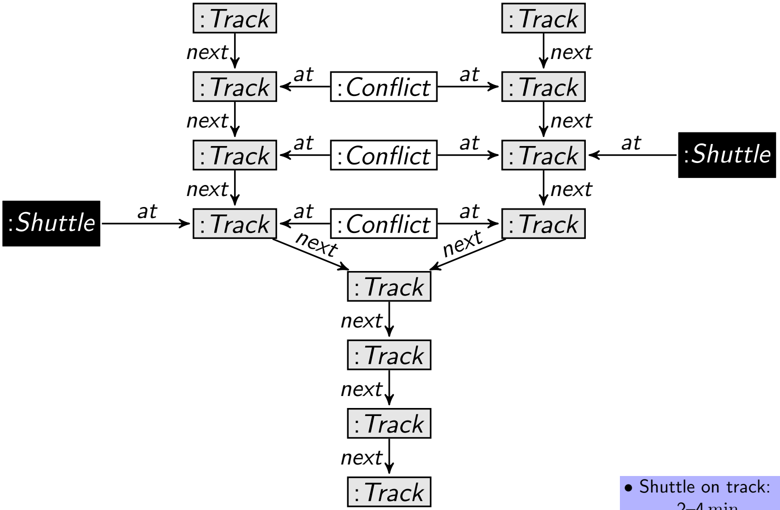
- Shuttle on track:
  2–4 min

Shuttle Scenario



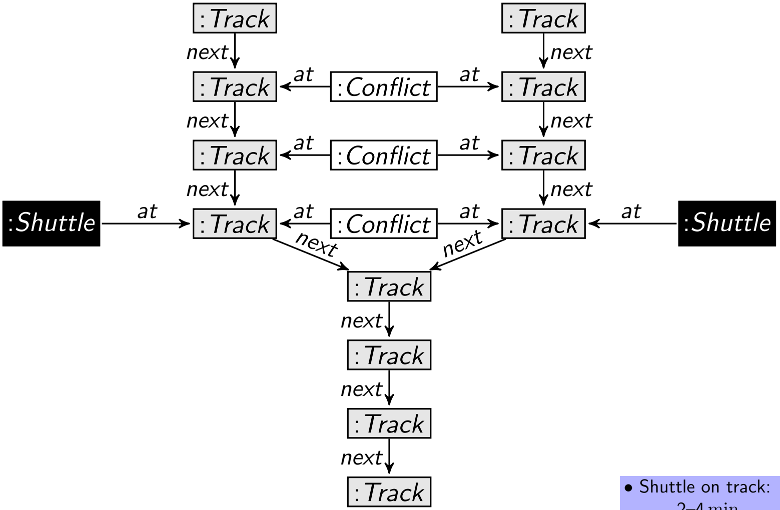
- Shuttle on track:
  2–4 min

Shuttle Scenario



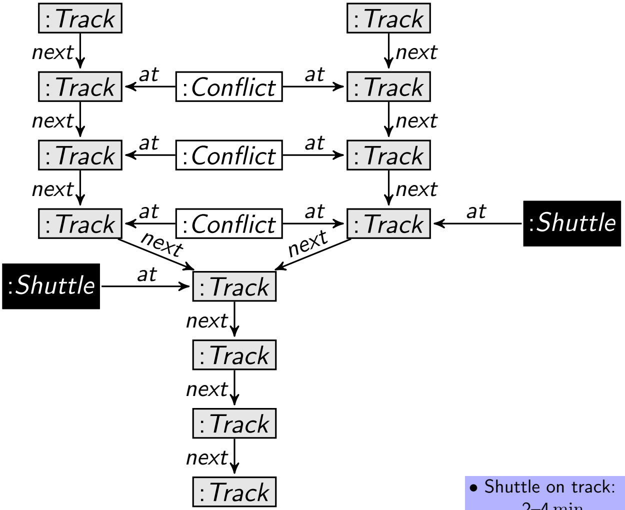
- Shuttle on track: 2–4 min

Shuttle Scenario



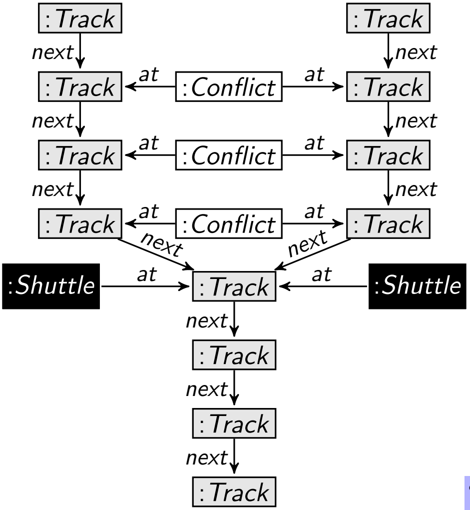
- Shuttle on track: 2–4 min

Shuttle Scenario



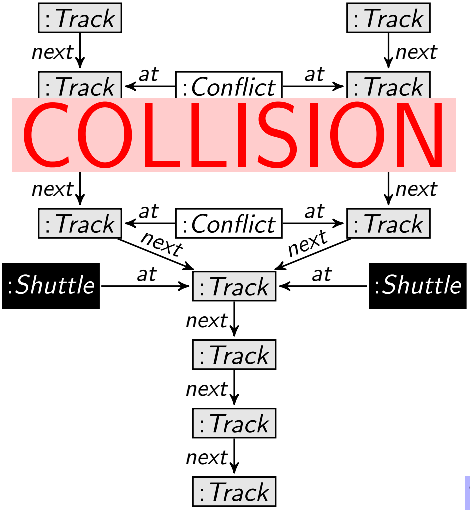
- Shuttle on track: 2–4 min

Shuttle Scenario



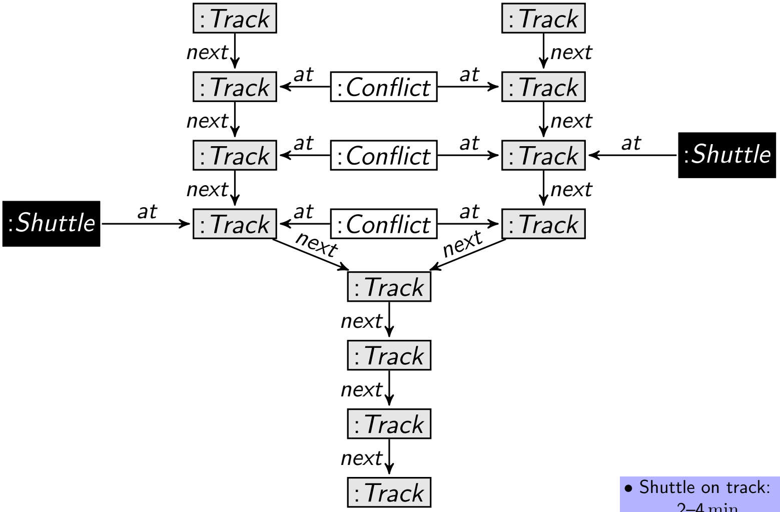
- Shuttle on track:
  2–4 min

Shuttle Scenario



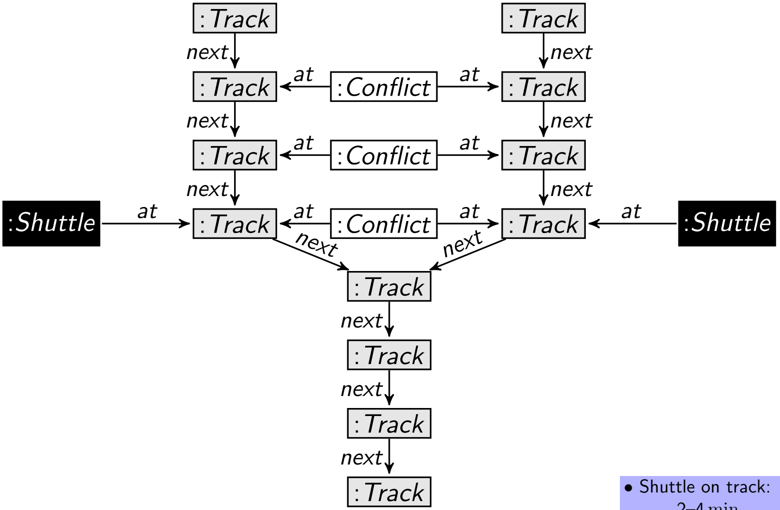
- Shuttle on track: 2–4 min

Shuttle Scenario



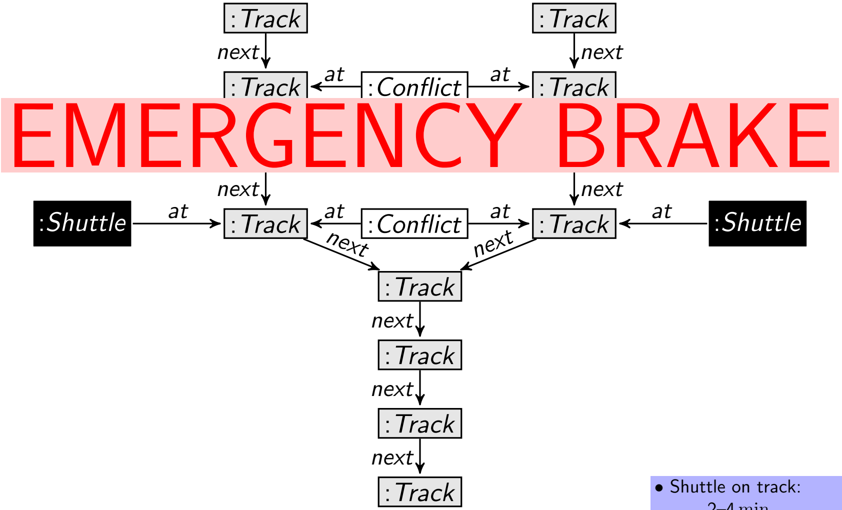
- Shuttle on track: 2–4 min

Shuttle Scenario



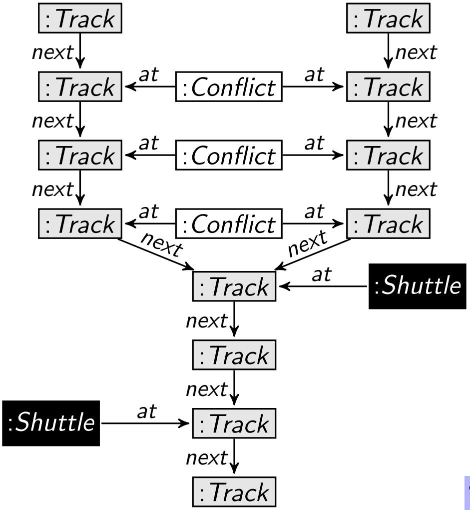
- Shuttle on track: 2–4 min

Shuttle Scenario



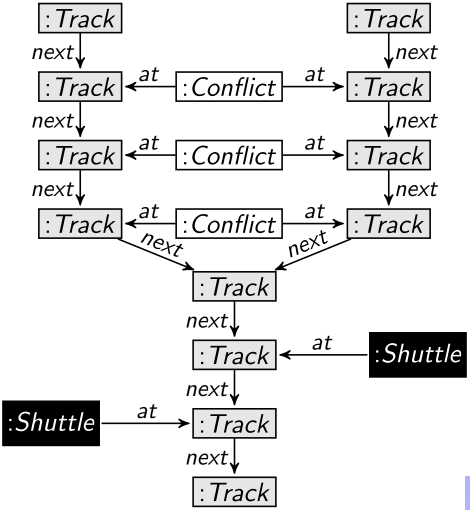
- Shuttle on track:
  2–4 min

Shuttle Scenario



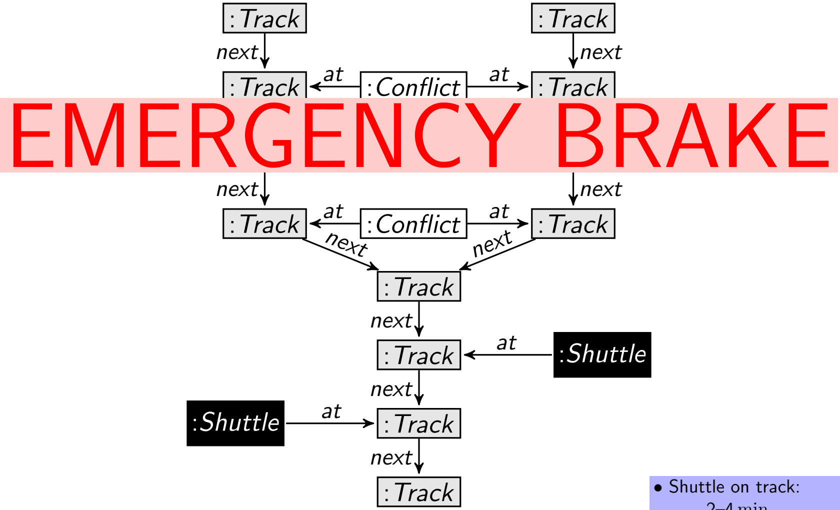
- Shuttle on track: 2–4 min

Shuttle Scenario



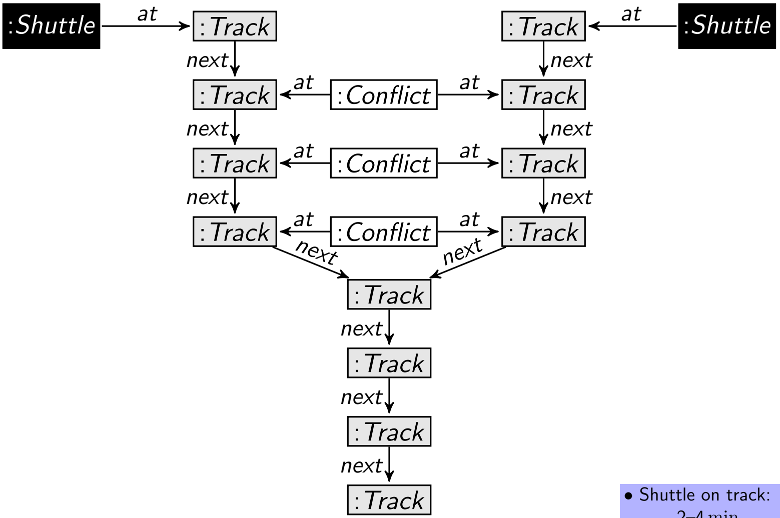
- Shuttle on track: 2–4 min

Shuttle Scenario



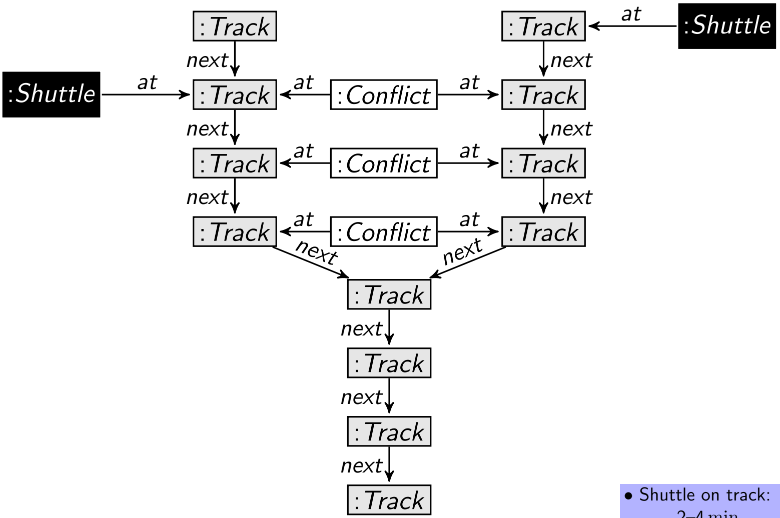
- Shuttle on track: 2–4 min

Shuttle Scenario



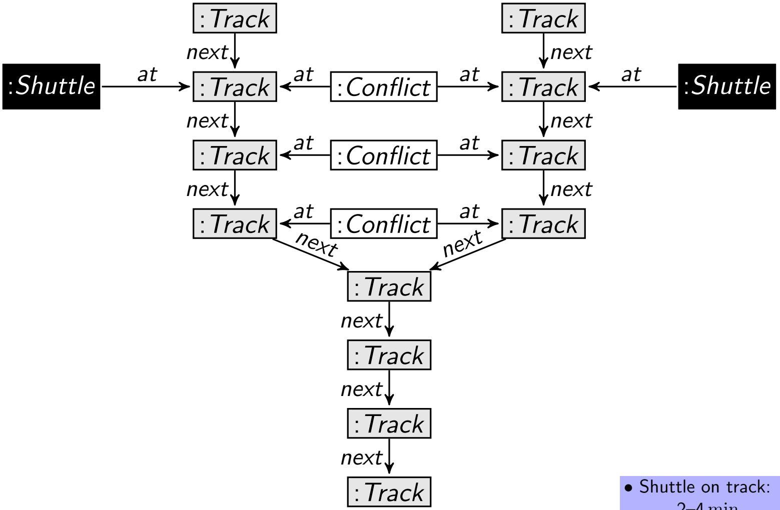
- Shuttle on track:
  2–4 min

Shuttle Scenario



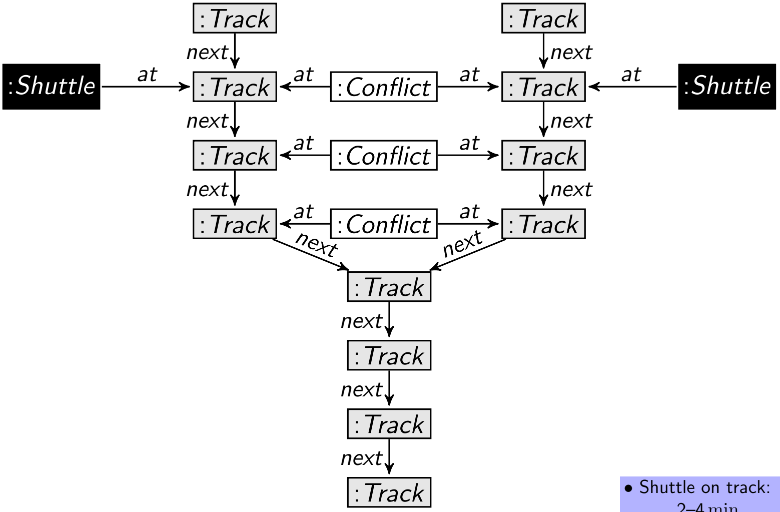
- Shuttle on track:
  2–4 min

Shuttle Scenario



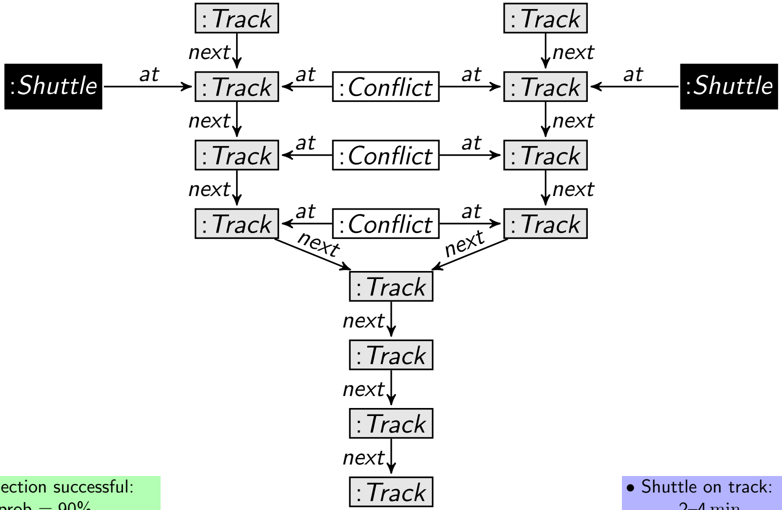
- Shuttle on track:
  2–4 min

Shuttle Scenario



- Shuttle on track:
  2–4 min
- Connection duration:
  2 min

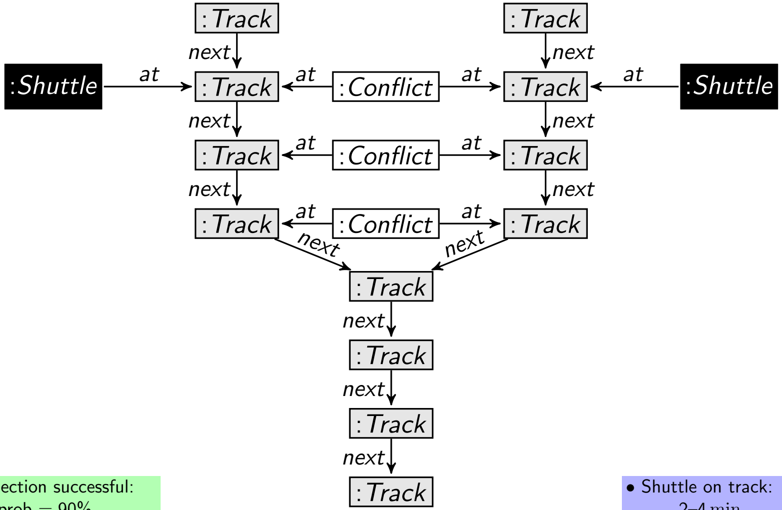
Shuttle Scenario



- Connection successful:
  prob = 90%

- Shuttle on track:
  2–4 min
- Connection duration:
  2 min

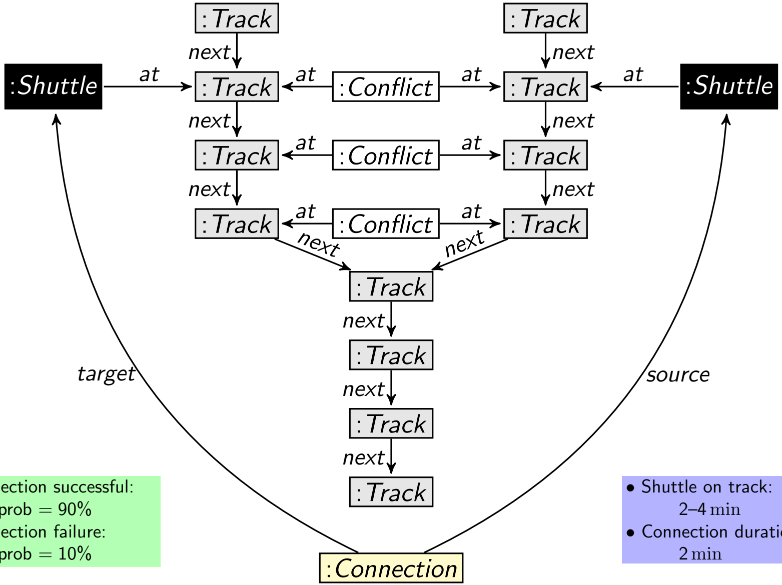
Shuttle Scenario



- Connection successful: prob = 90%
- Connection failure: prob = 10%

- Shuttle on track: 2–4 min
- Connection duration: 2 min

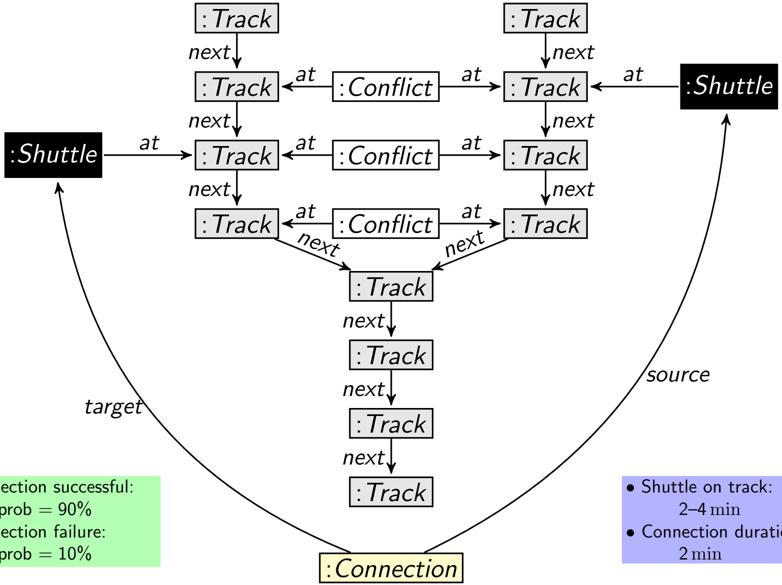
Shuttle Scenario



- Connection successful:
  prob = 90%
- Connection failure:
  prob = 10%

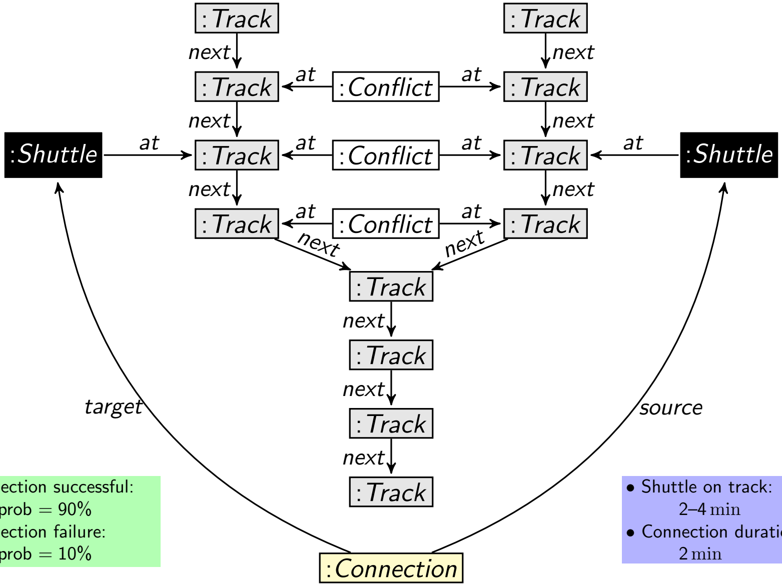
- Shuttle on track:
  2–4 min
- Connection duration:
  2 min

Shuttle Scenario



- Connection successful:
  prob = 90%
- Connection failure:
  prob = 10%

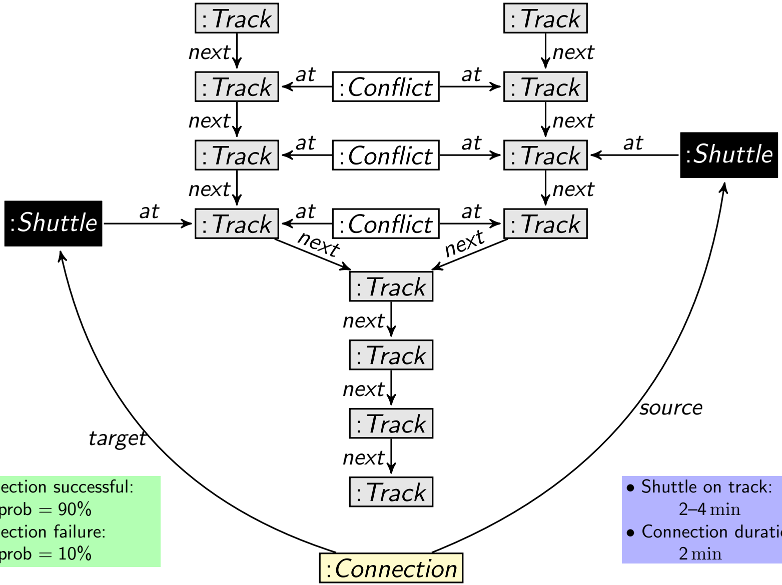
- Shuttle on track:
  2–4 min
- Connection duration:
  2 min



- Connection successful:
  prob = 90%
- Connection failure:
  prob = 10%

- Shuttle on track:
  2–4 min
- Connection duration:
  2 min

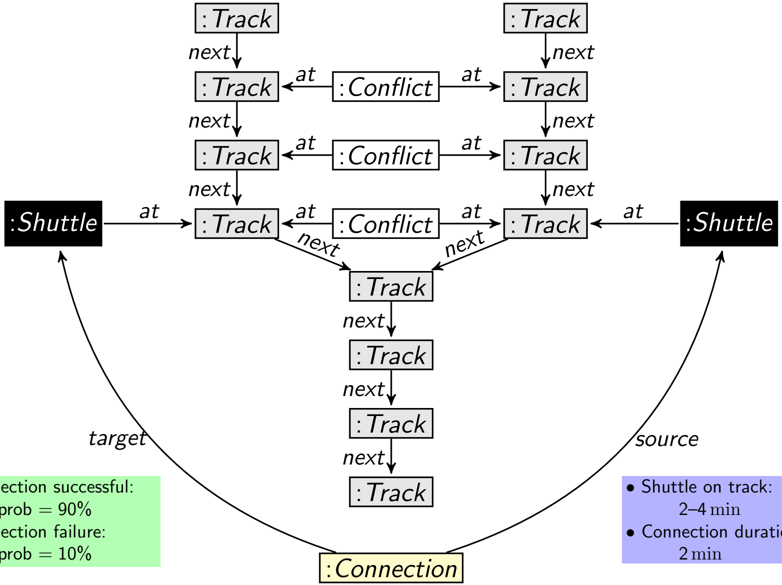
Shuttle Scenario



- Connection successful:
  prob = 90%
- Connection failure:
  prob = 10%

- Shuttle on track:
  2–4 min
- Connection duration:
  2 min

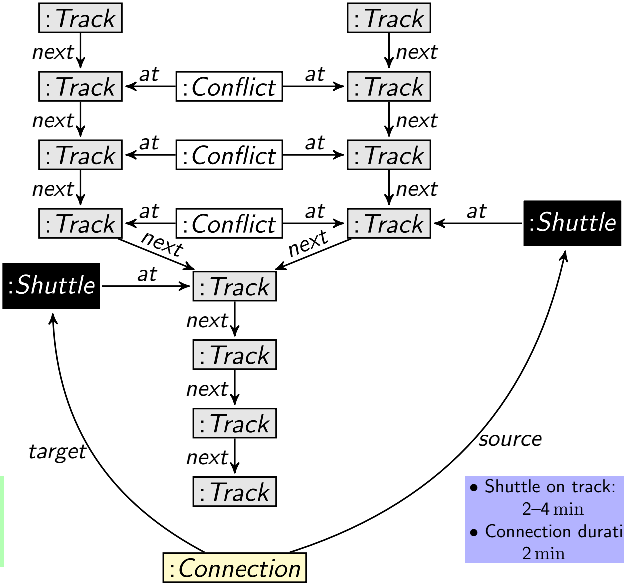
Shuttle Scenario



- Connection successful:
  prob = 90%
- Connection failure:
  prob = 10%

- Shuttle on track:
  2–4 min
- Connection duration:
  2 min

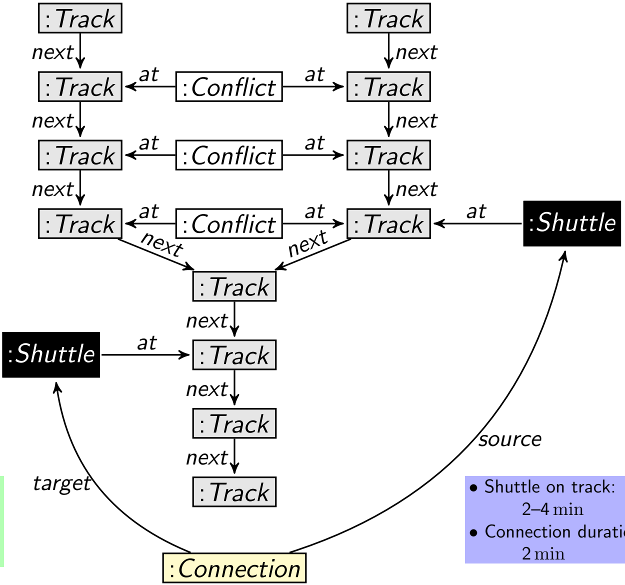
Shuttle Scenario



- Connection successful:
  prob = 90%
- Connection failure:
  prob = 10%

- Shuttle on track:
  2–4 min
- Connection duration:
  2 min

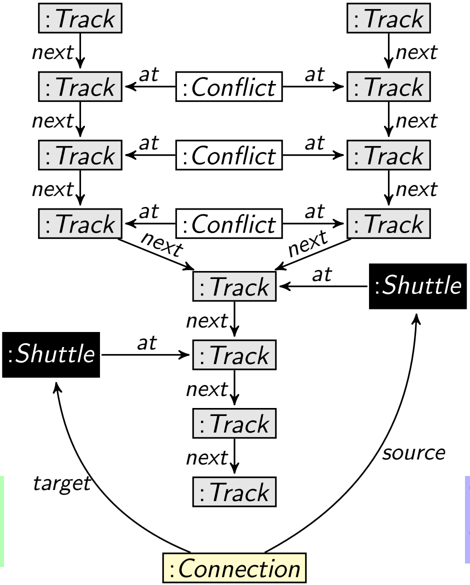
Shuttle Scenario



- Connection successful: prob = 90%
- Connection failure: prob = 10%

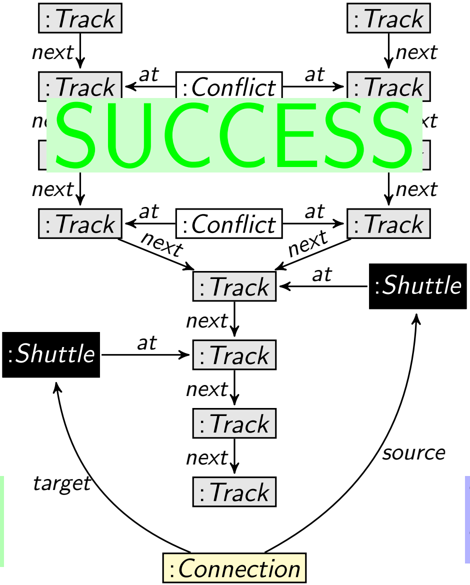
- Shuttle on track: 2–4 min
- Connection duration: 2 min

Shuttle Scenario



- Connection successful:
  prob = 90%
- Connection failure:
  prob = 10%

- Shuttle on track:
  2–4 min
- Connection duration:
  2 min



- Connection successful:
  prob = 90%
- Connection failure:
  prob = 10%

- Shuttle on track:
  2–4 min
- Connection duration:
  2 min

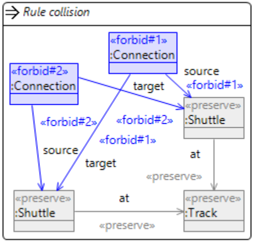
Experiment 1 (Possibility of Collision)

Can the considered shuttle scenario under any circumstances exhibit a collision?

Experiment 1 (Possibility of Collision)

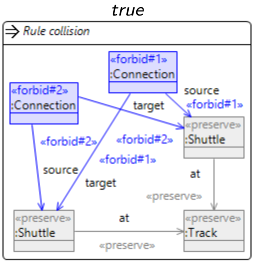
Can the considered shuttle scenario under any circumstances exhibit a collision?

true

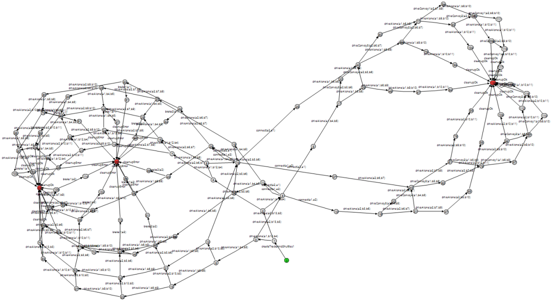


Atomic proposition *collision*

Can the considered shuttle scenario under any circumstances exhibit a collision?



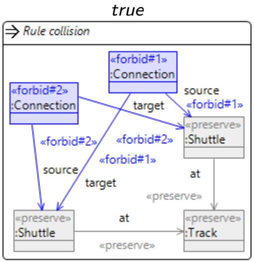
Atomic proposition *collision*



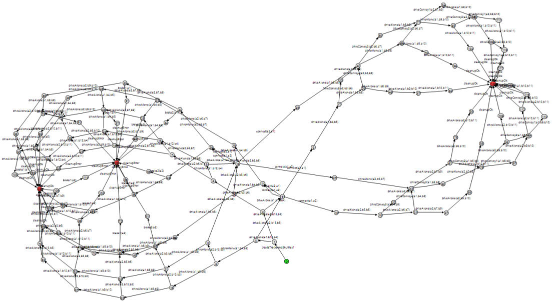
State space for the topology with 3 conflict nodes

Experiment 1 (Possibility of Collision)

Can the considered shuttle scenario under any circumstances exhibit a collision?



Atomic proposition *collision*



State space for the topology with 3 conflict nodes

Result: No collisions are possible.

Experiment 2 (Probability for Emergency Brake)

What is the maximal probability that a shuttle executes an emergency brake?

Experiment 2 (Probability for Emergency Brake)

What is the maximal probability that a shuttle executes an emergency brake?

- State spaces for topologies with 2–6 conflict nodes generated by HENSHIN

Experiment 2 (Probability for Emergency Brake)

What is the maximal probability that a shuttle executes an emergency brake?

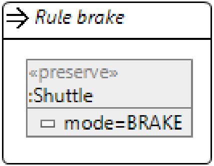
- State spaces for topologies with 2–6 conflict nodes generated by HENSHIN
- State spaces exported to PRISM

Experiment 2 (Probability for Emergency Brake)

What is the maximal probability that a shuttle executes an emergency brake?

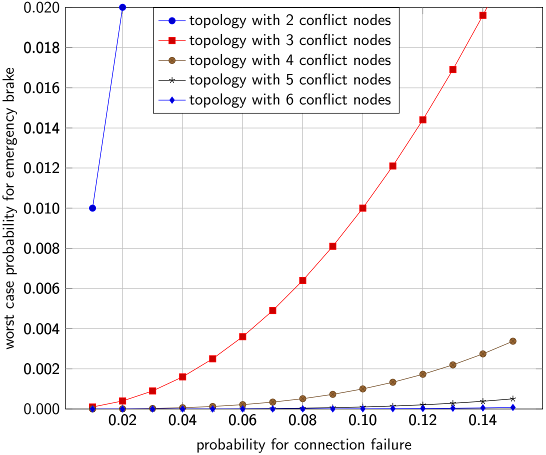
- State spaces for topologies with 2–6 conflict nodes generated by HENSHIN
- State spaces exported to PRISM
- Property for braking behavior in PRISM notation: $Pmax = ?\,[F\ „brake“]$

true

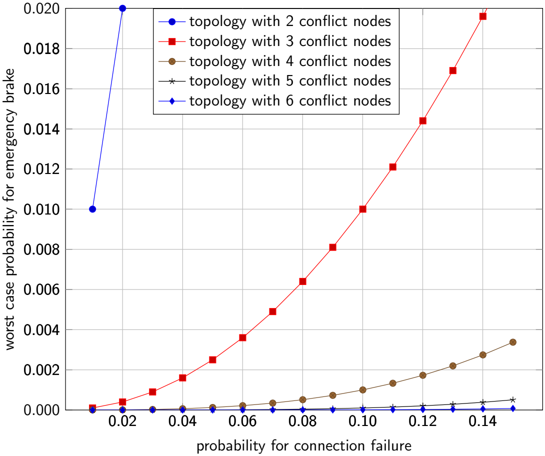


Atomic proposition *brake*

Experiment 2 (Probability for Emergency Brake)



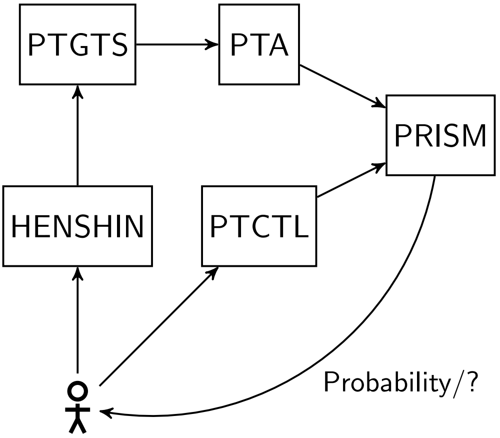
Experiment 2 (Probability for Emergency Brake)



Result: For a desired worst case probability (y-axis) and a communication service quality (x-axis) we can determine the minimal required number of conflict nodes.

Table of Contents

Conclusion



Future Work

- Extension of PTGTSs to interval probabilities (and therefore to IPTA)
- Extension of PTCTL to path properties to specify structure dynamics
- Extension of PRISM to verify path properties of PTGTSs

Thank You For Your Attention