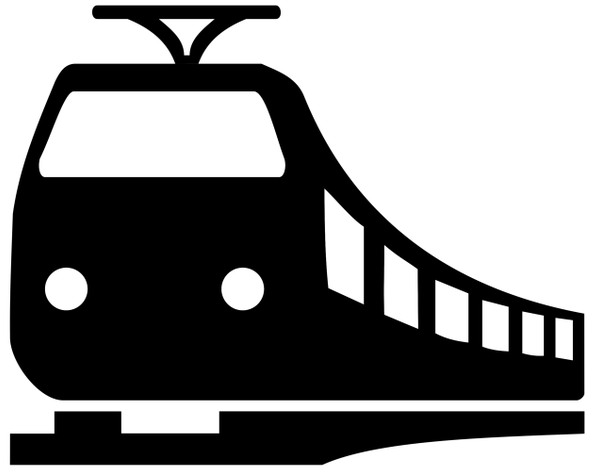
Network Navigation with Online Delays is PSPACE-complete

Thomas Depian, Christoph Kern, **Sebastian Röder**, Manuel Sorge, Soeren Terziadis, Markus Wallinger

04.06.2025



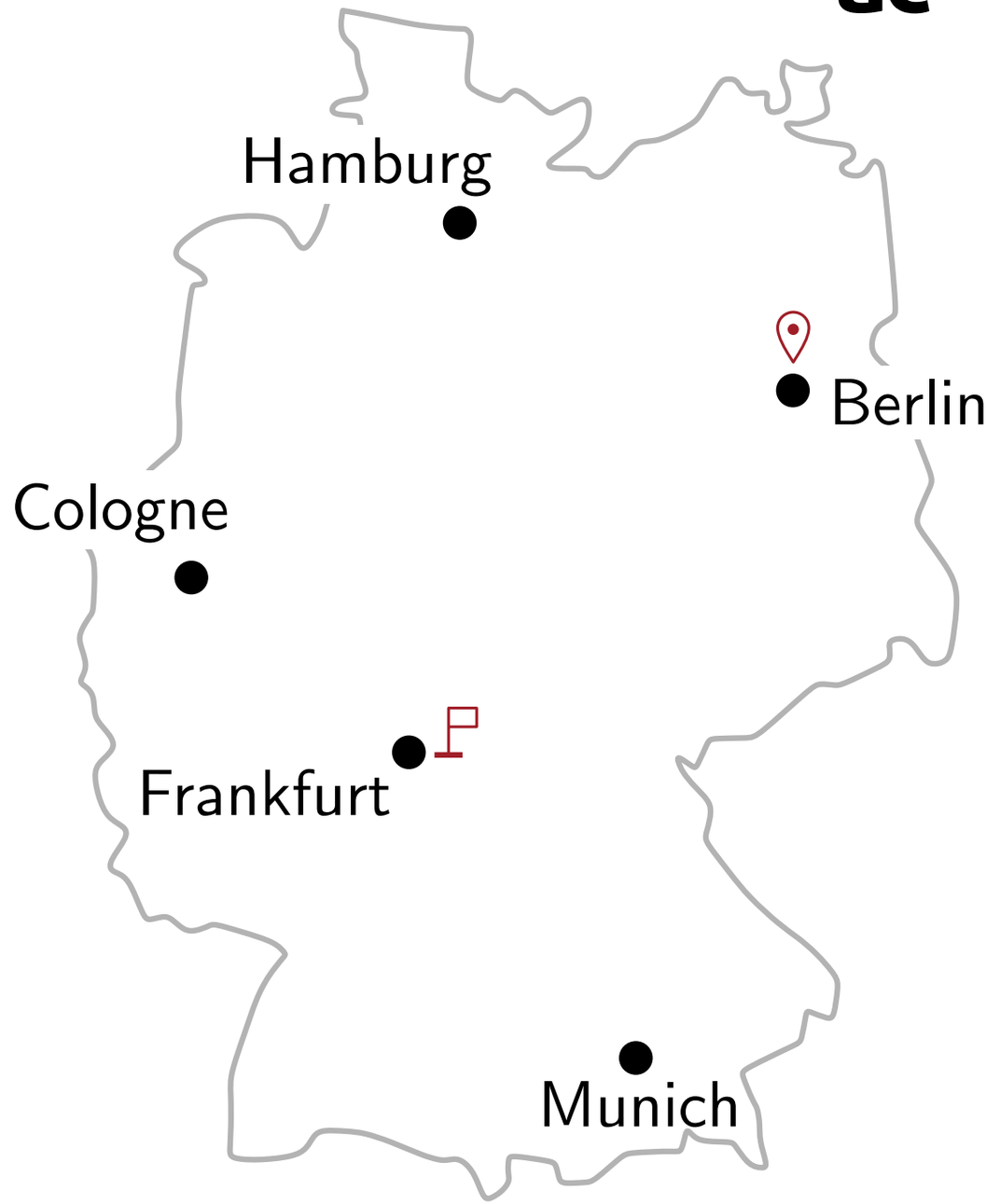
Introduction and Motivation





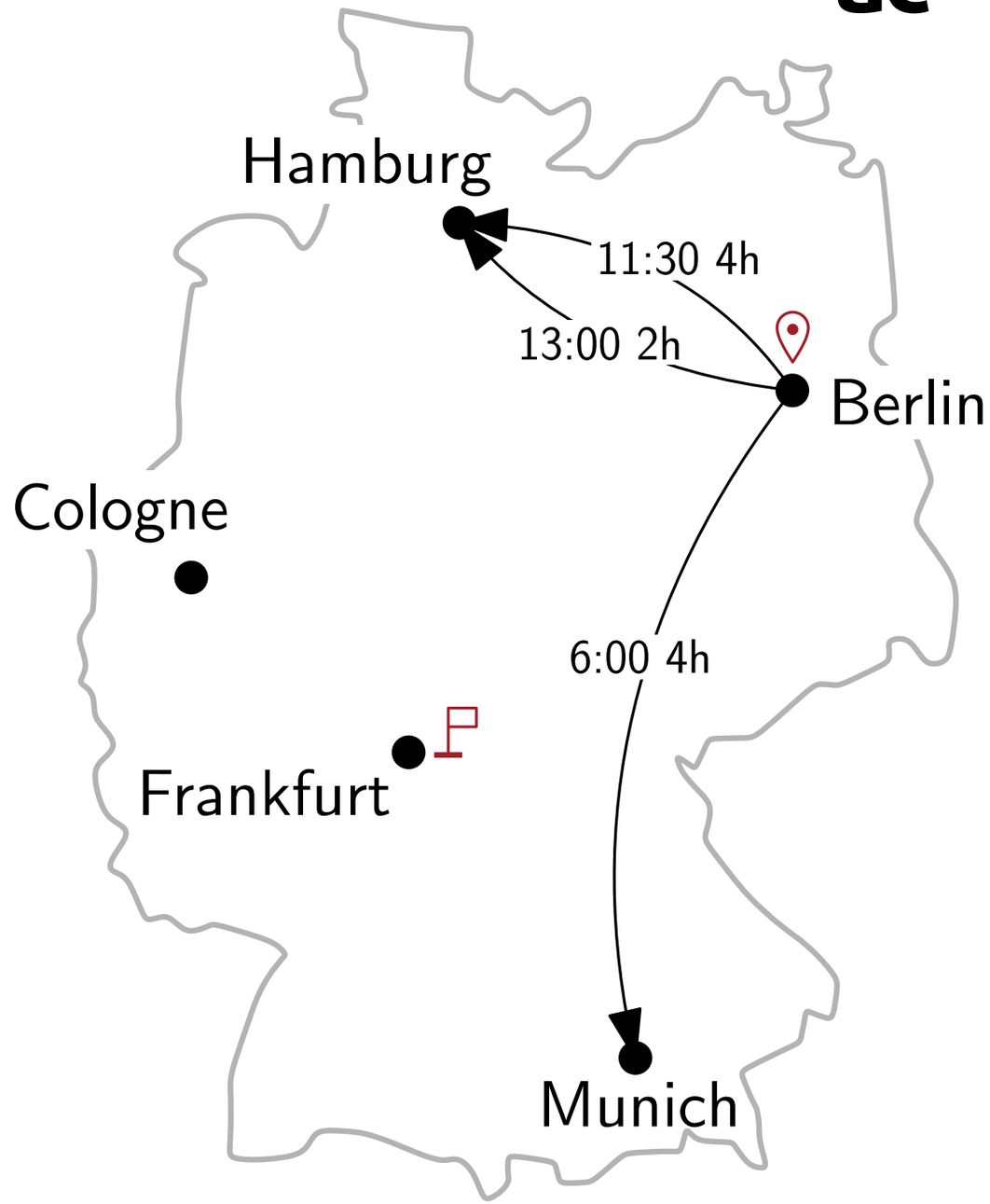
Introduction and Motivation





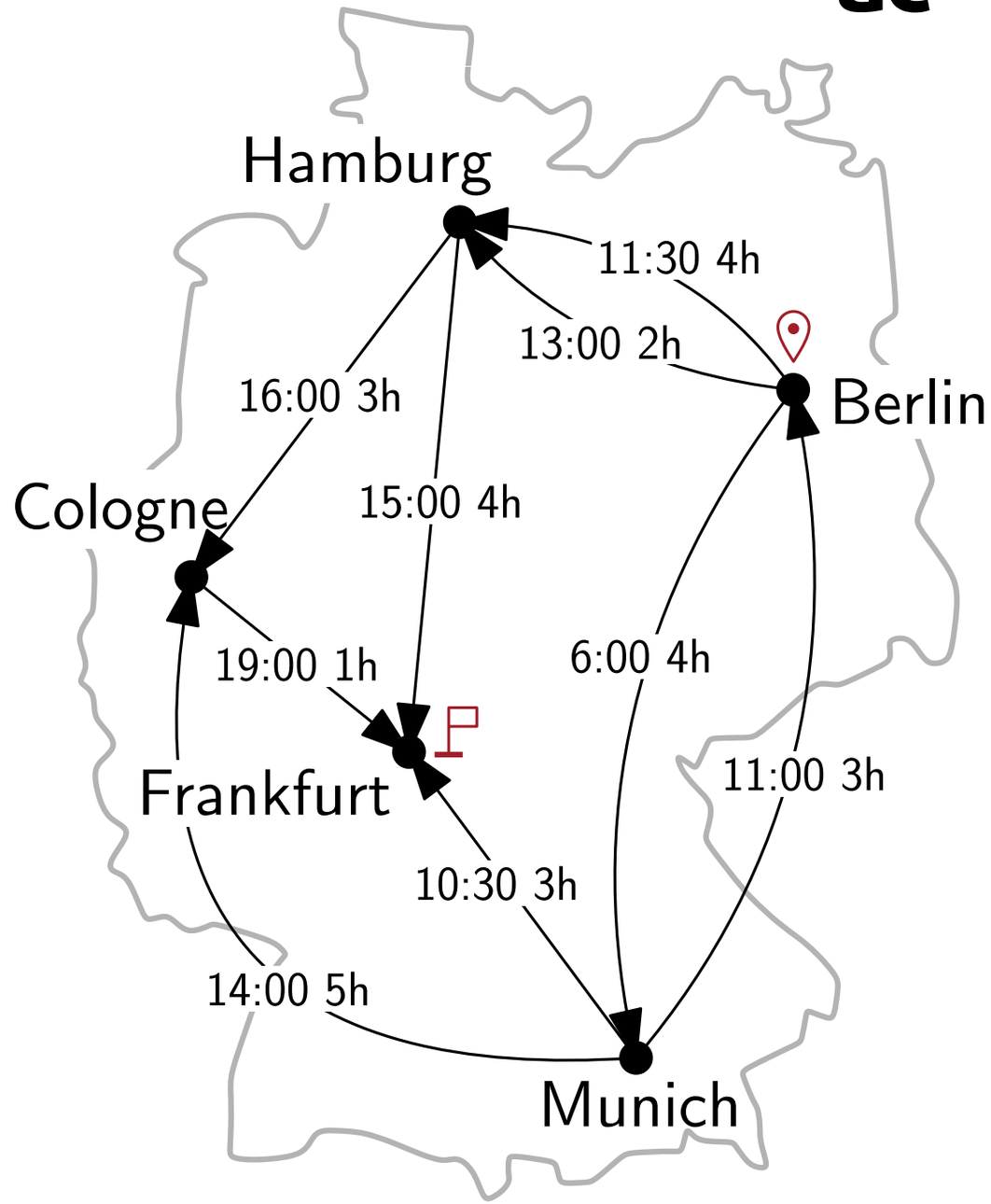
Introduction and Motivation





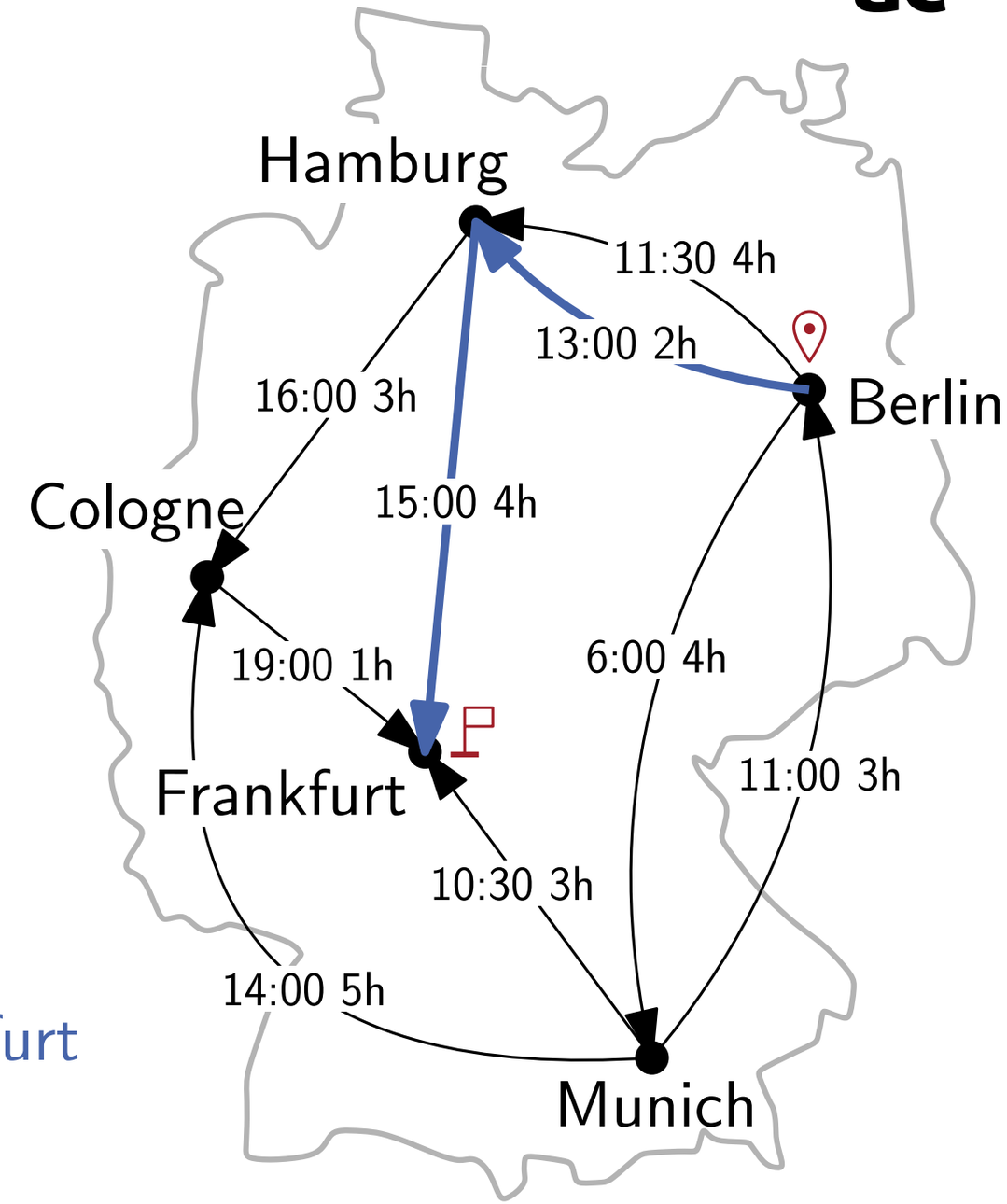
Introduction and Motivation





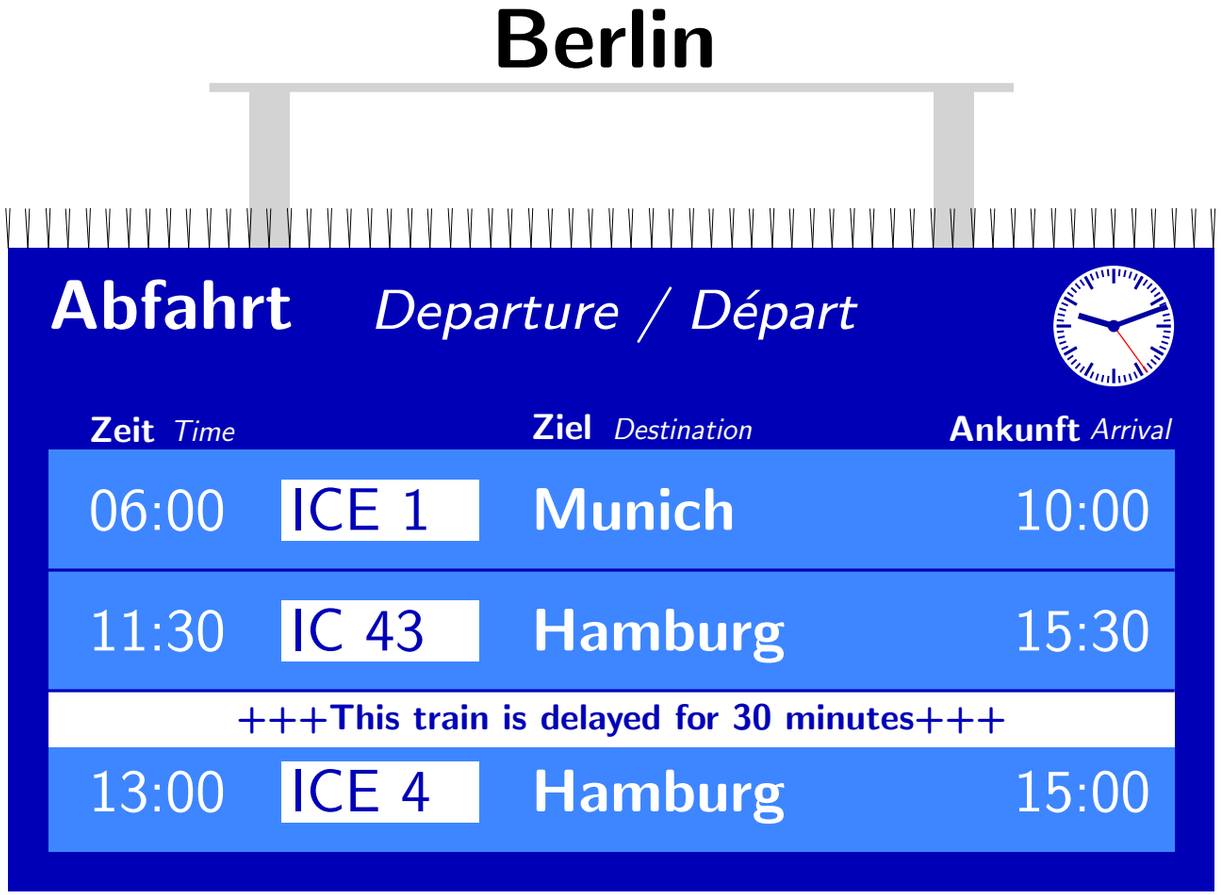
Introduction and Motivation

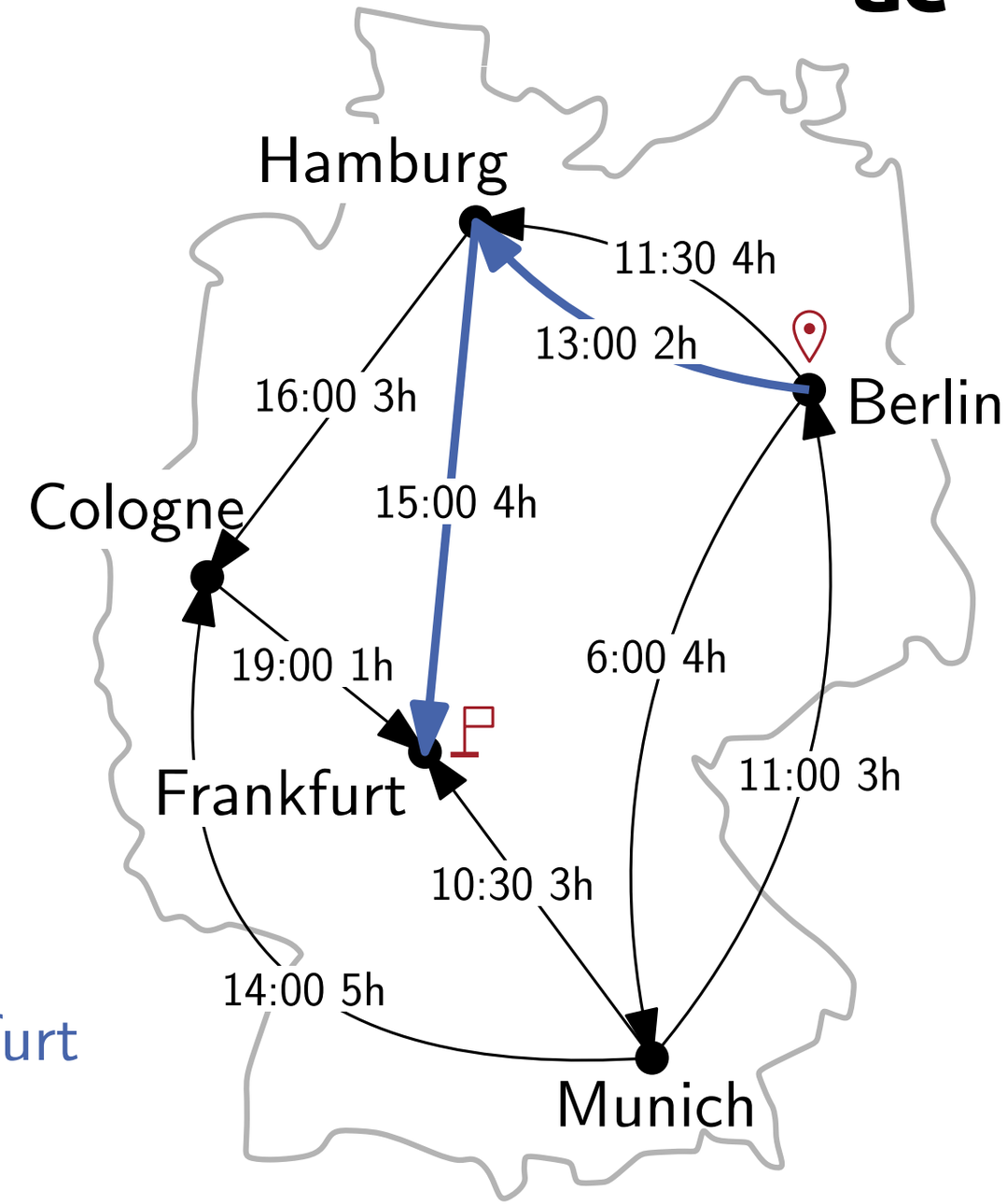




Berlin $\xrightarrow{13:00\ 2h}$ Hamburg $\xrightarrow{15:00\ 4h}$ Frankfurt

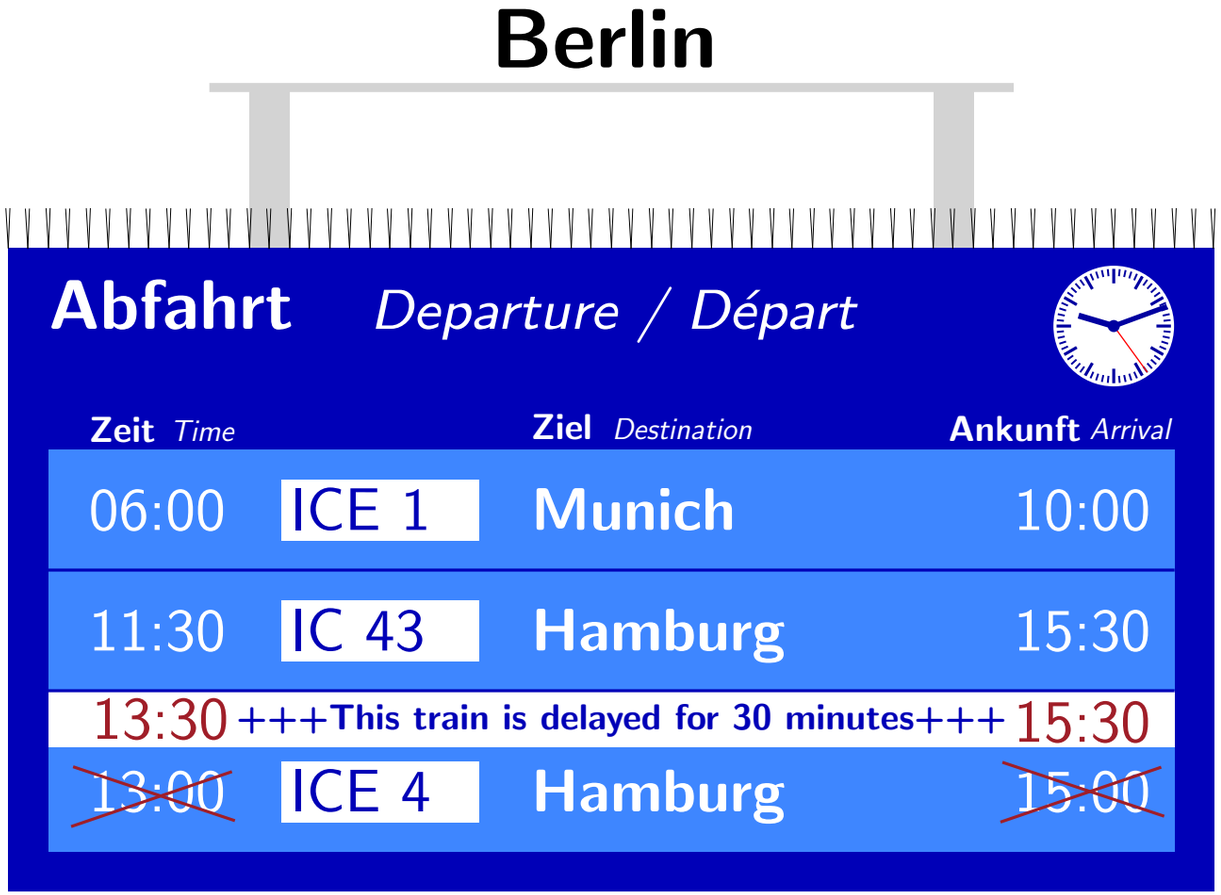
Introduction and Motivation

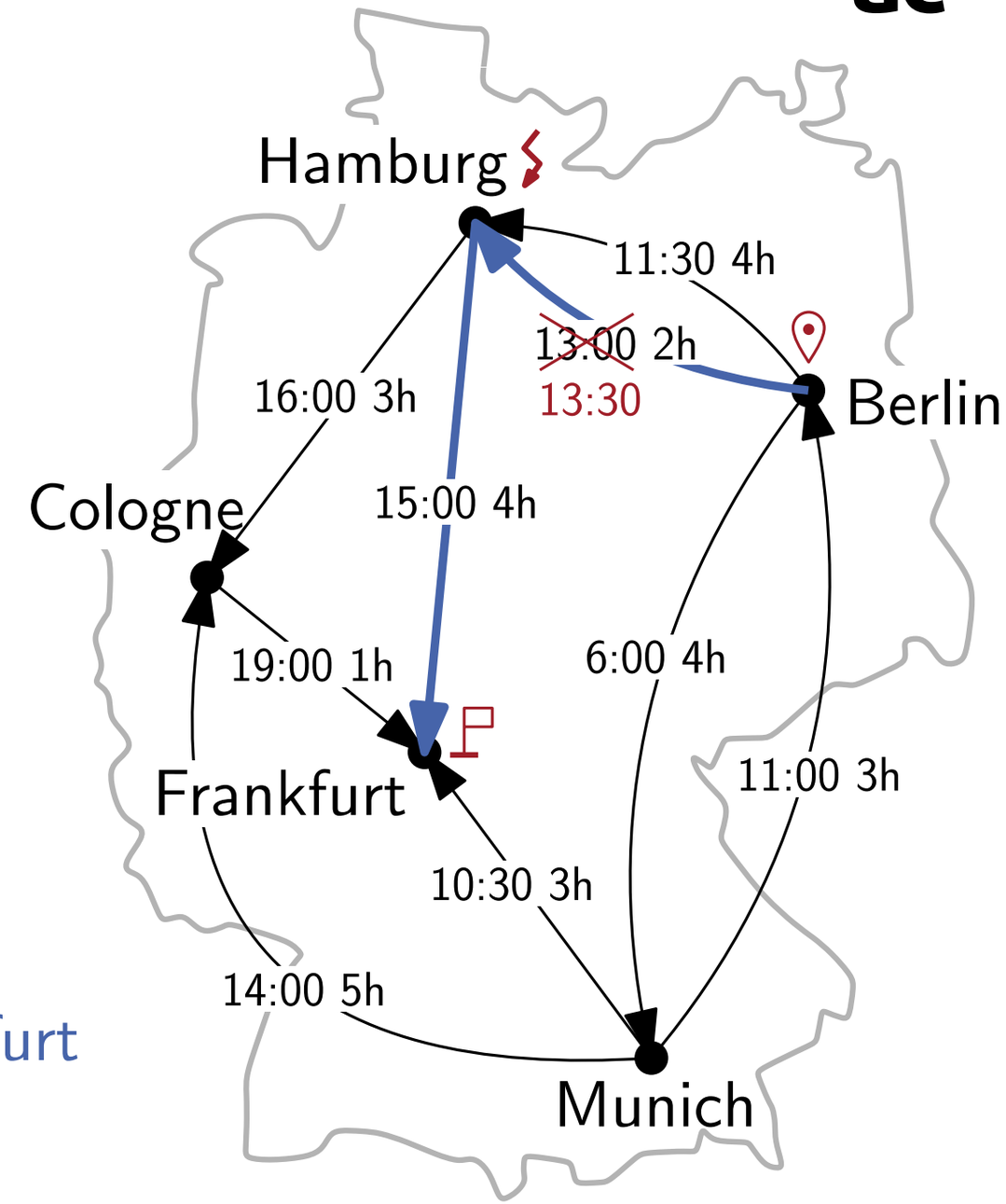




Berlin $\xrightarrow{13:00\ 2h}$ Hamburg $\xrightarrow{15:00\ 4h}$ Frankfurt

Introduction and Motivation





Berlin $\xrightarrow{\substack{13:30 \\ \cancel{13:00}\ 2h}}$ Hamburg $\xrightarrow{15:00\ 4h}$ Frankfurt





DB Punctuality – Long-Distance Trains

Delay	April 2024	April 2025
$\leq$ 6min	64.3	61.9
$\leq$ 15min	70.2	67.0

[Deutsche Bahn, May 2025]









DB Punctuality – Long-Distance Trains

Delay	April 2024	April 2025
$\leq$ 6min	64.3	61.9
$\leq$ 15min	70.2	67.0

[Deutsche Bahn, May 2025]







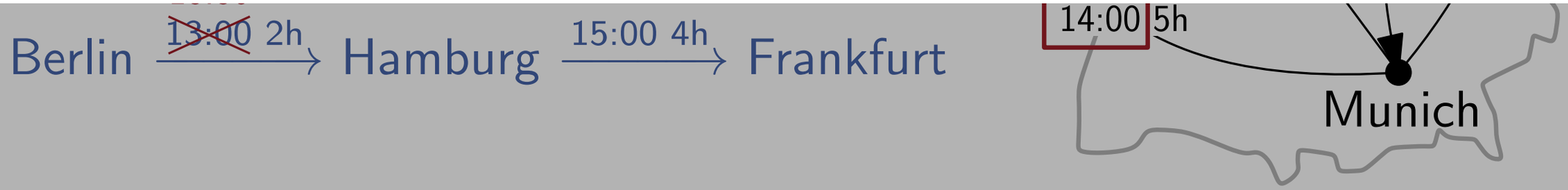


DB Punctuality – Long-Distance Trains

Delay	April 2024	April 2025
$\leq$ 6min	64.3	61.9
$\leq$ 15min	70.2	67.0

How can we incorperate delays?

[Deutsche Bahn, May 2025]



Related Work

Problem	**Definition**	**Complexity**
Delay-Robust Route [Füchsle et al., 2022a]	∃ route ∀ delays	**NP**-complete

Related Work

Problem	**Definition**	**Complexity**
Delay-Robust Route [Füchsle et al., 2022a]	∃ route ∀ delays	**NP**-complete
Delay-Robust Connection [Füchsle et al., 2022b]	∀ delays ∃ route	poly-time

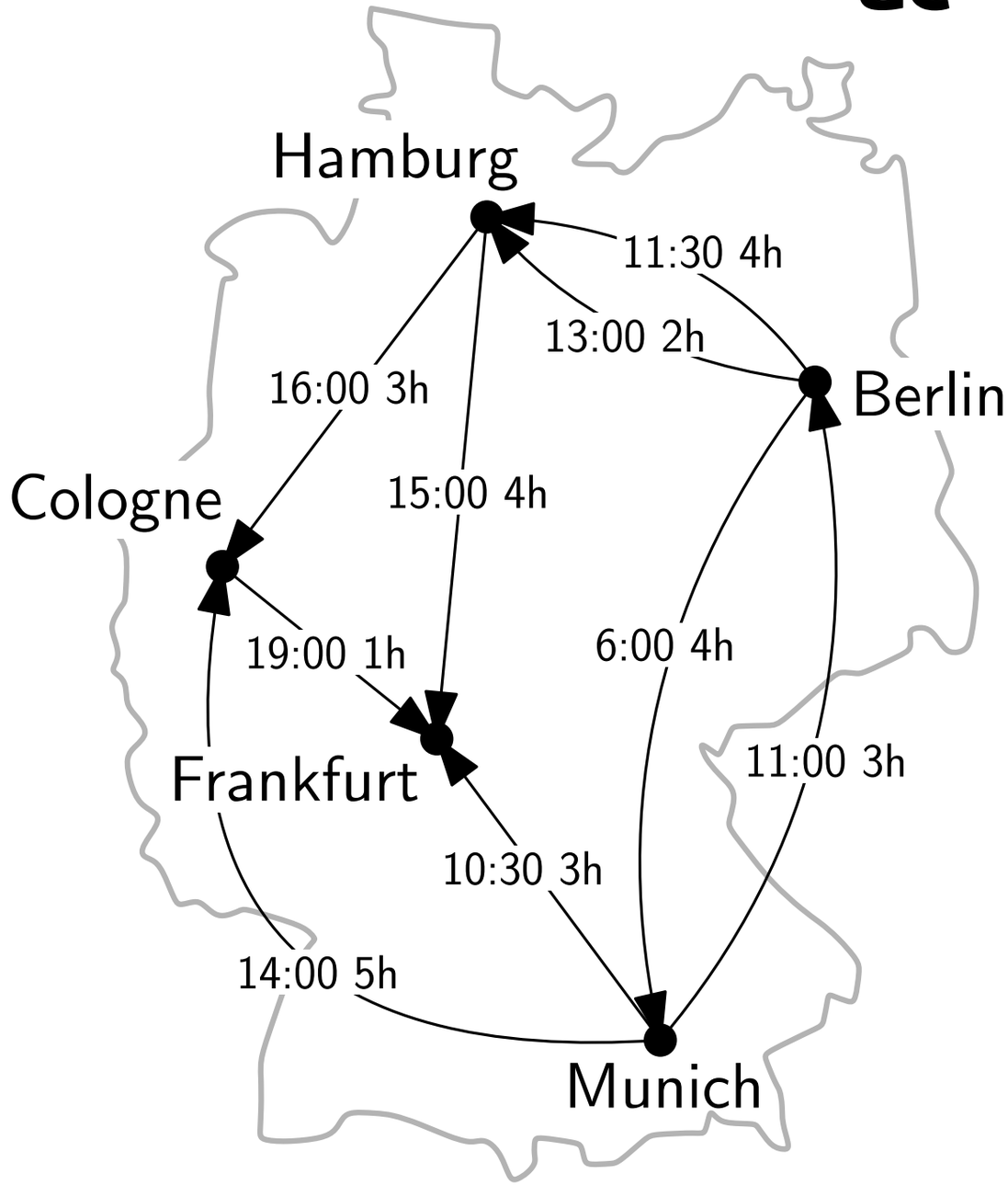
Related Work

Problem	**Definition**	**Complexity**
Delay-Robust Route [Füchsle et al., 2022a]	∃ route ∀ delays	**NP**-complete
Delay-Robust Connection [Füchsle et al., 2022b]	∀ delays ∃ route	poly-time
Robust Connection Game	Given start and end Delays anounced online	**PSPACE**-complete

Problem Definition

Setup:

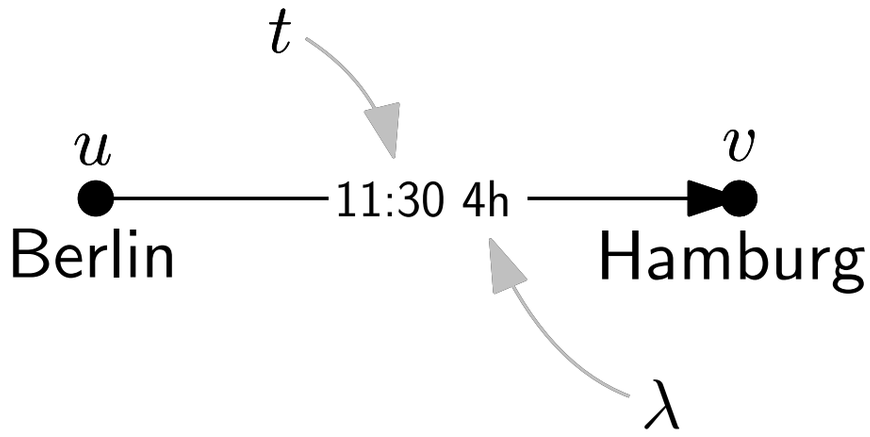
- Temporal graph $G = (V, E)$

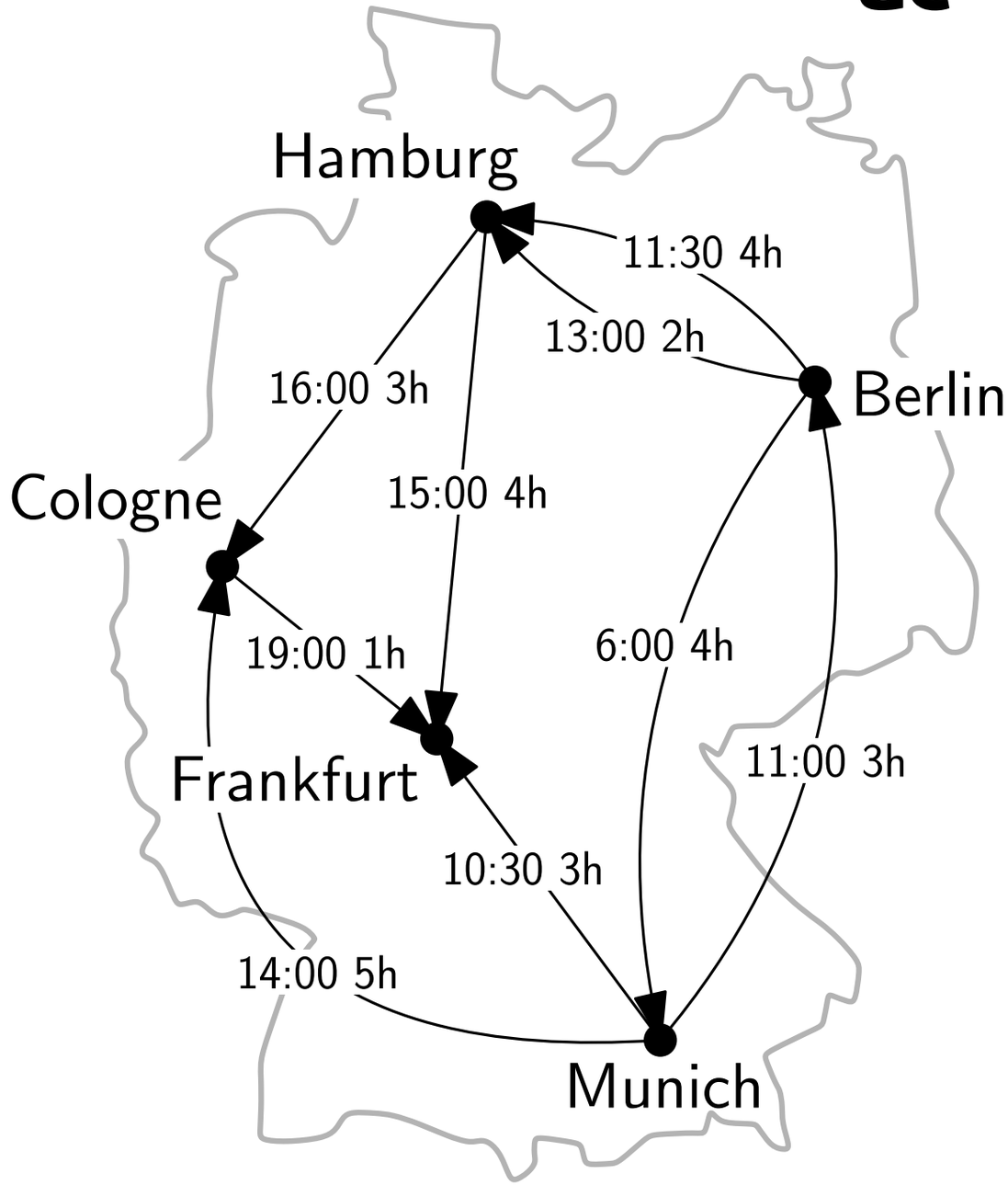


Problem Definition

Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$

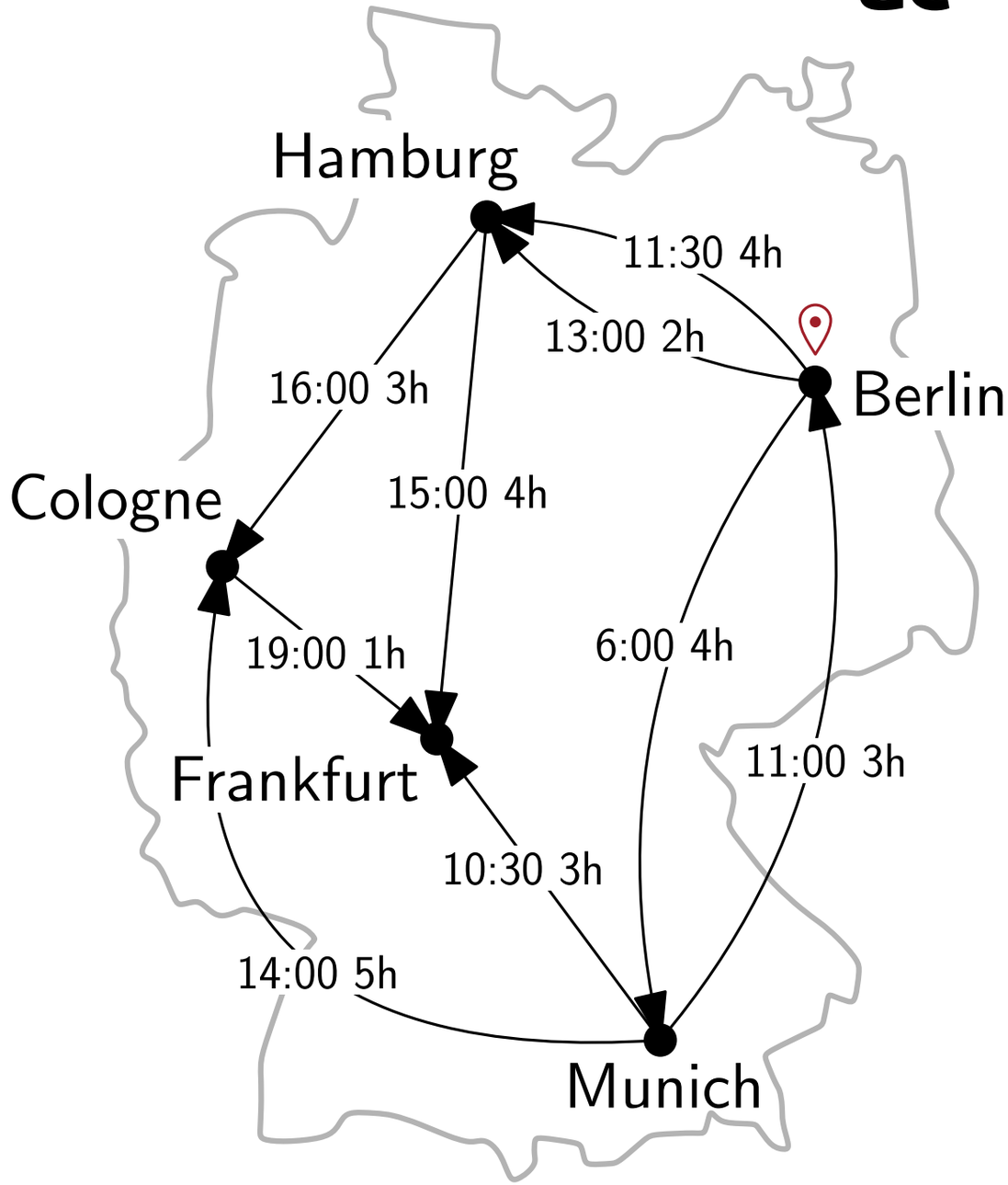




Problem Definition

Setup:

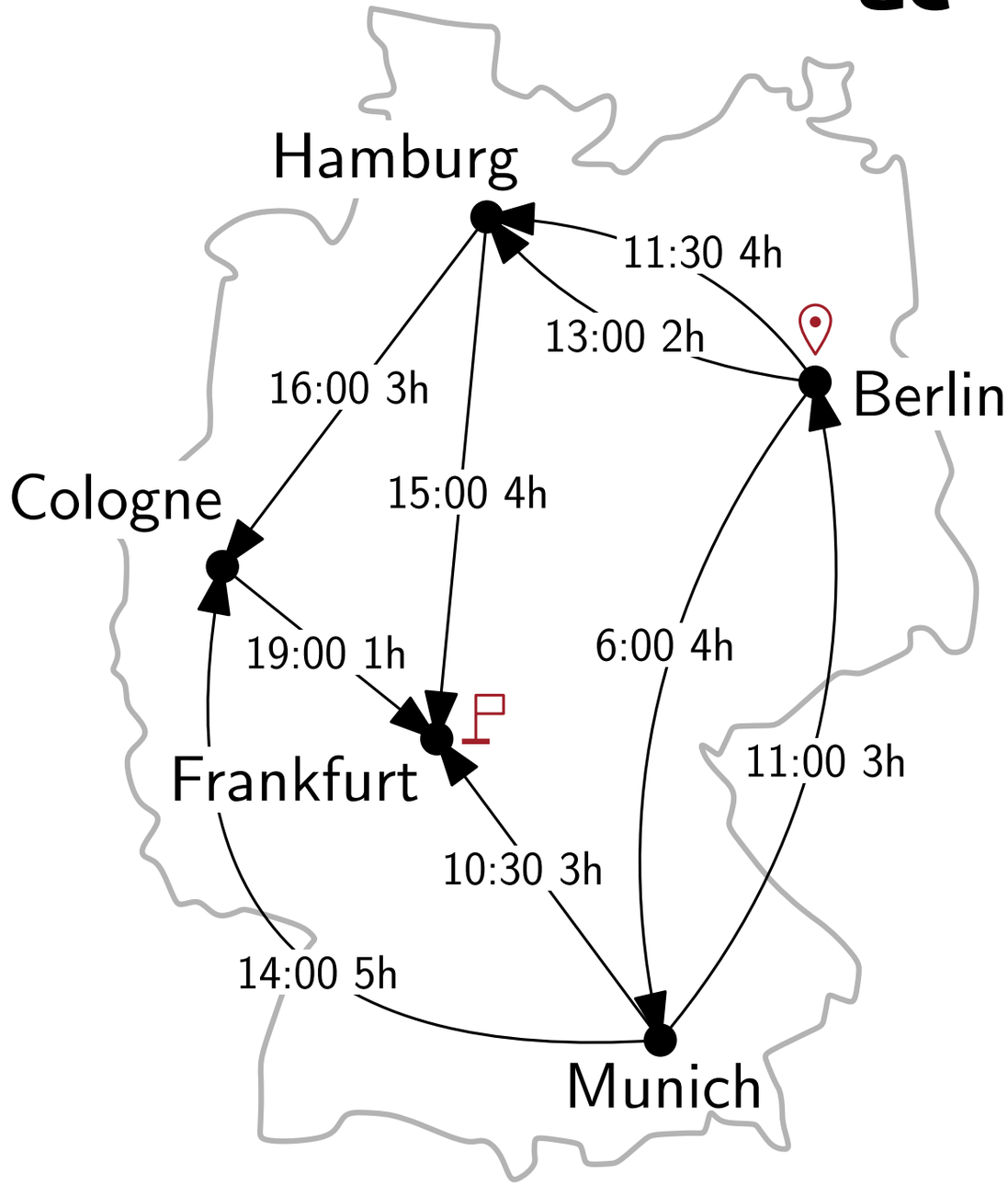
- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$



Problem Definition

Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$

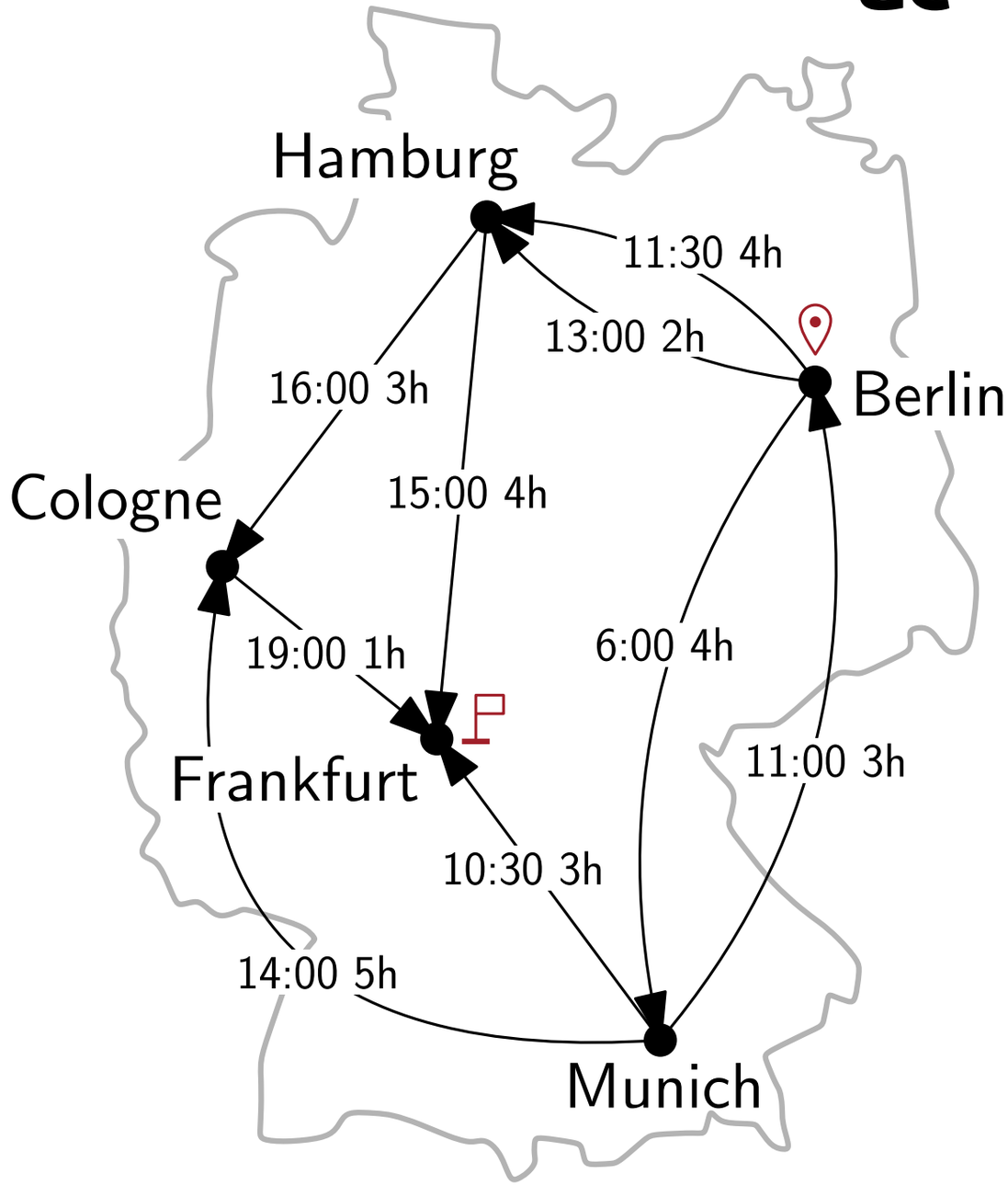


Problem Definition

Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$

- Traveler: wants to get from s to z
- Adversary: places delays on network



Problem Definition

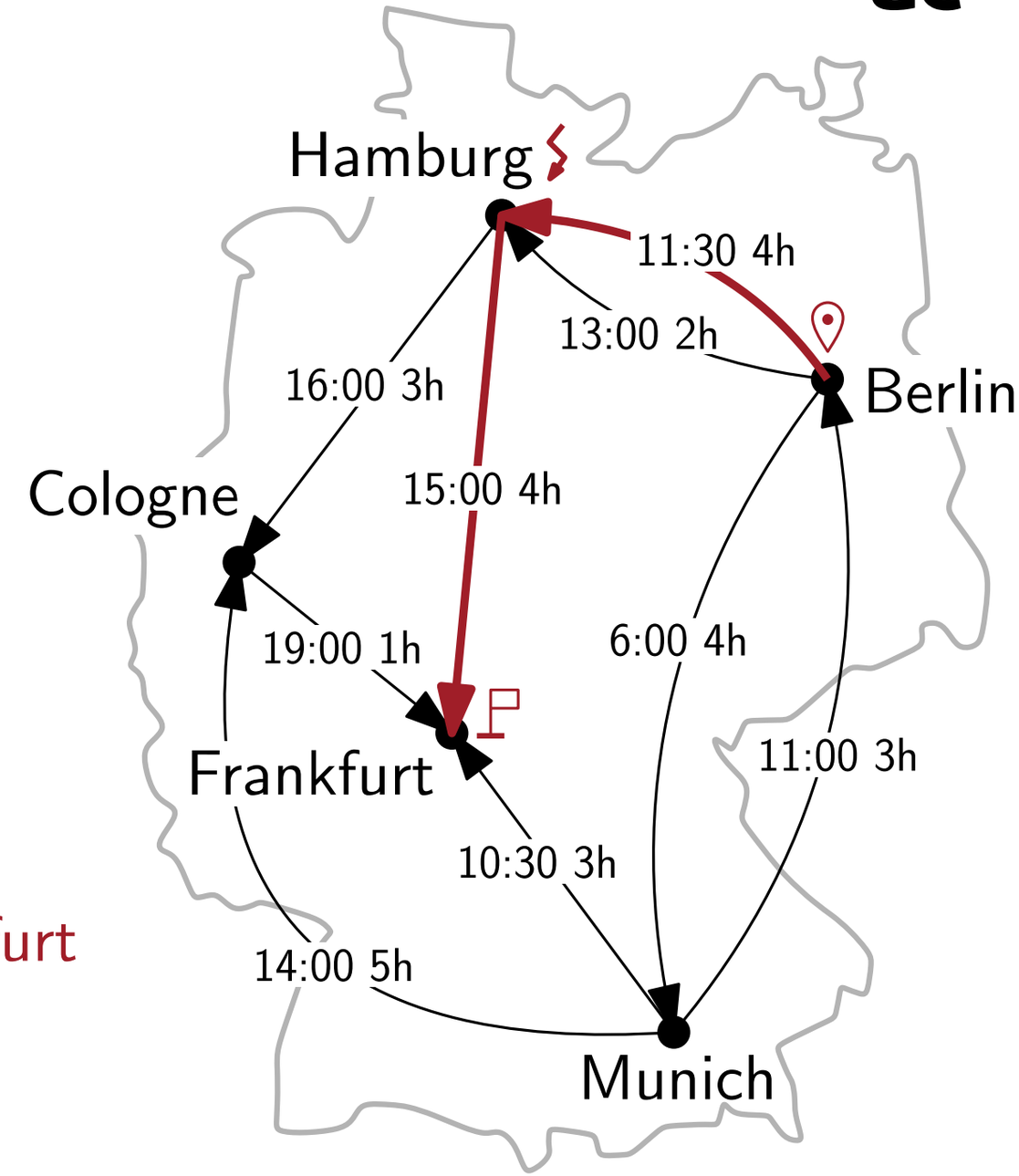
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$

- Traveler: wants to get from s to z
- Adversary: places delays on network

Invalid route:

Berlin $\xrightarrow{11:30\ 4h}$ Hamburg $\xrightarrow{15:00\ 4h}$ Frankfurt



Problem Definition

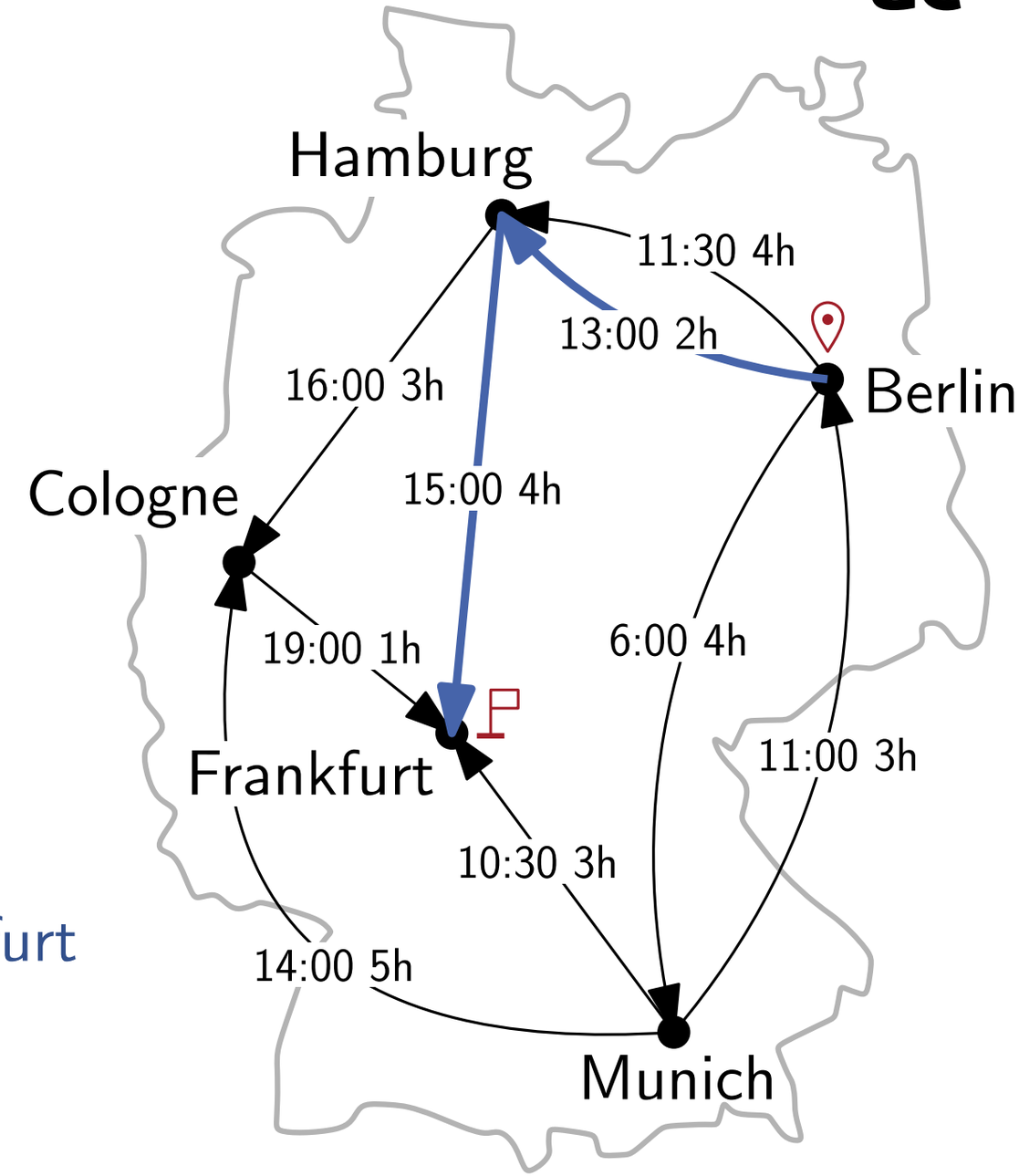
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$

- Traveler: wants to get from s to z
- Adversary: places delays on network

Valid route:

Berlin $\xrightarrow{\text{13:00 2h}}$ Hamburg $\xrightarrow{\text{15:00 4h}}$ Frankfurt



Problem Definition

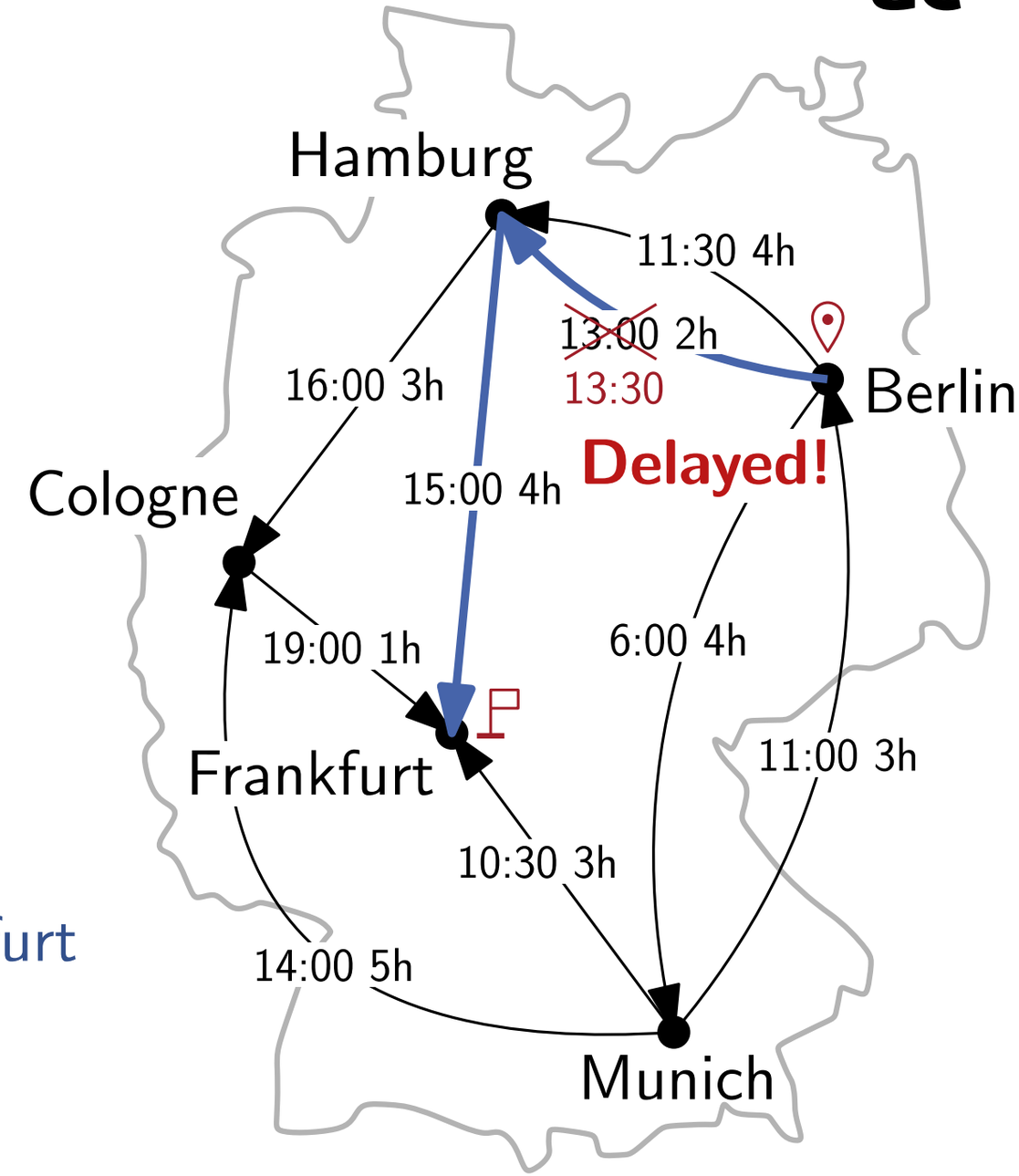
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$

- Traveler: wants to get from s to z
- Adversary: places delays on network

Valid route:

Berlin $\xrightarrow{\substack{13{:}30 \\ \cancel{13{:}00}\ 2h}}$ Hamburg $\xrightarrow{15{:}00\ 4h}$ Frankfurt



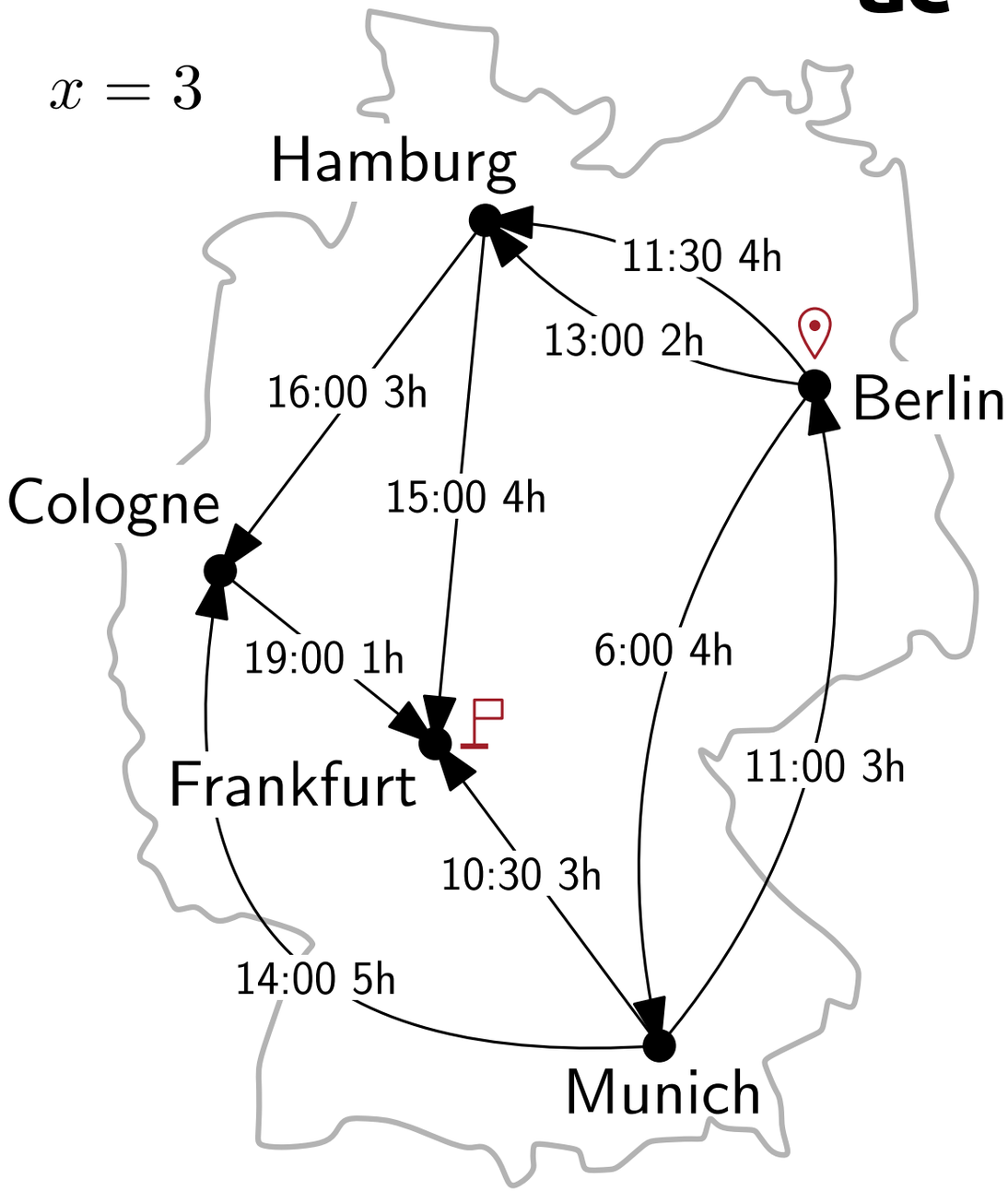
Problem Definition

Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$

- Traveler: wants to get from s to z
- Adversary: places delays on network

- Total delay budget x



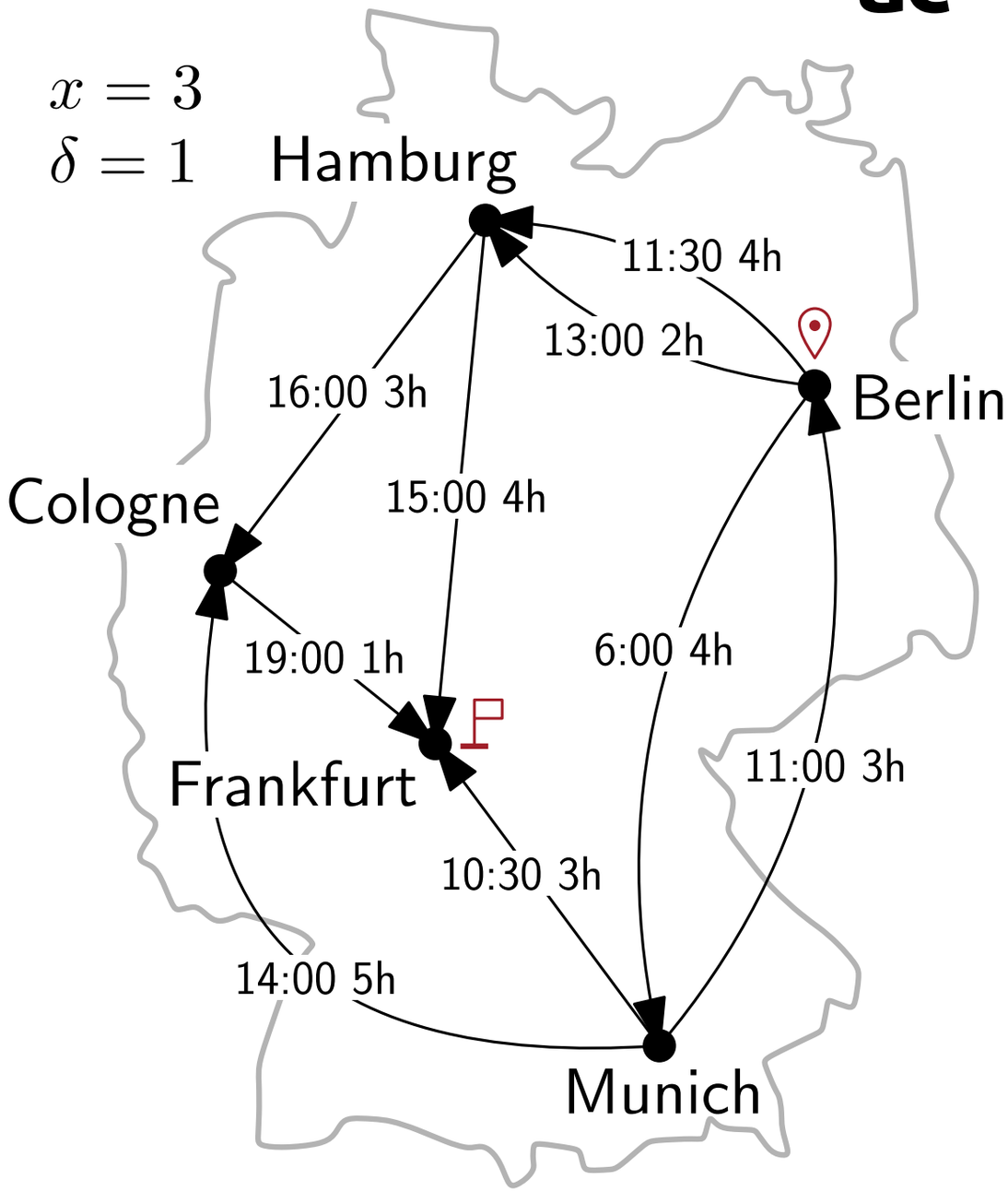
Problem Definition

Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$

- Traveler: wants to get from s to z
- Adversary: places delays on network

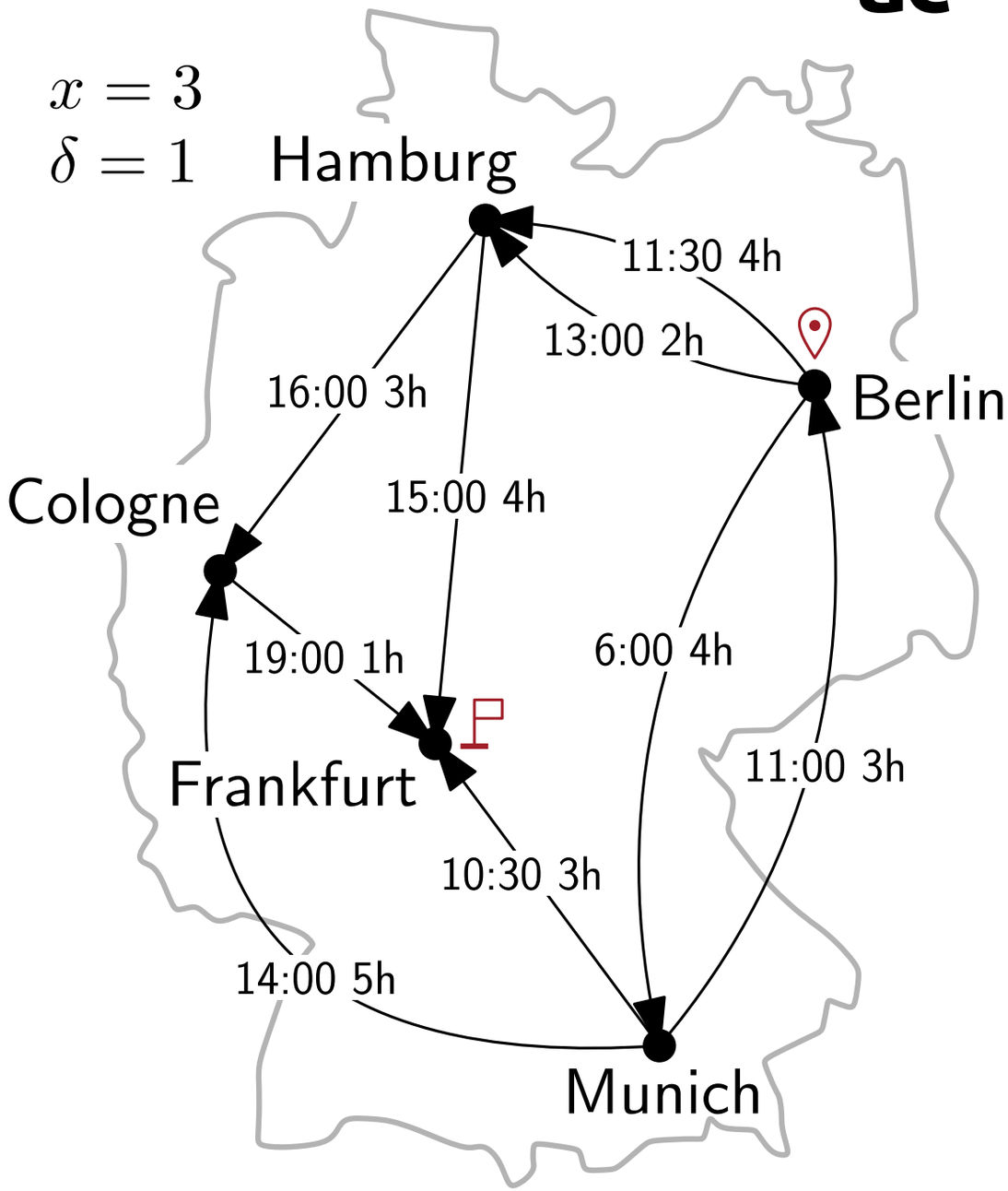
- Total delay budget x
- Delay time δ



Problem Definition

Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ



Problem Definition

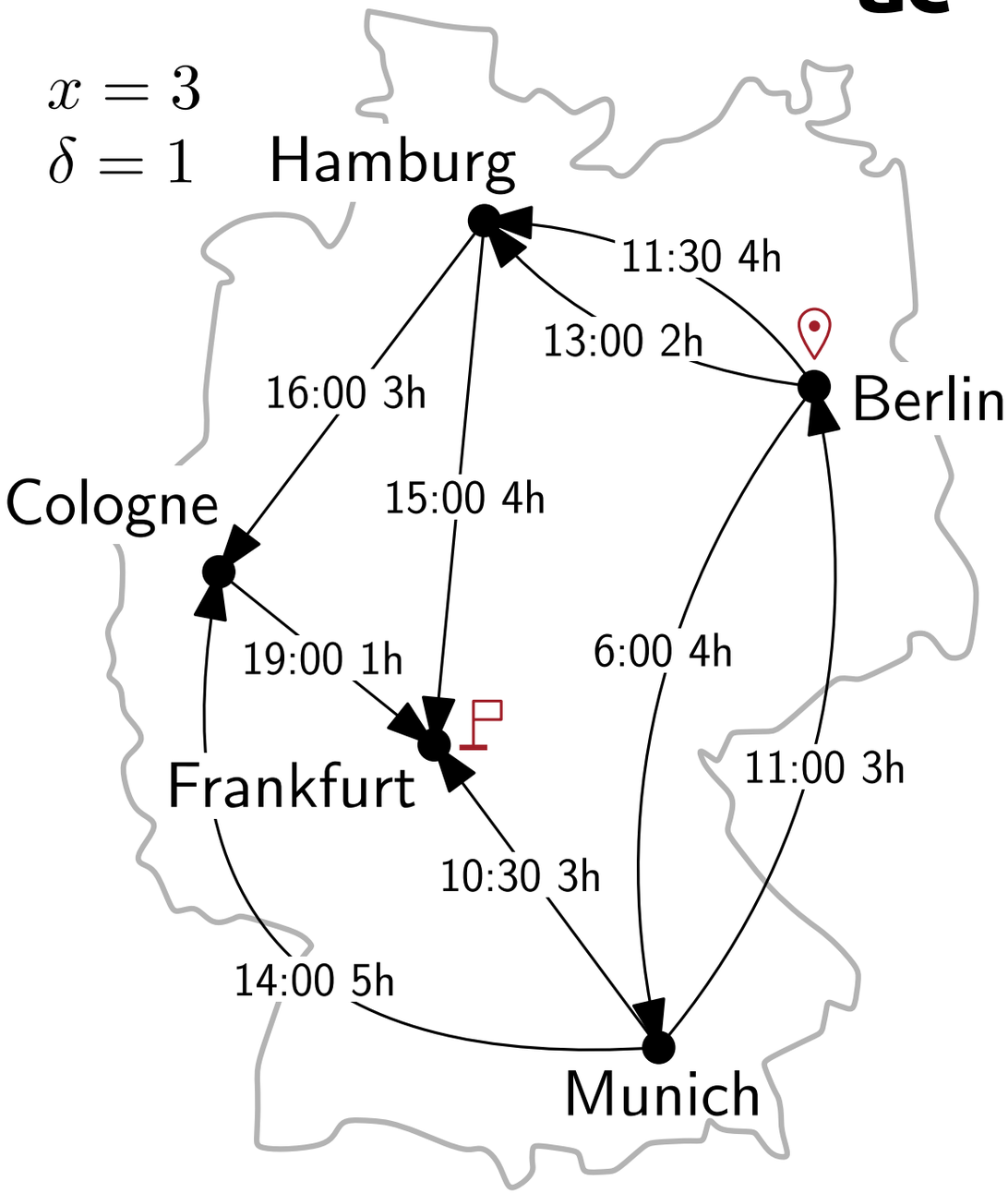
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ

ROBUST CONNECTION GAME

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

$\Rightarrow$ Does the Traveler have a winning strategy?



Problem Definition

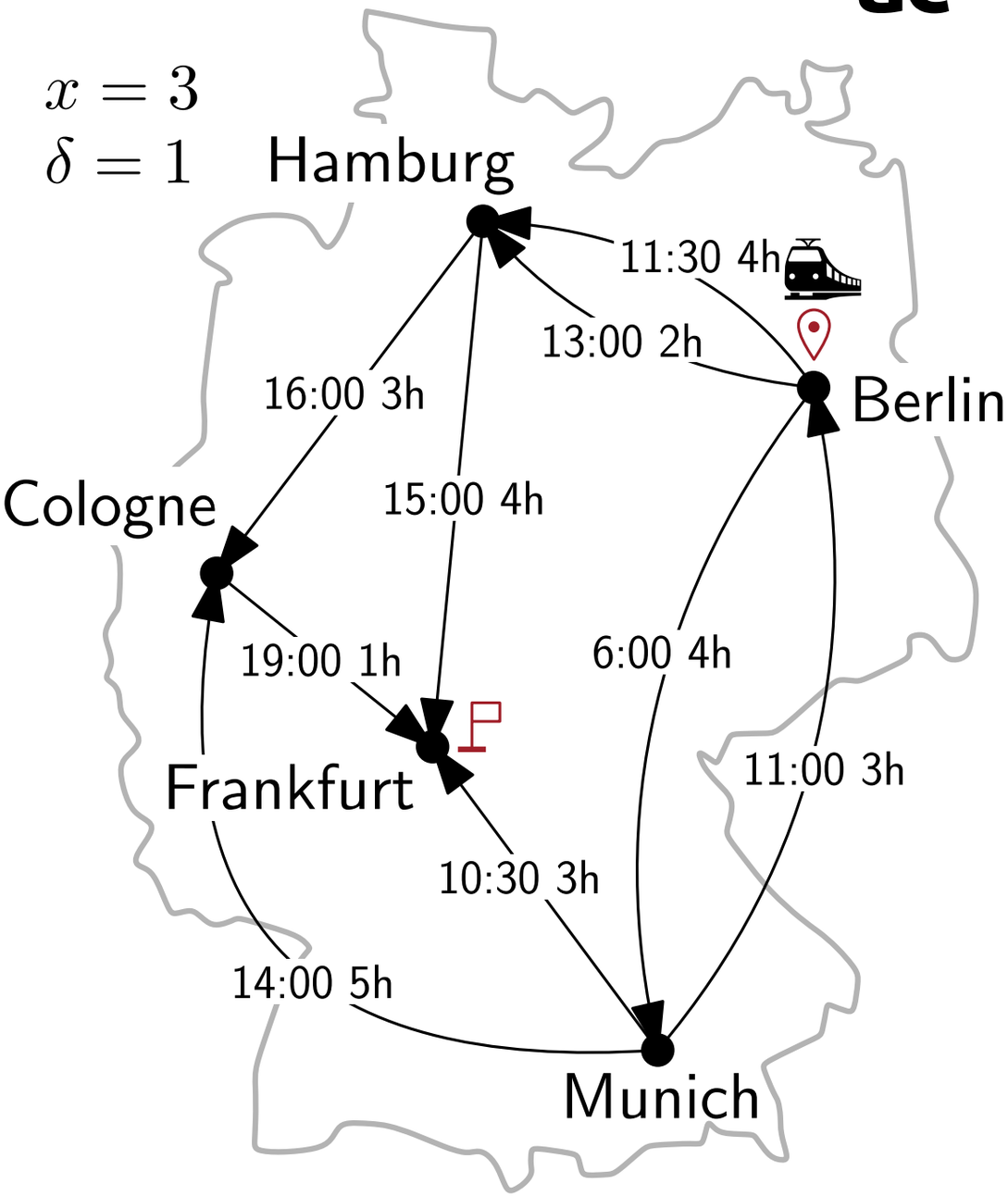
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ

Robust Connection Game

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

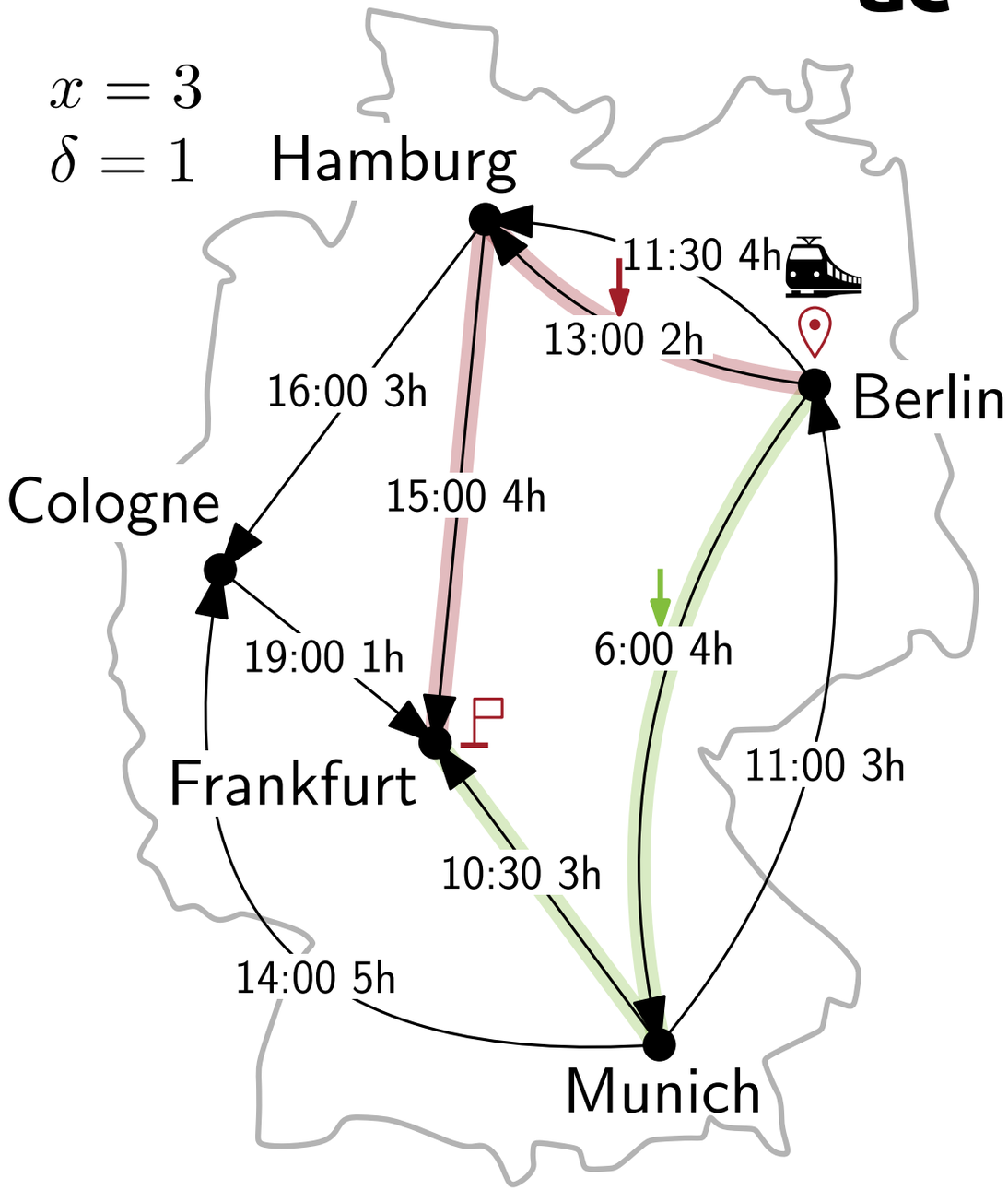
$\Rightarrow$ Does the Traveler have a winning strategy?



Problem Definition

Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ



ROBUST CONNECTION GAME

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

$\Rightarrow$ Does the Traveler have a winning strategy?

Problem Definition

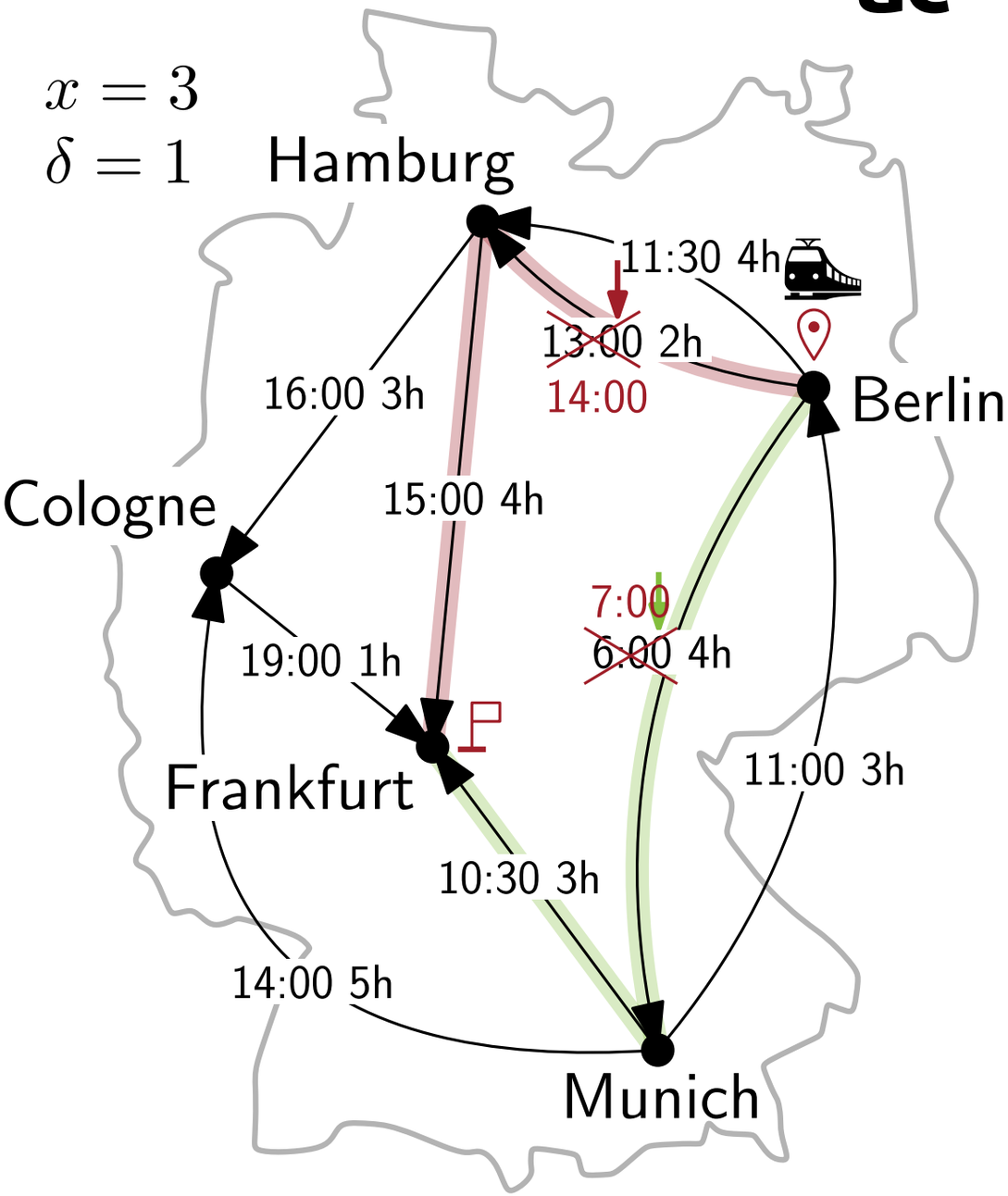
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ

Robust Connection Game

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

$\Rightarrow$ Does the Traveler have a winning strategy?



Problem Definition

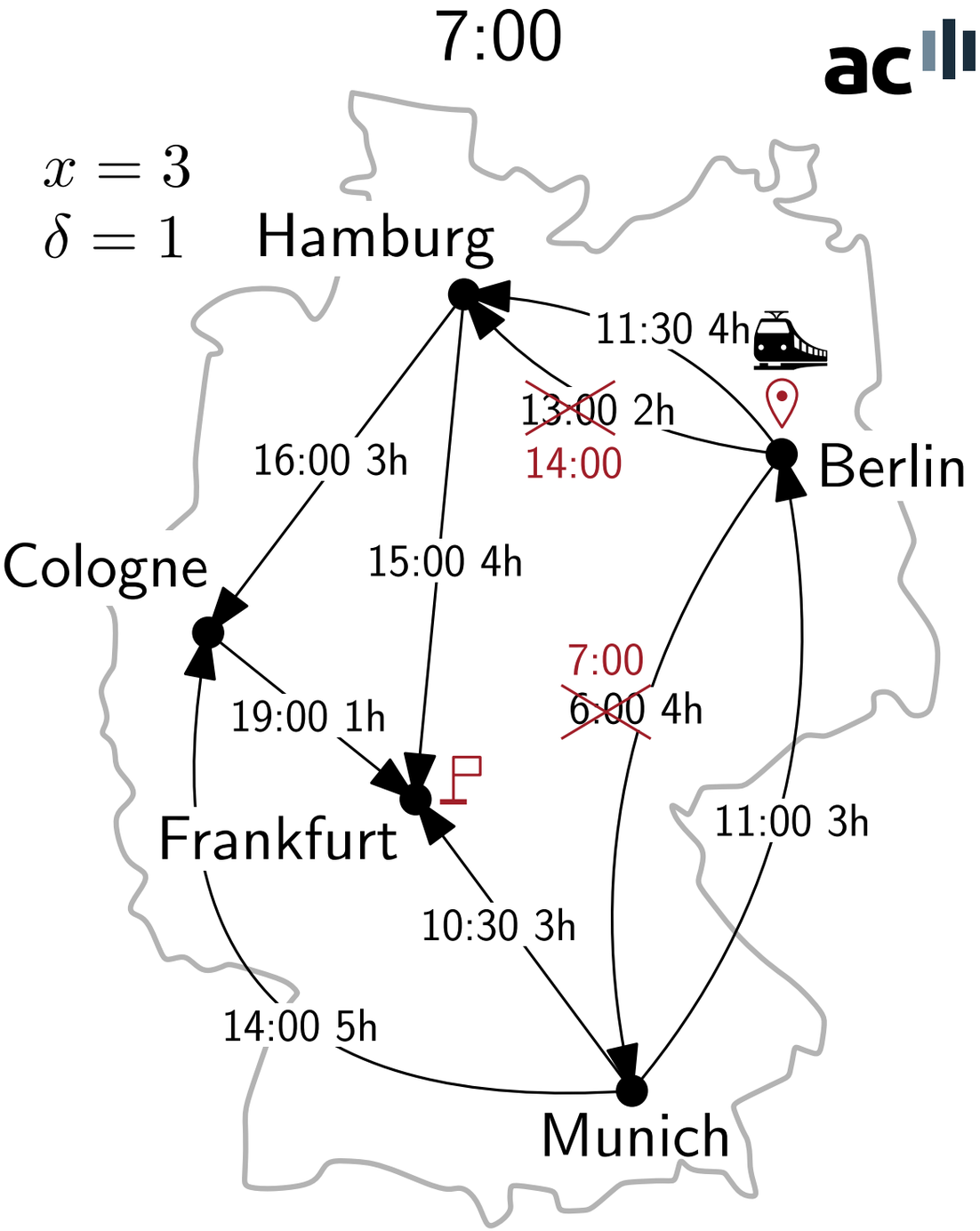
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ

Robust Connection Game

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

$\Rightarrow$ Does the Traveler have a winning strategy?



Problem Definition

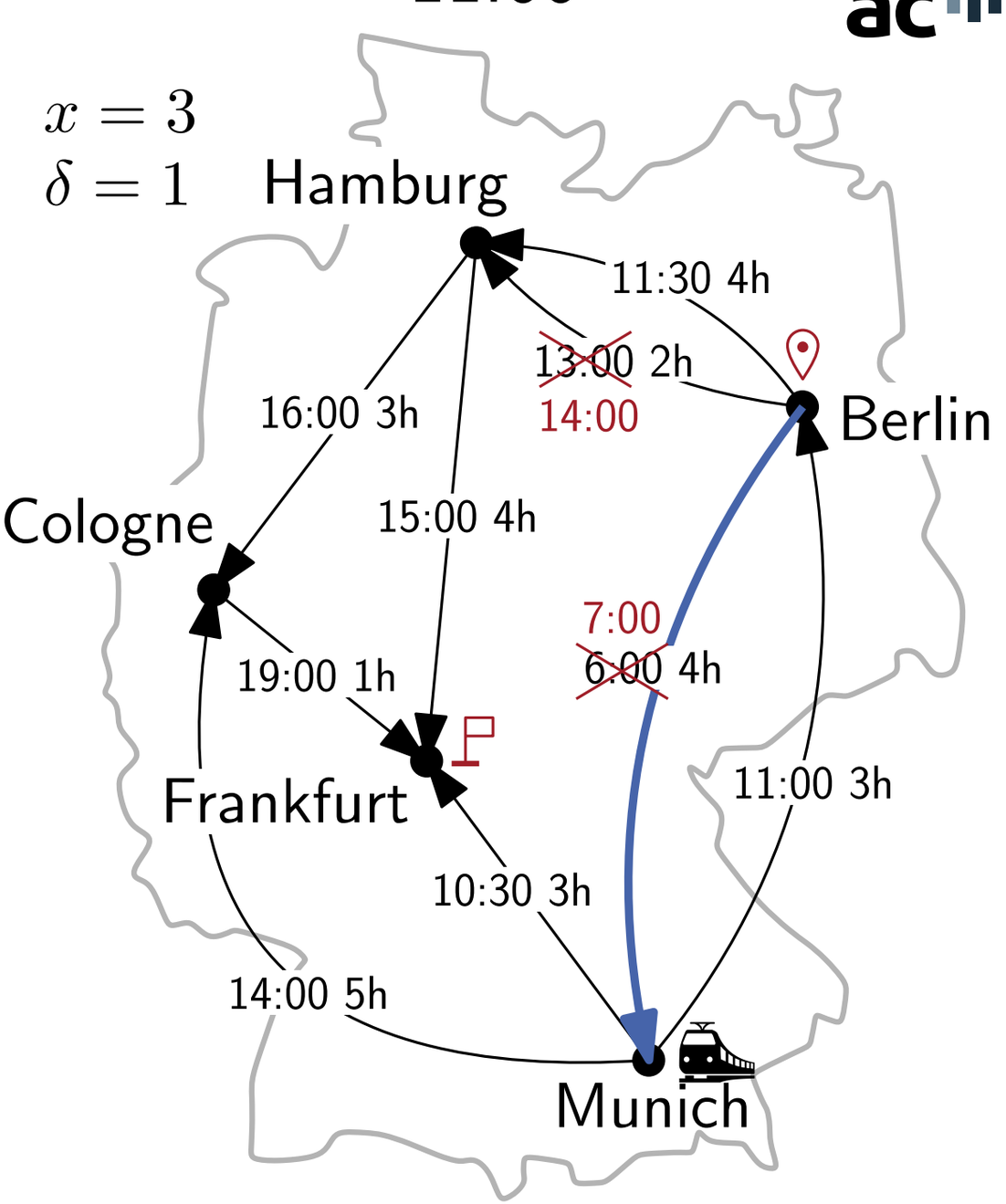
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ

Robust Connection Game

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

$\Rightarrow$ Does the Traveler have a winning strategy?



Problem Definition

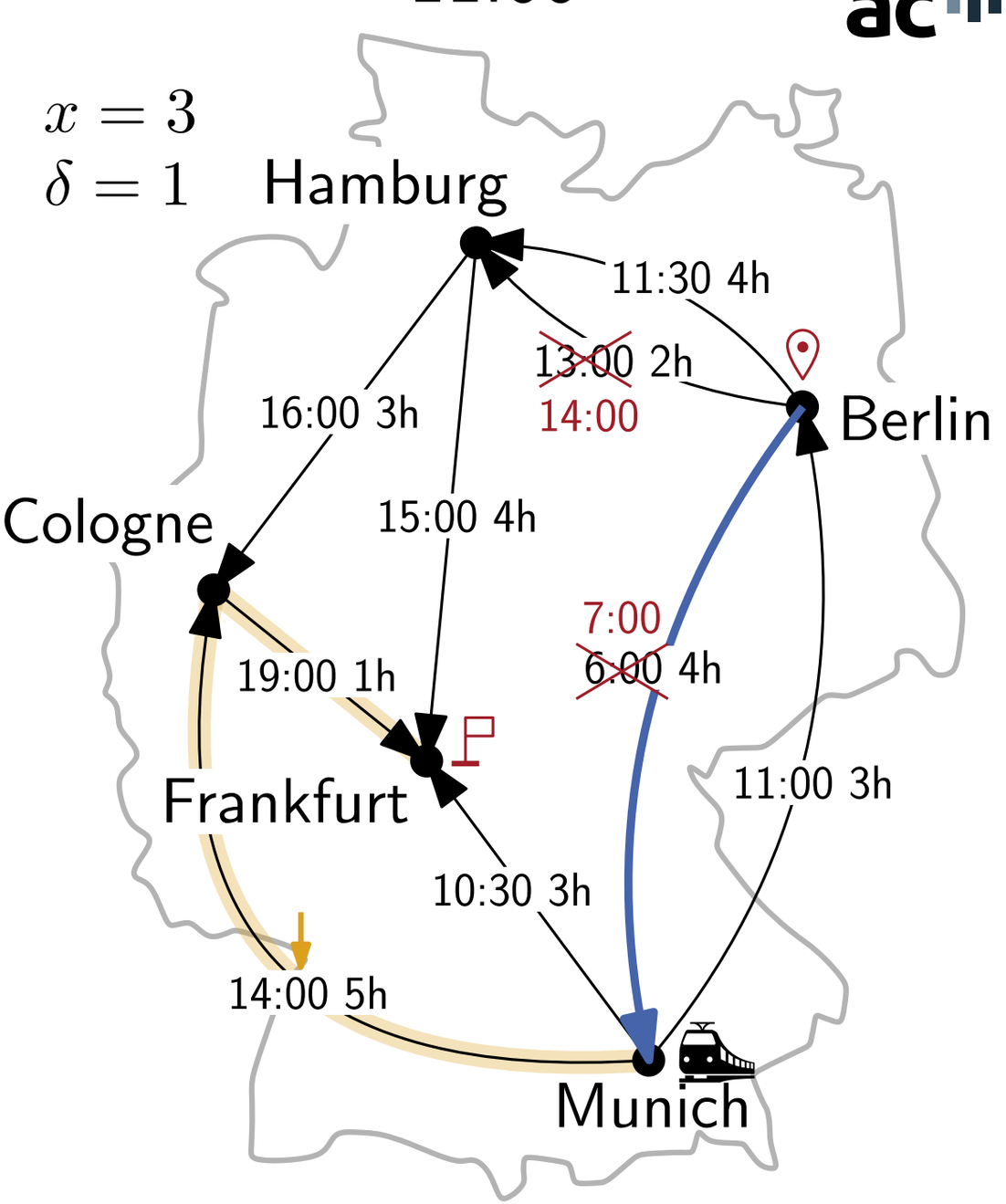
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ

Robust Connection Game

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

$\Rightarrow$ Does the Traveler have a winning strategy?



Problem Definition

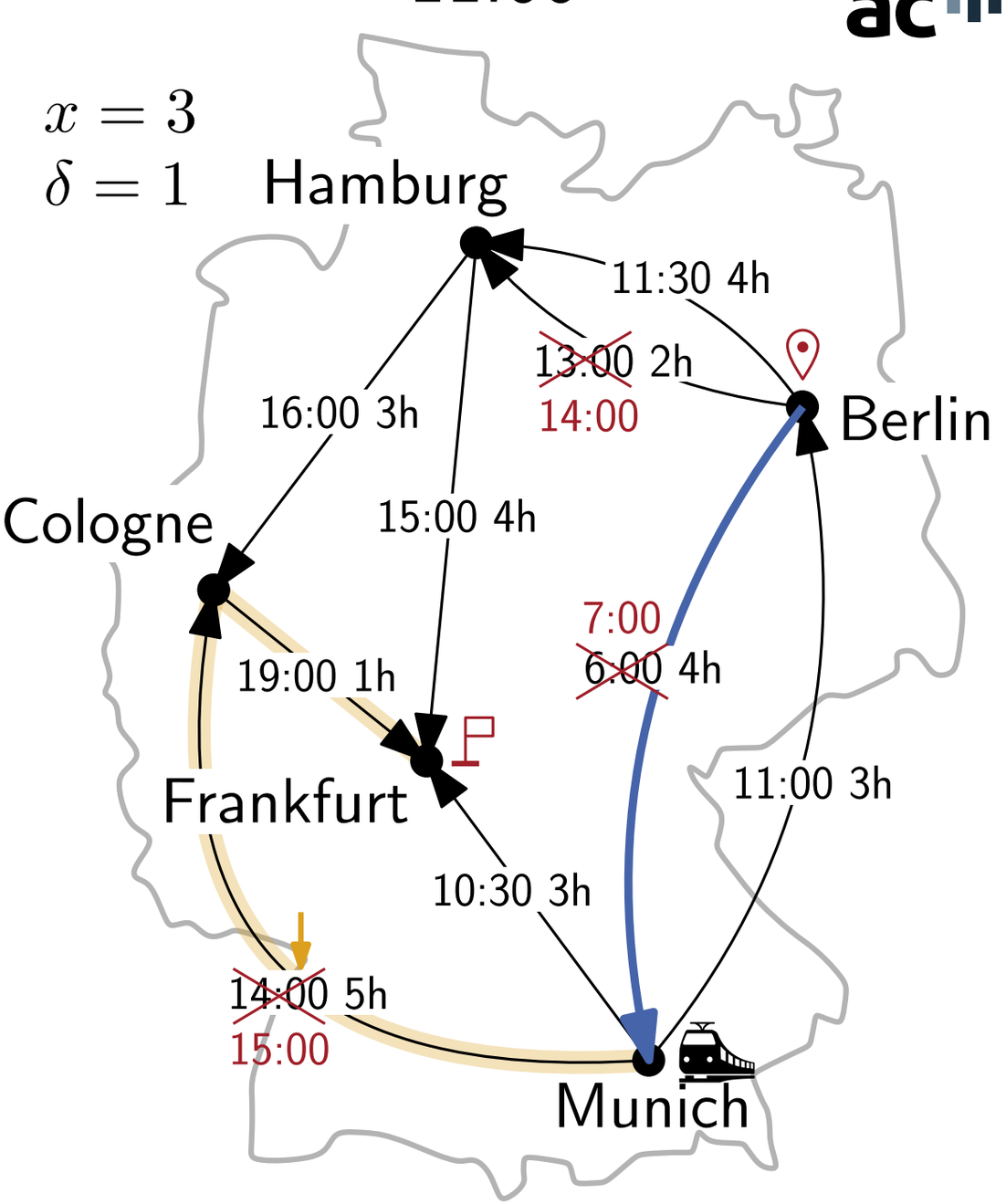
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ

Robust Connection Game

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

$\Rightarrow$ Does the Traveler have a winning strategy?



Problem Definition

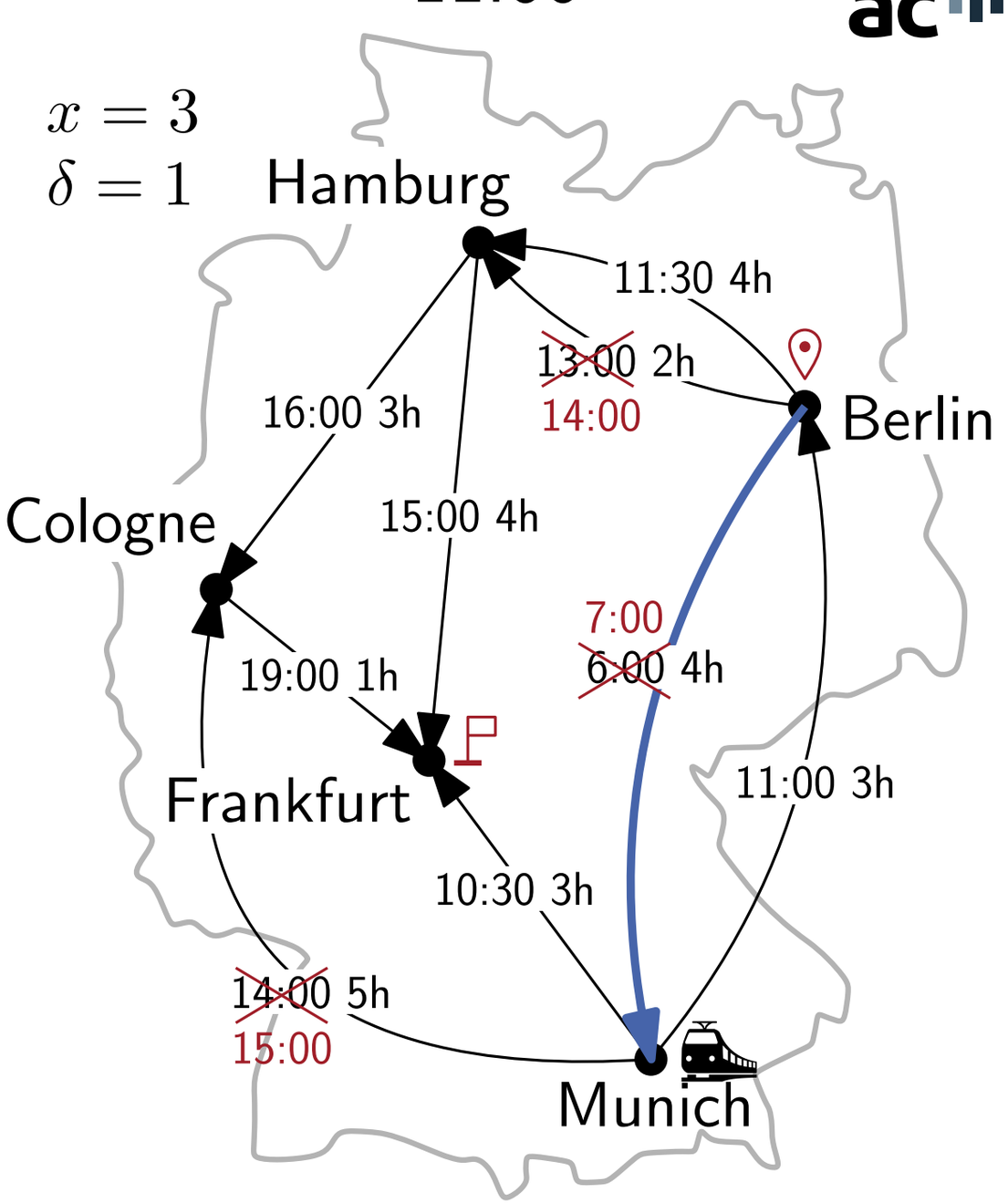
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ

Robust Connection Game

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

$\Rightarrow$ Does the Traveler have a winning strategy?



Problem Definition

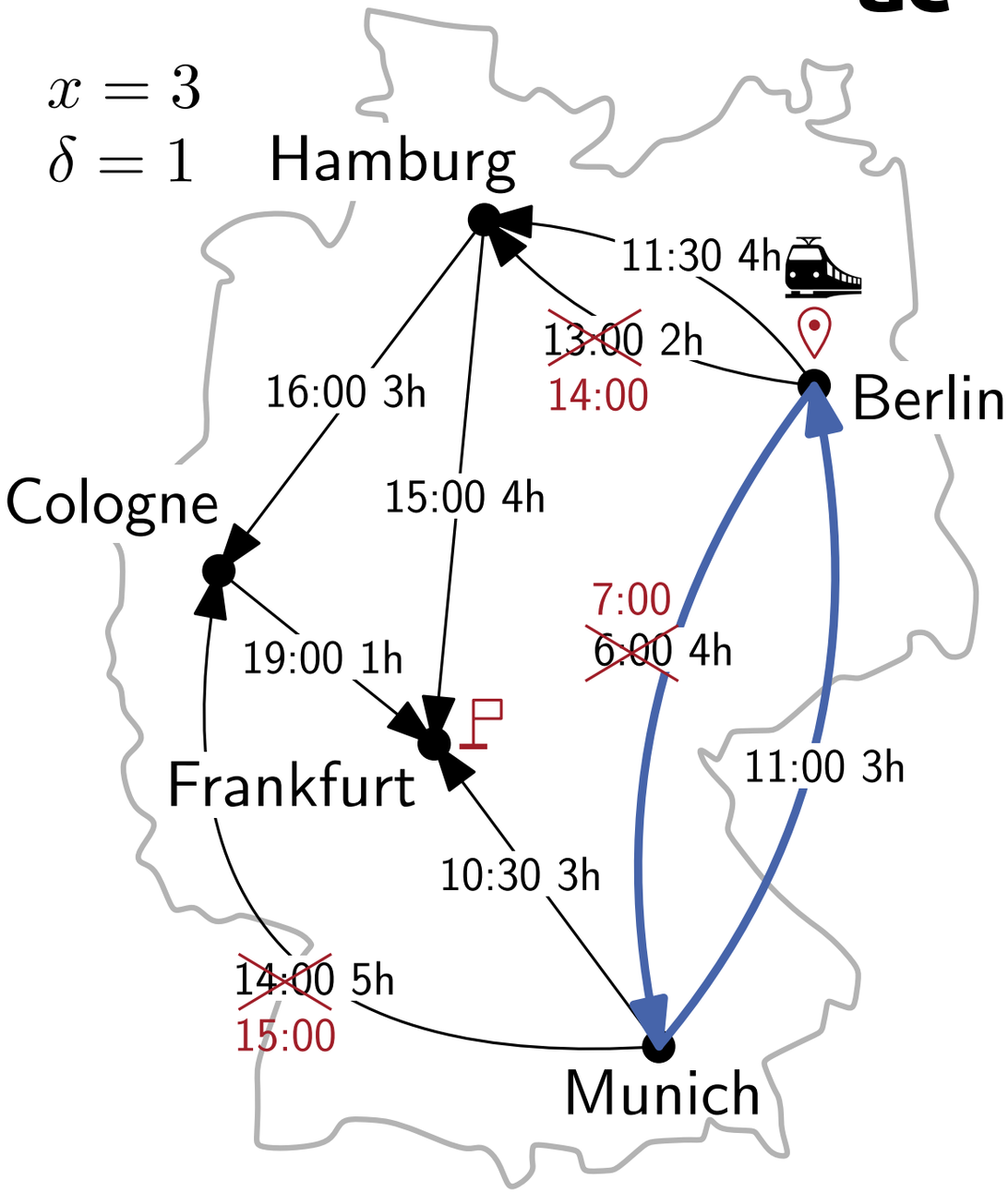
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ

Robust Connection Game

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

$\Rightarrow$ Does the Traveler have a winning strategy?



Problem Definition

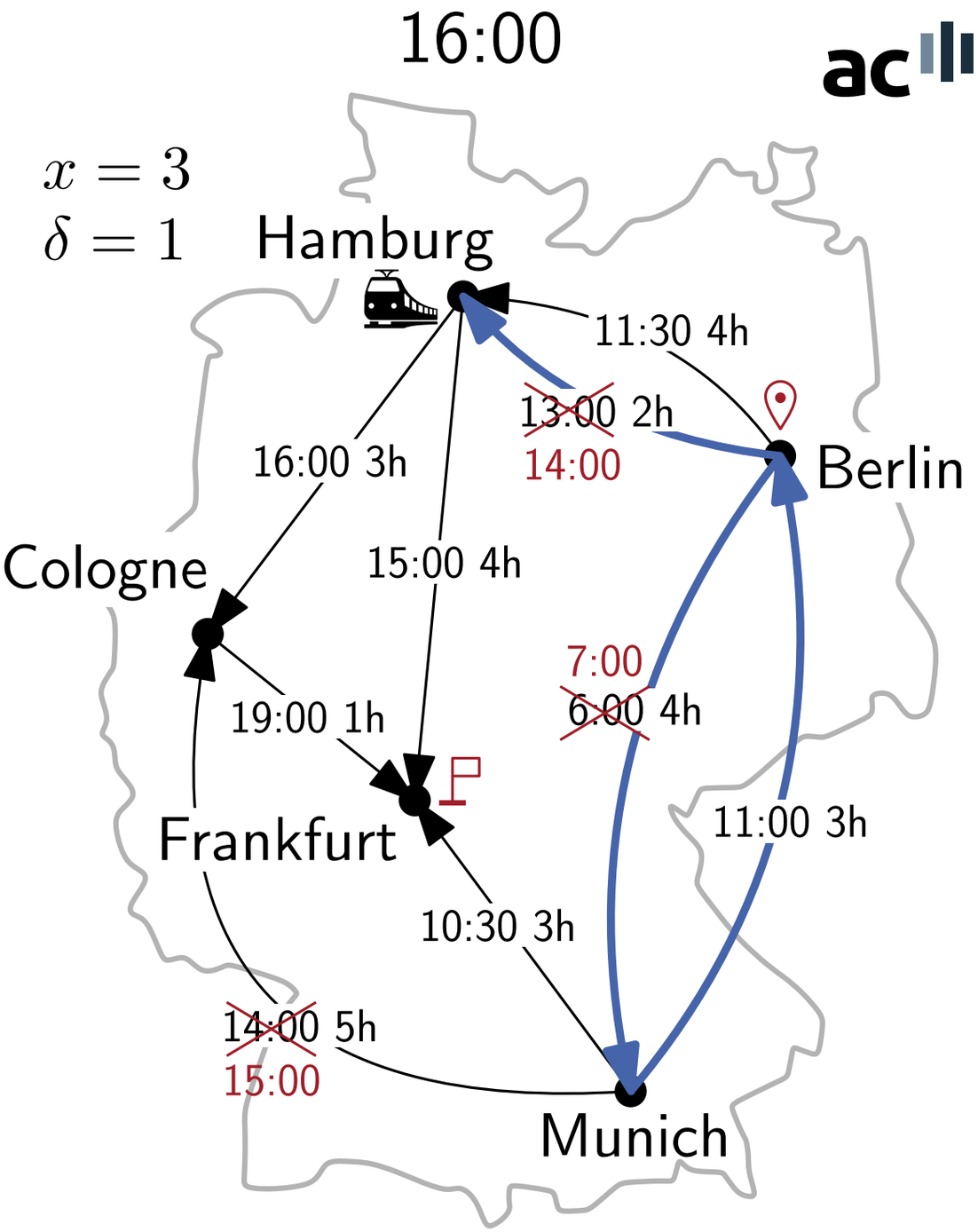
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ

Robust Connection Game

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

$\Rightarrow$ Does the Traveler have a winning strategy?



Problem Definition

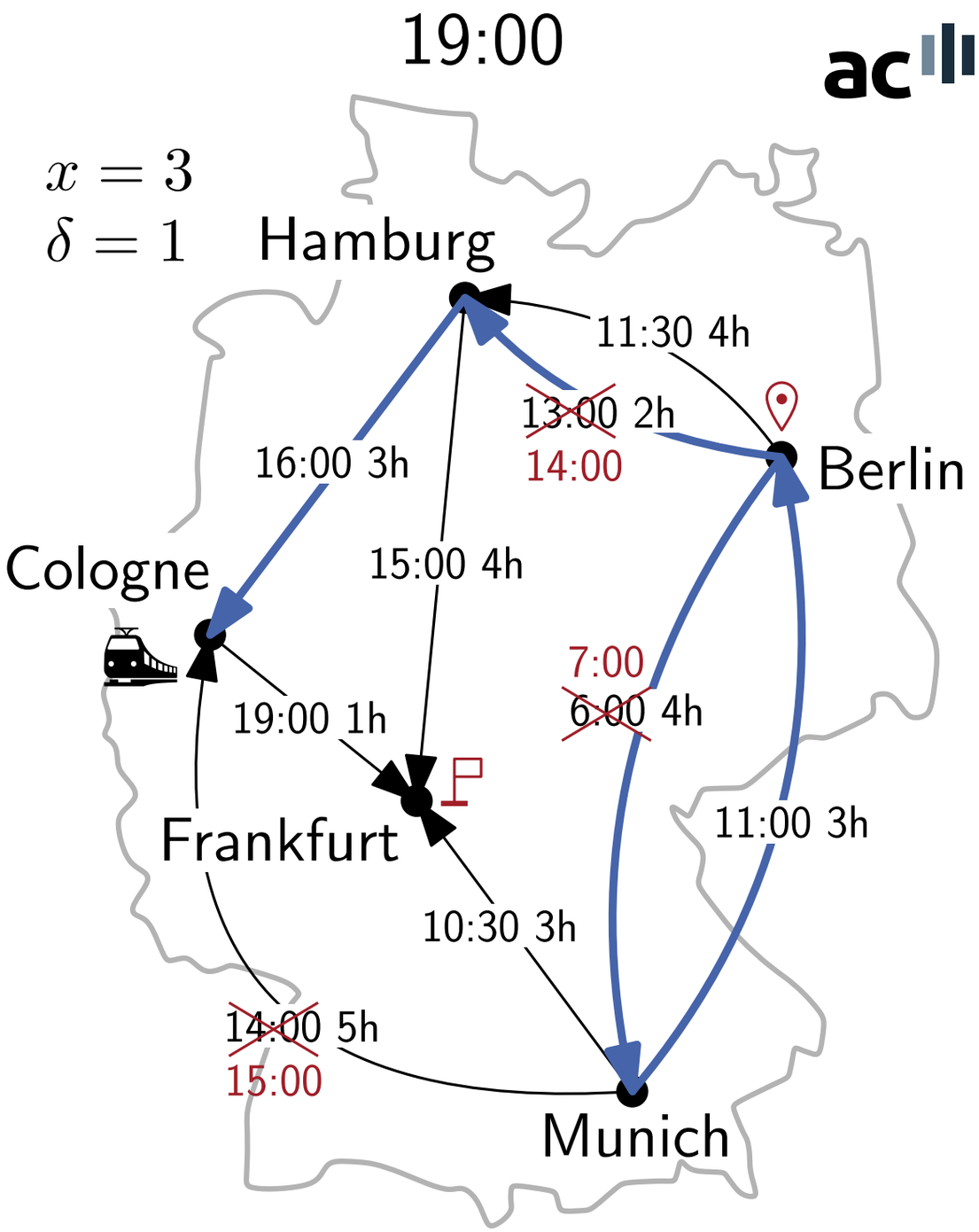
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ

Robust Connection Game

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

$\Rightarrow$ Does the Traveler have a winning strategy?



Problem Definition

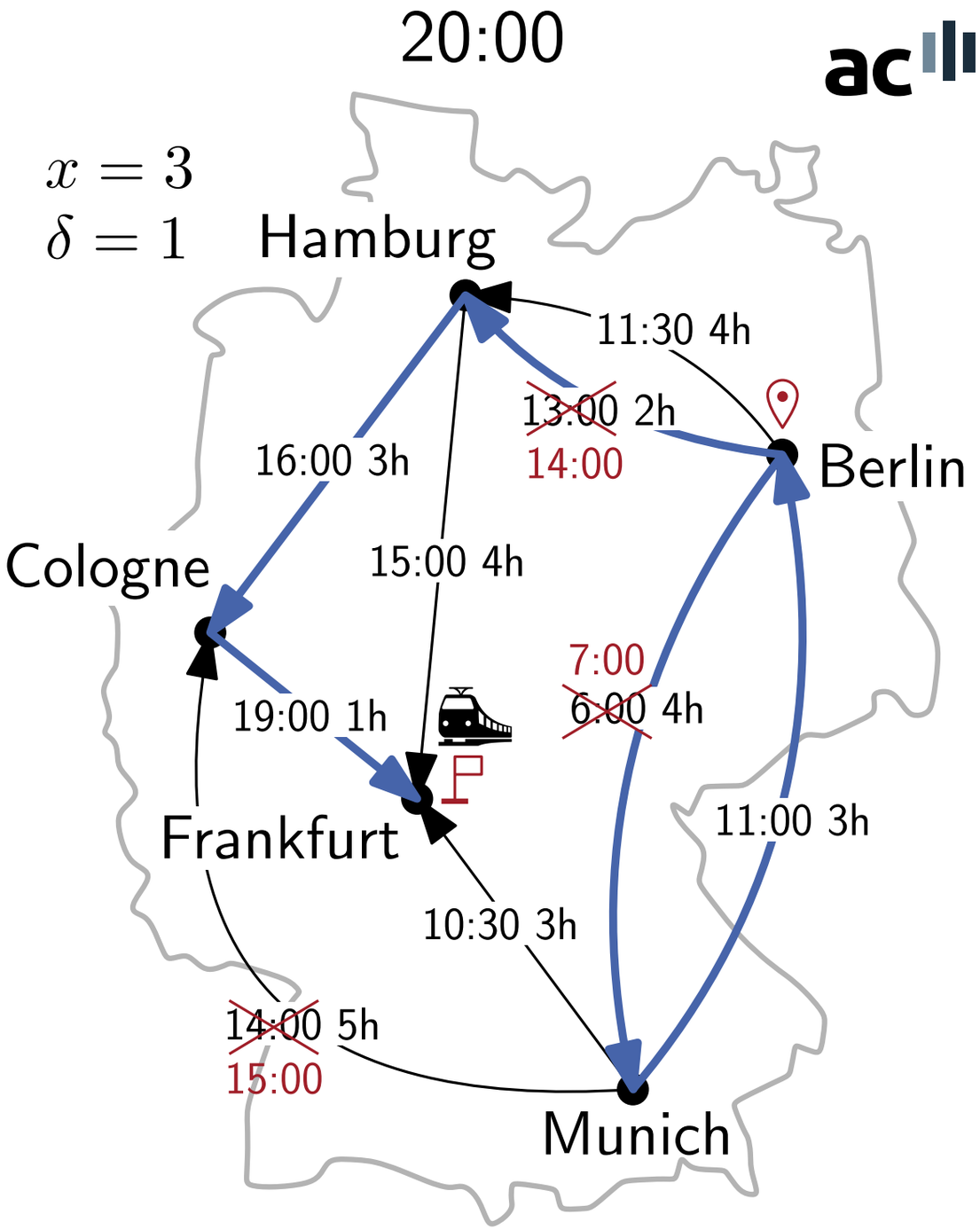
Setup:

- Temporal graph $G = (V, E)$
  - $e = (u, v, t, \lambda) \in E$
- Start vertex $s \in V$
- Goal vertex $z \in V$
- Total delay budget x
- Delay time δ

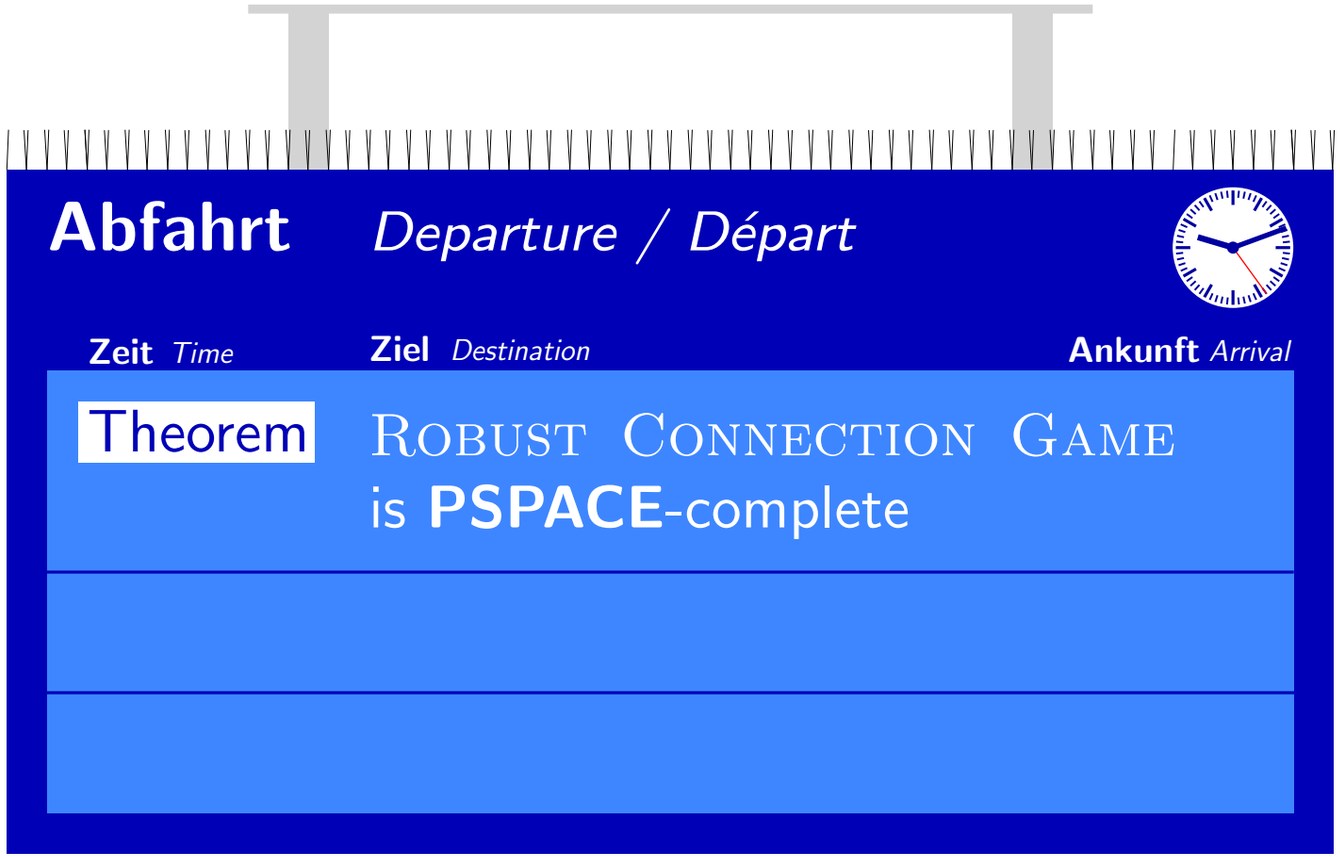
Robust Connection Game

- Adversary announces delays on upcoming outgoing arcs
- Traveler picks arc

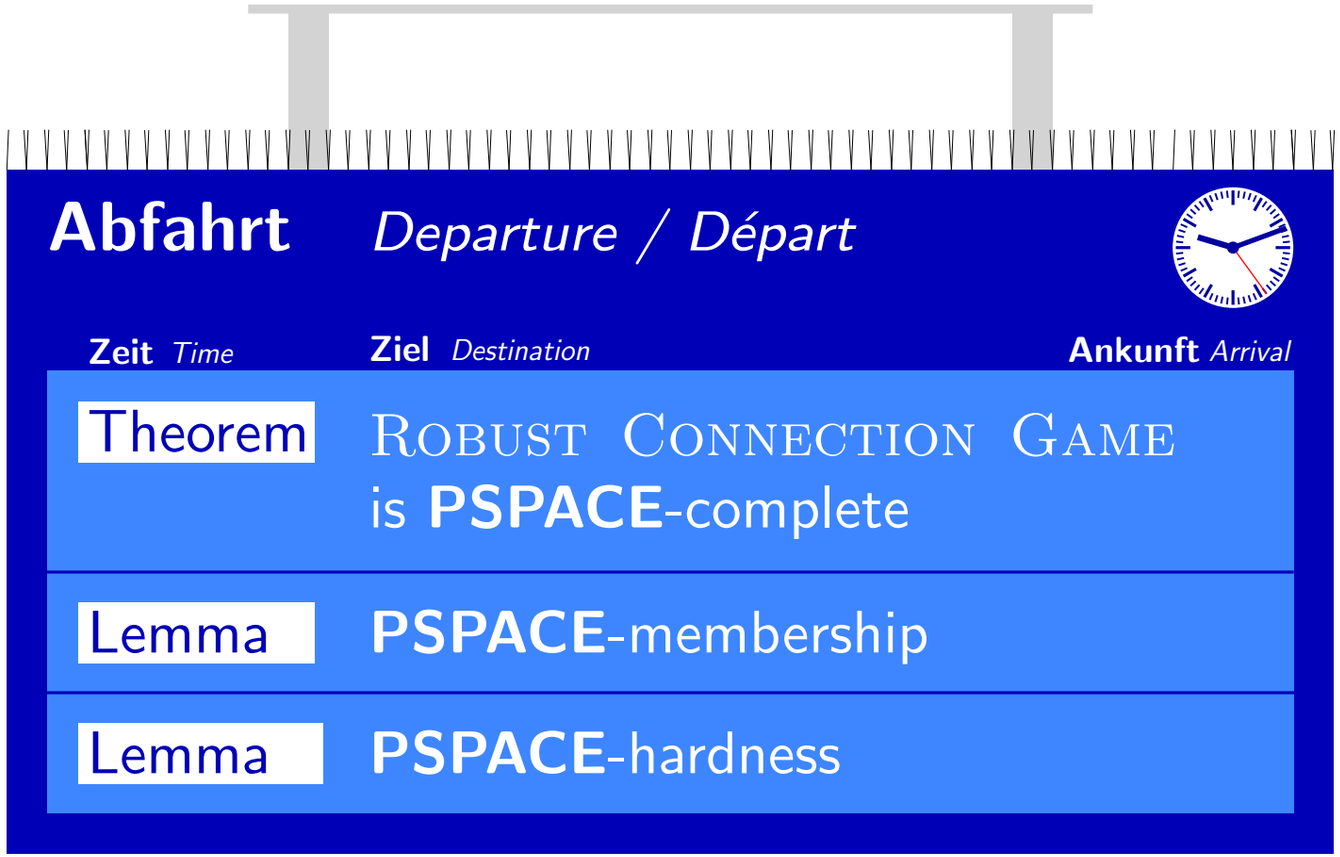
$\Rightarrow$ Does the Traveler have a winning strategy?



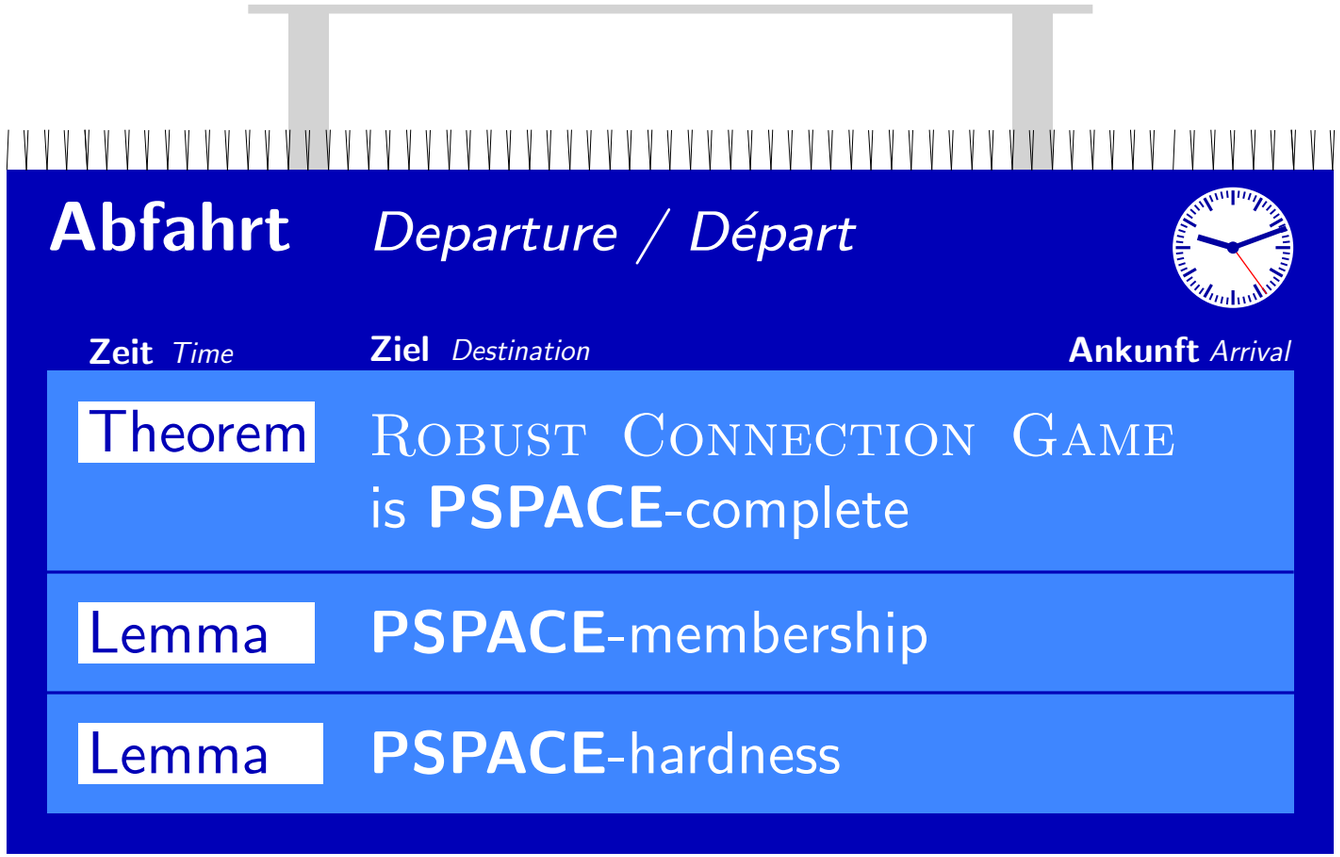
Overview


Abfahrt
Departure / Départ
Zeit Time
Ziel Destination
Ankunft Arrival
Theorem
Robust Connection Game
is PSPACE-complete

Overview


Abfahrt
Departure / Départ
Zeit Time
Ziel Destination
Ankunft Arrival
Theorem
Robust Connection Game
is PSPACE-complete
Lemma
PSPACE-membership
Lemma
PSPACE-hardness

Overview


Abfahrt
Departure / Départ
Zeit Time
Ziel Destination
Ankunft Arrival
Theorem
Robust Connection Game
is PSPACE-complete
Lemma
PSPACE-membership
Lemma
PSPACE-hardness

PSPACE-Membership of Robust Connection Game

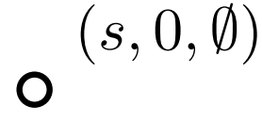
- Dynamic programming

PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D

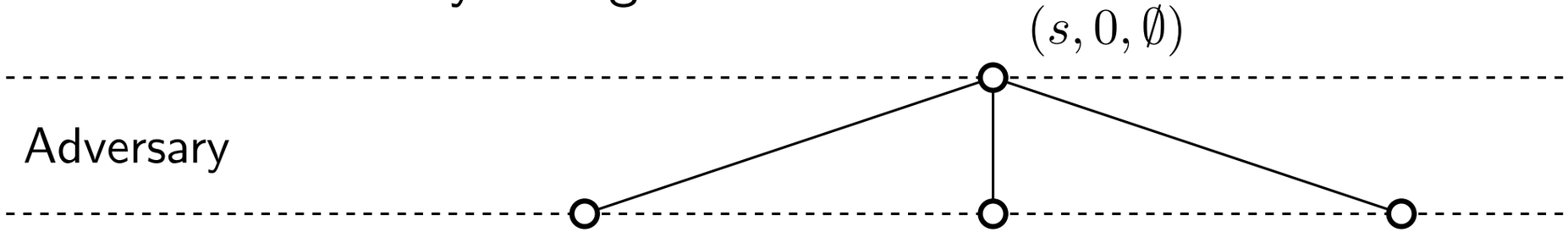
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



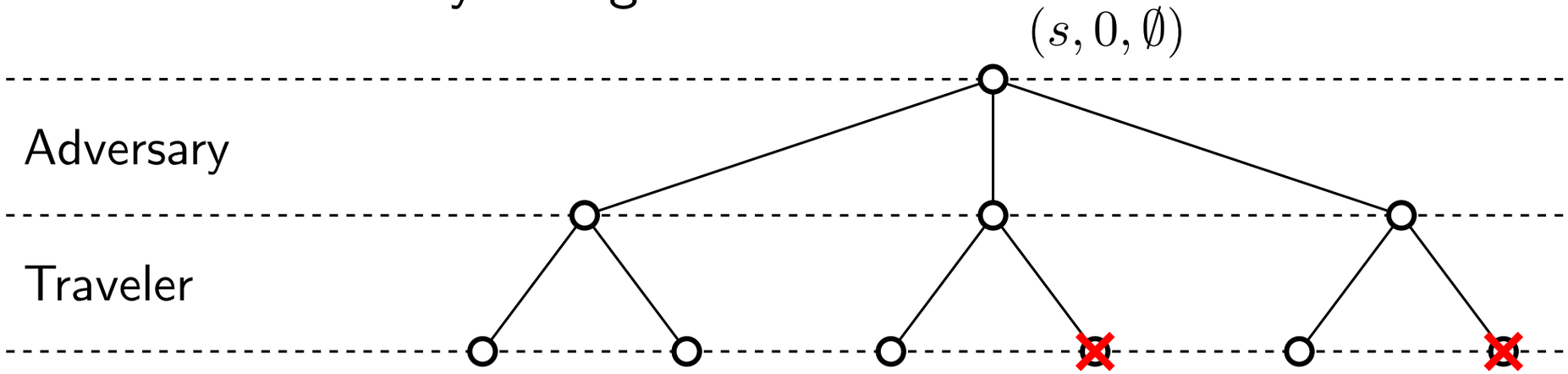
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



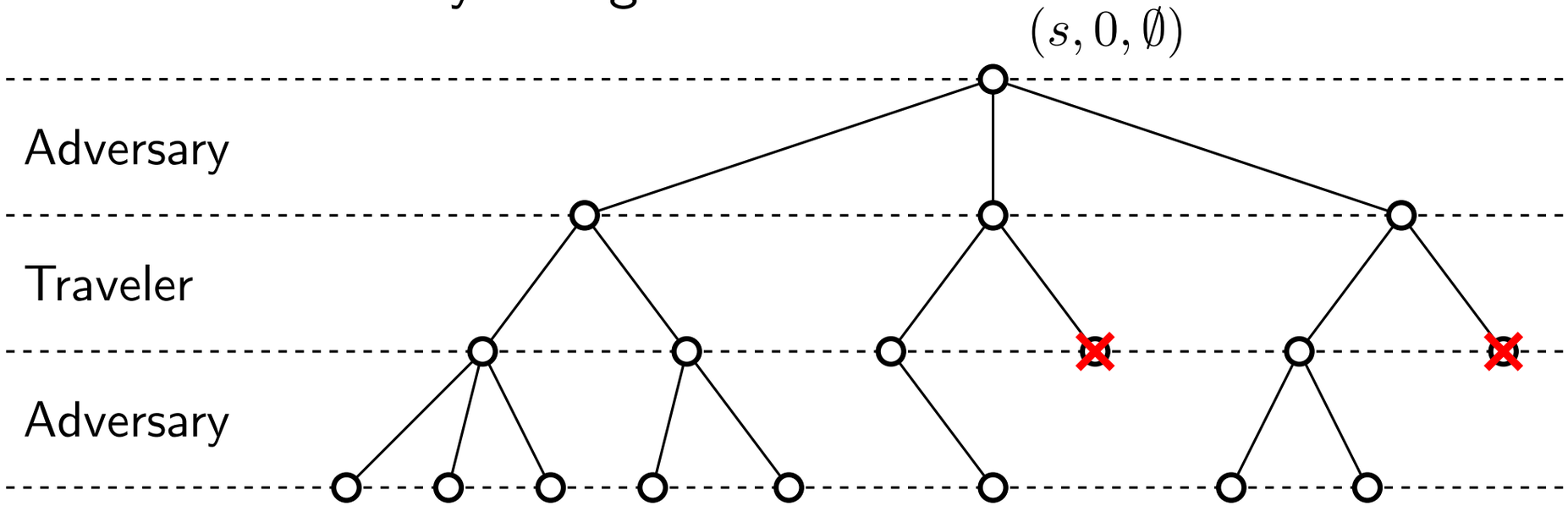
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



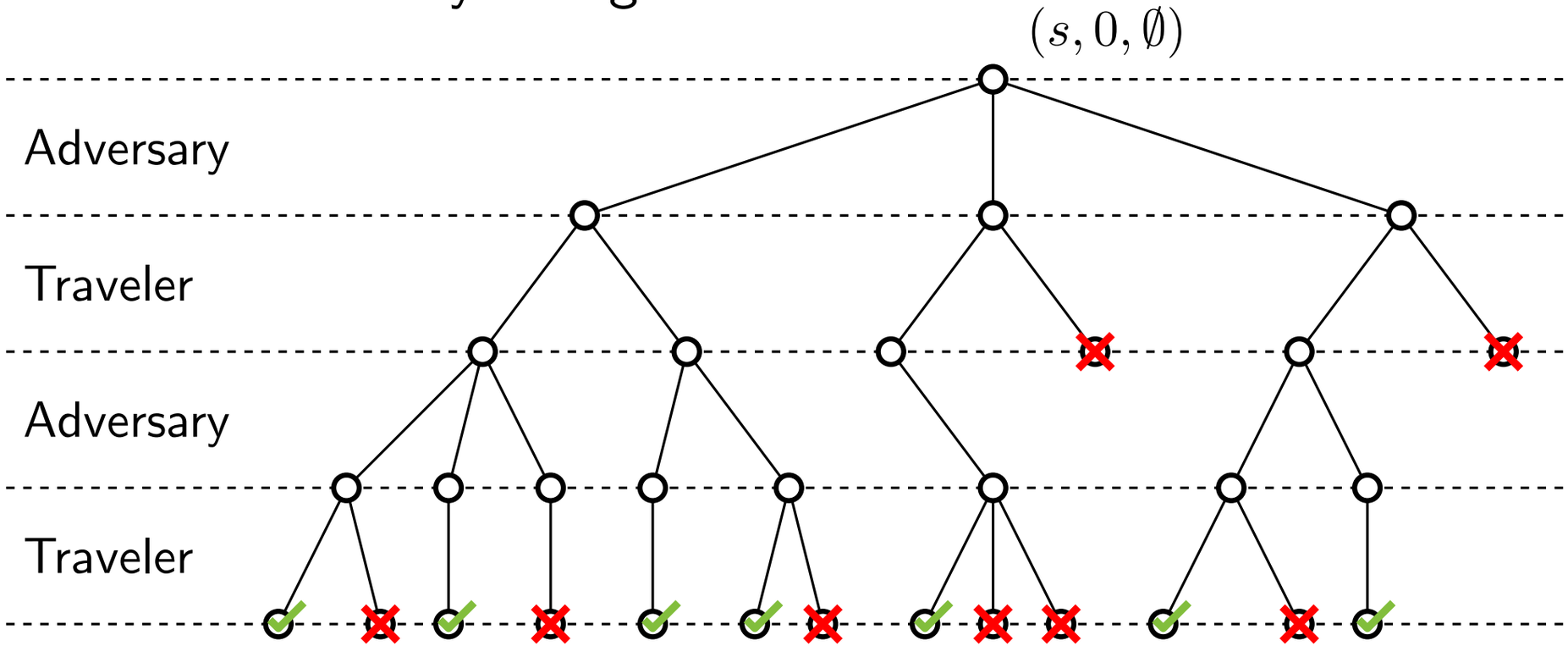
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



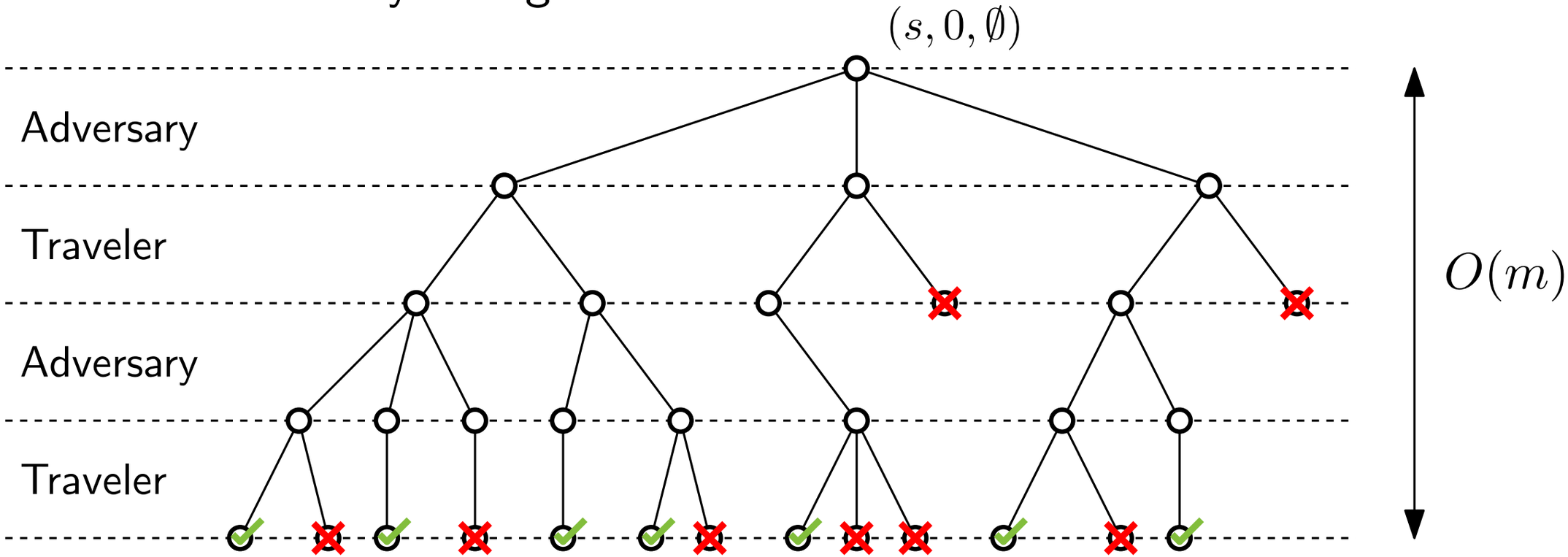
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



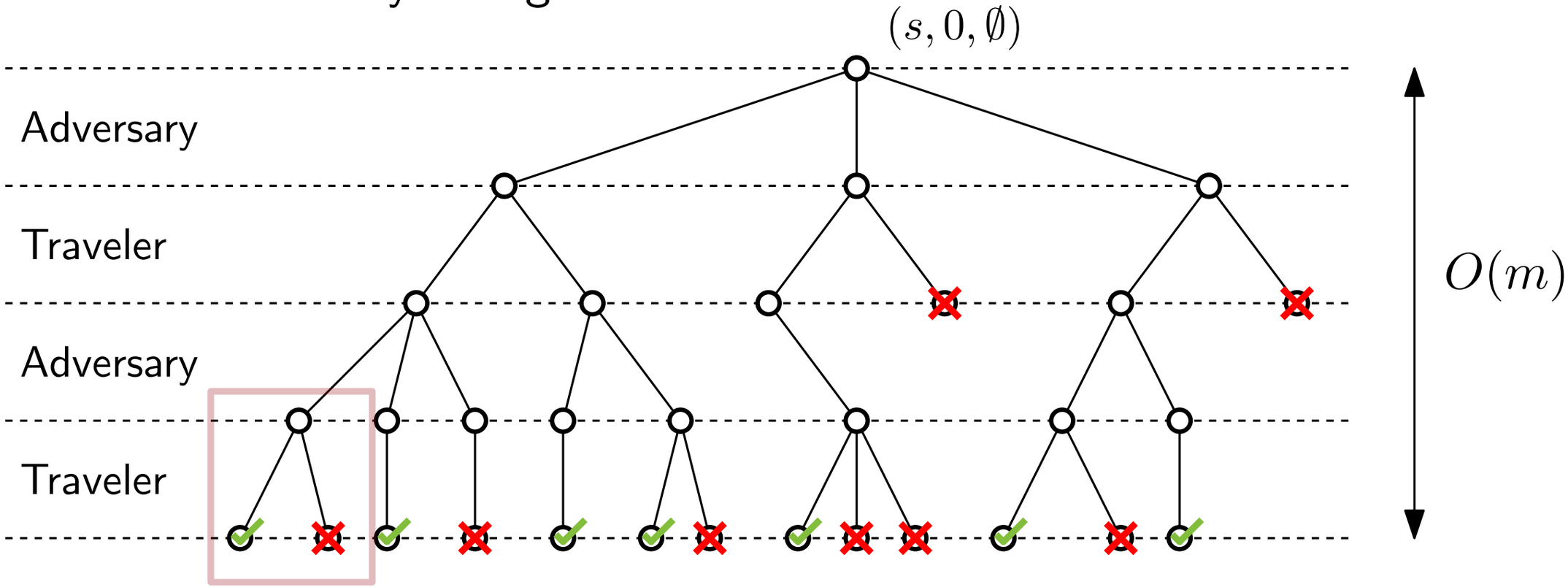
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



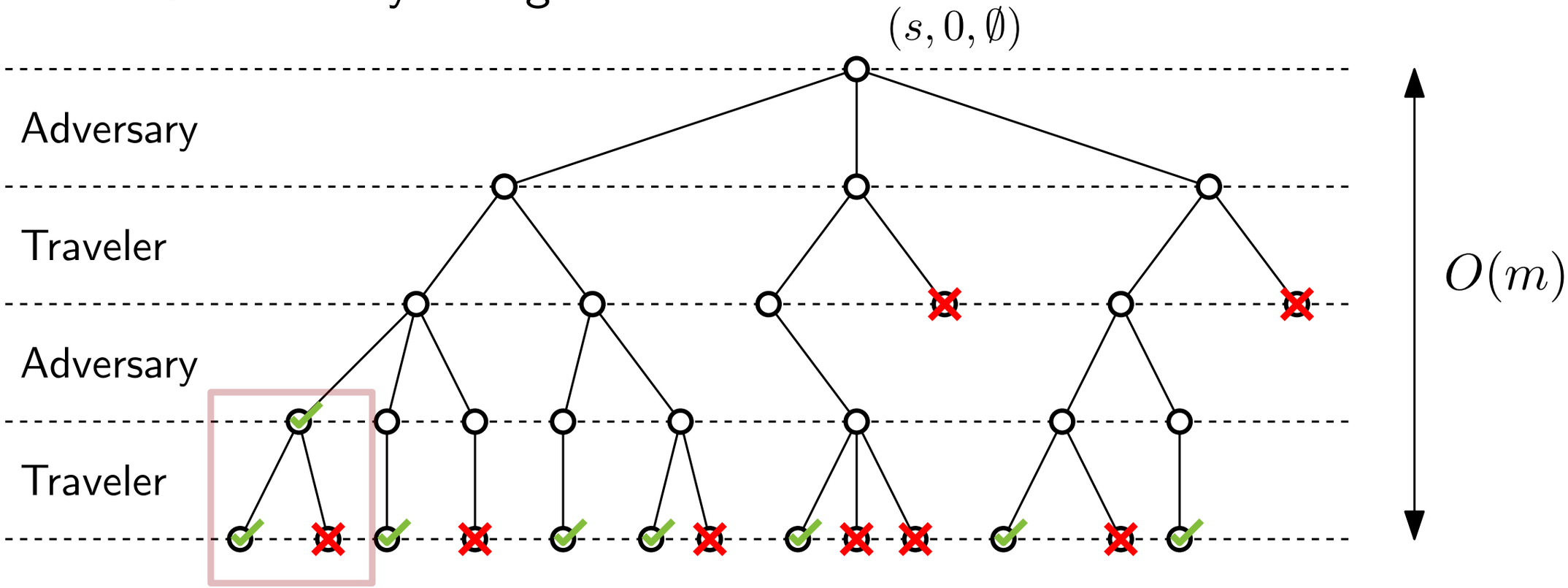
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



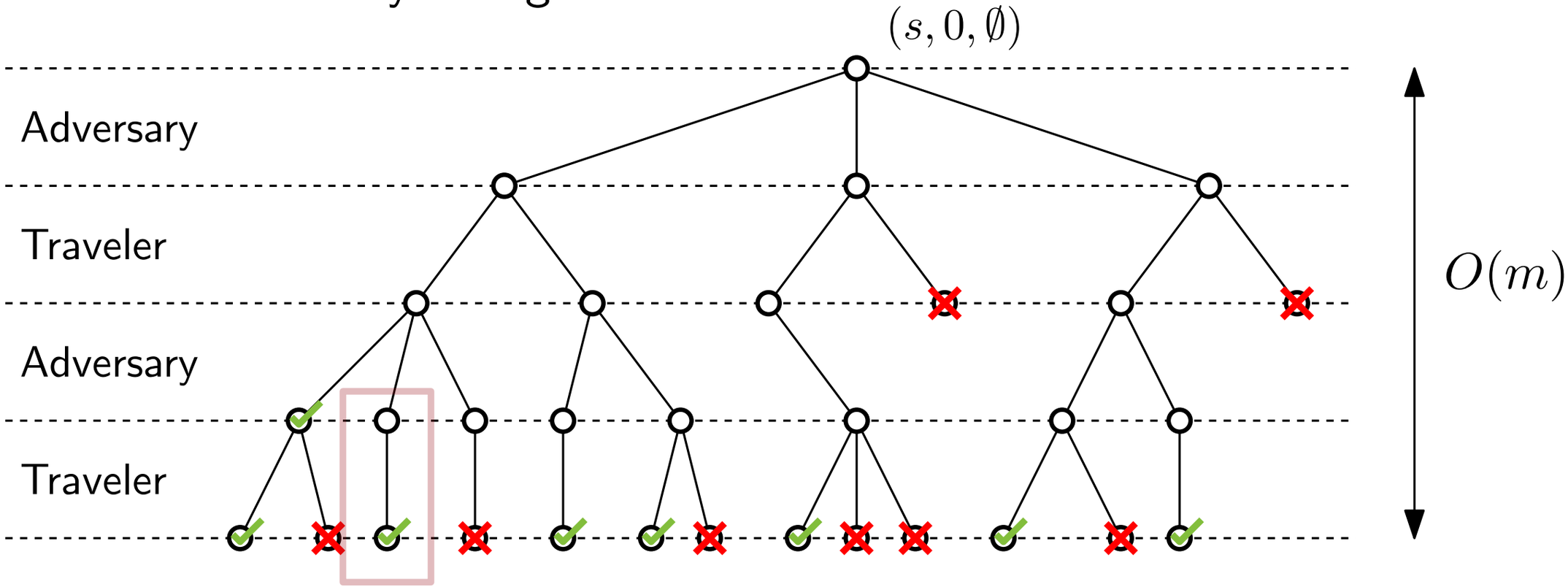
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



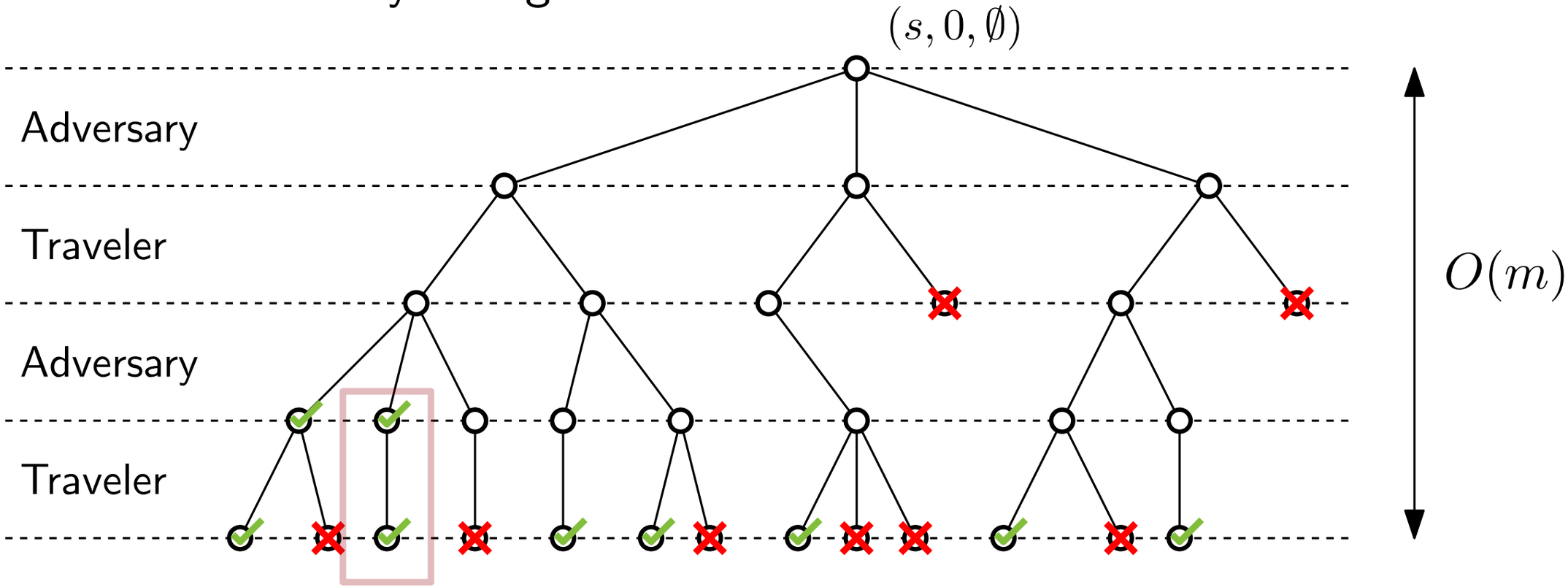
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



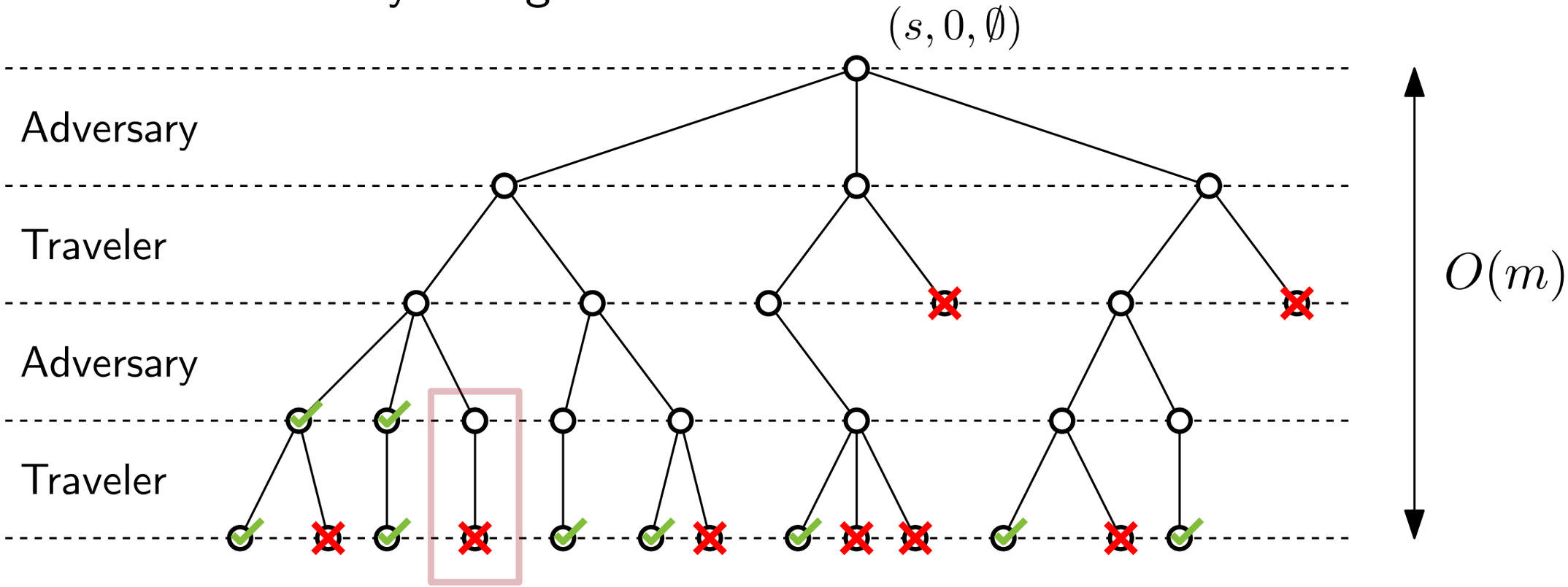
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



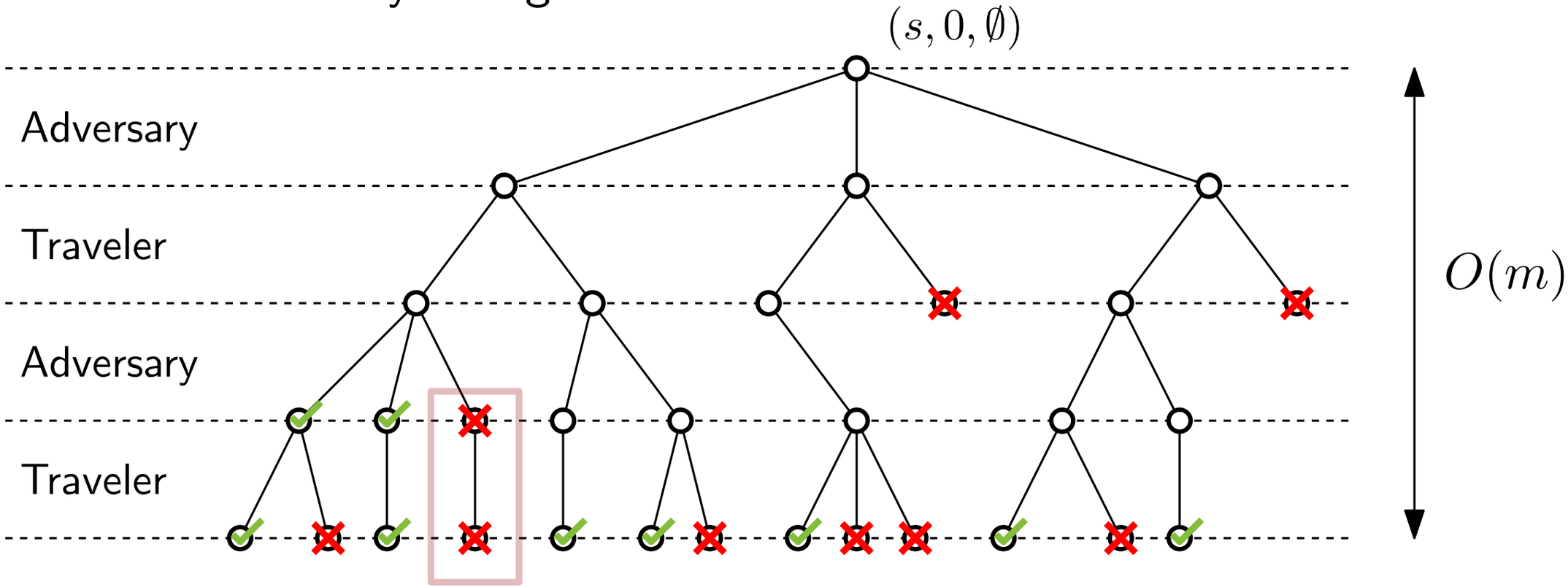
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



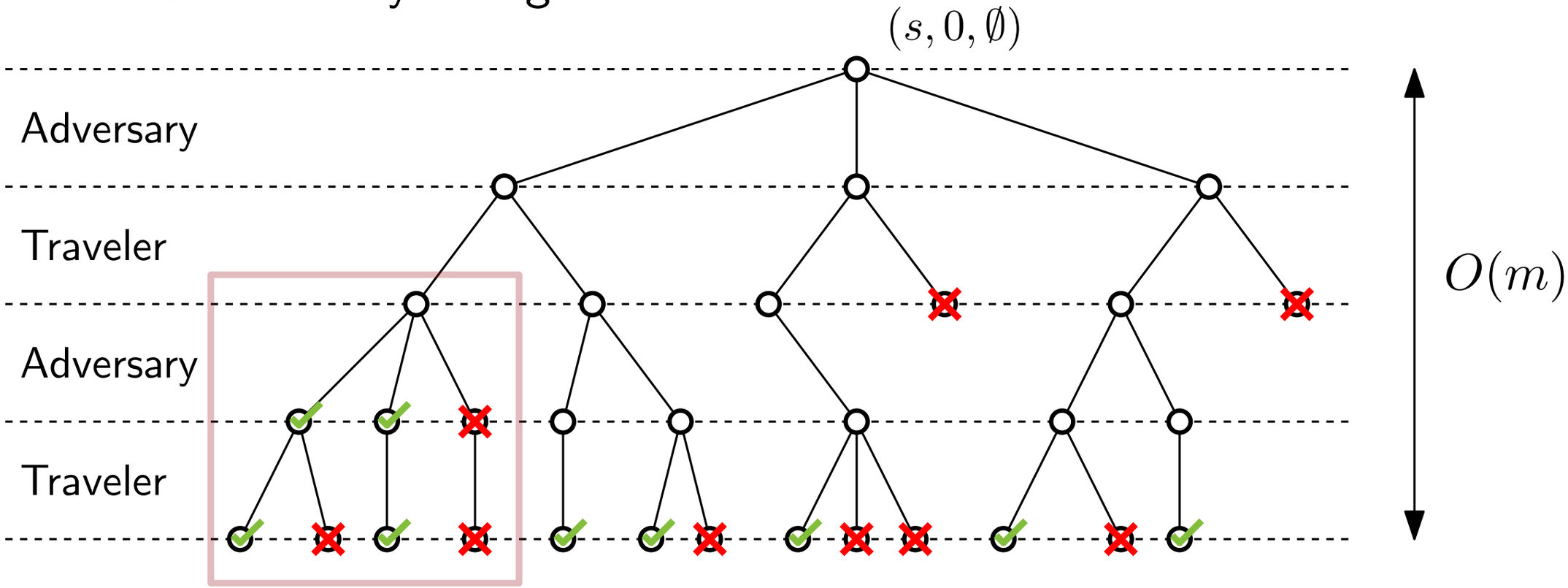
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



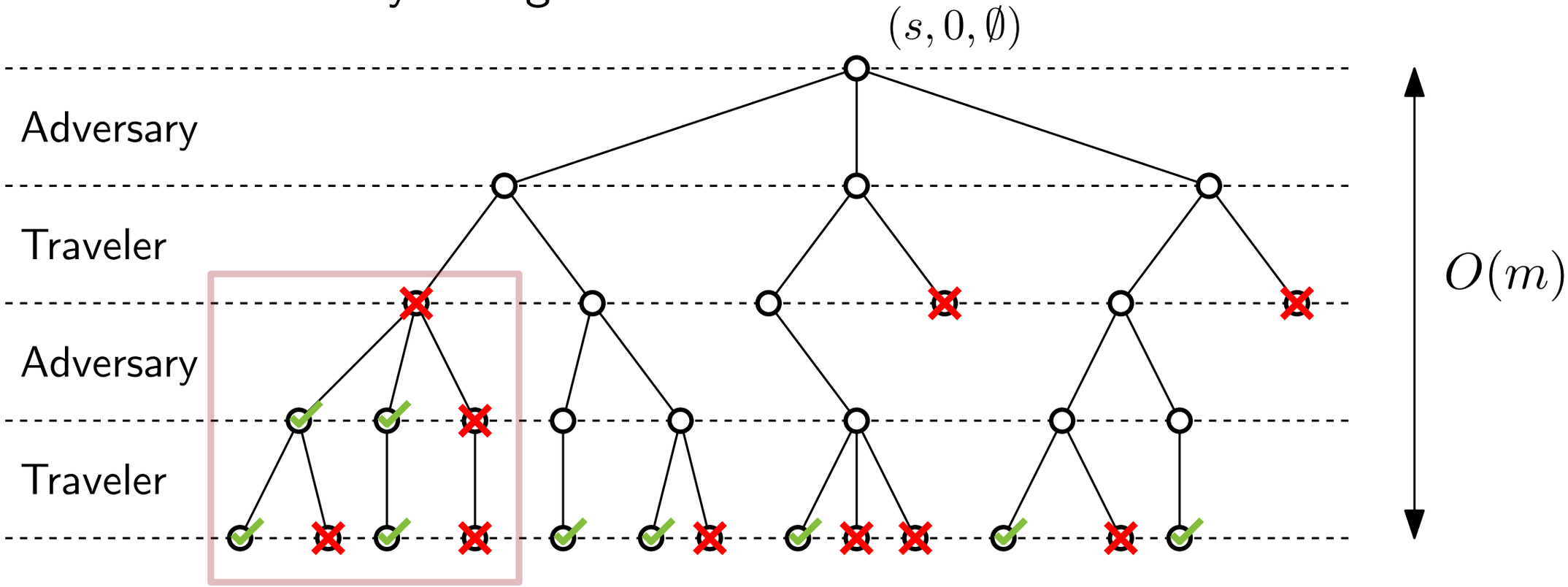
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



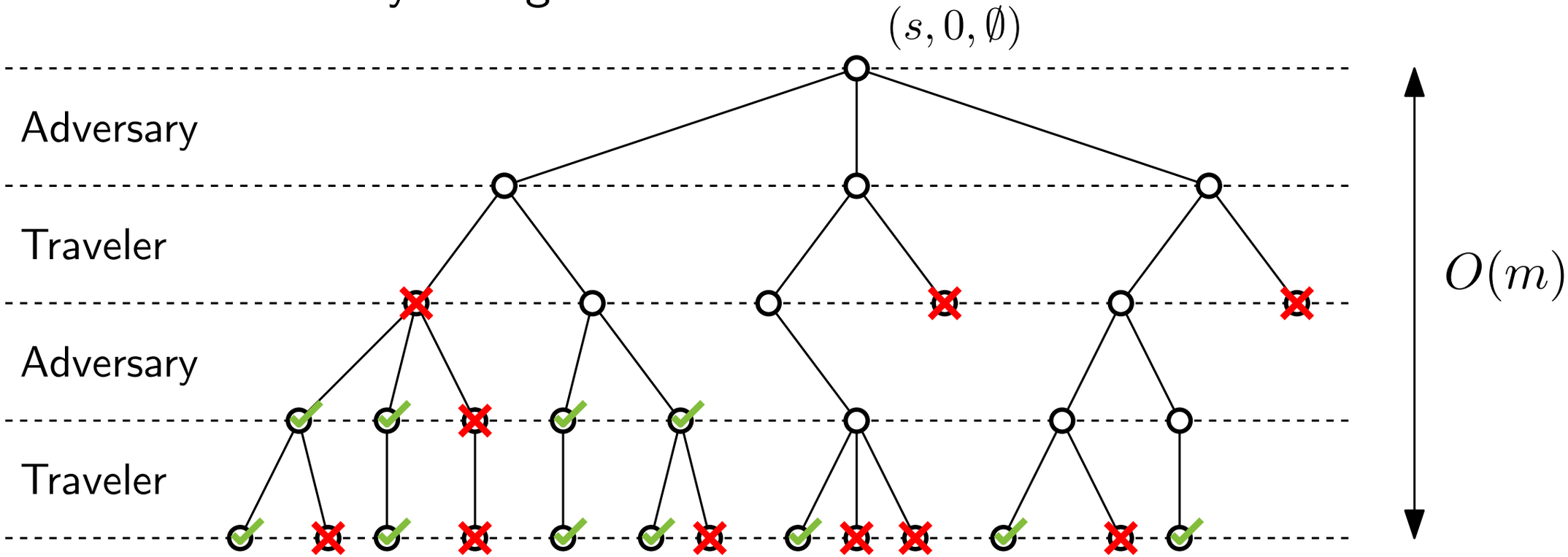
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



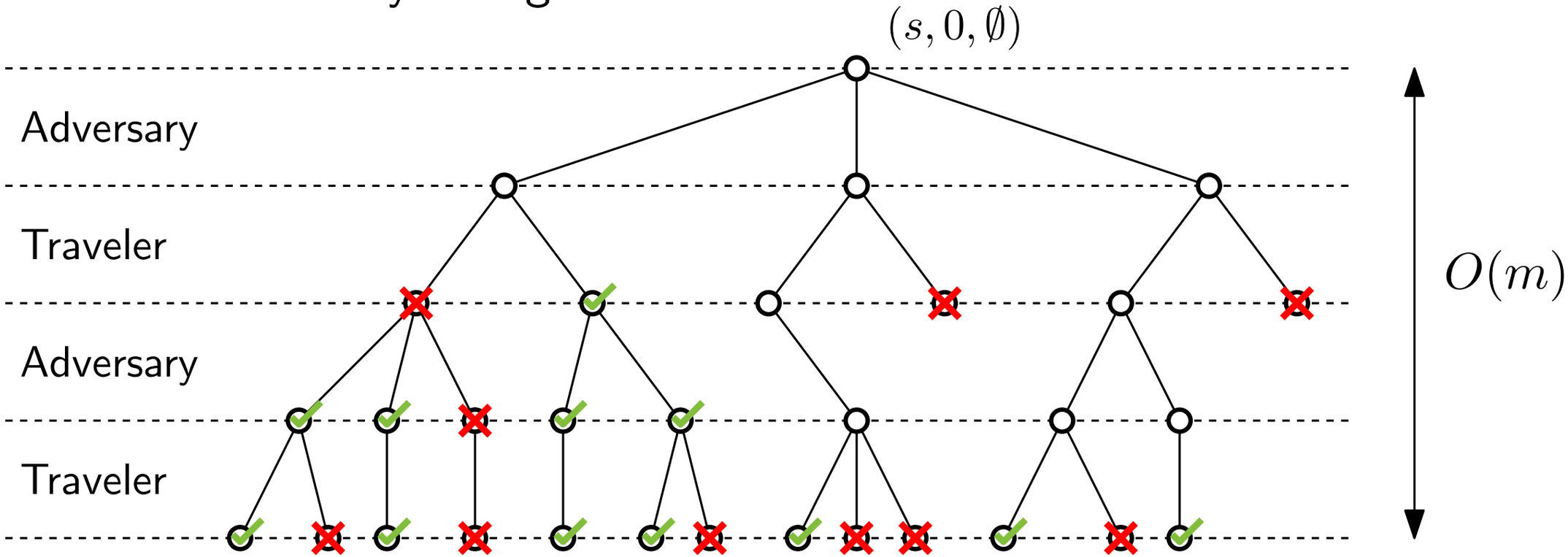
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



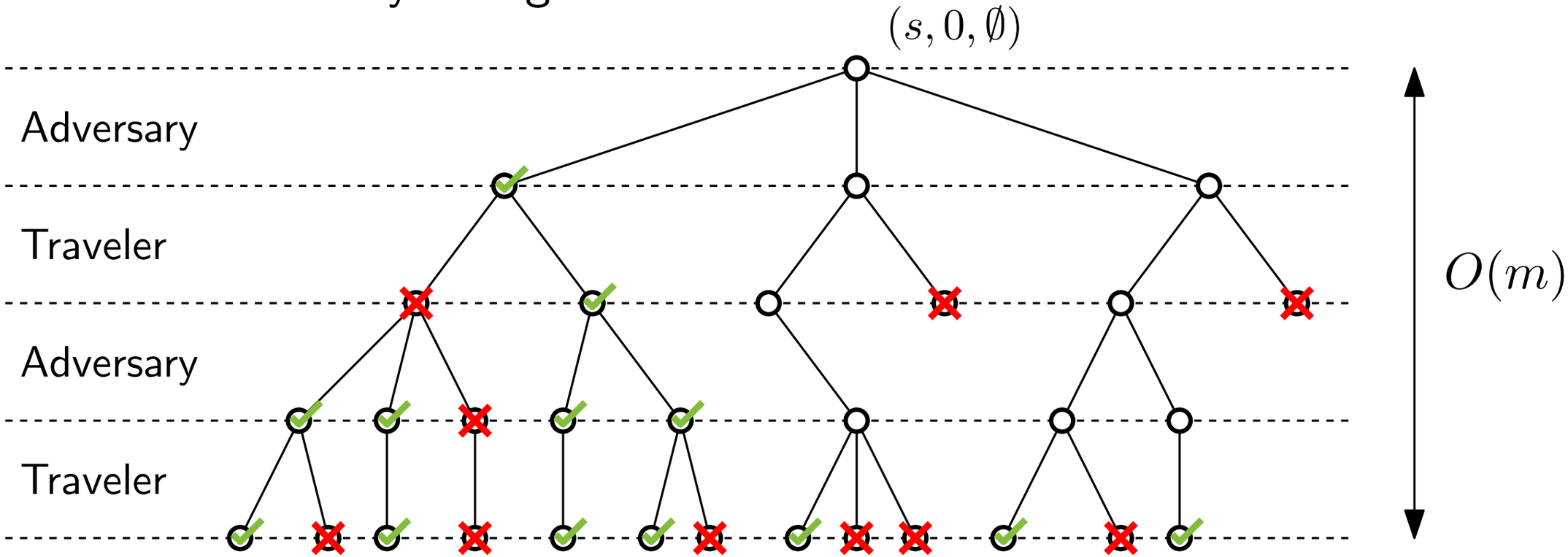
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



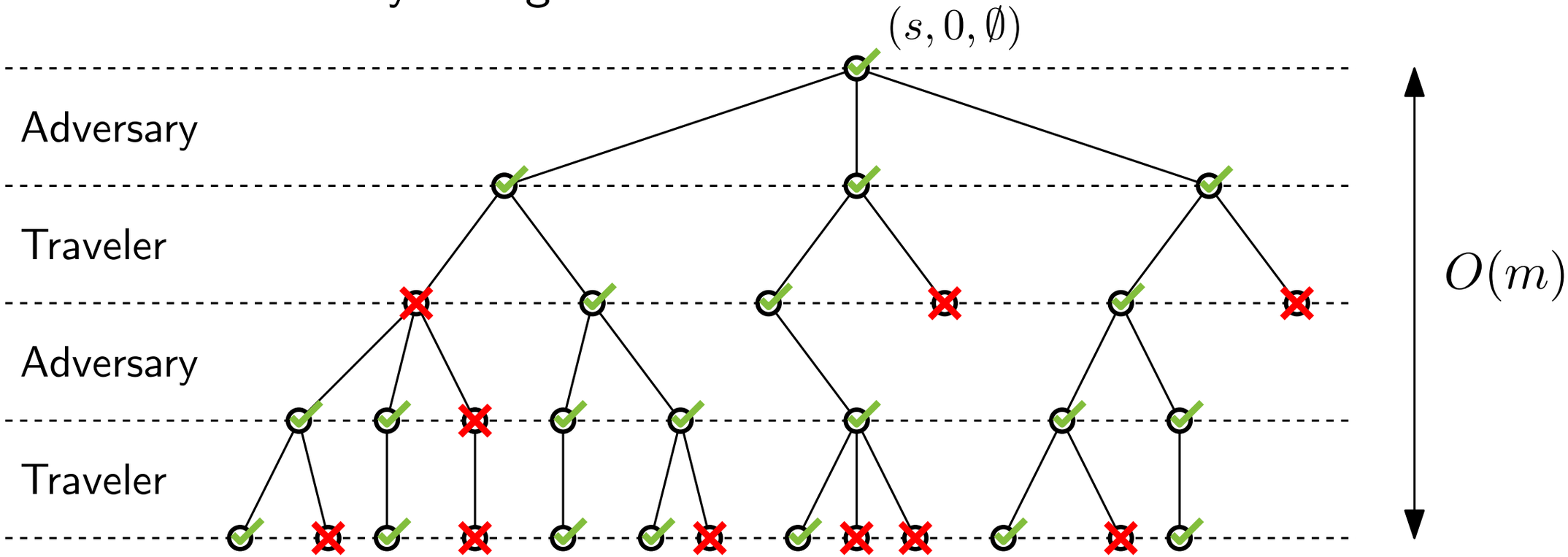
PSPACE-Membership of Robust Connection Game

- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D

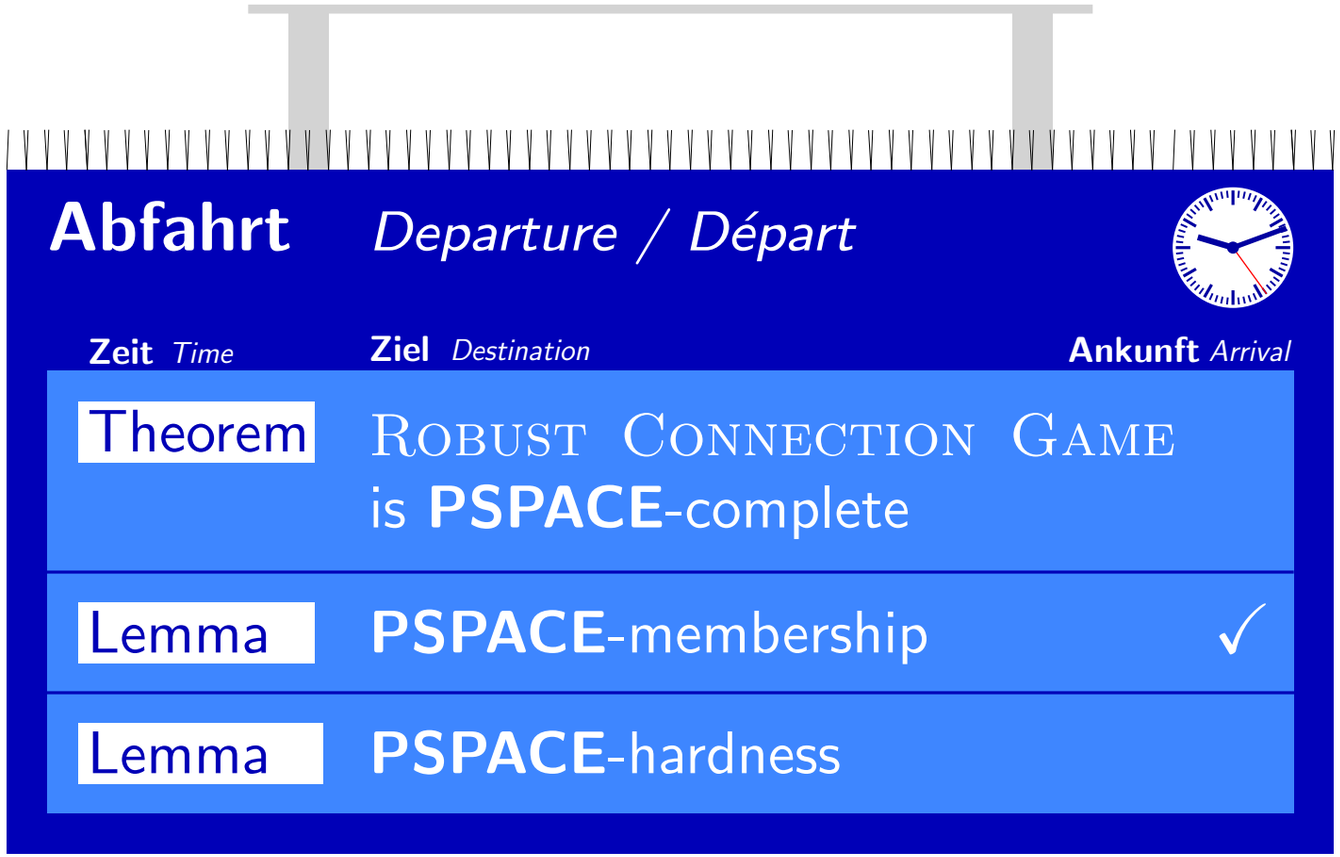


PSPACE-Membership of Robust Connection Game

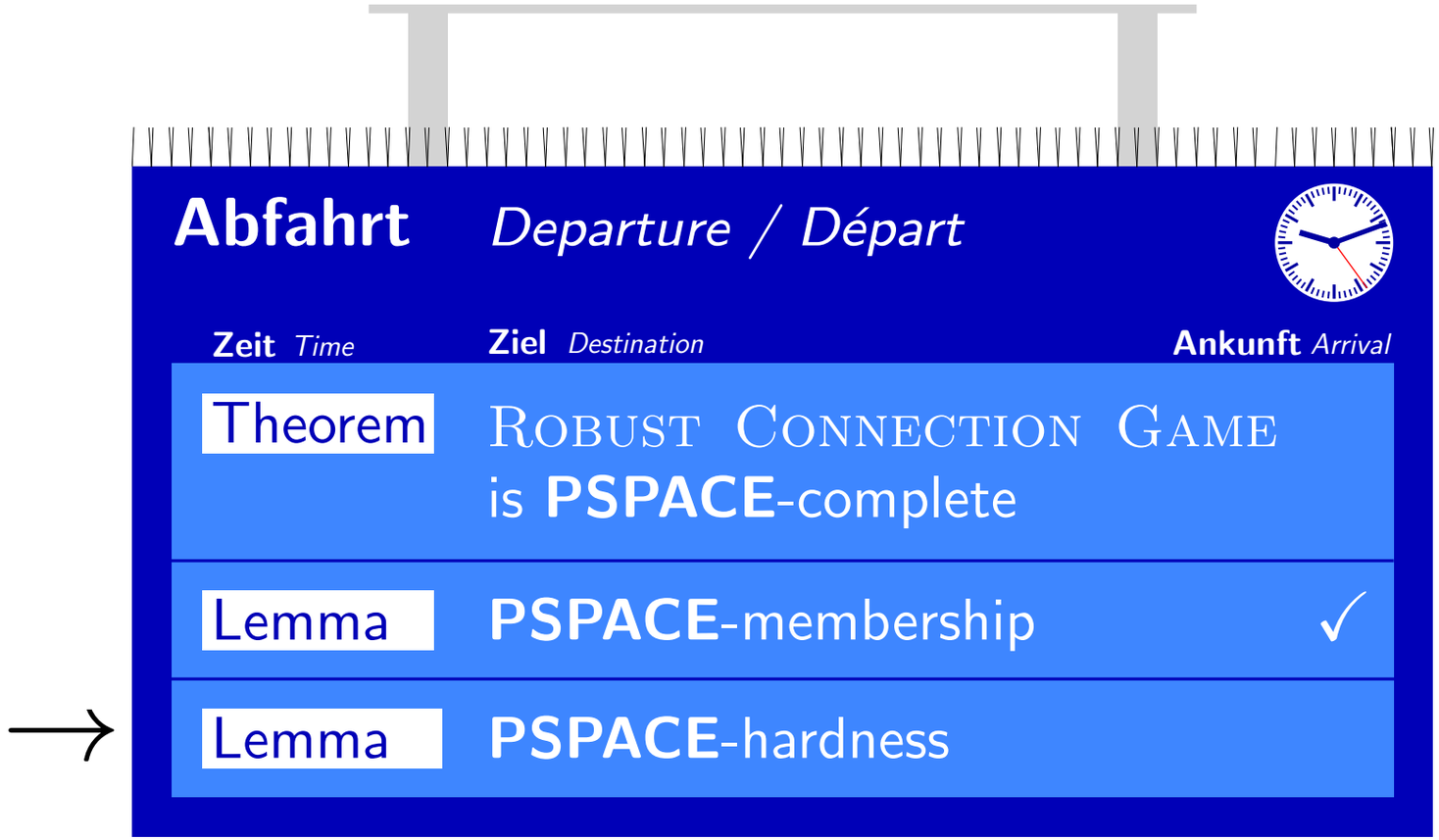
- Dynamic programming
- Game state (v, t, D)
  - Current vertex v
  - Current time t
  - Set of delayed edges D



Overview


Abfahrt
Departure / Départ
Zeit Time
Ziel Destination
Ankunft Arrival
Theorem
Robust Connection Game is PSPACE-complete
Lemma
PSPACE-membership ✓
Lemma
PSPACE-hardness

Overview


Abfahrt
Departure / Départ
Zeit Time
Ziel Destination
Ankunft Arrival
Theorem
Robust Connection Game
is PSPACE-complete
Lemma
PSPACE-membership
✓
Lemma
PSPACE-hardness

PSPACE-Hardness of Robust Connection Game

- QBF-Game
  Input: a QBF $\phi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$ with $Q_i \in \{\exists, \forall\}$
  Game: 2-players ($\exists$- and $\forall$-player), $\exists$-player wins if φ is satisfied
  Question: Does the $\exists$-player have a winning strategy?

PSPACE-Hardness of Robust Connection Game

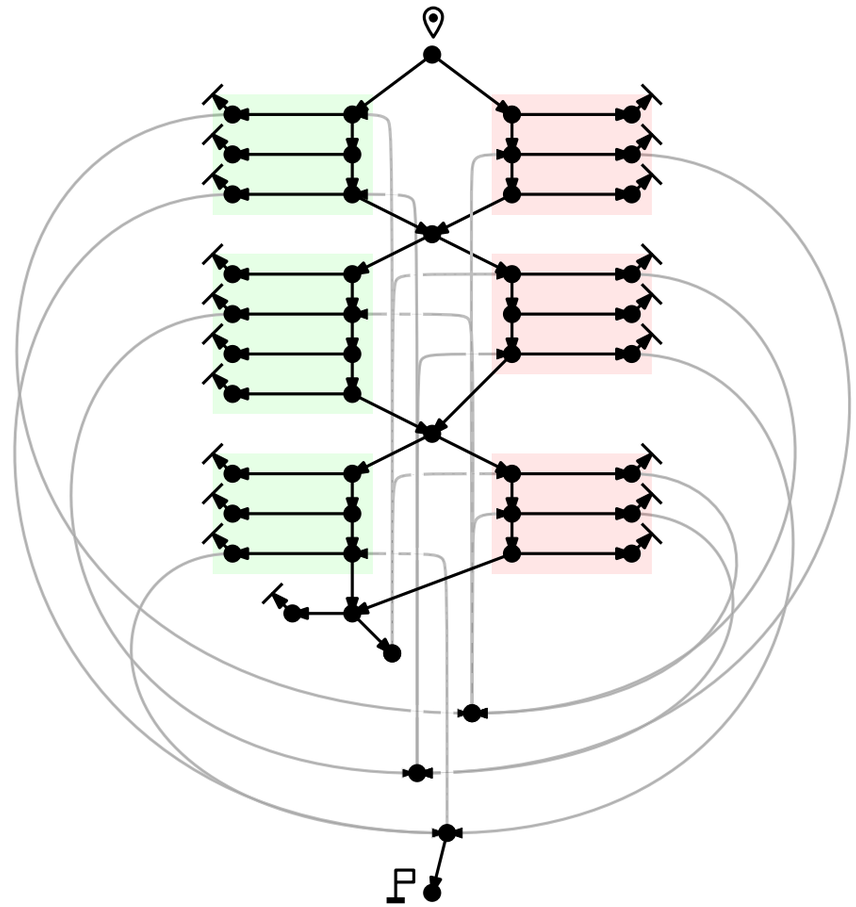
- QBF-Game
  Input: a QBF $\phi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$ with $Q_i \in \{\exists, \forall\}$
  Game: 2-players ($\exists$- and $\forall$-player), $\exists$-player wins if φ is satisfied
  Question: Does the $\exists$-player have a winning strategy?
- $\varphi = C_1 \wedge C_2 \wedge \dots C_m$

PSPACE-Hardness of Robust Connection Game

- QBF-Game
  Input: a QBF $\phi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$ with $Q_i \in \{\exists, \forall\}$
  Game: 2-players ($\exists$- and $\forall$-player), $\exists$-player wins if φ is satisfied
  Question: Does the $\exists$-player have a winning strategy?
- $\varphi = C_1 \wedge C_2 \wedge \dots C_m$

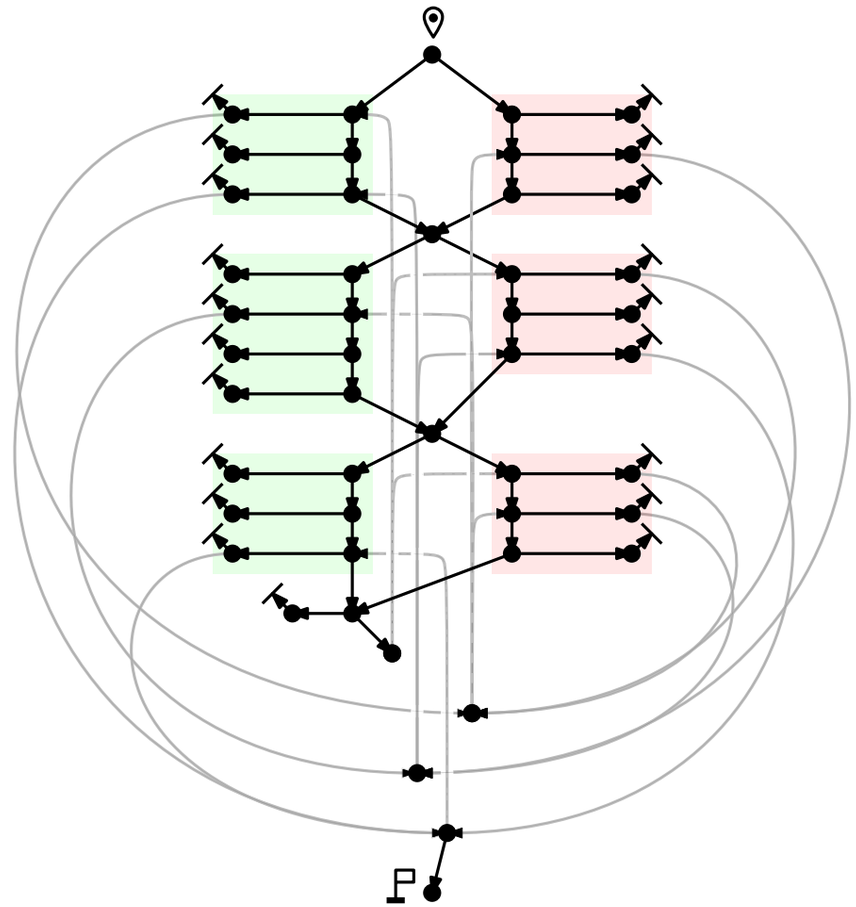
PSPACE-Hardness of Robust Connection Game

- QBF-Game
  Input: a QBF $\phi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$ with $Q_i \in \{\exists, \forall\}$
  Game: 2-players ($\exists$- and $\forall$-player), $\exists$-player wins if φ is satisfied
  Question: Does the $\exists$-player have a winning strategy?
- $\varphi = C_1 \wedge C_2 \wedge \dots C_m$



PSPACE-Hardness of Robust Connection Game

- QBF-Game
  Input: a QBF $\phi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$ with $Q_i \in \{\exists, \forall\}$
  Game: 2-players ($\exists$- and $\forall$-player), $\exists$-player wins if φ is satisfied
  Question: Does the $\exists$-player have a winning strategy?
- $\varphi = C_1 \wedge C_2 \wedge \dots C_m$
- Travel time $\lambda(e) = 1 \; \forall e$
  Delay time $\delta = 1$



PSPACE-Hardness of Robust Connection Game

- QBF-Game

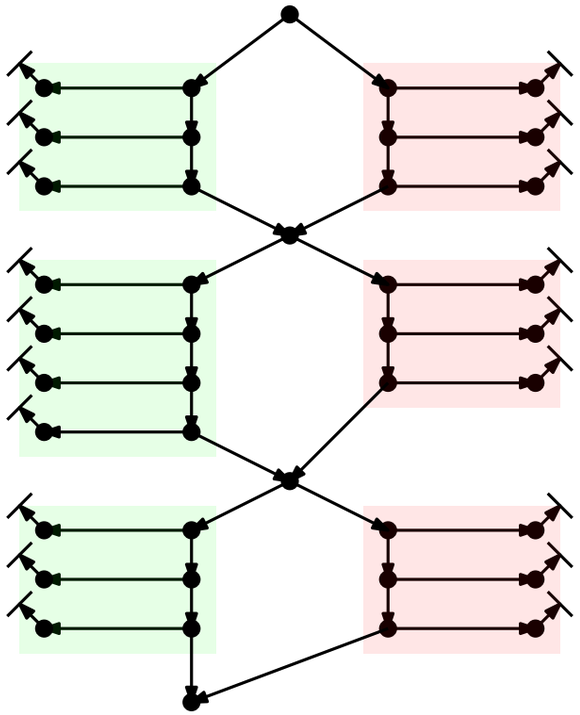
$Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$ with $Q_i \in$

and ∀-player), ∃-player wins if

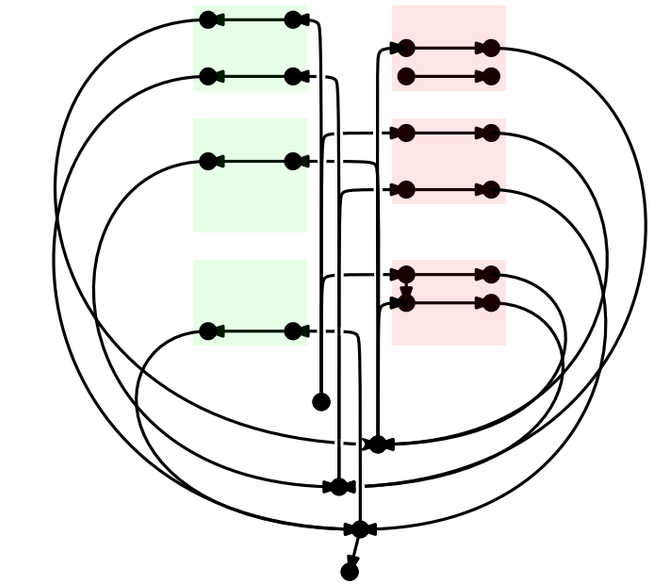
∃-player have a winning strate

m

$\forall e$



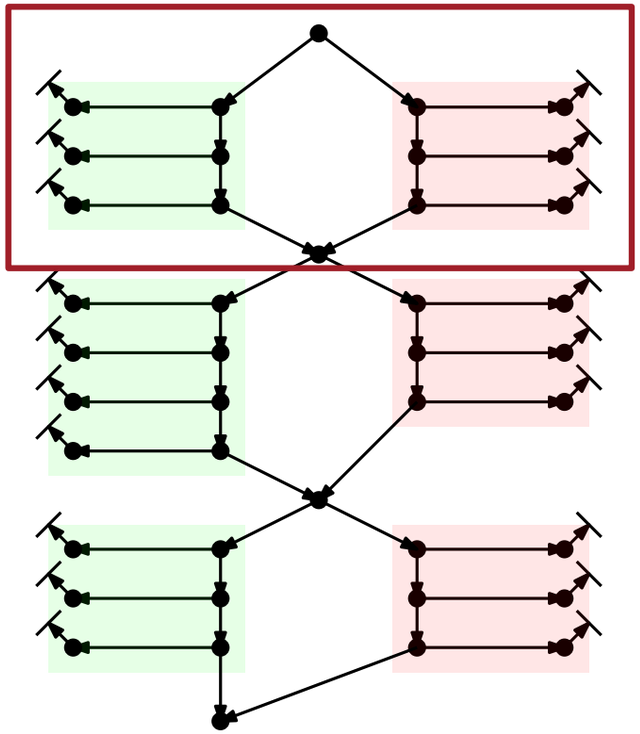
Assignment Phase



Evaluation Phase

PSPACE-Hardness of Robust Connection Game

- QBF-Game



Assignment Phase

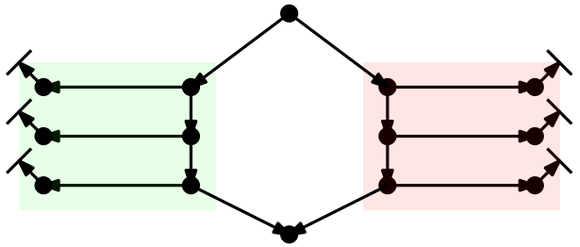
$Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$ with $Q_i \in$

an

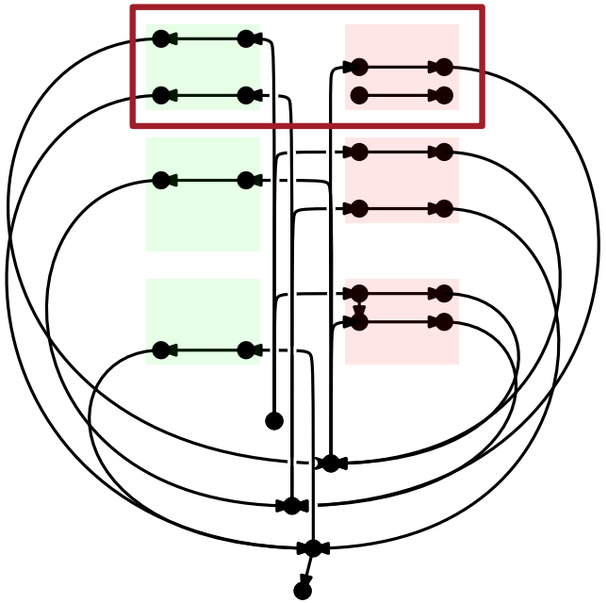
s if

∃-p

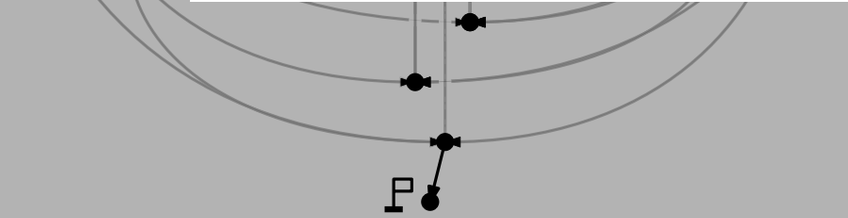
rate



Variable Gadget



Evaluation Phase



PSPACE-Hardness of Robust Connection Game

- QBF-Game

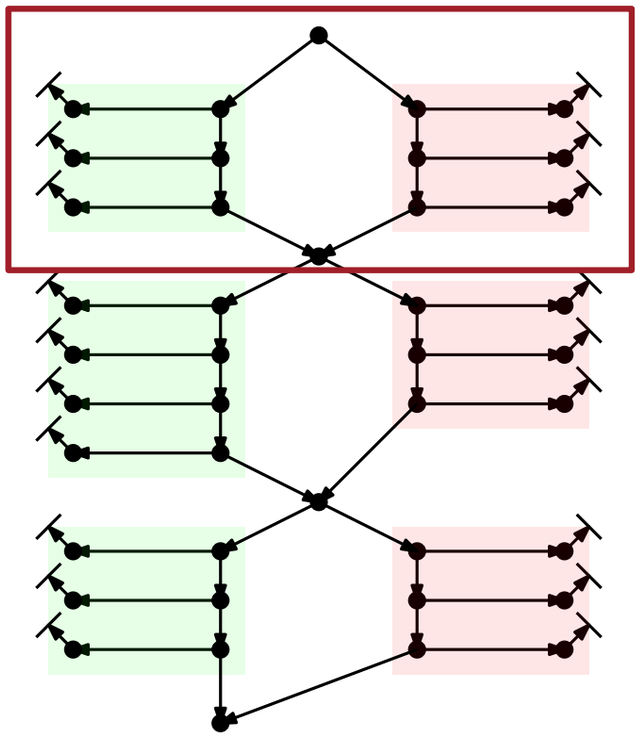
$Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$ with $Q_i \in$

an

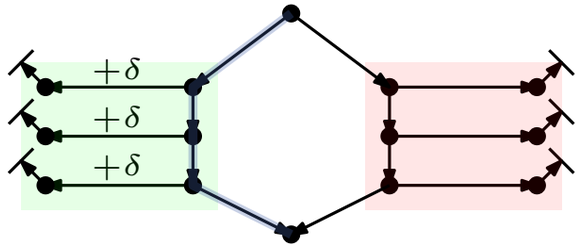
s if

∃-p

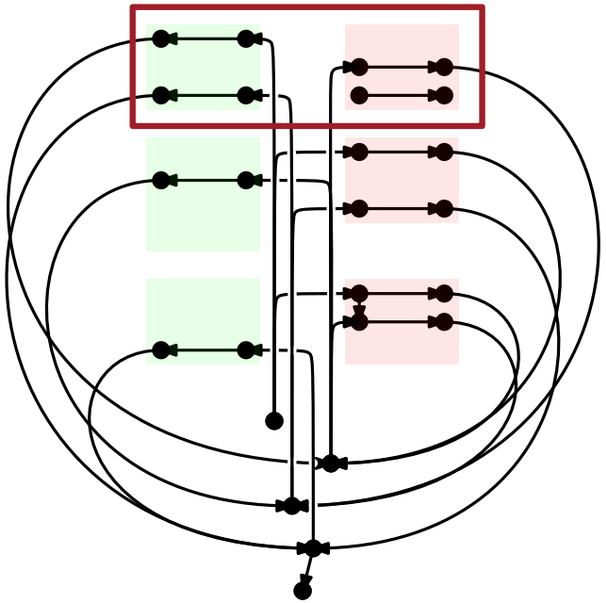
rate



Assignment Phase



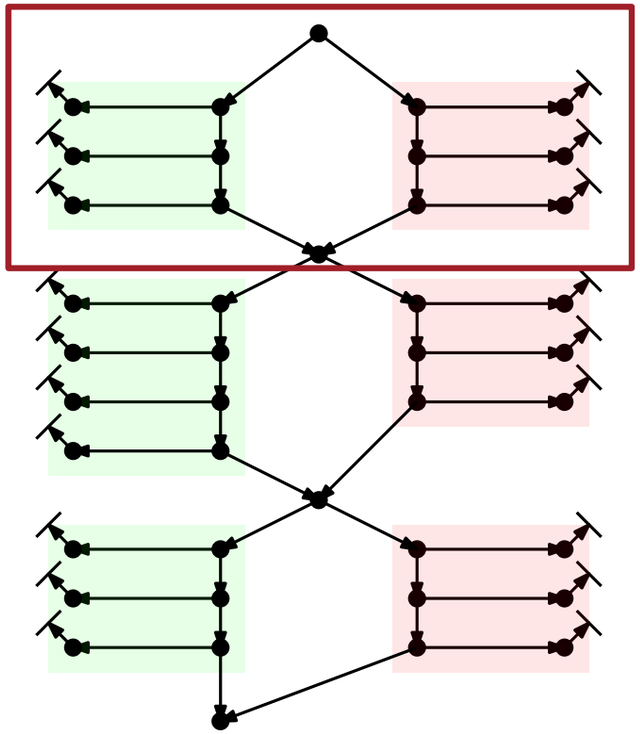
Variable Gadget



Evaluation Phase

PSPACE-Hardness of Robust Connection Game

- QBF-Game



Assignment Phase

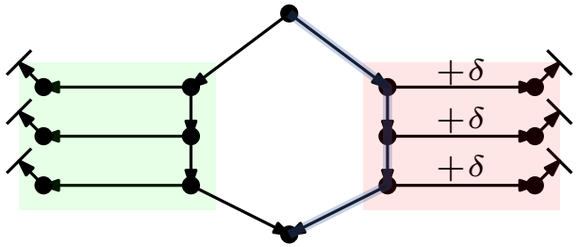
$Q_1x_1Q_2x_2\ldots Q_nx_n\varphi$ with $Q_i \in$

an

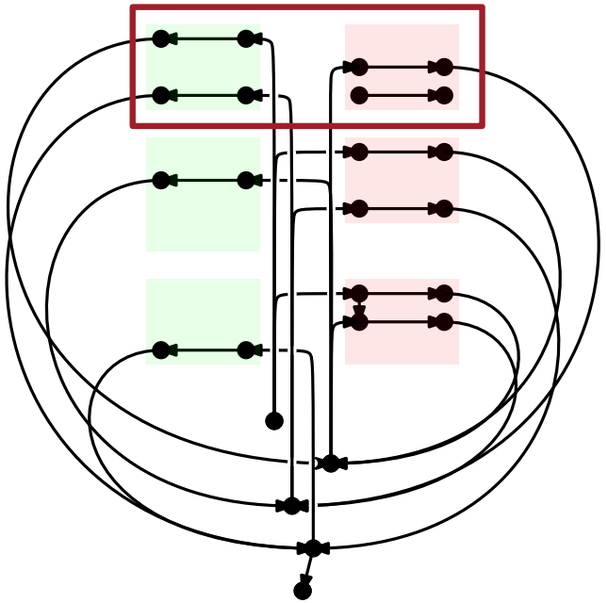
s if

∃-p

rate



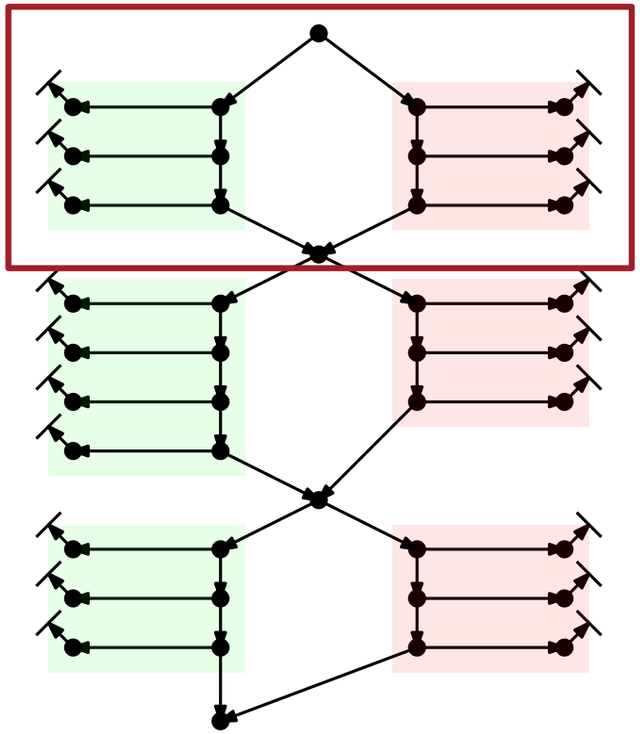
Variable Gadget



Evaluation Phase

PSPACE-Hardness of Robust Connection Game

- QBF-Game



Assignment Phase

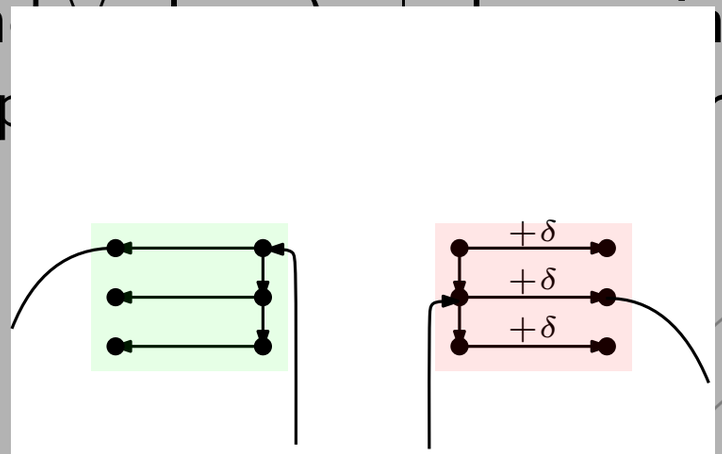
$Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$ with $Q_i \in$

an

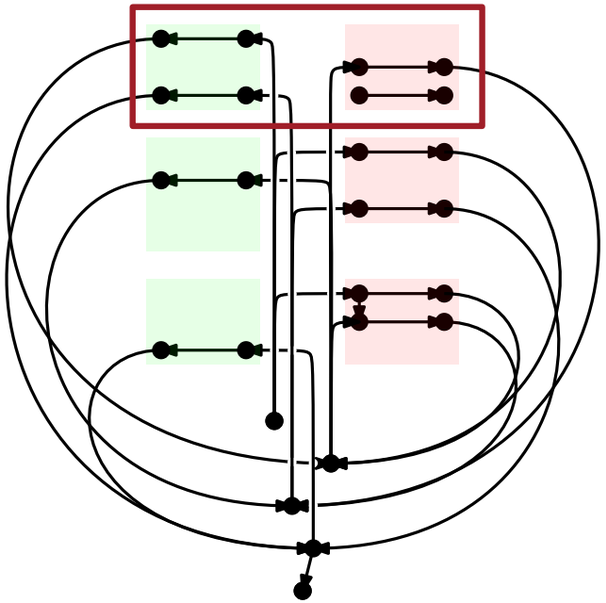
s if

∃-p

rate



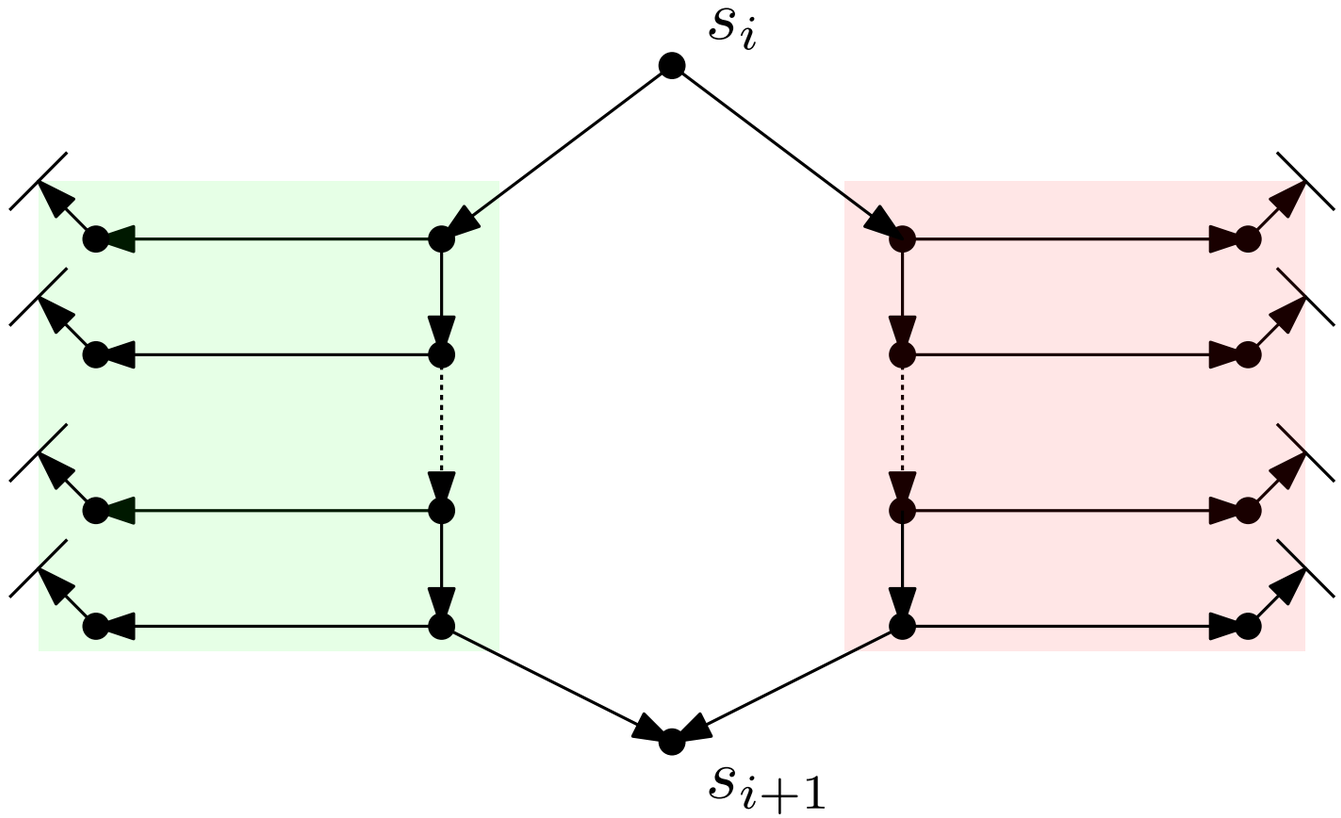
Variable Gadget



Evaluation Phase

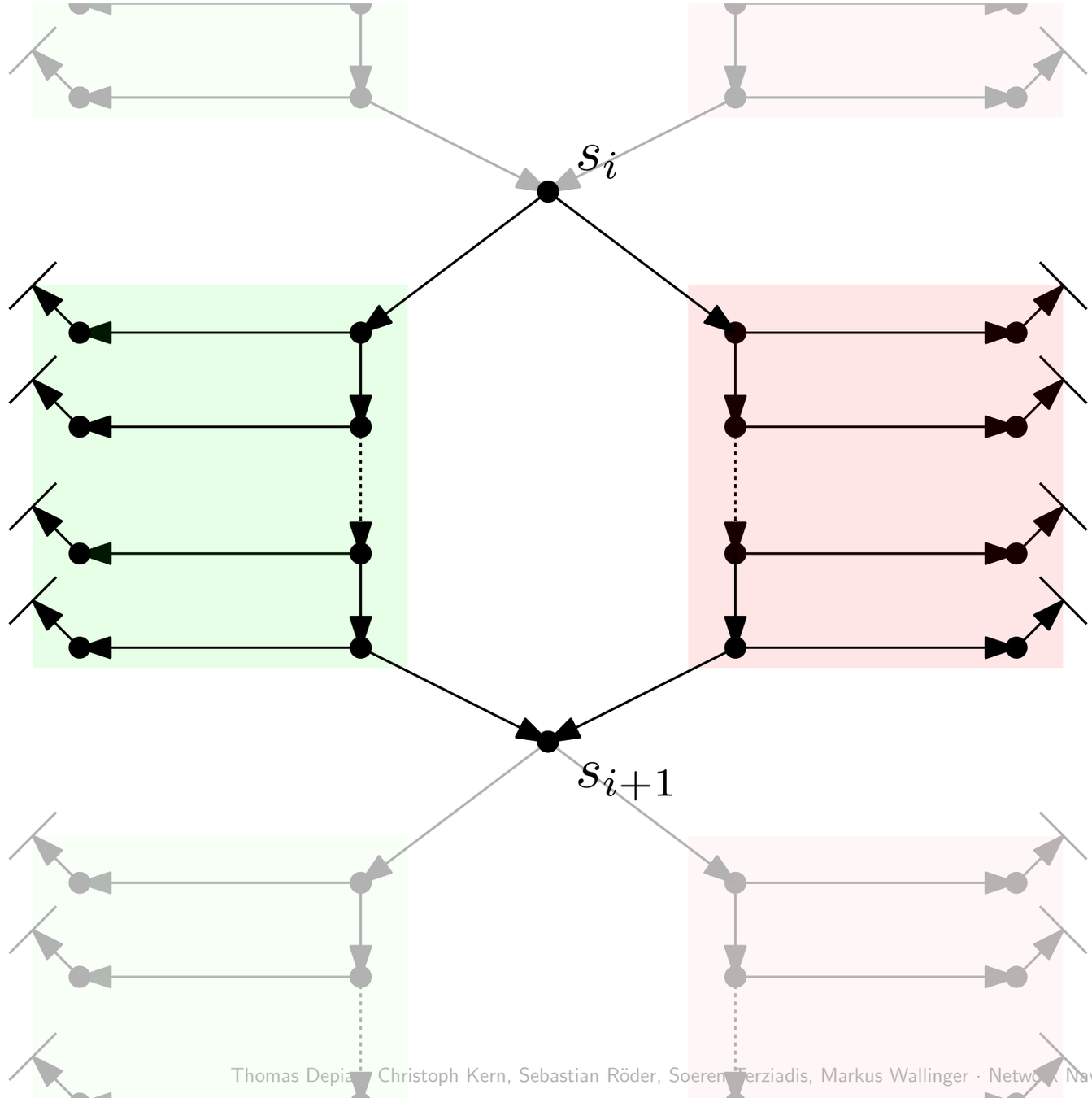
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



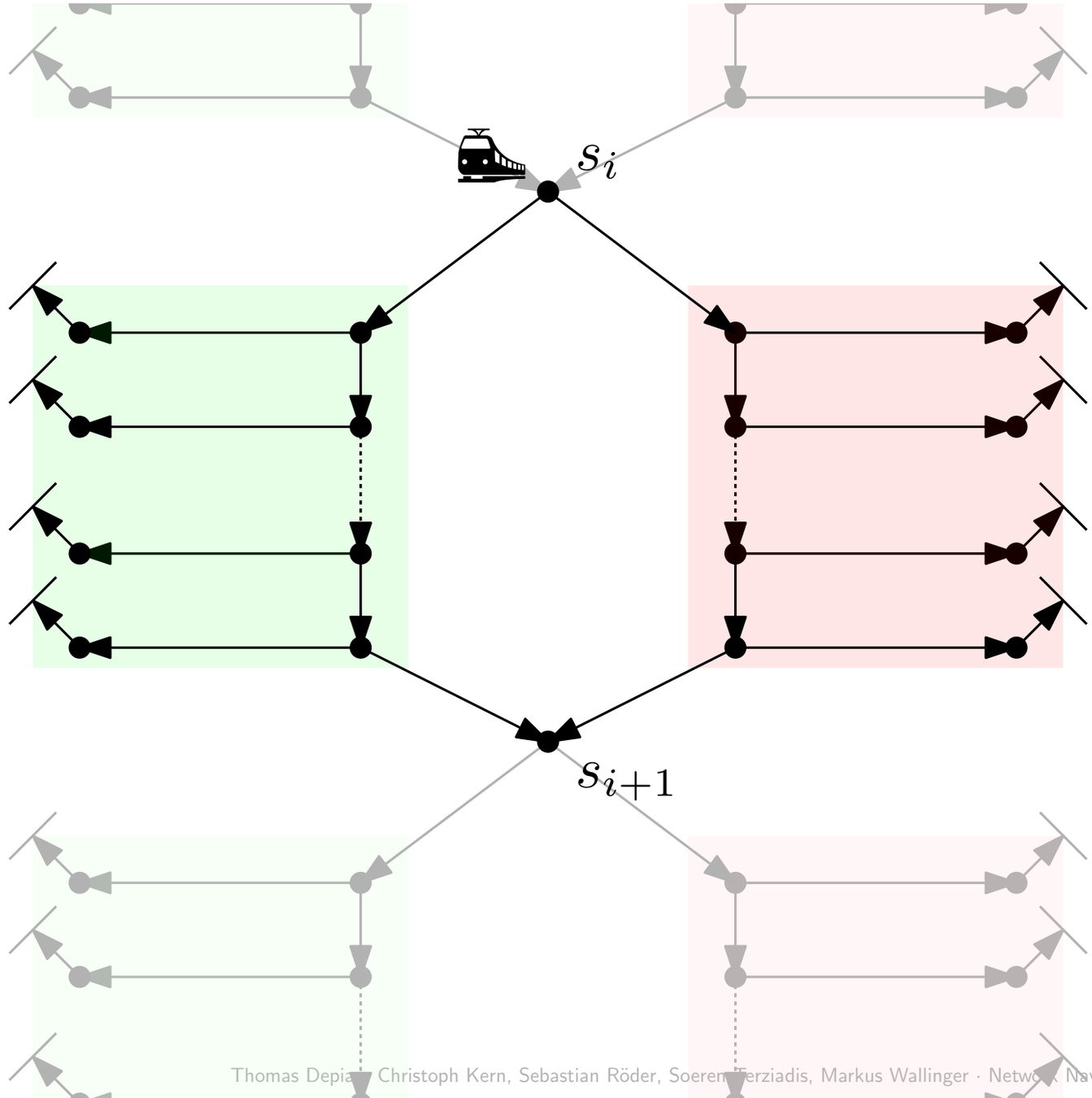
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



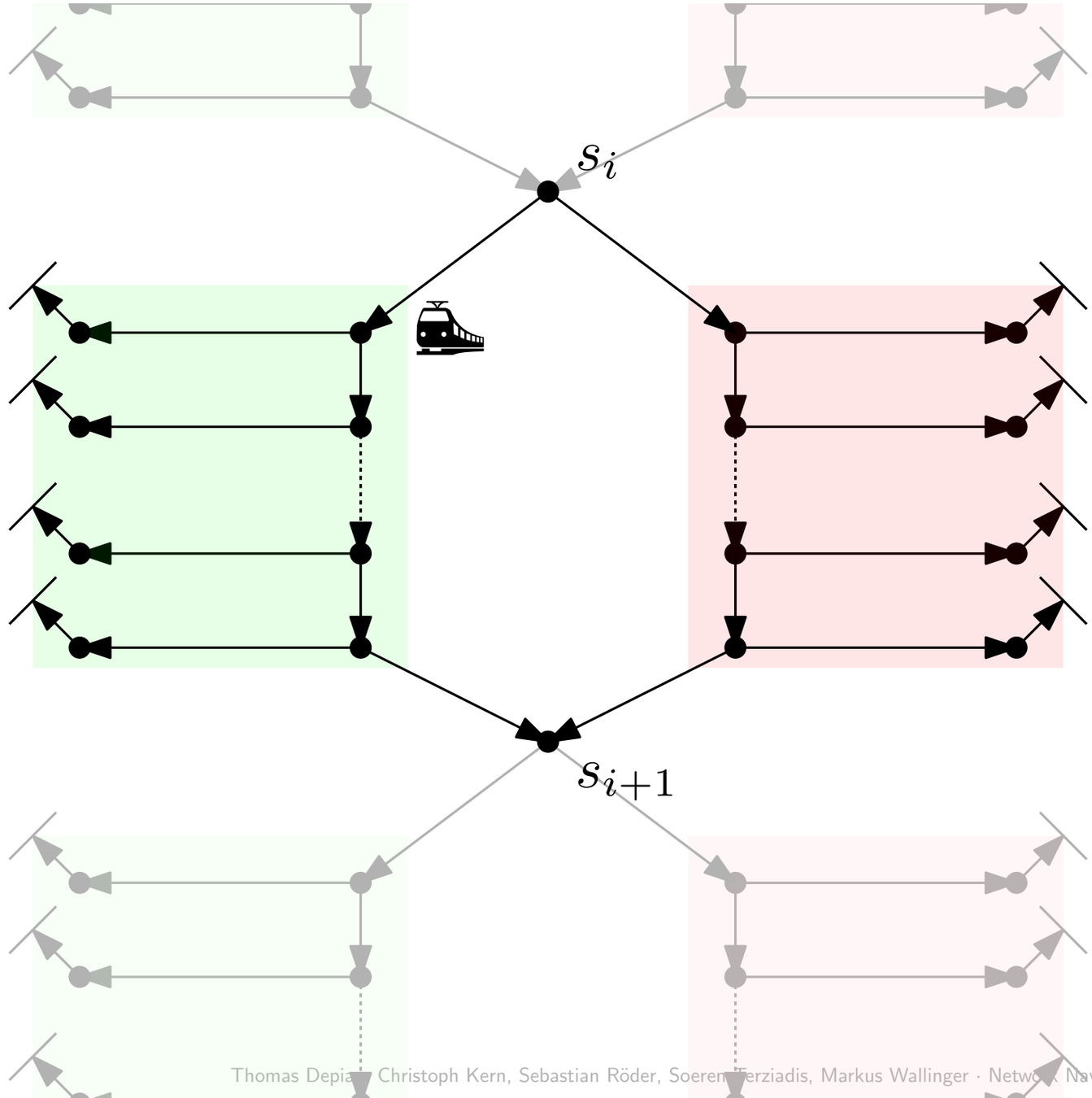
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



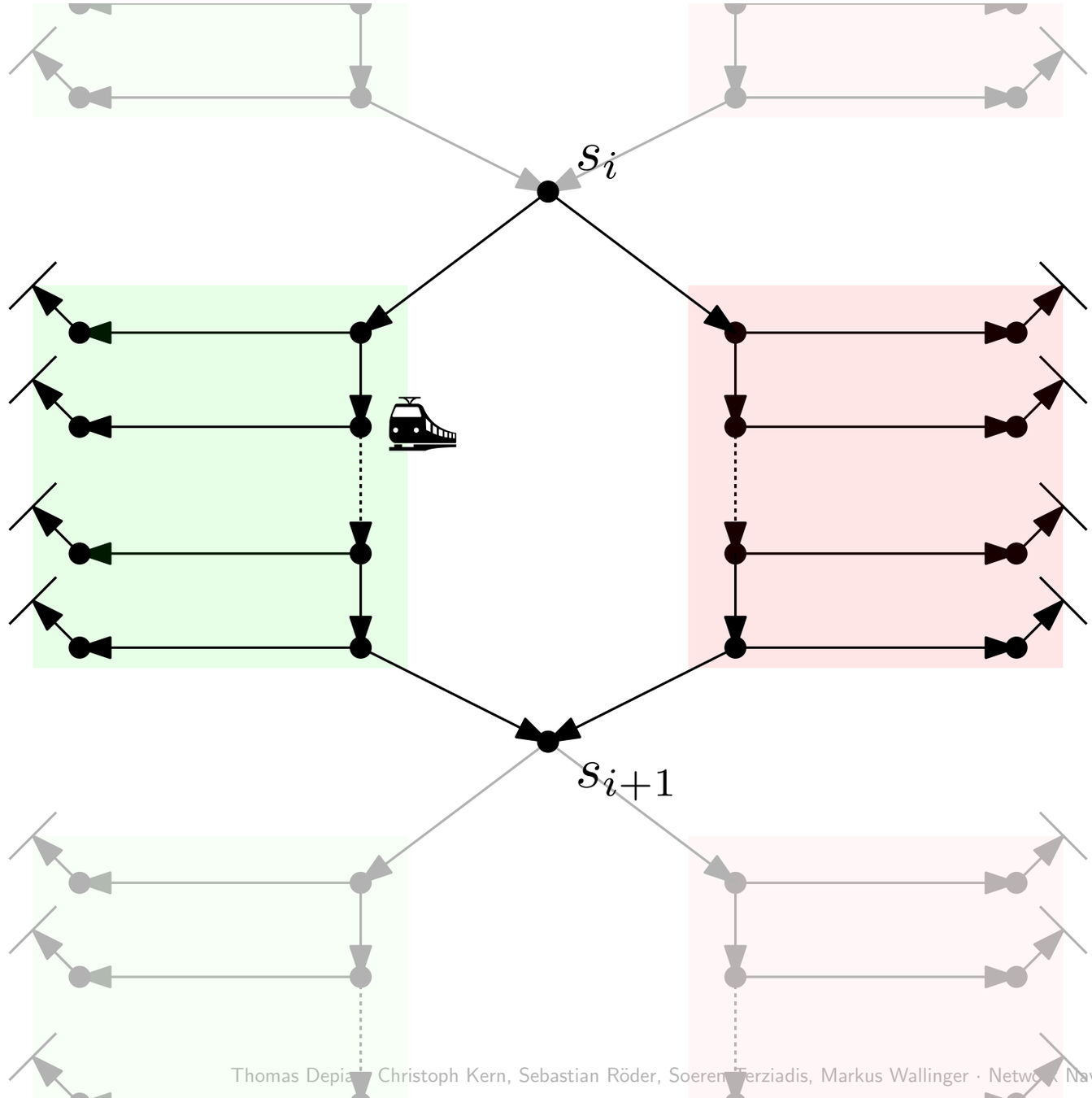
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



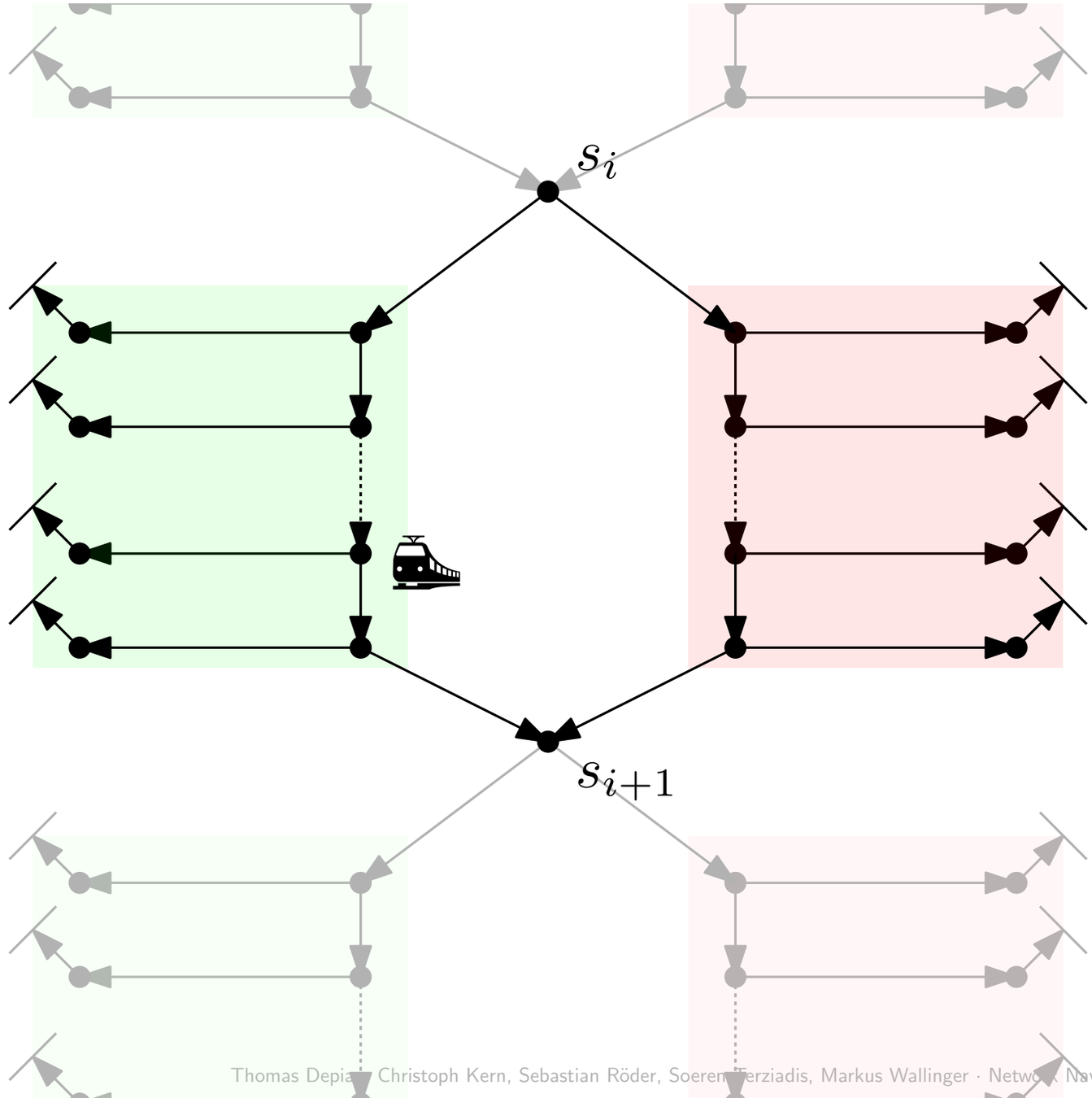
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



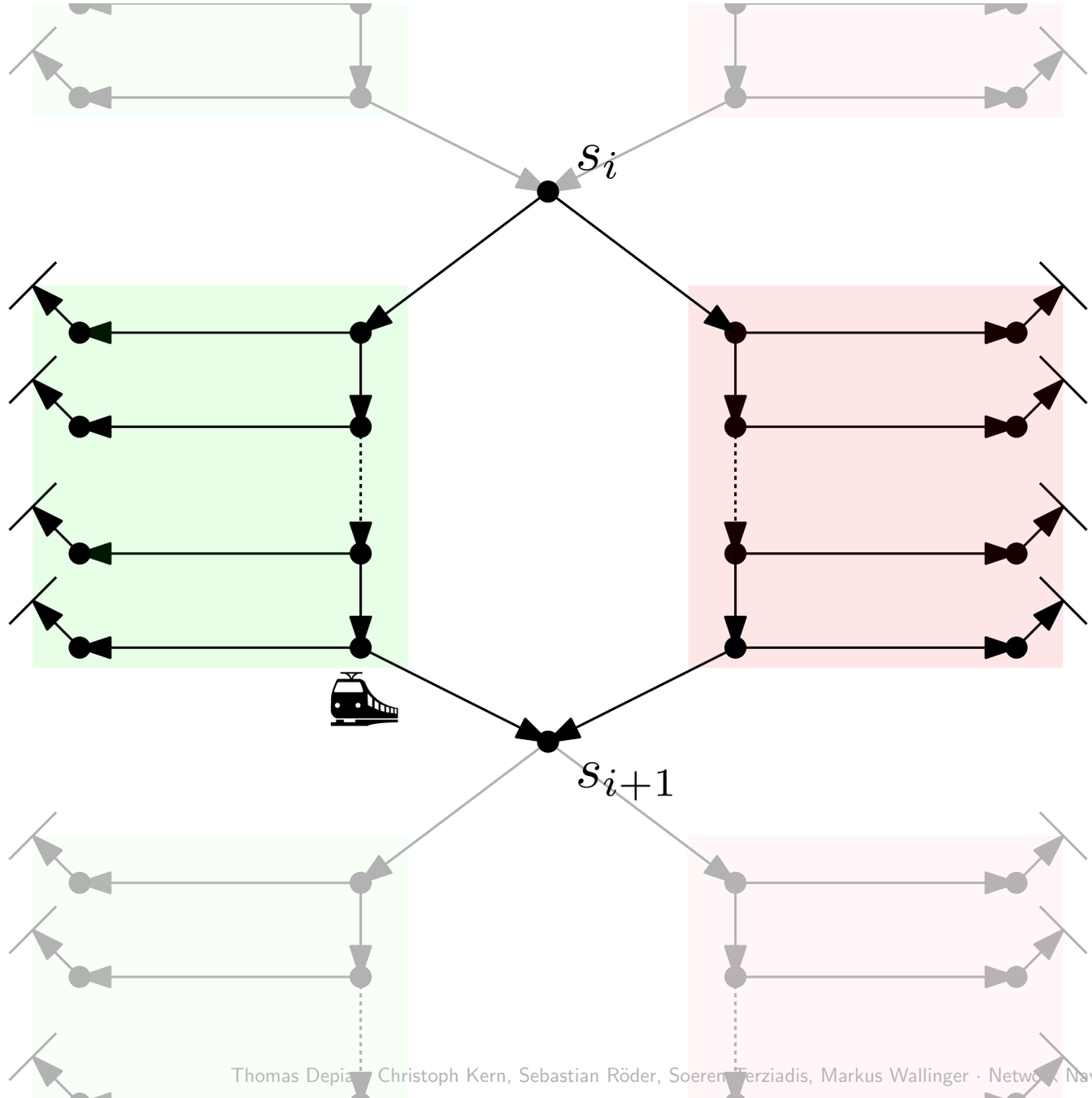
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



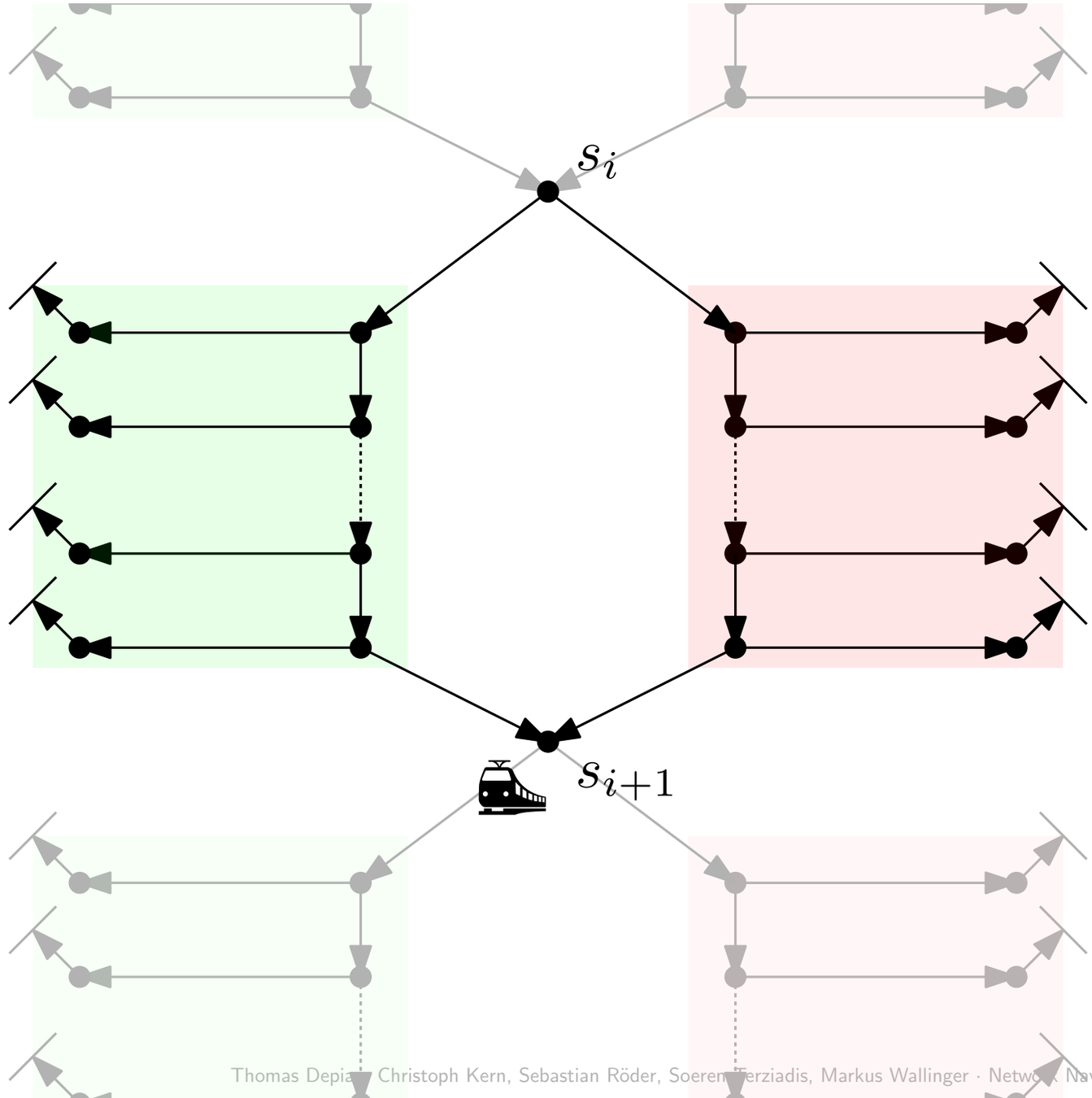
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



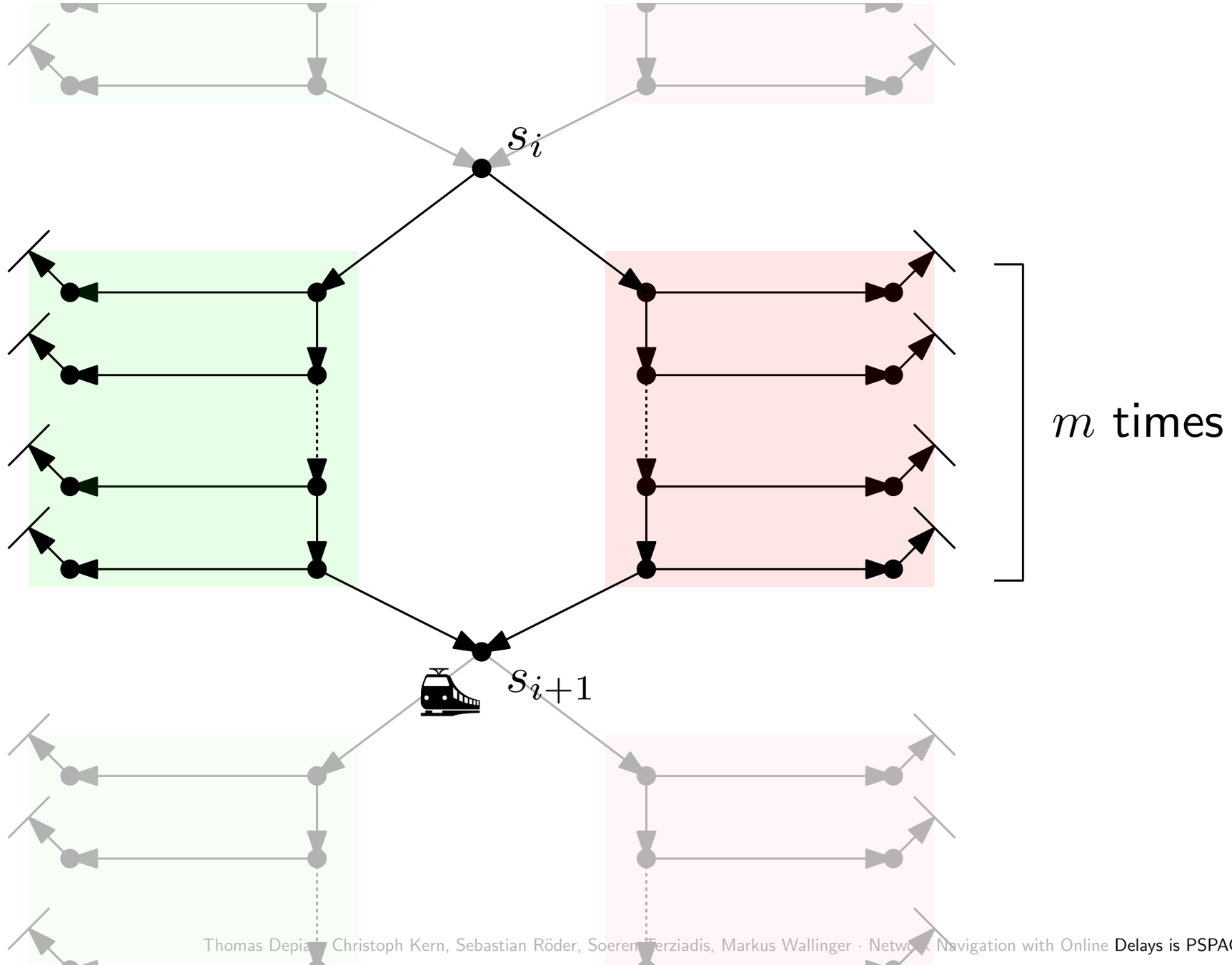
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



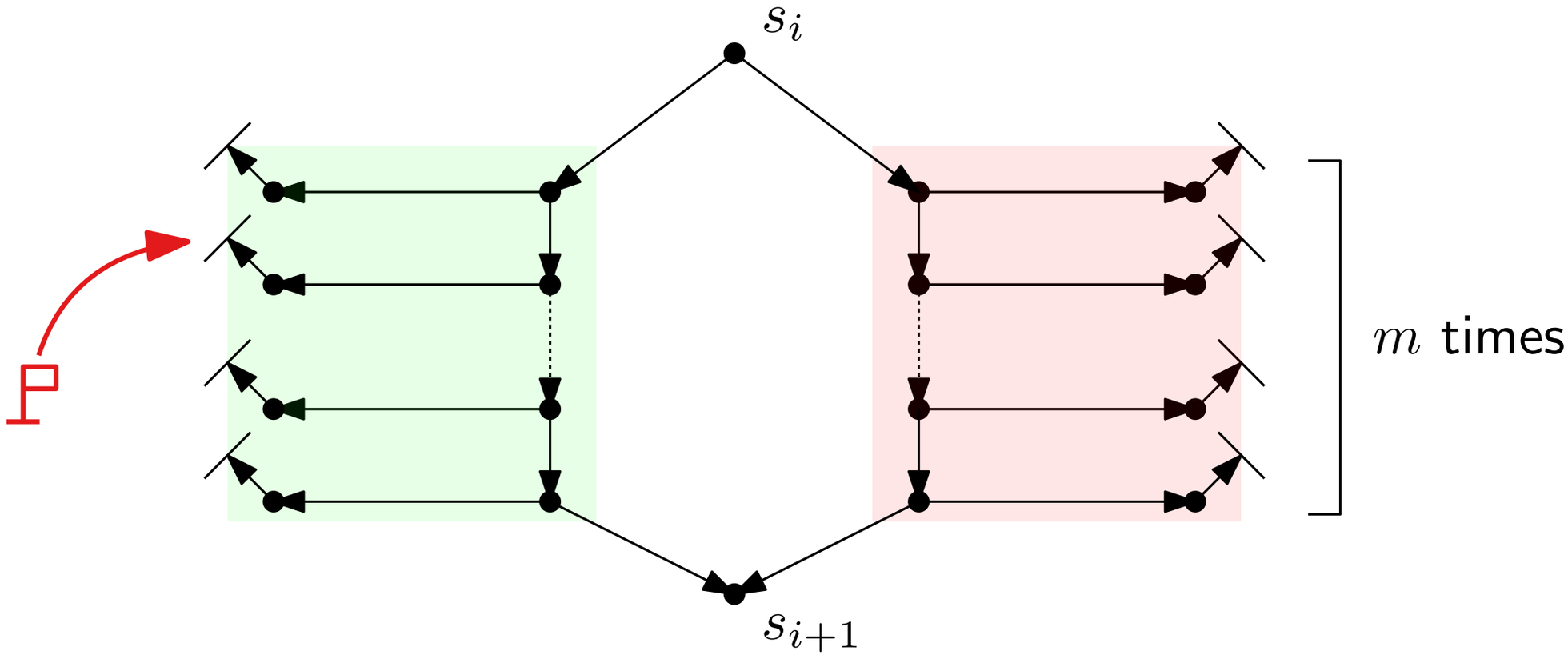
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



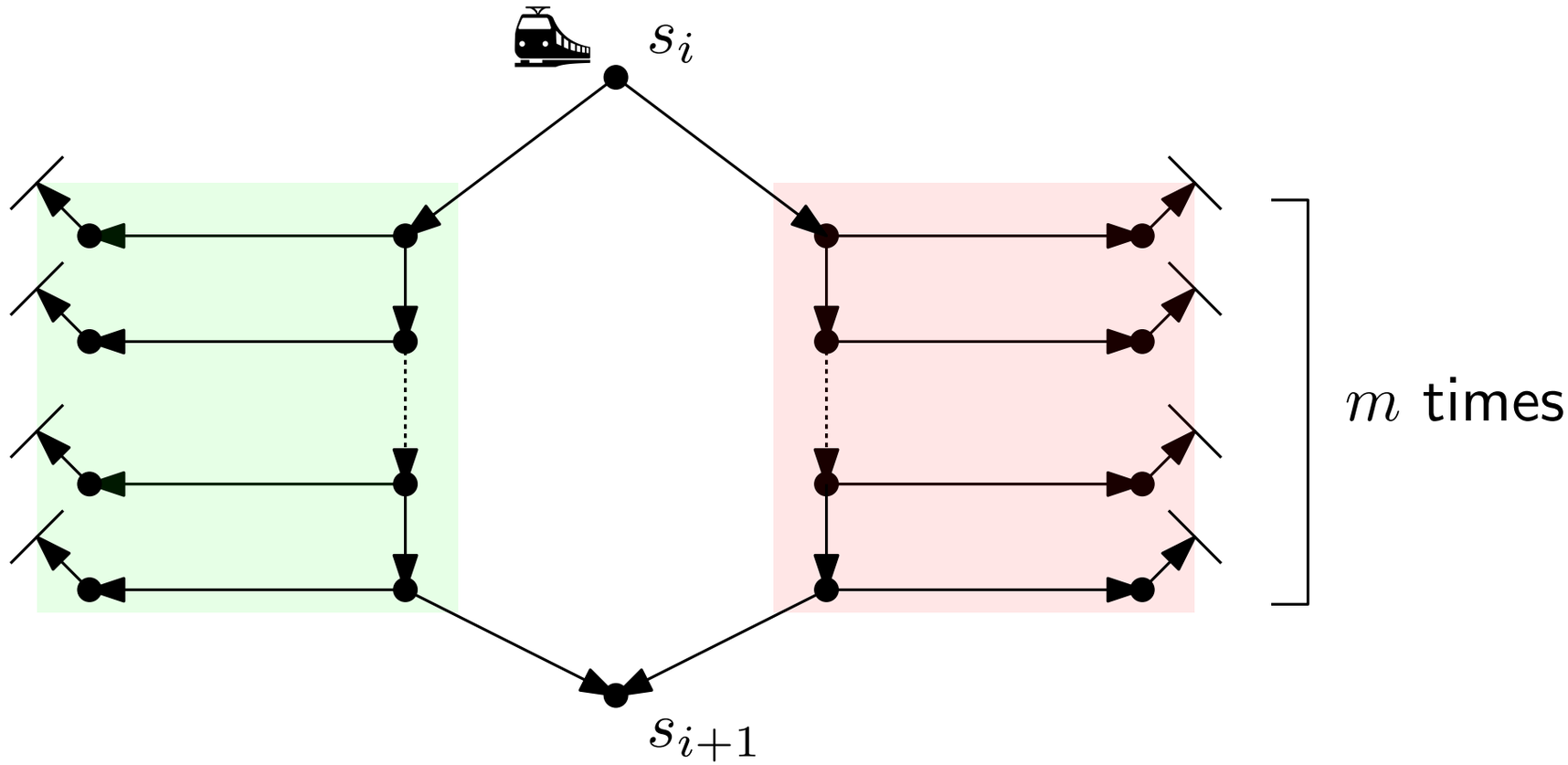
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



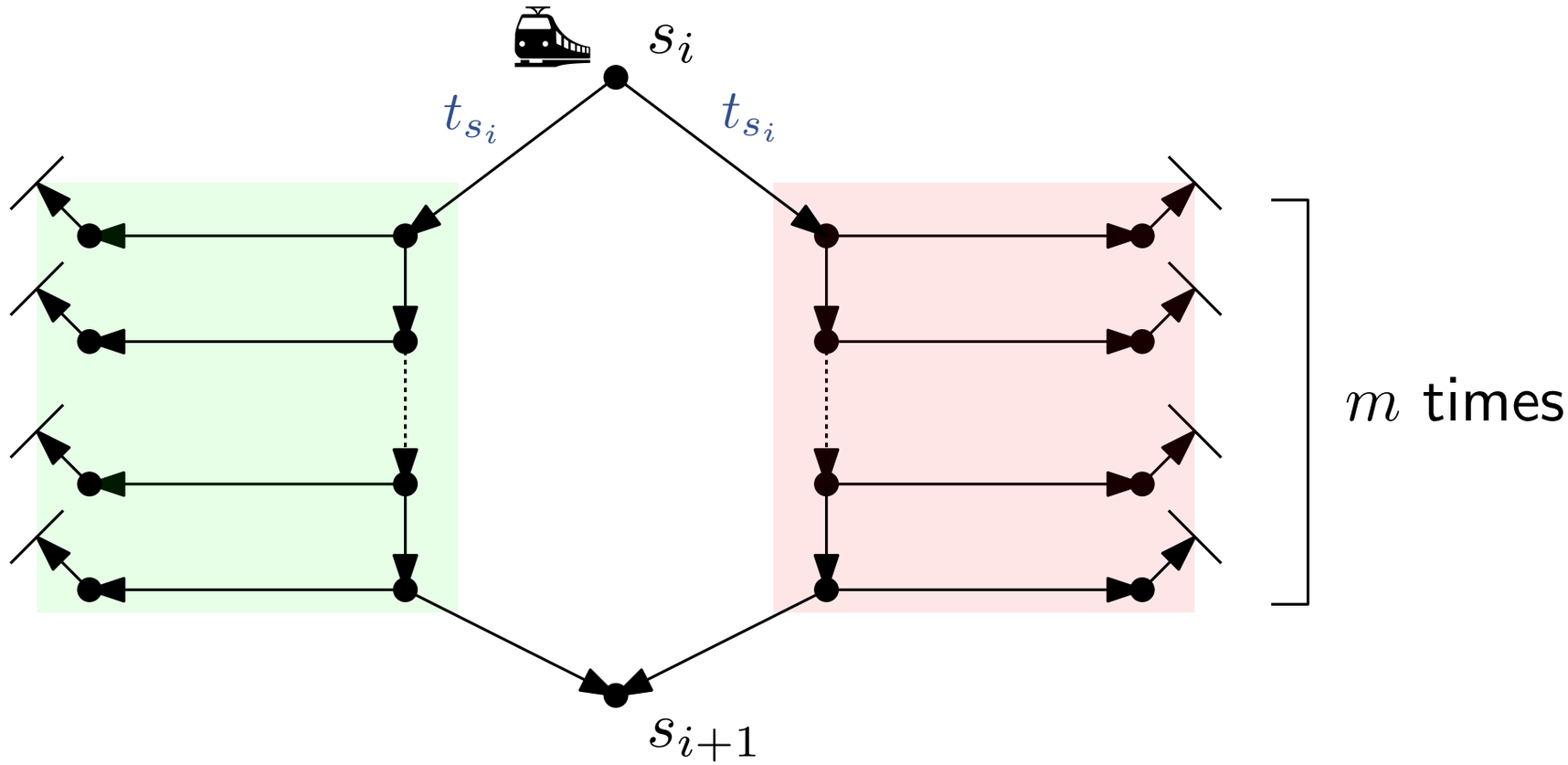
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



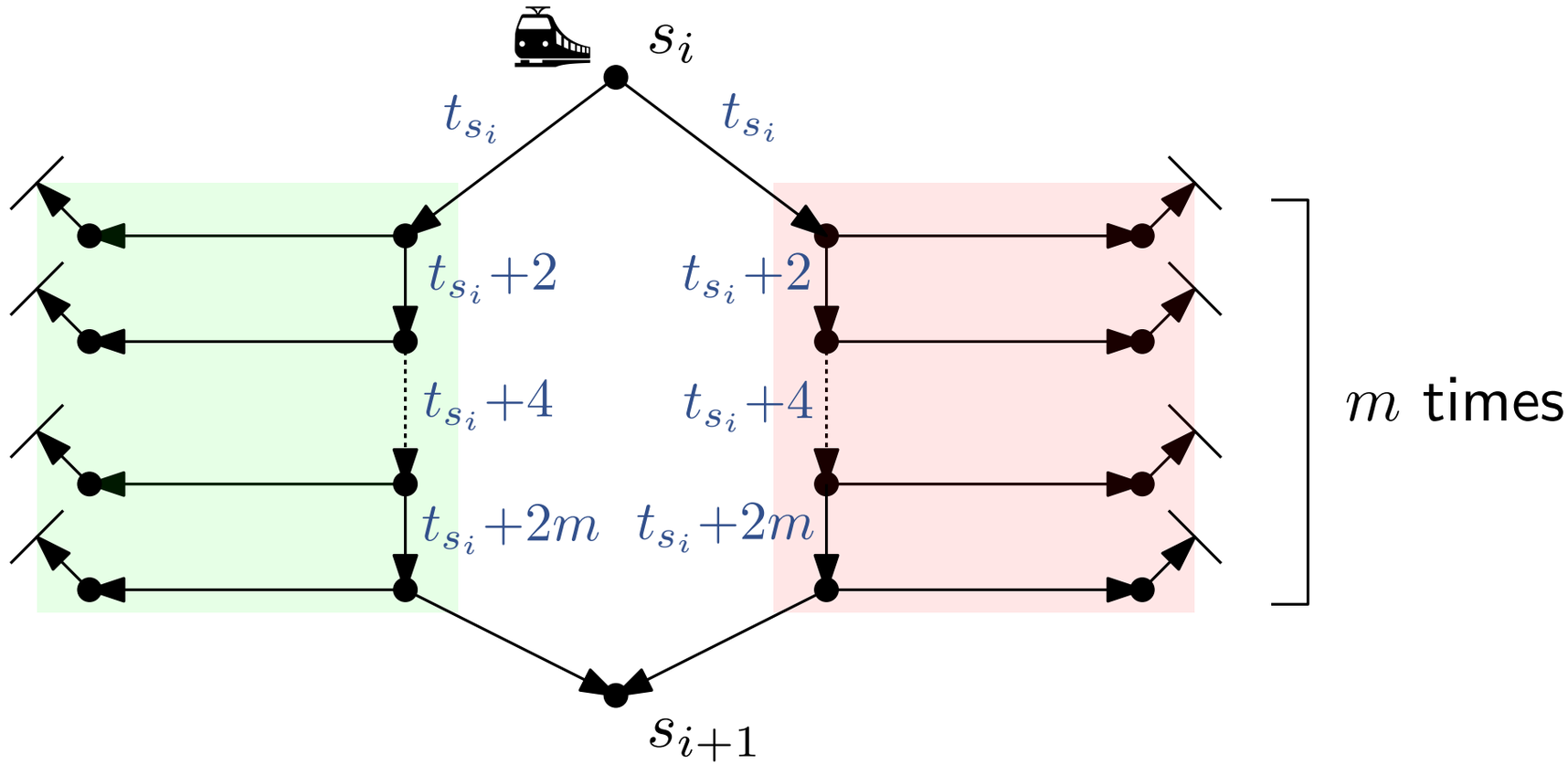
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



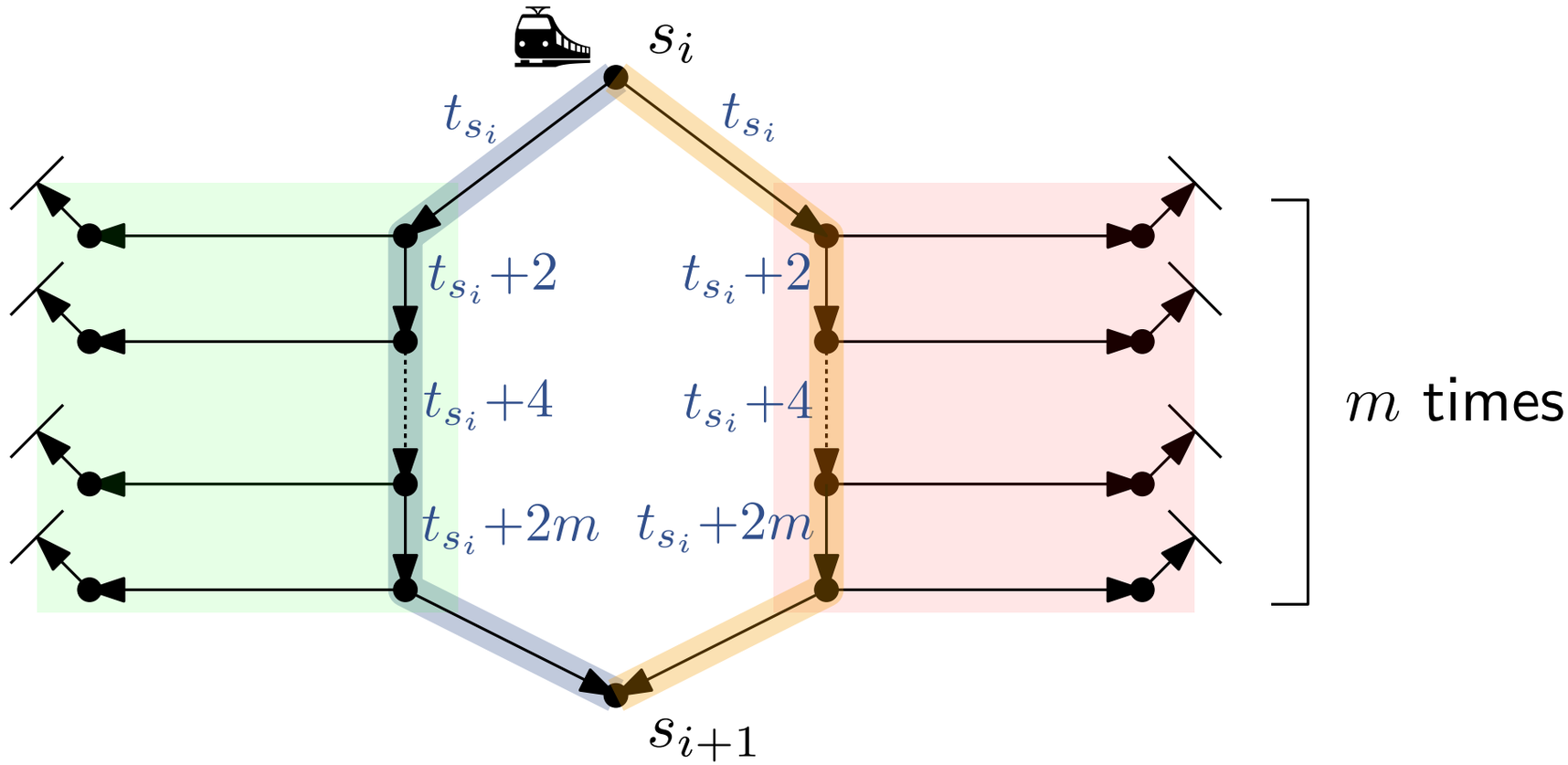
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



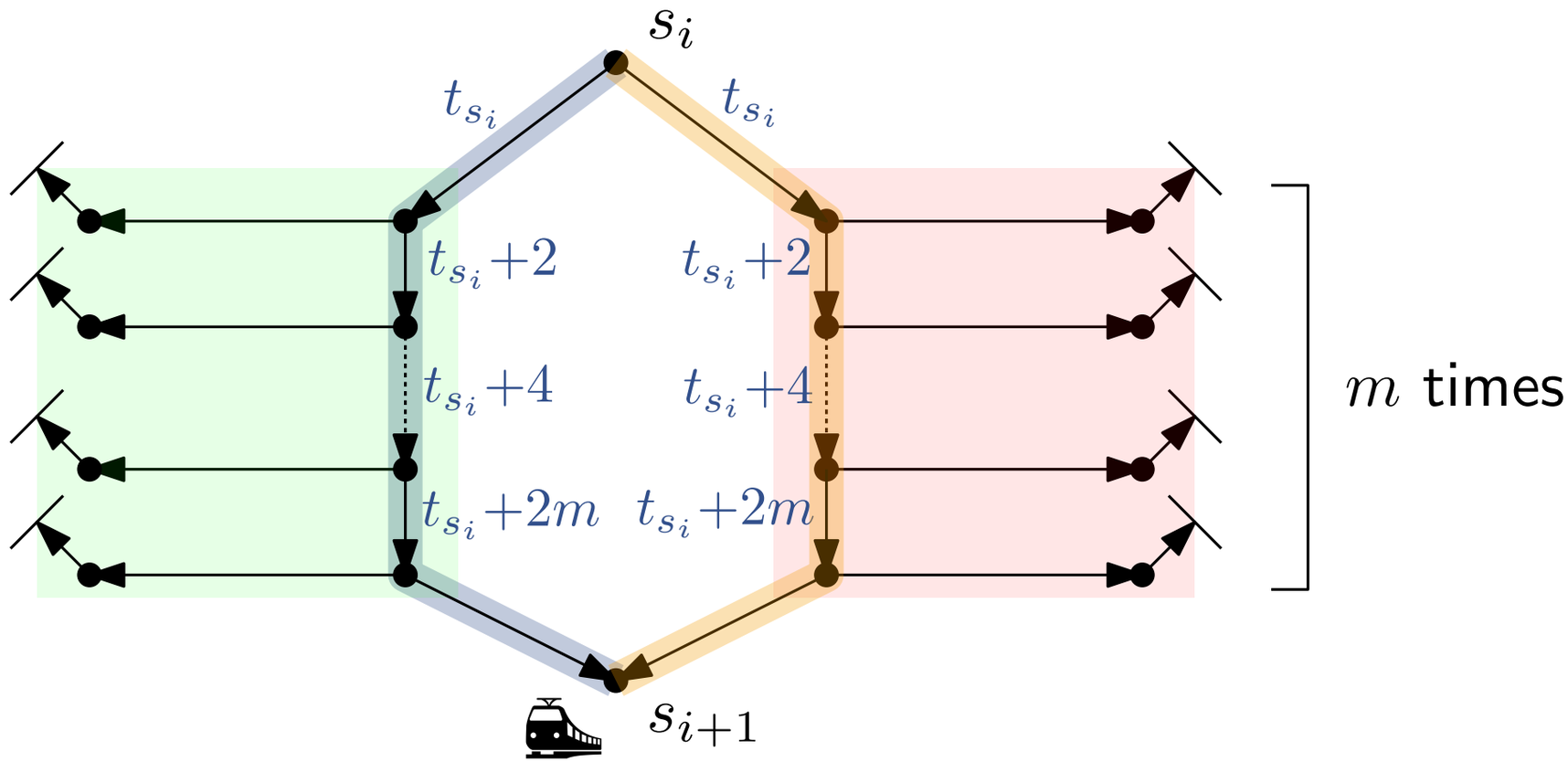
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



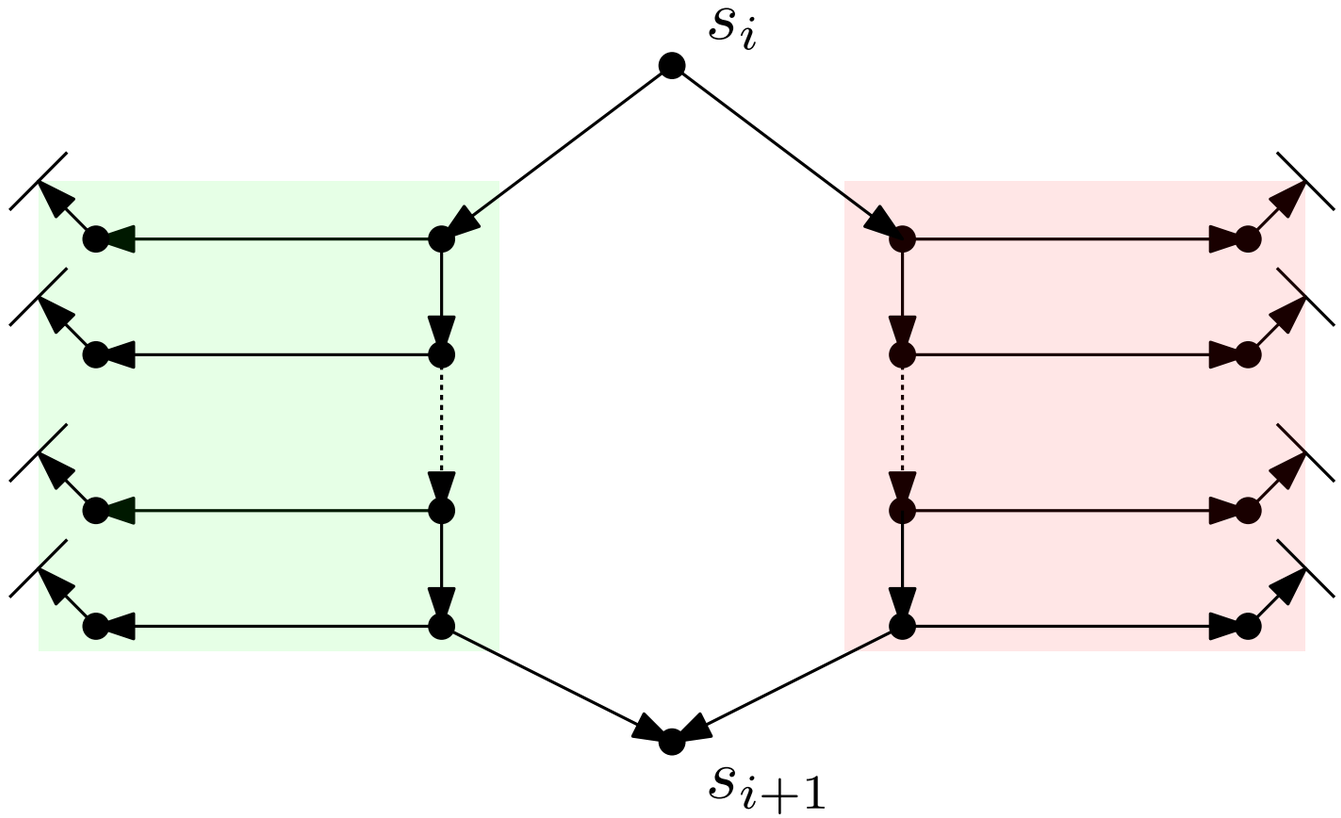
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



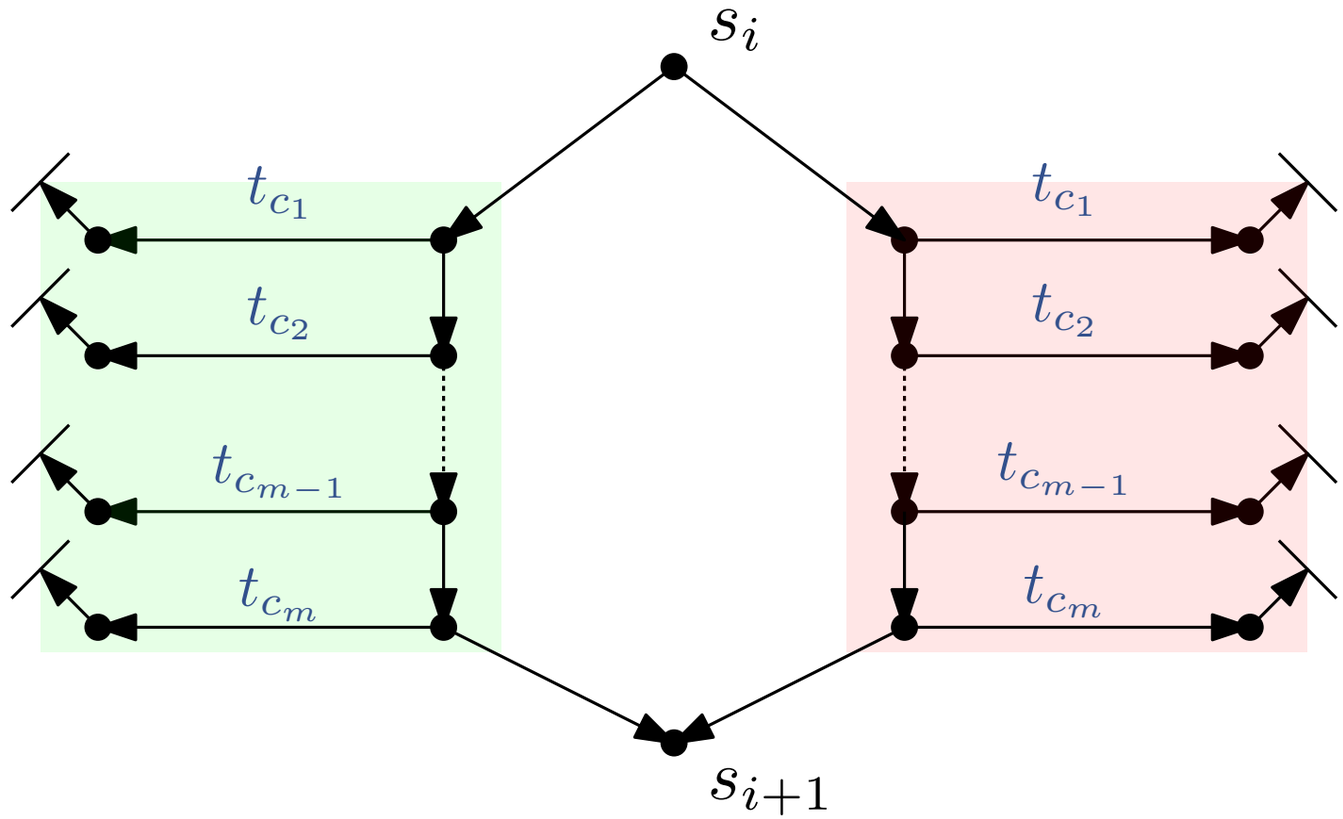
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



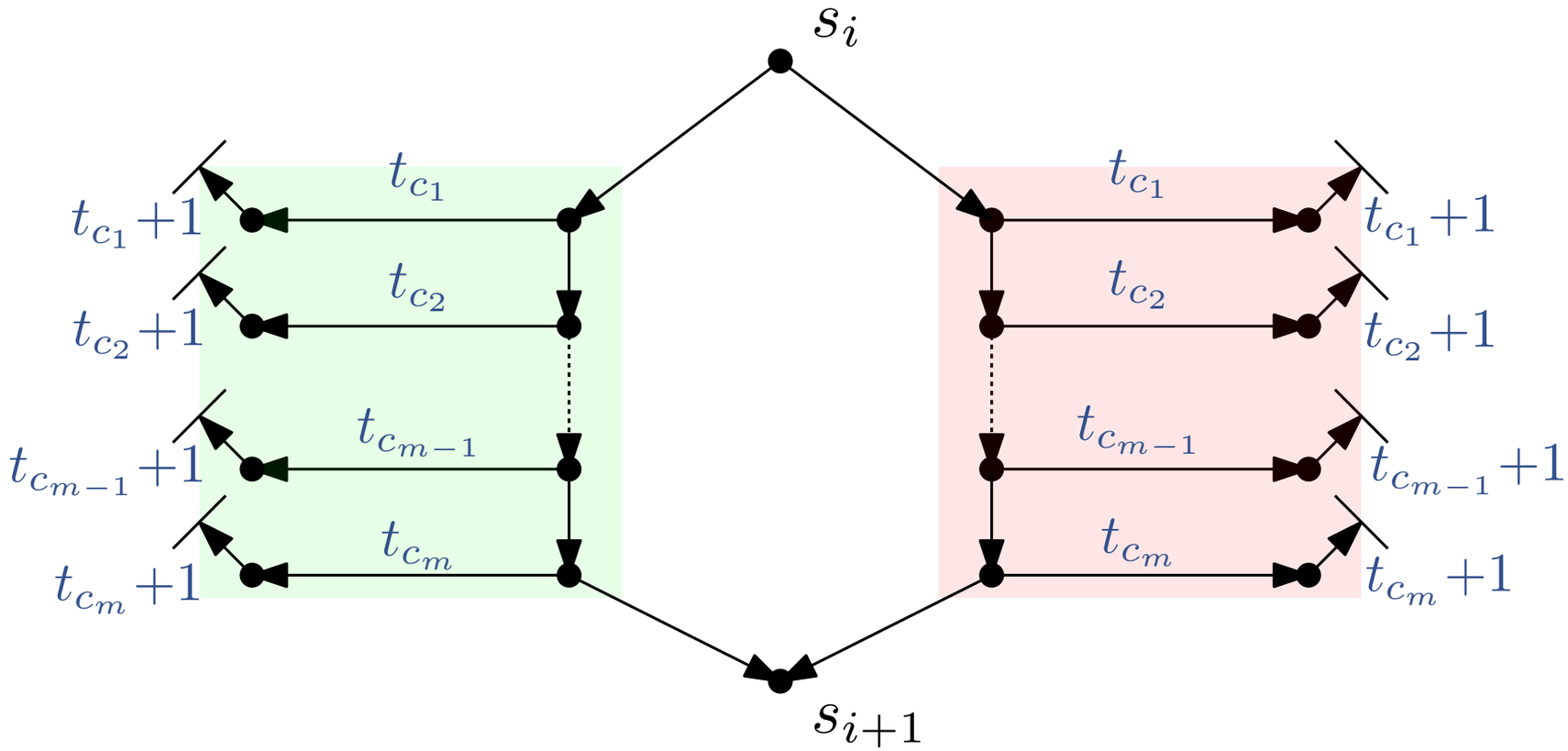
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



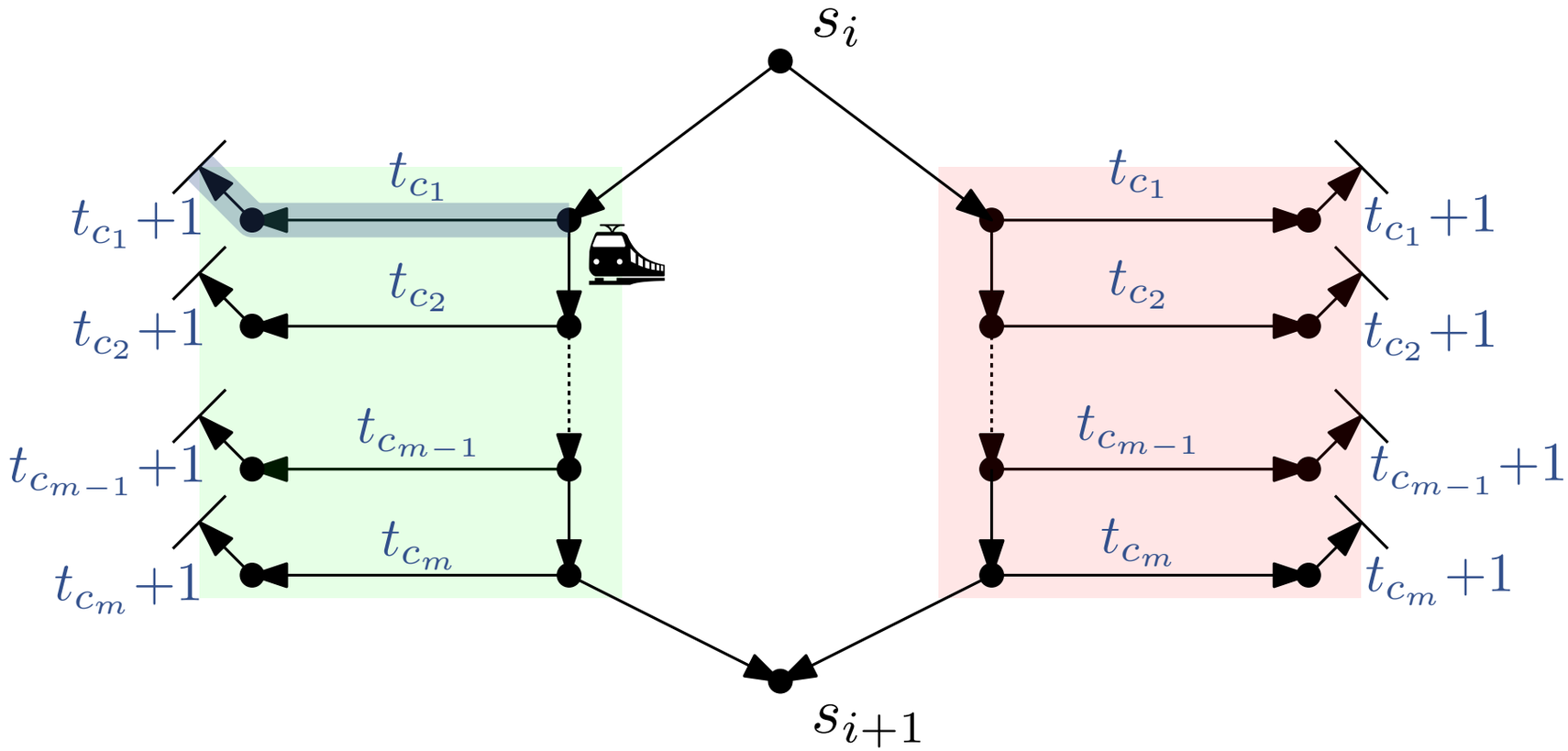
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



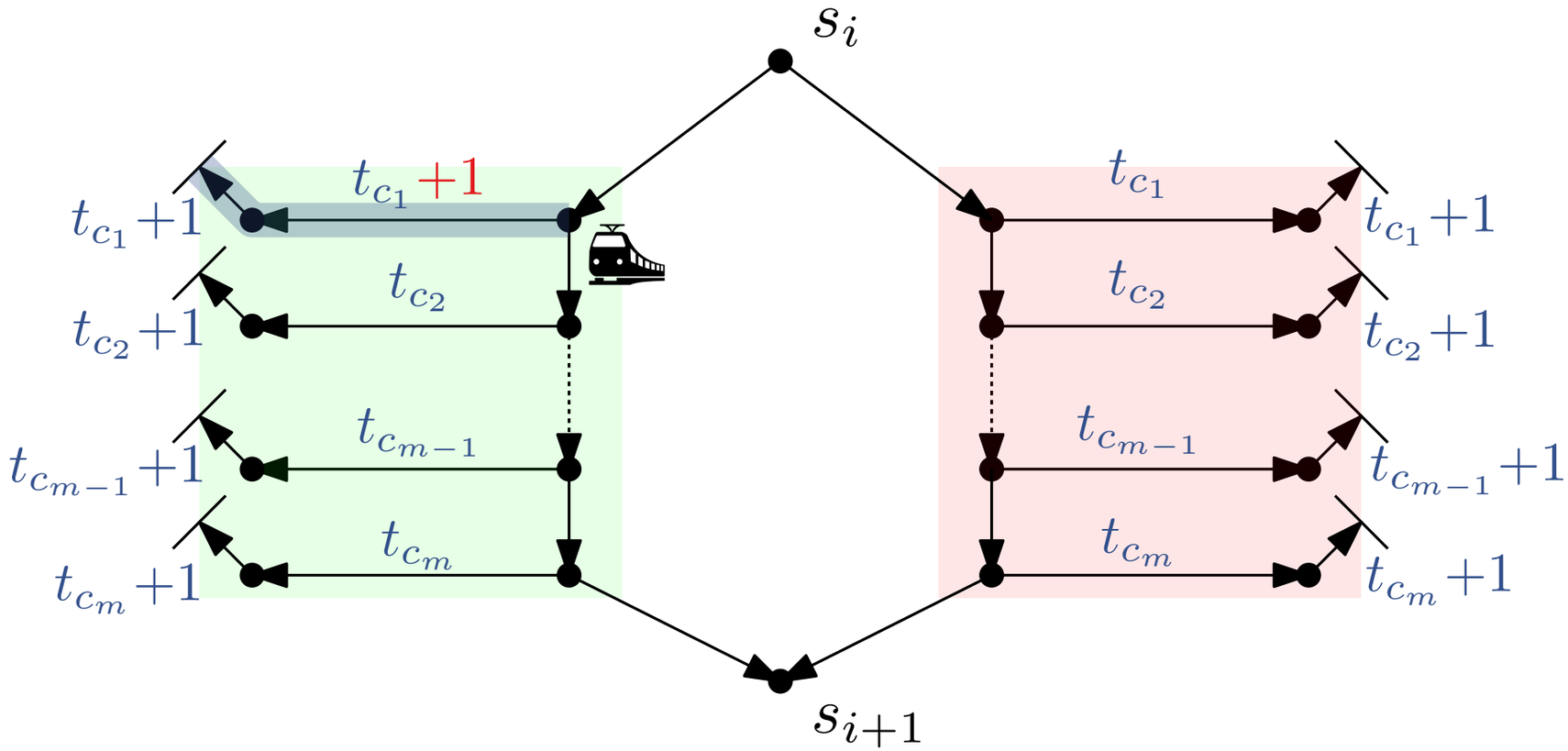
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



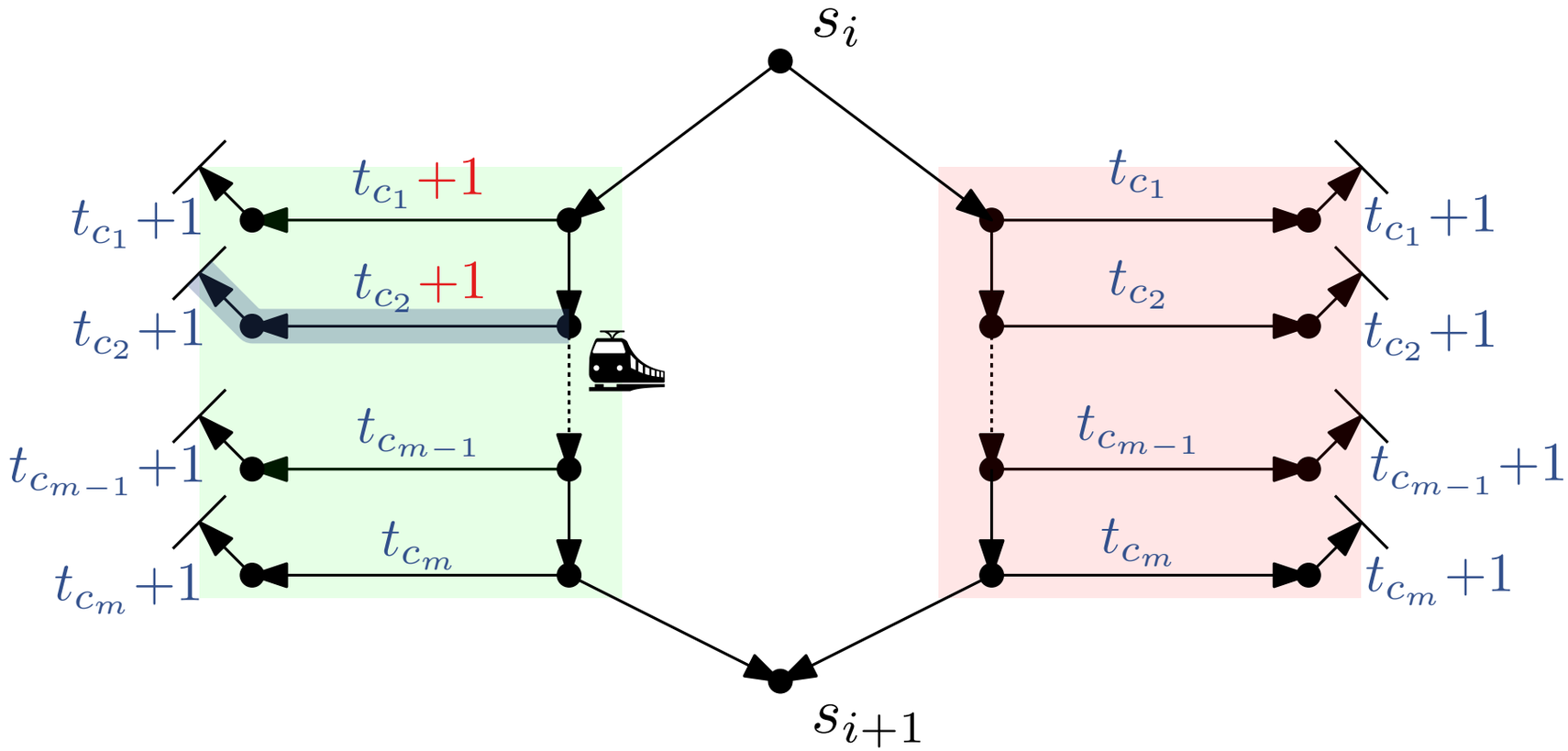
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



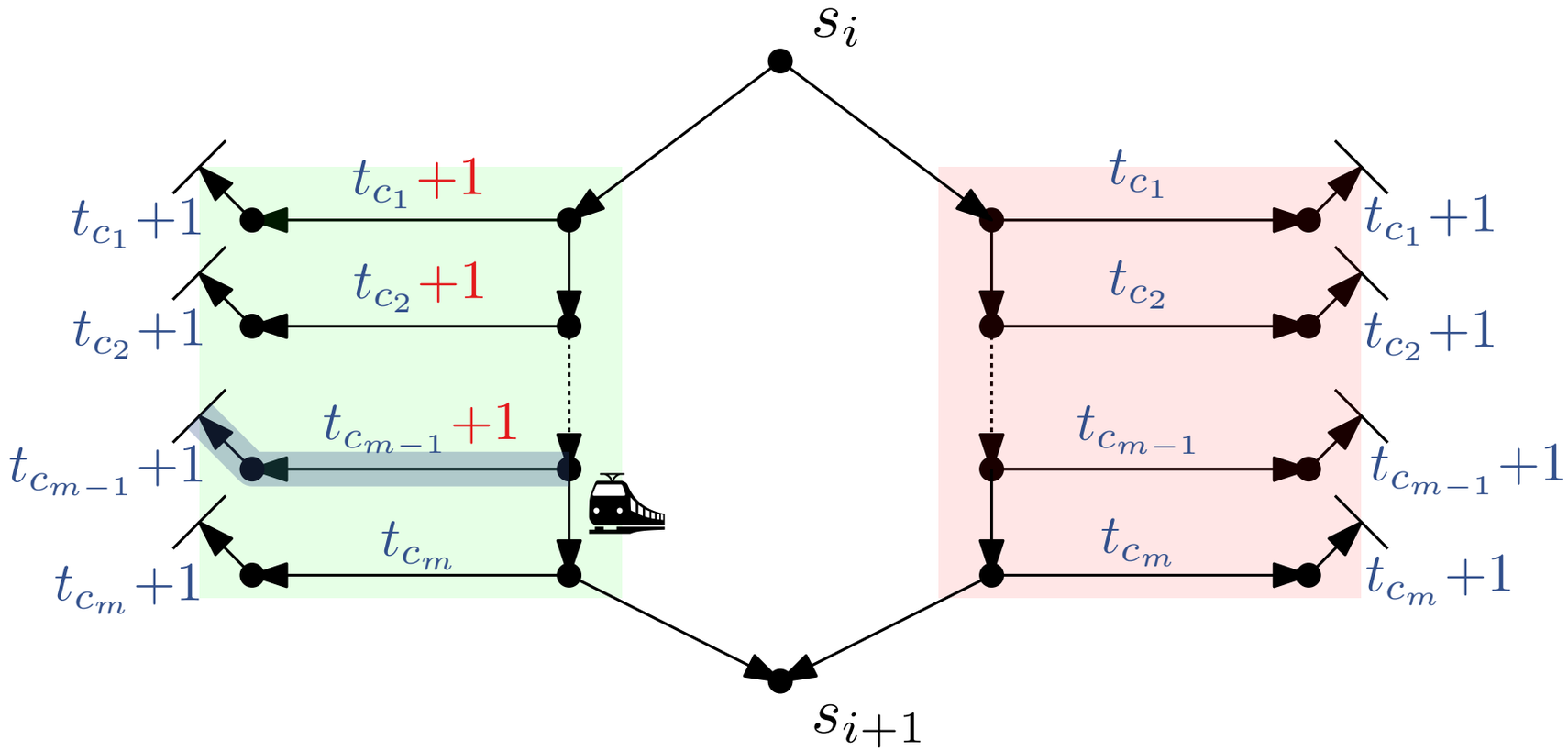
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



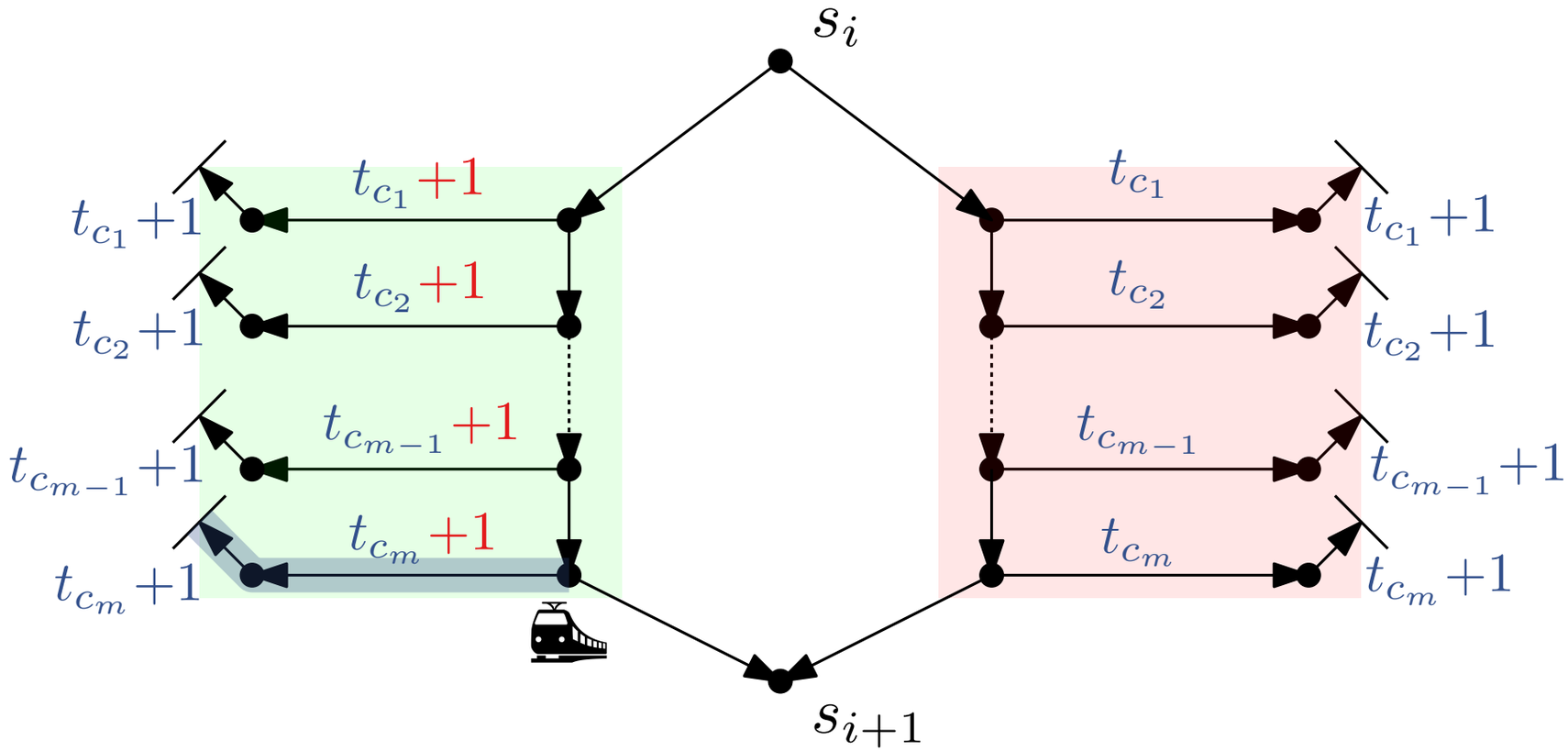
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



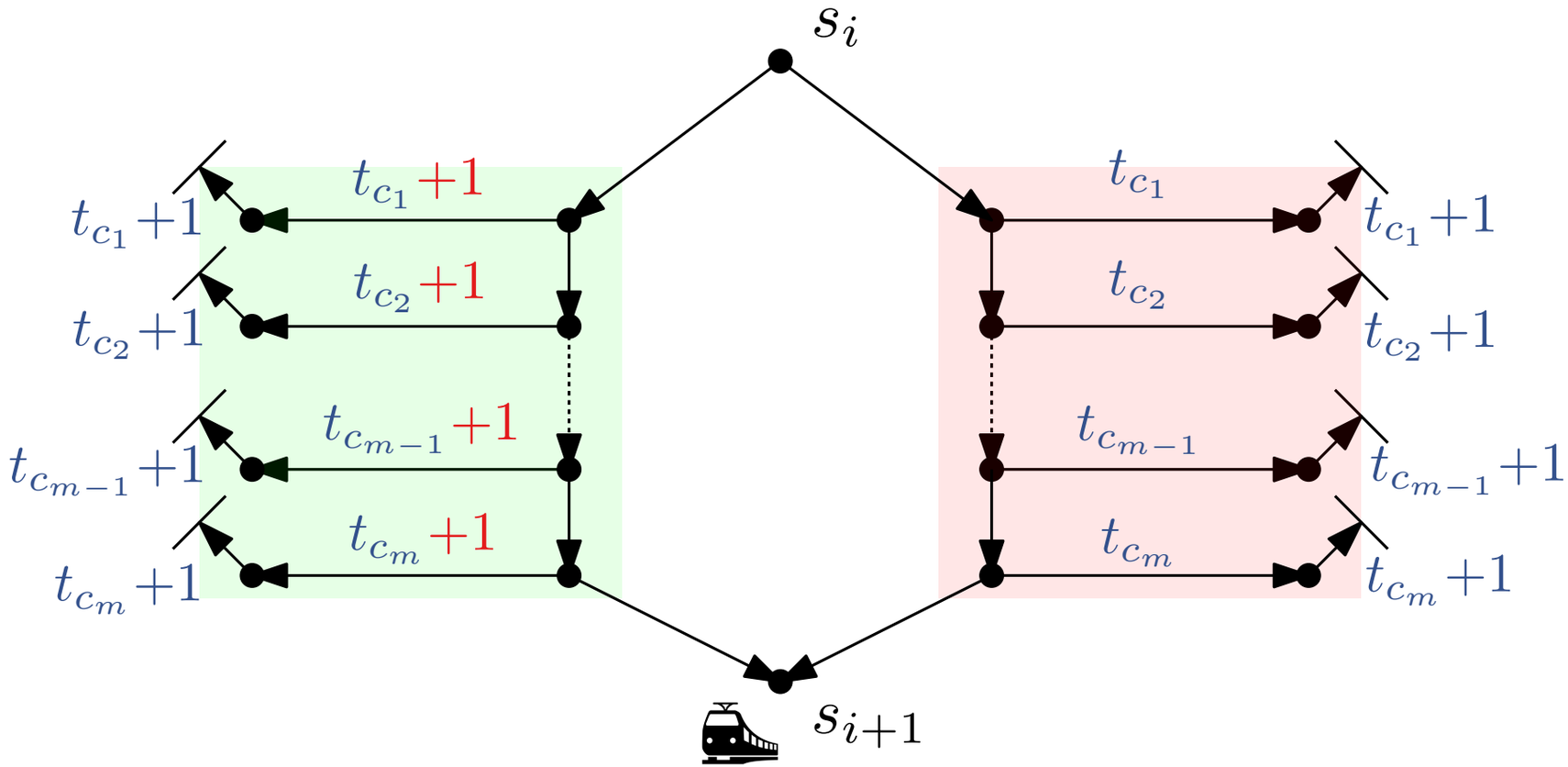
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



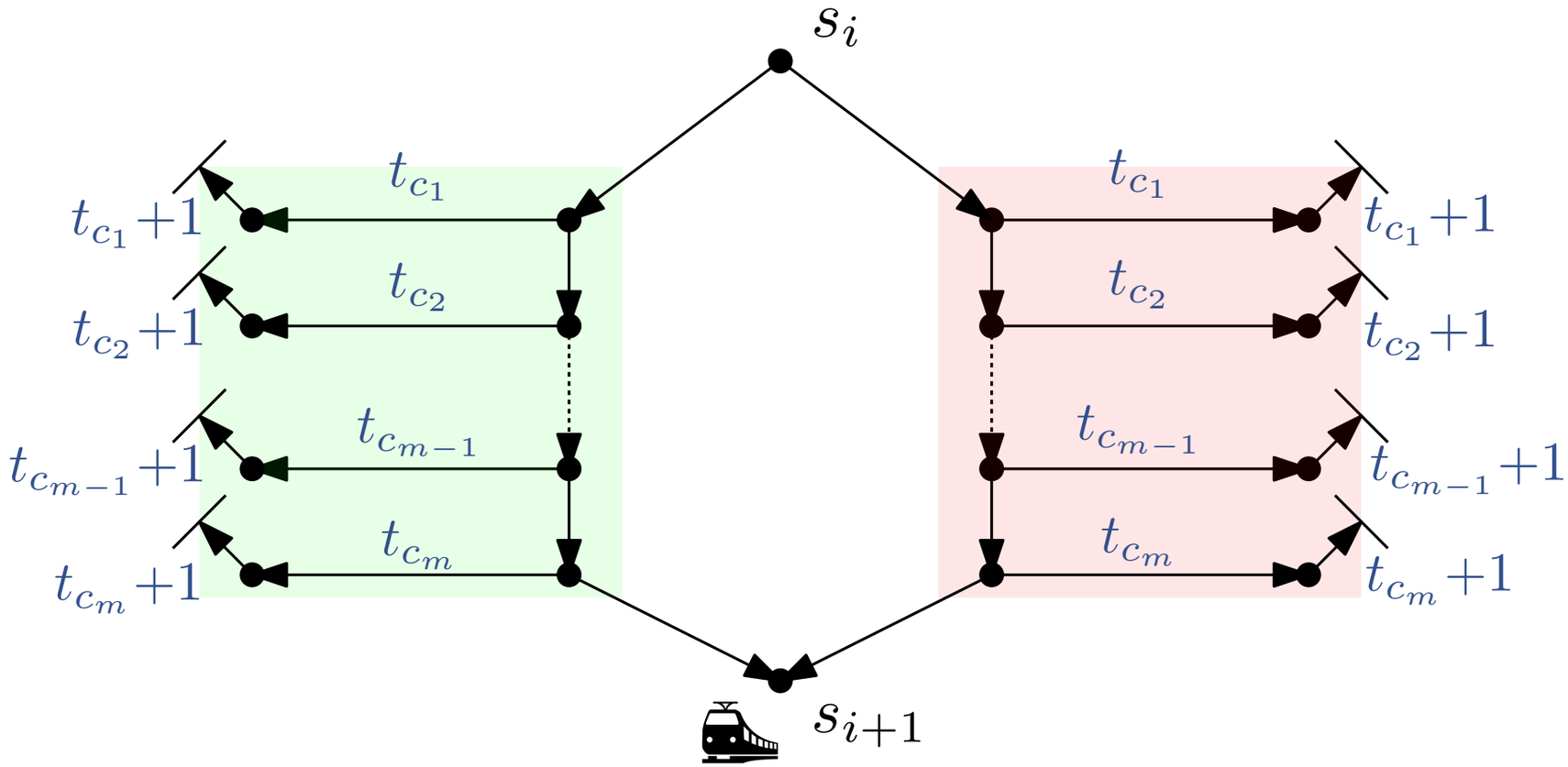
Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$



Existential Variable Gadget

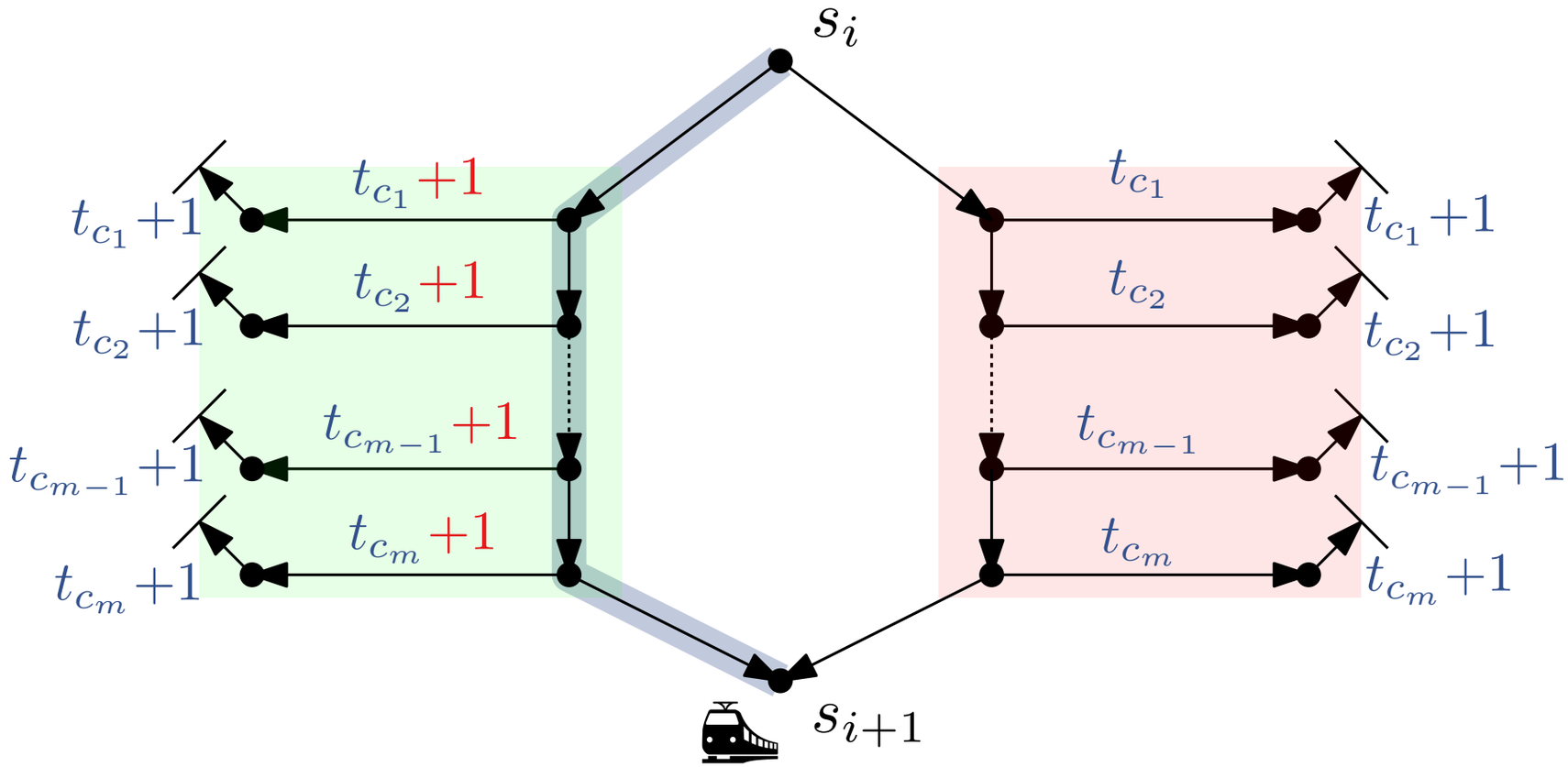
- $Q_1 x_1 = \exists x_i$



- **2 possibilities**:

Existential Variable Gadget

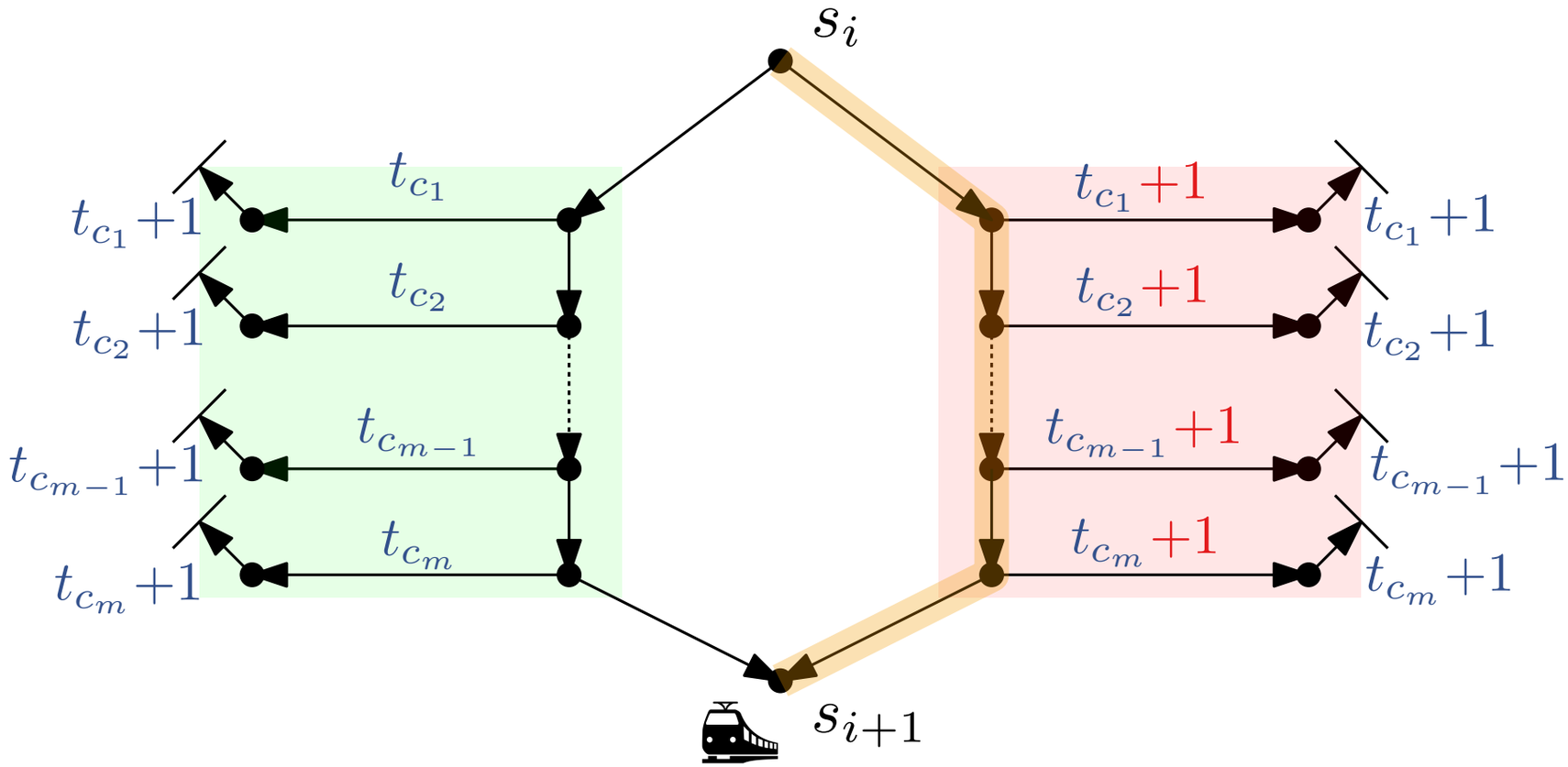
- $Q_1 x_1 = \exists x_i$



- **2 possibilities**:
  - m positive edges delayed

Existential Variable Gadget

- $Q_1 x_1 = \exists x_i$

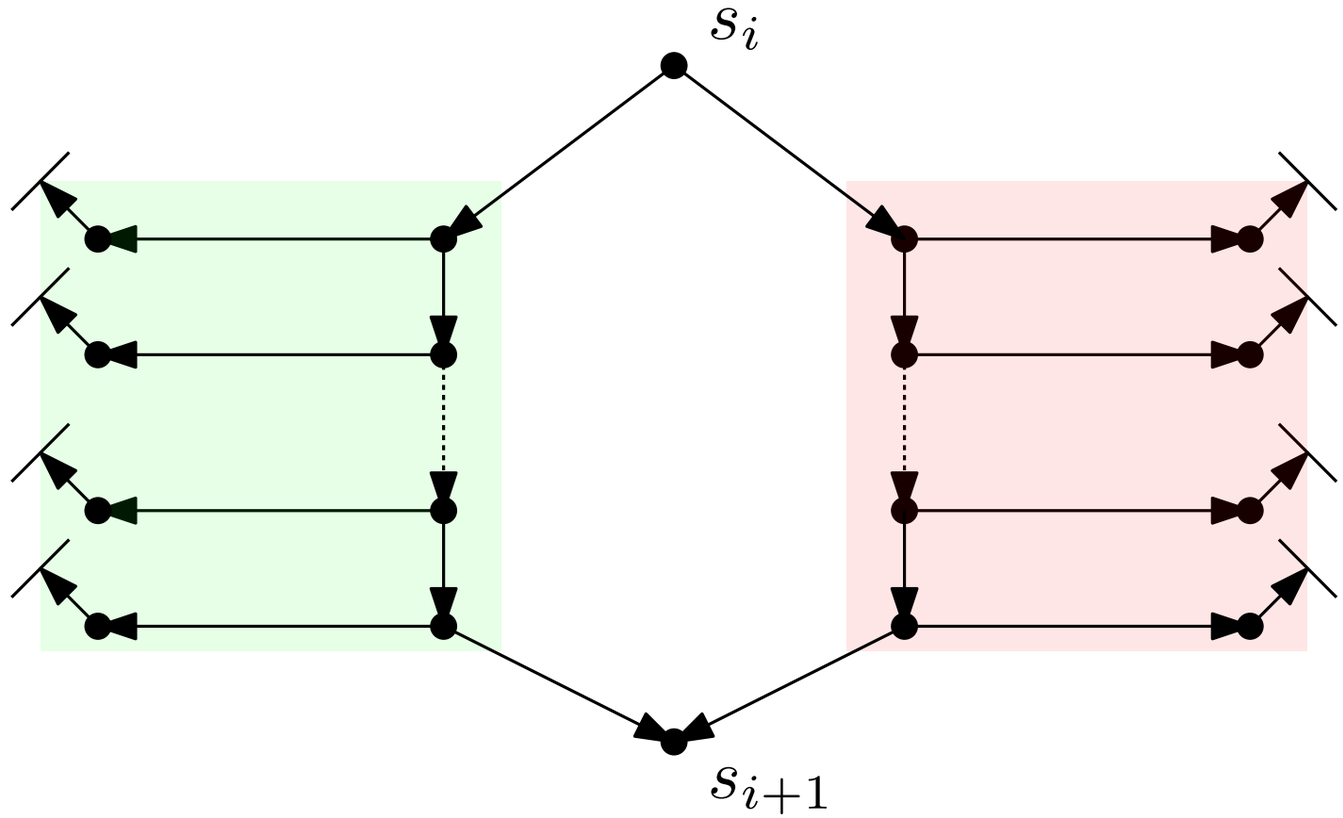


- **2 possibilities**:
  - m positive edges delayed
  - m negative edges delayed

Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

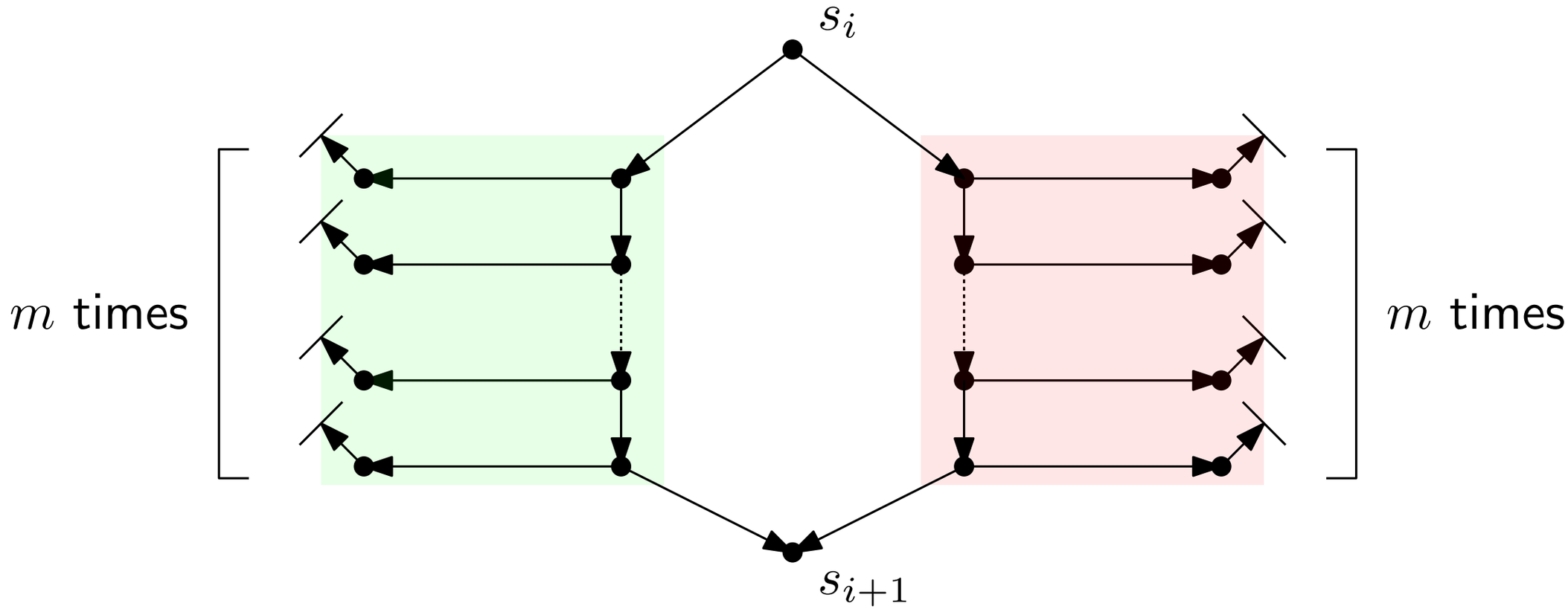
$\delta = 1$



Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

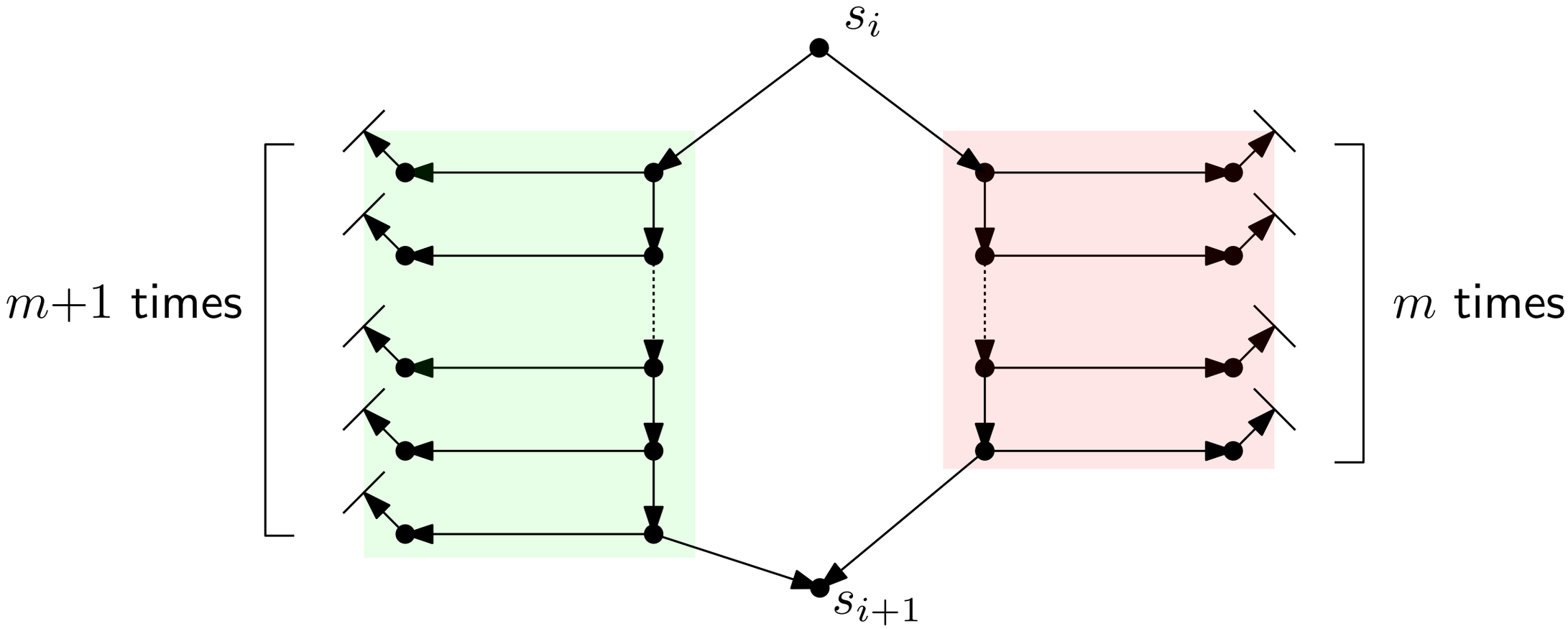
$\delta = 1$



Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

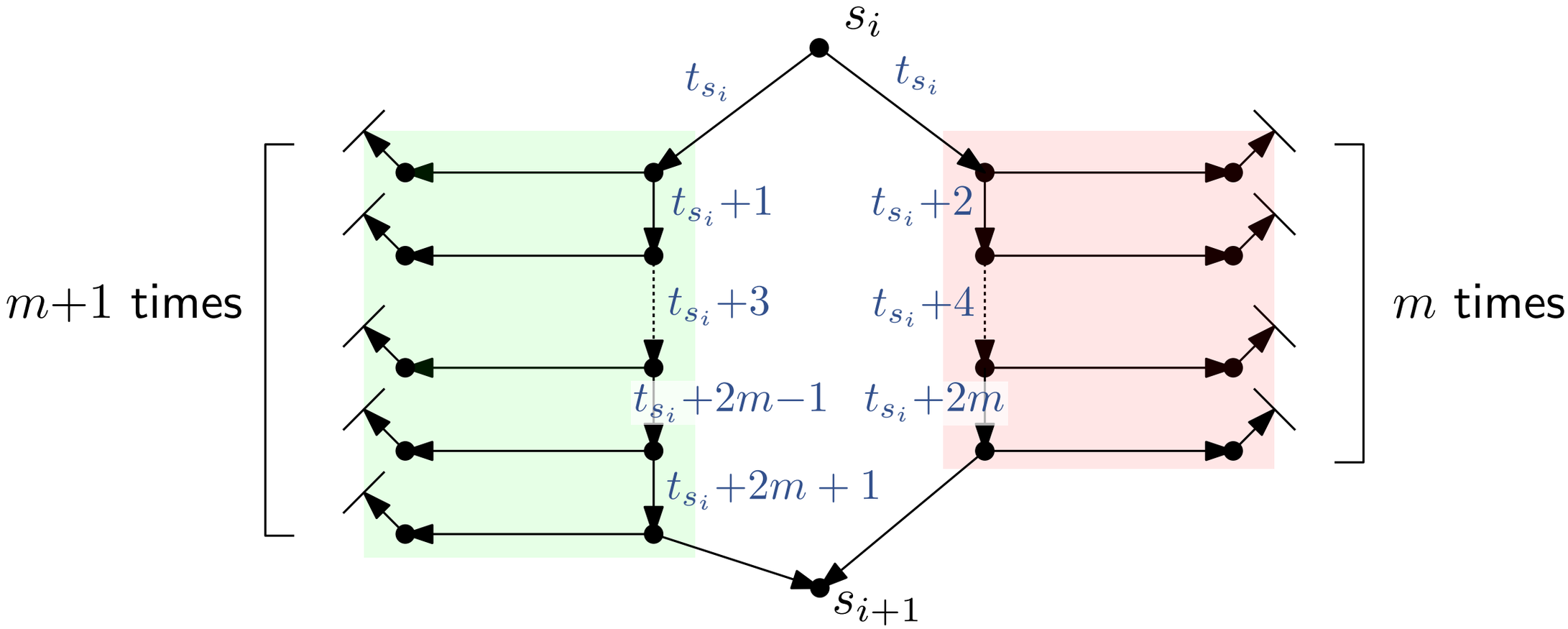
$\delta = 1$



Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

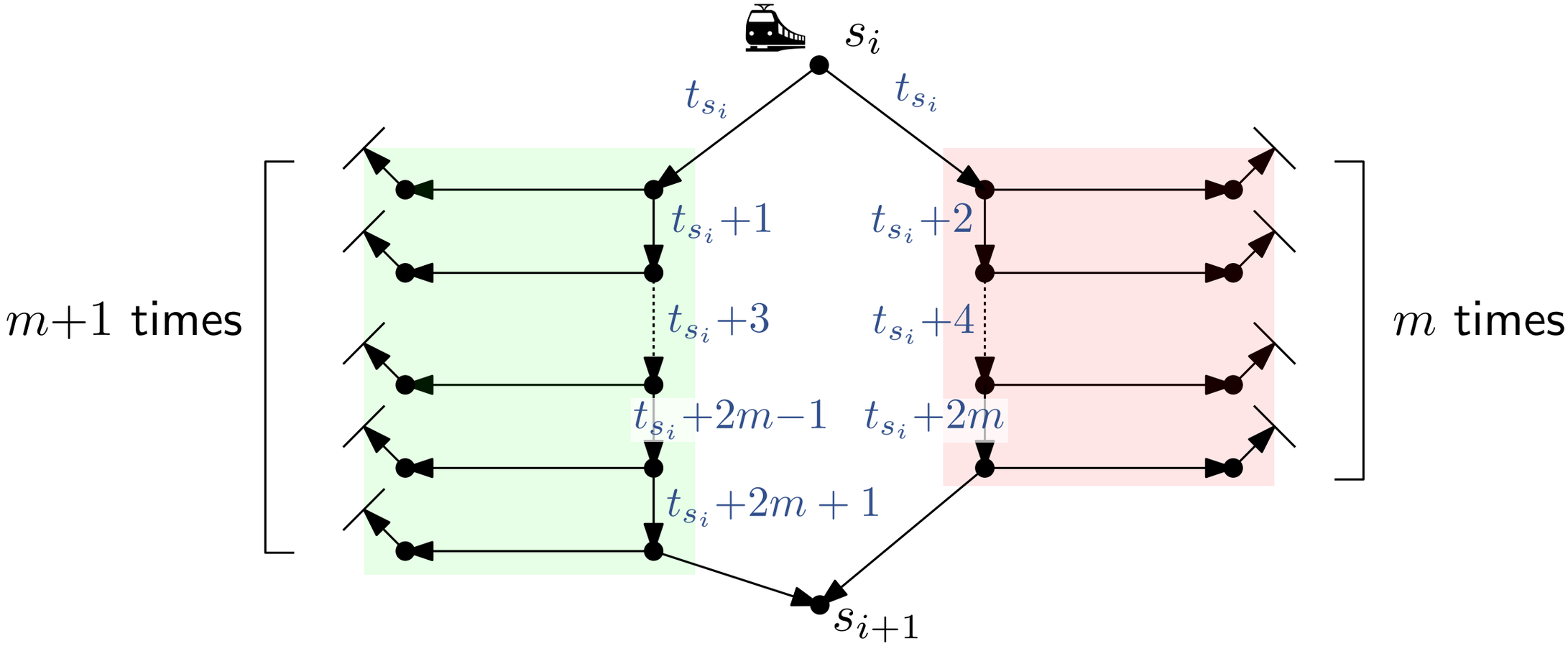
$\delta = 1$



Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

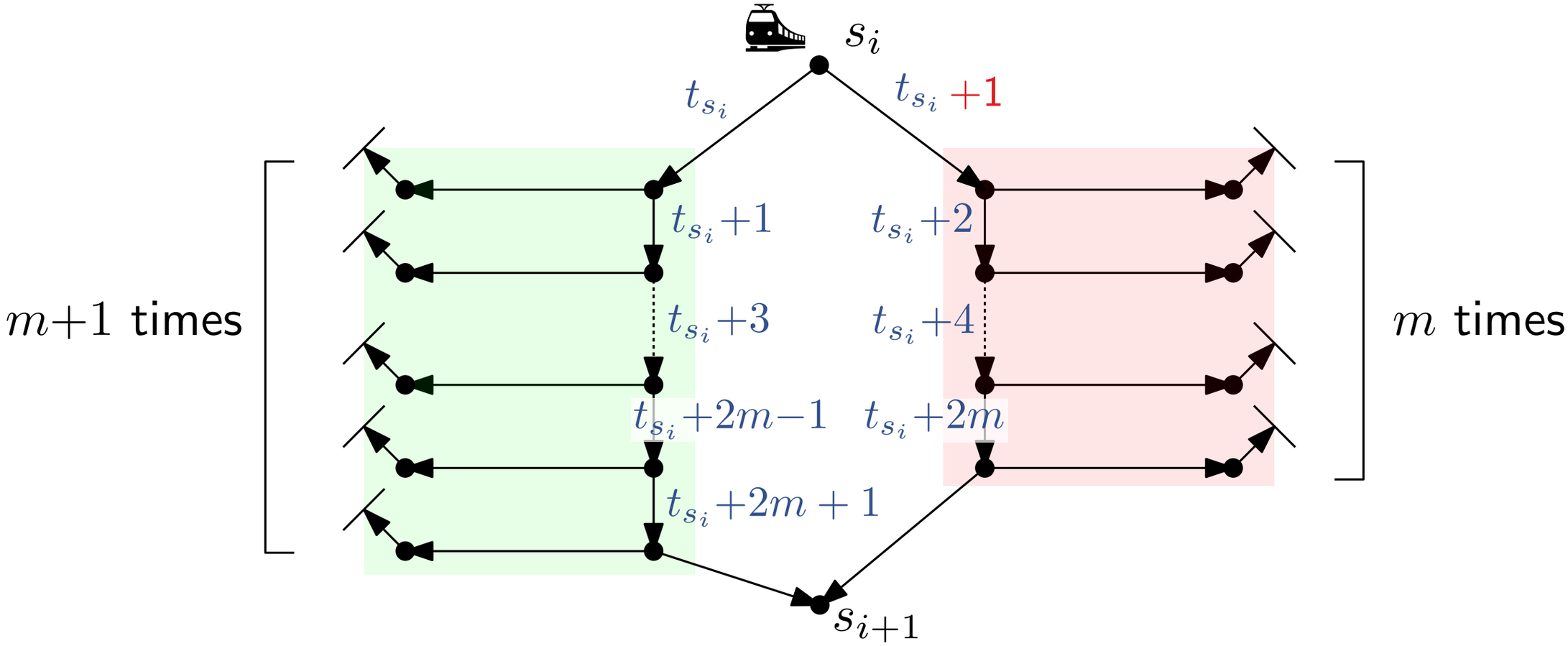
$\delta = 1$



Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

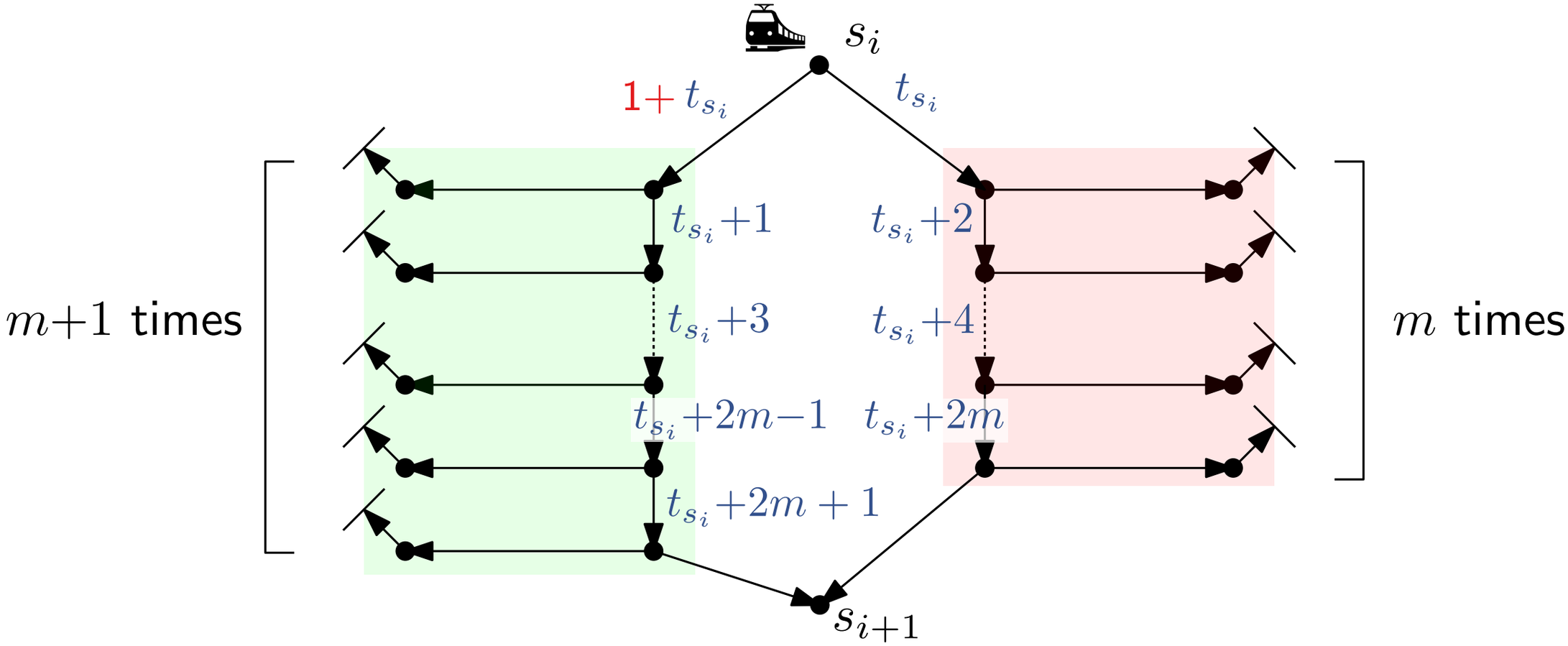
$\delta = 1$



Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

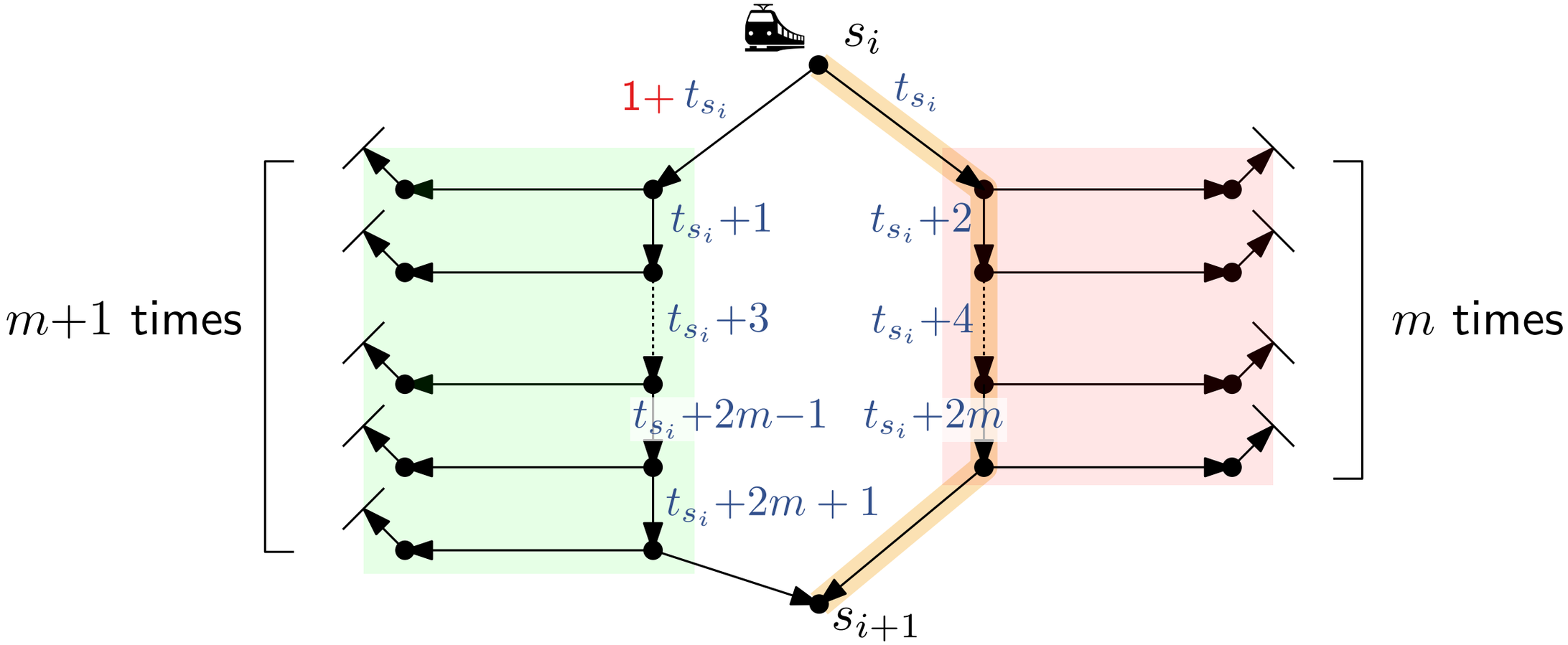
$\delta = 1$



Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

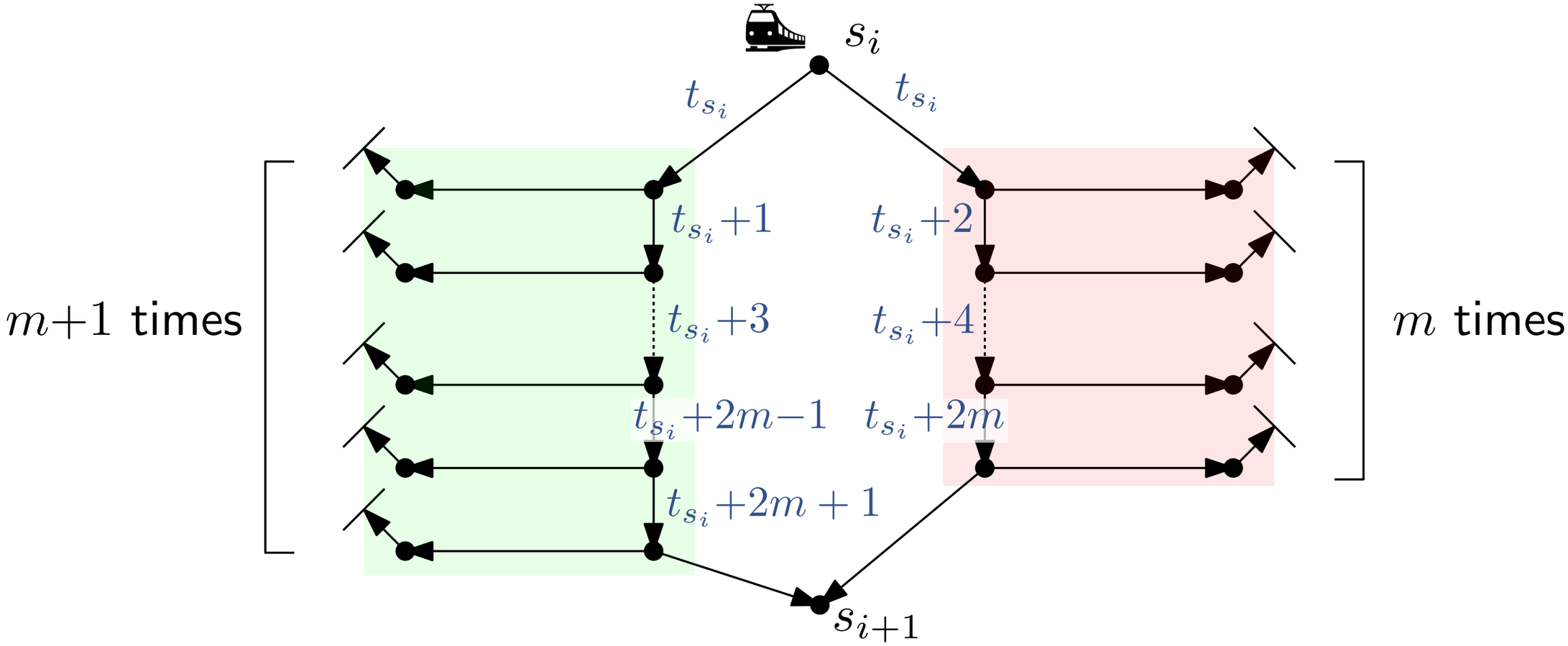
$\delta = 1$



Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

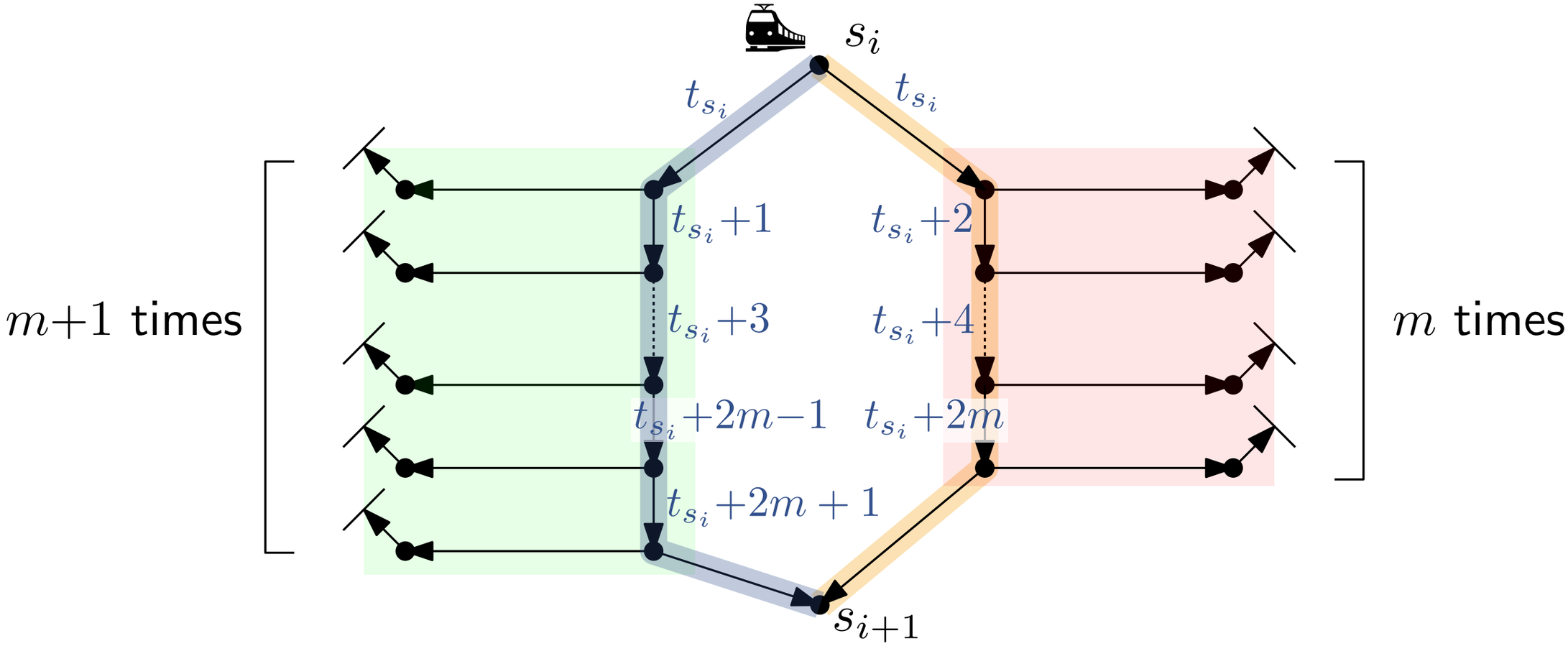
$\delta = 1$



Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

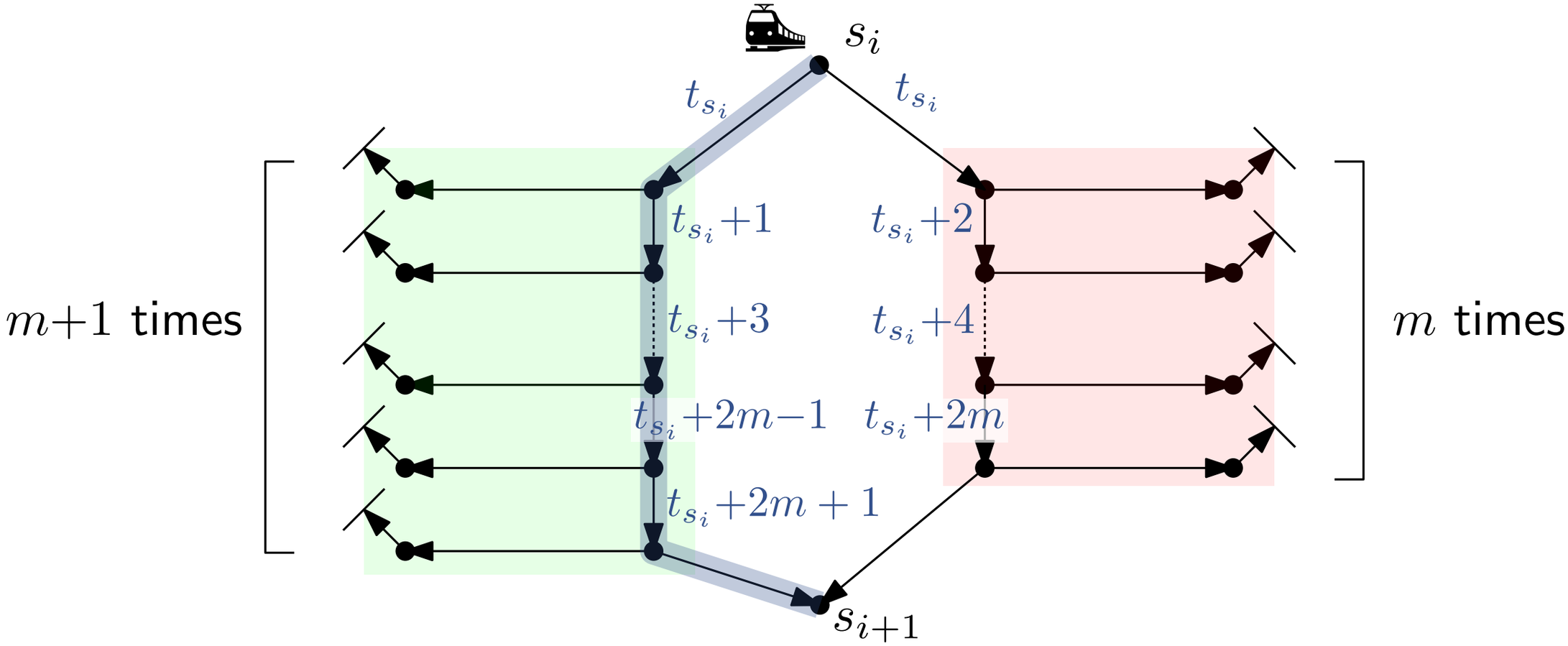
$\delta = 1$



Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

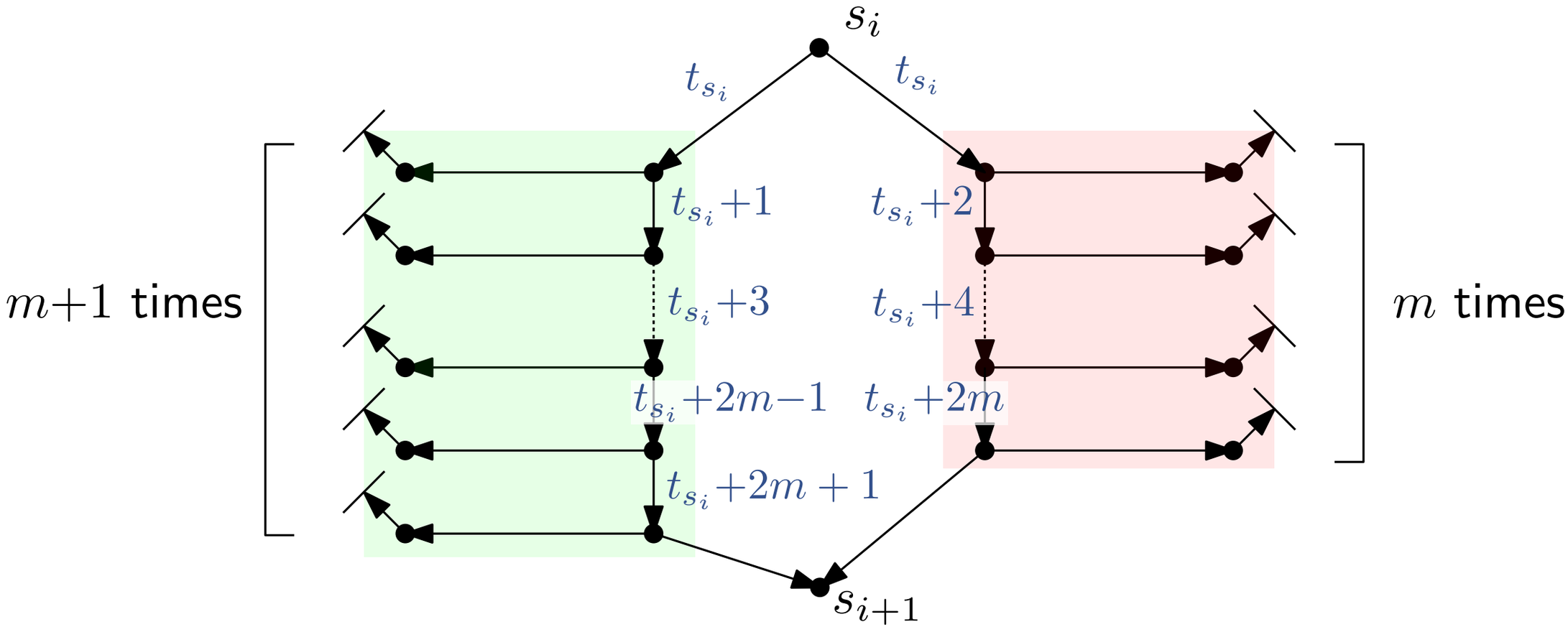
$\delta = 1$



Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

$\delta = 1$

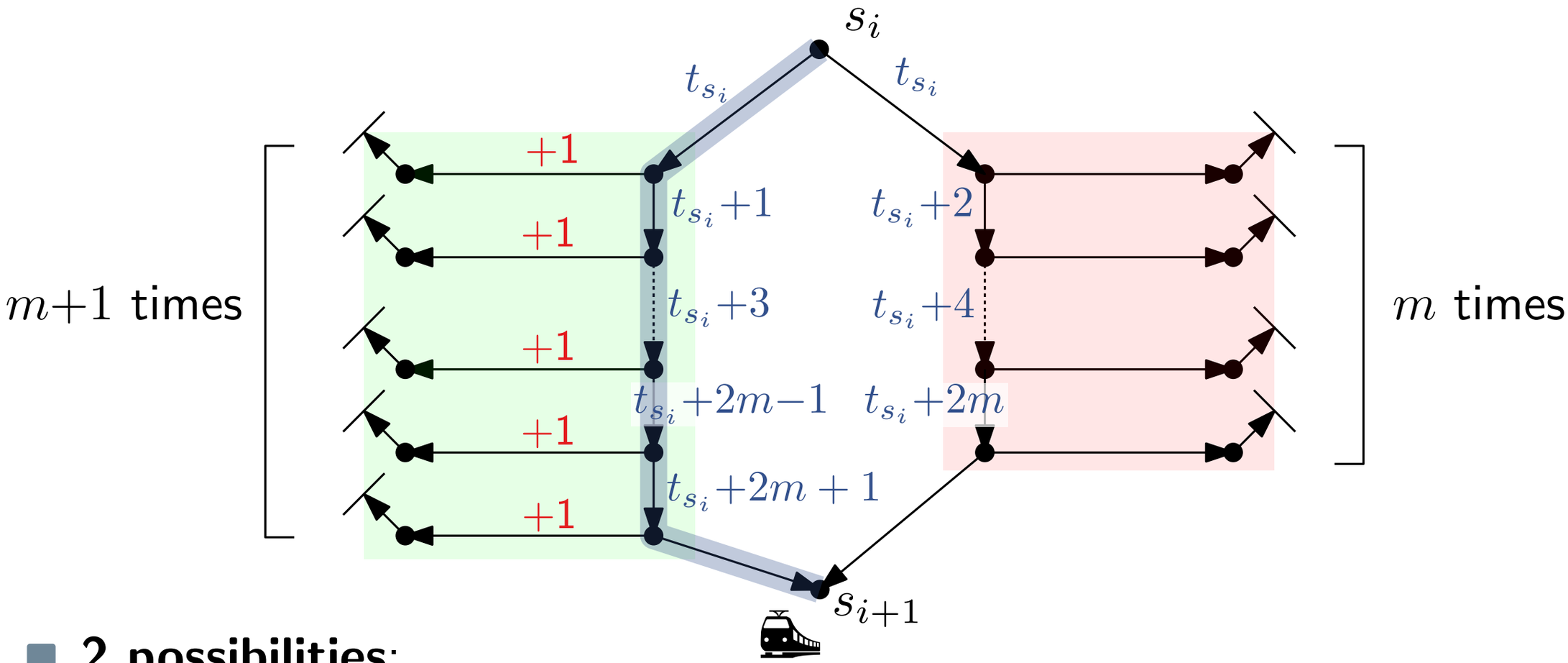


- **2 possibilities**:

Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

$\delta = 1$

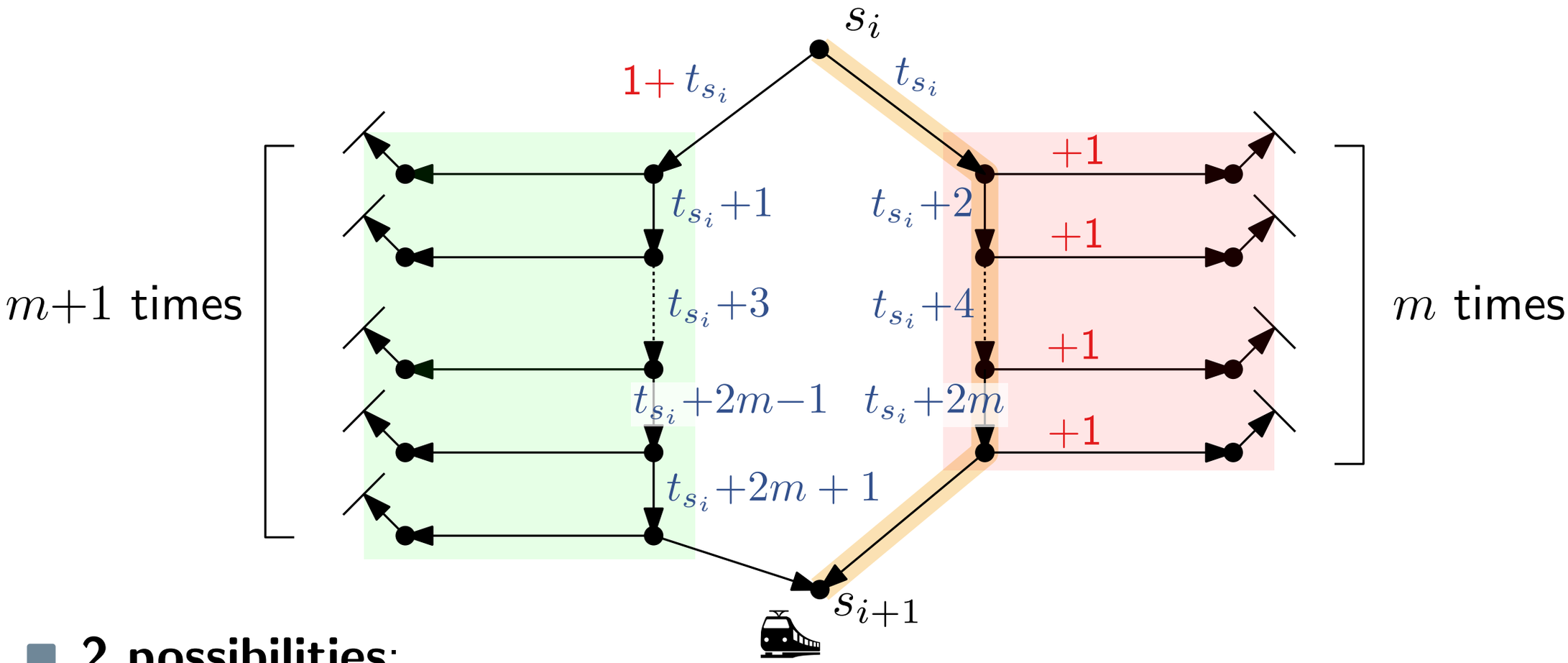


- **2 possibilities**:
  - $m+1$ positive edges delayed

Universal Variable Gadget

- $Q_1 x_1 = \forall x_i$

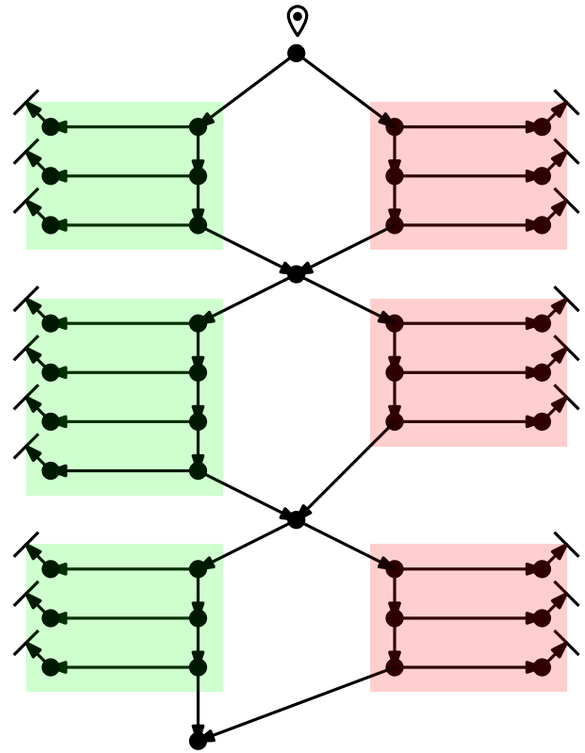
$\delta = 1$



- **2 possibilities**:
  - $m+1$ positive edges delayed
  - m negative edges delayed $+1$ extra edge

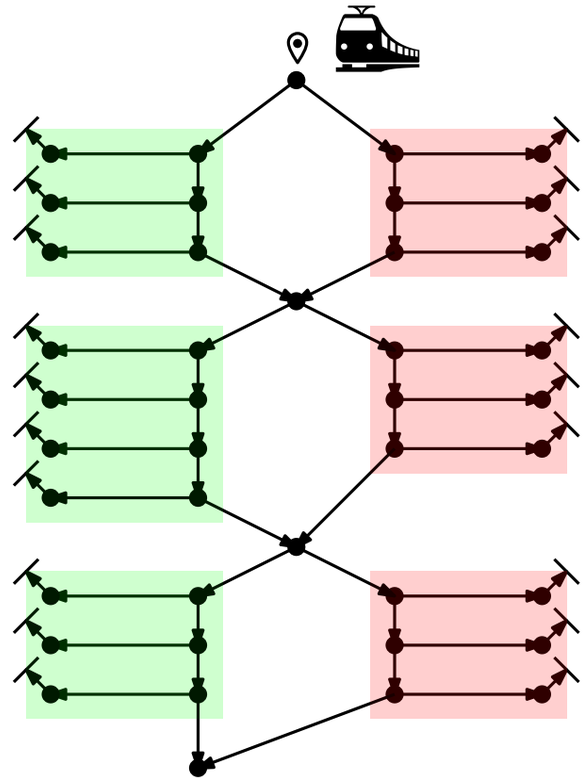
Clause Gadget

$$\begin{aligned}\phi = &\exists x_1 \forall x_2 \exists x_3 \\ &((x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\ &(\neg x_1 \vee x_2 \vee x_3) \wedge \\ &(x_1 \vee \neg x_2 \vee x_3))\end{aligned}$$



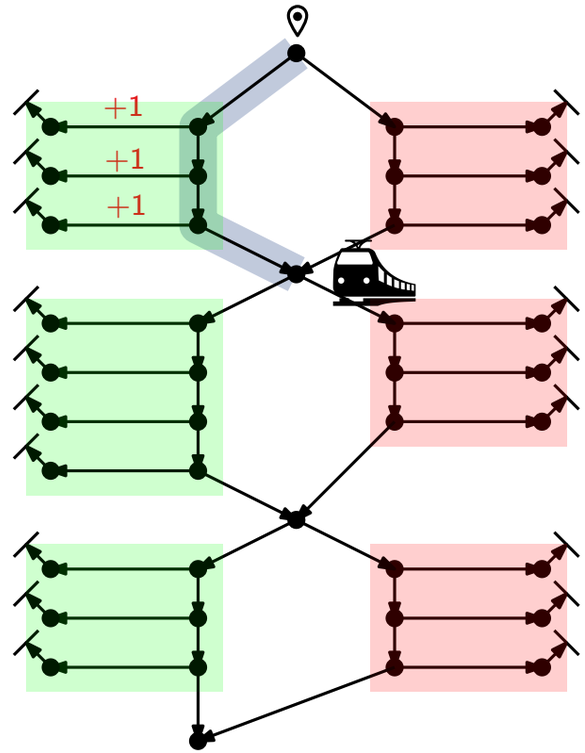
Clause Gadget

$$\begin{aligned}\phi = &\exists x_1 \forall x_2 \exists x_3 \\ &((x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\ &(\neg x_1 \vee x_2 \vee x_3) \wedge \\ &(x_1 \vee \neg x_2 \vee x_3))\end{aligned}$$



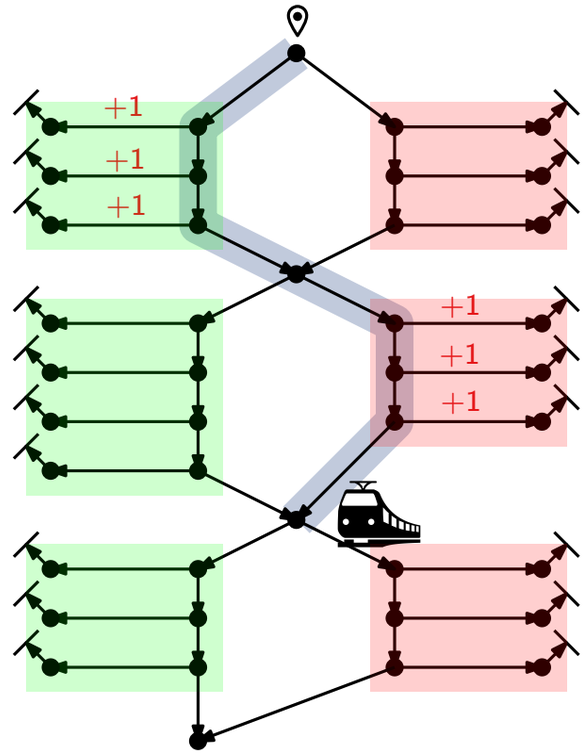
Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \\ ((x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\ (\neg x_1 \vee x_2 \vee x_3) \wedge \\ (x_1 \vee \neg x_2 \vee x_3))$$



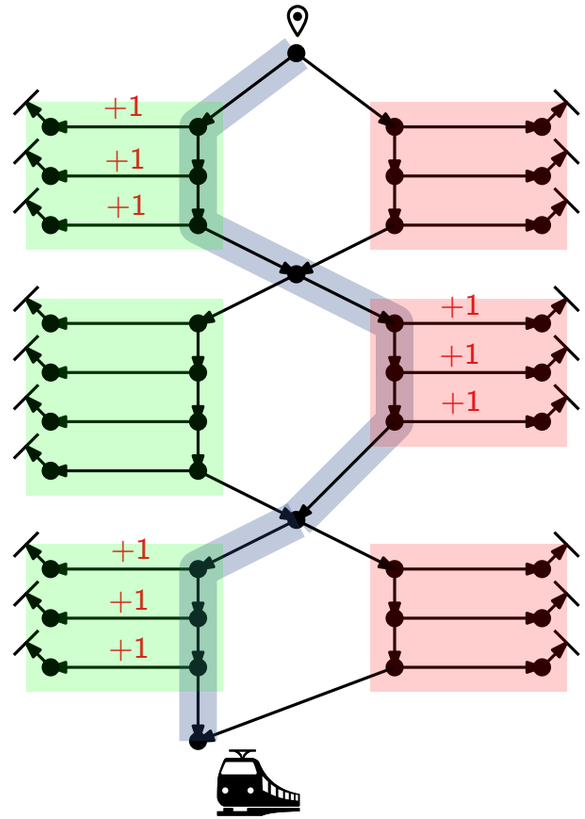
Clause Gadget

$$\begin{aligned}\phi = &\exists x_1 \forall x_2 \exists x_3 \\ &((x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\ &(\neg x_1 \vee x_2 \vee x_3) \wedge \\ &(x_1 \vee \neg x_2 \vee x_3))\end{aligned}$$



Clause Gadget

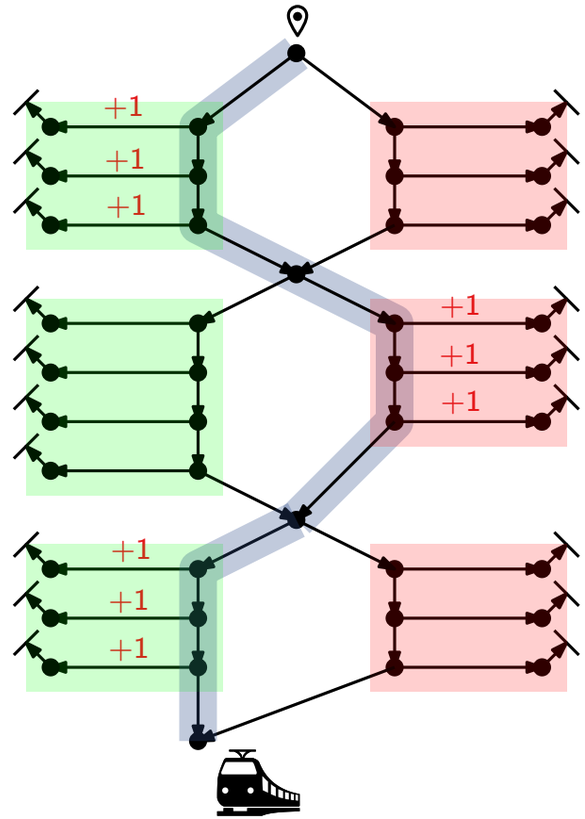
$$\phi = \exists x_1 \forall x_2 \exists x_3 \\ ((x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\ (\neg x_1 \vee x_2 \vee x_3) \wedge \\ (x_1 \vee \neg x_2 \vee x_3))$$



Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \\ ((x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\ (\neg x_1 \vee x_2 \vee x_3) \wedge \\ (x_1 \vee \neg x_2 \vee x_3))$$

- no budget left



Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \big((x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)\big)$$

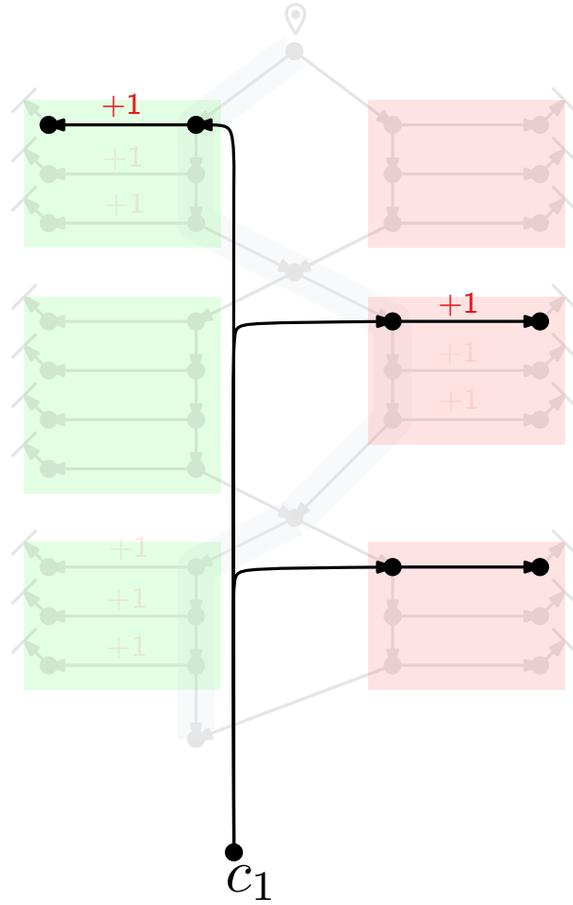
- no budget left

c_1

Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \big((x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)\big)$$

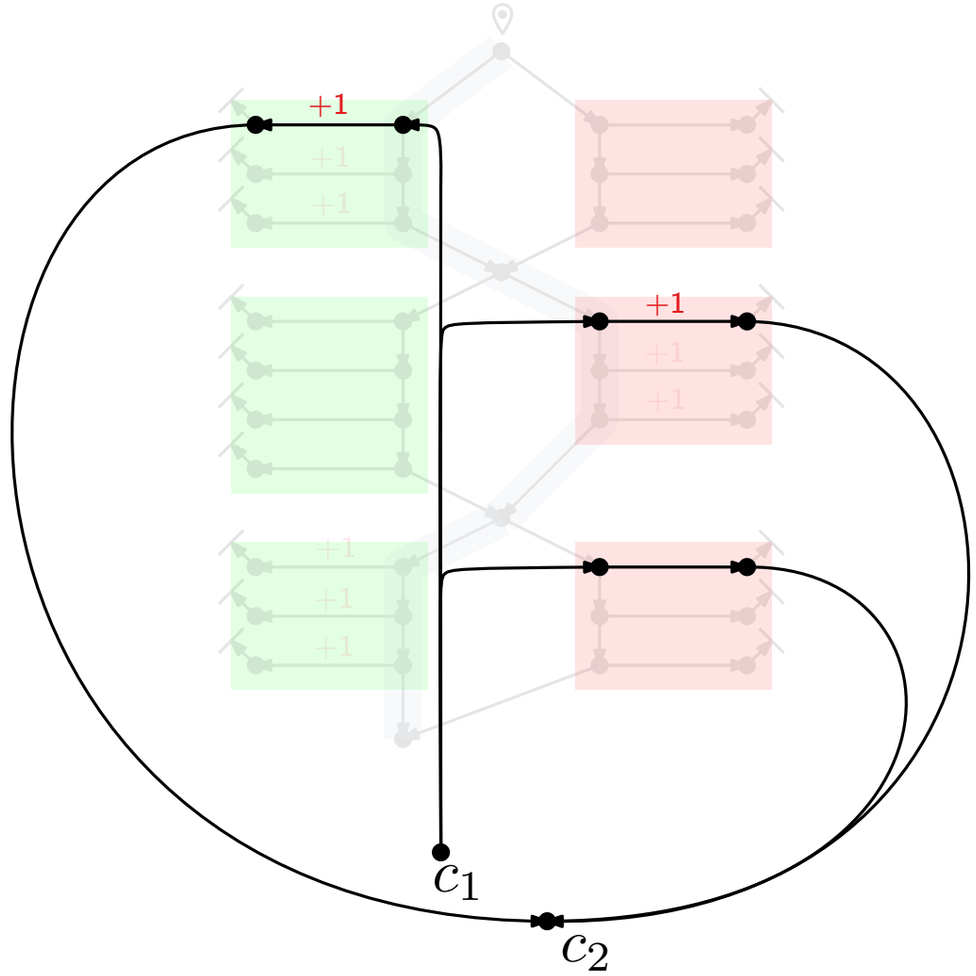
- no budget left



Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \big((x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)\big)$$

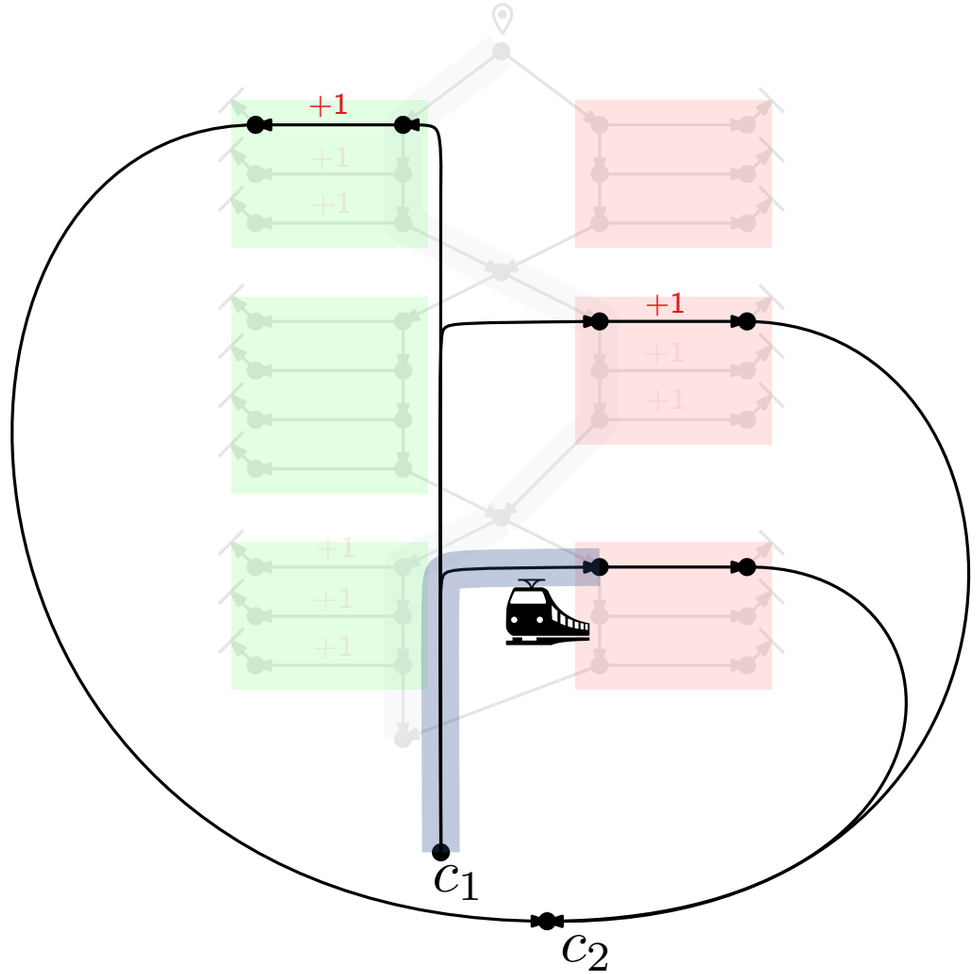
- no budget left



Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \big((x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)\big)$$

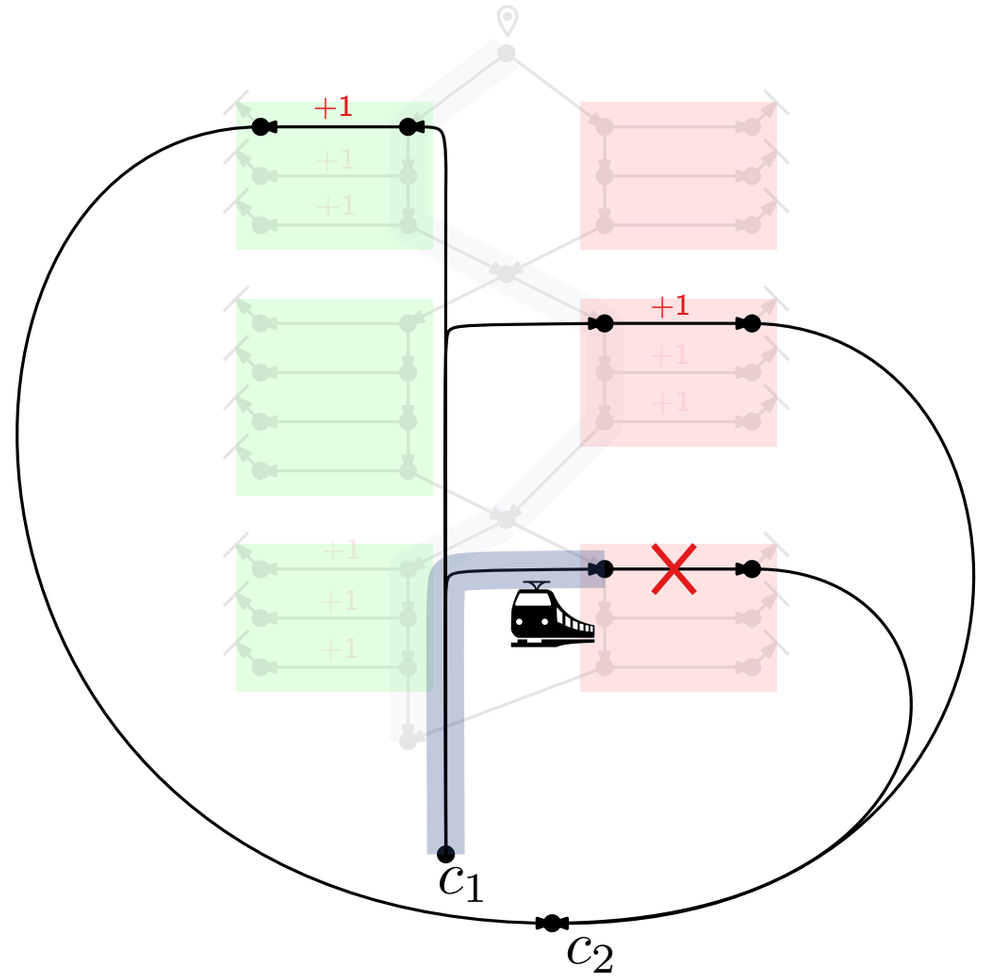
- no budget left



Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \big((x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)\big)$$

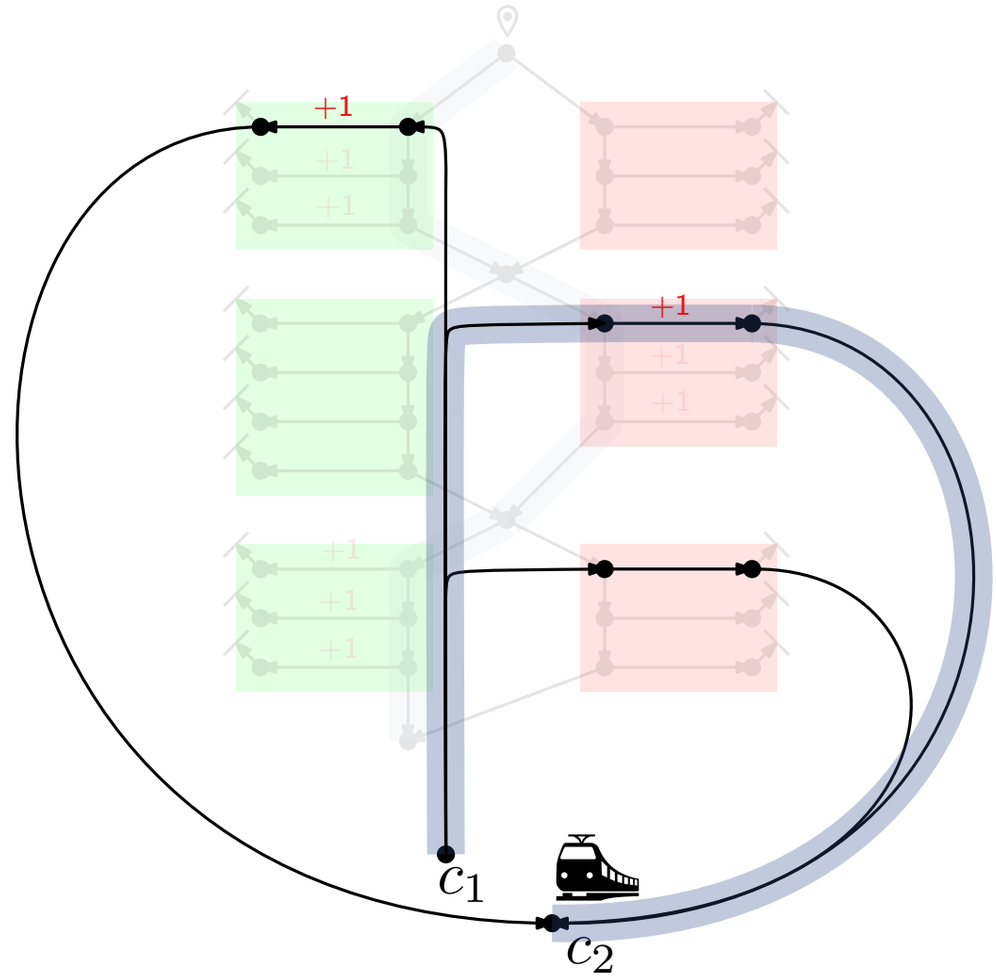
- no budget left



Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \big((x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)\big)$$

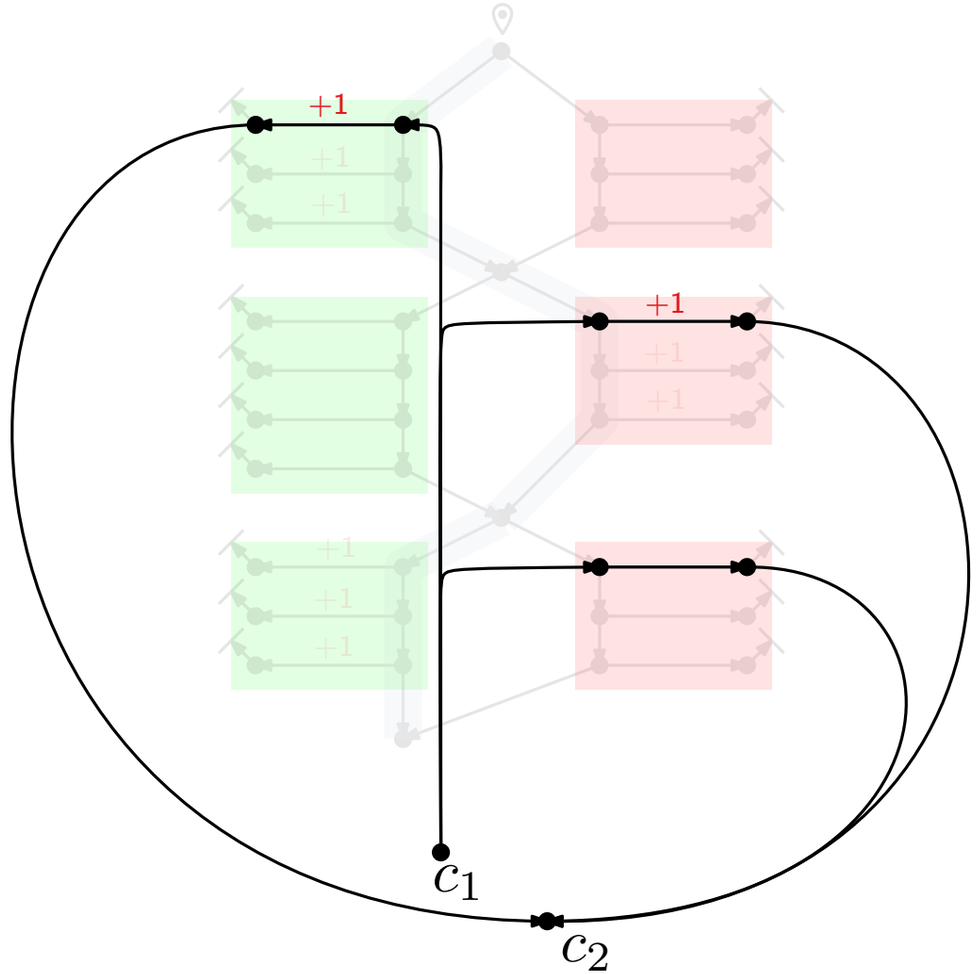
- no budget left



Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \big((x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)\big)$$

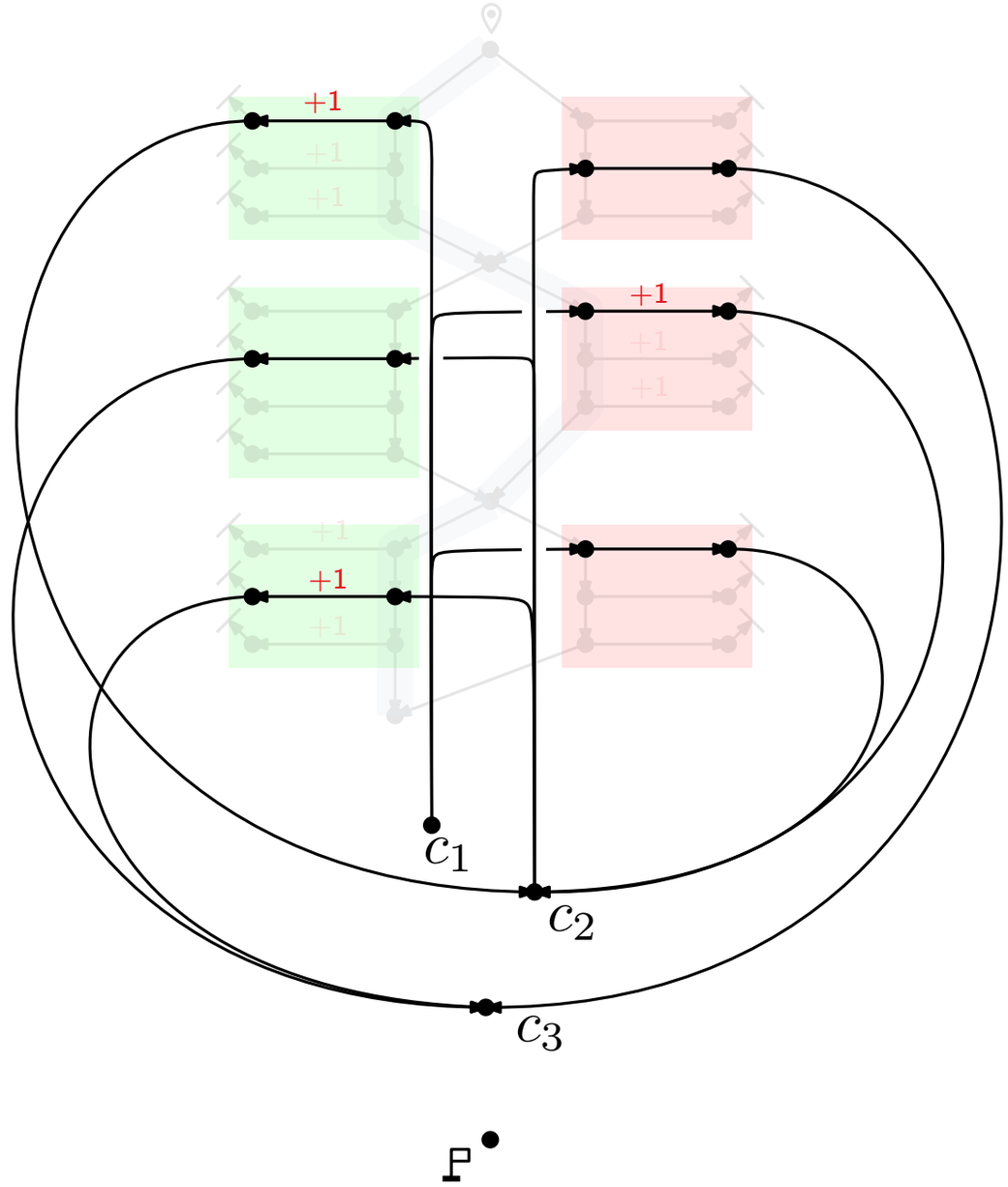
- no budget left



Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \big((x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)\big)$$

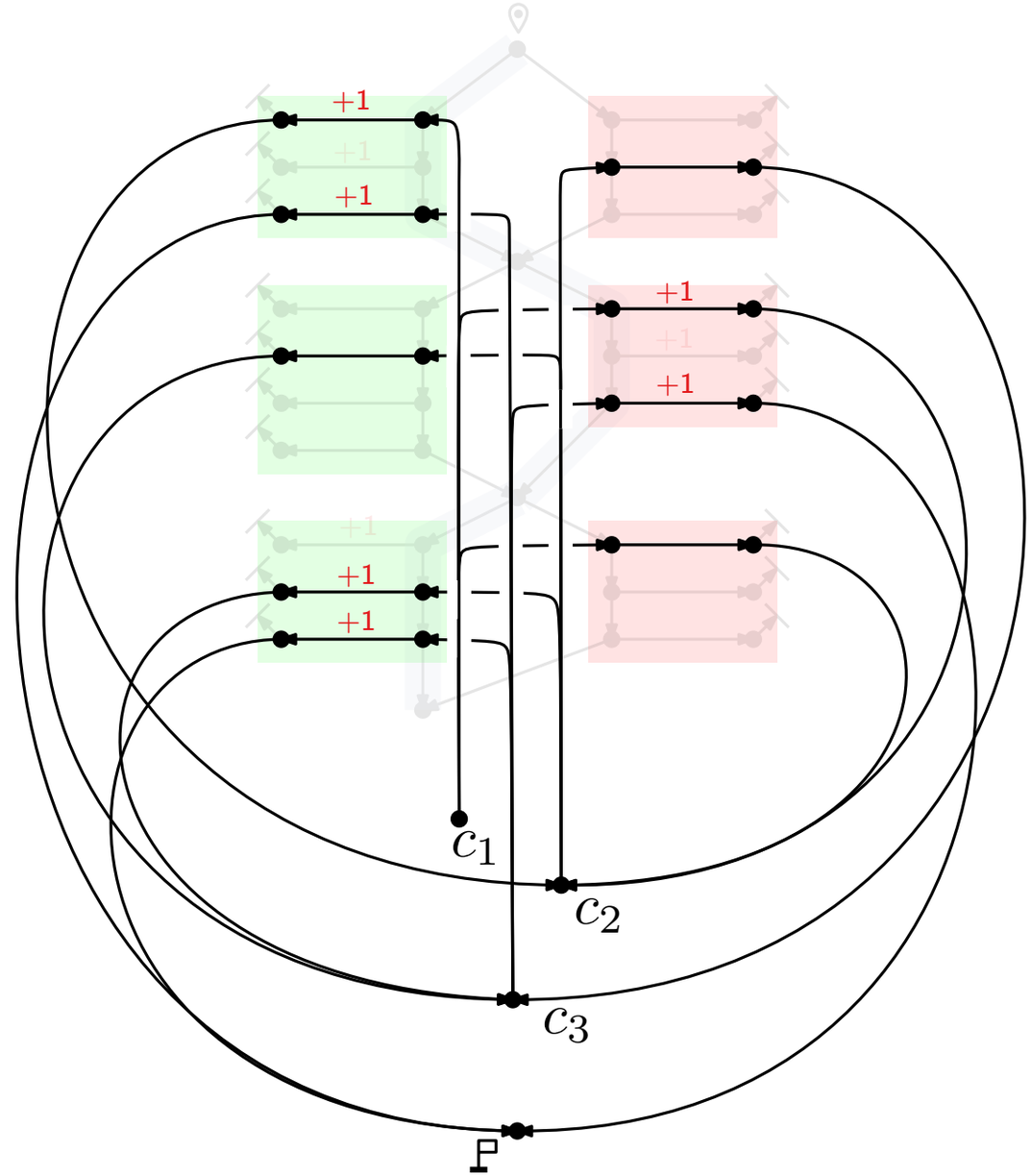
- no budget left



Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \; ((x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3))$$

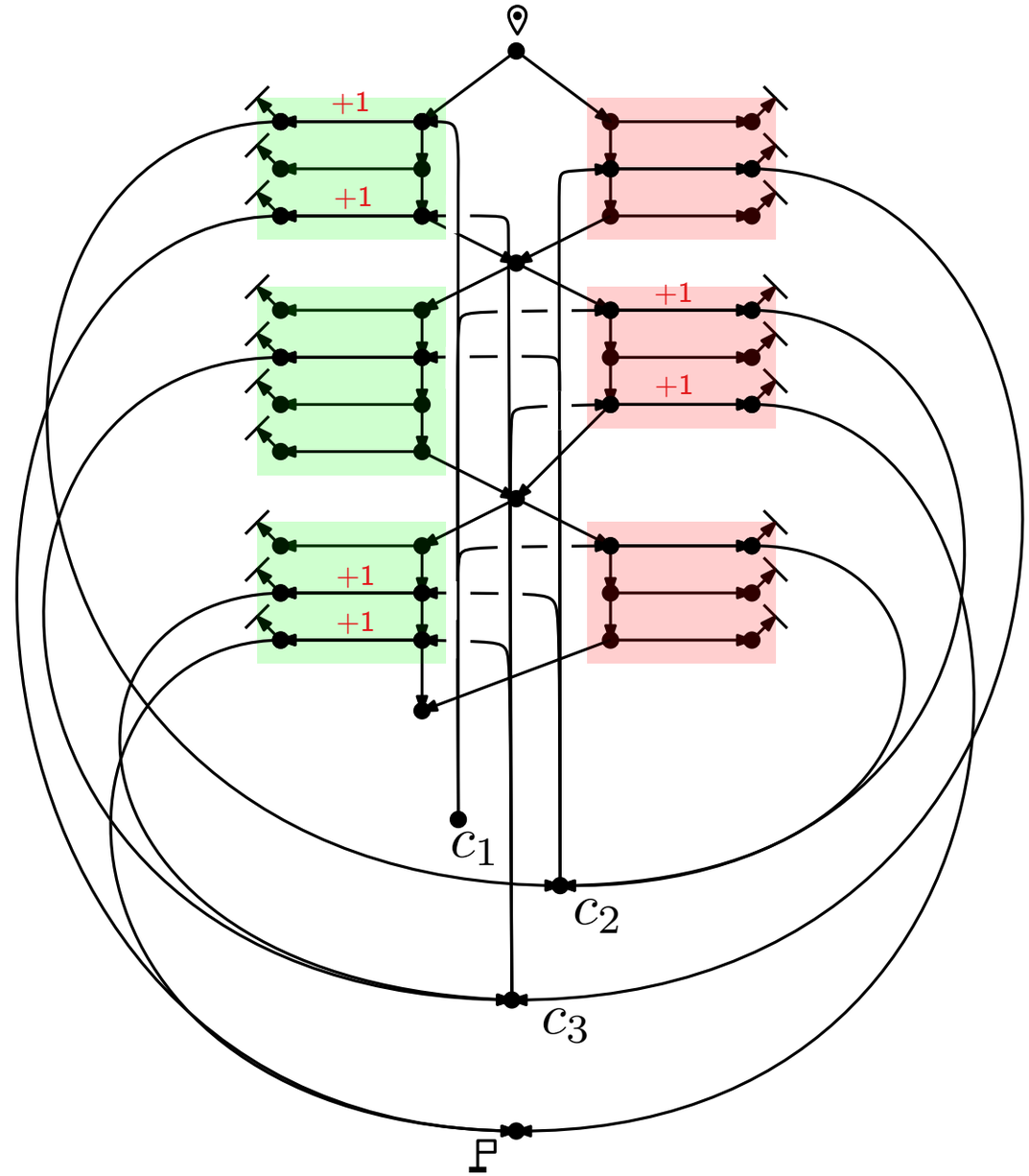
- no budget left



Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \\ ((x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\ (\neg x_1 \vee x_2 \vee x_3) \wedge \\ (x_1 \vee \neg x_2 \vee x_3))$$

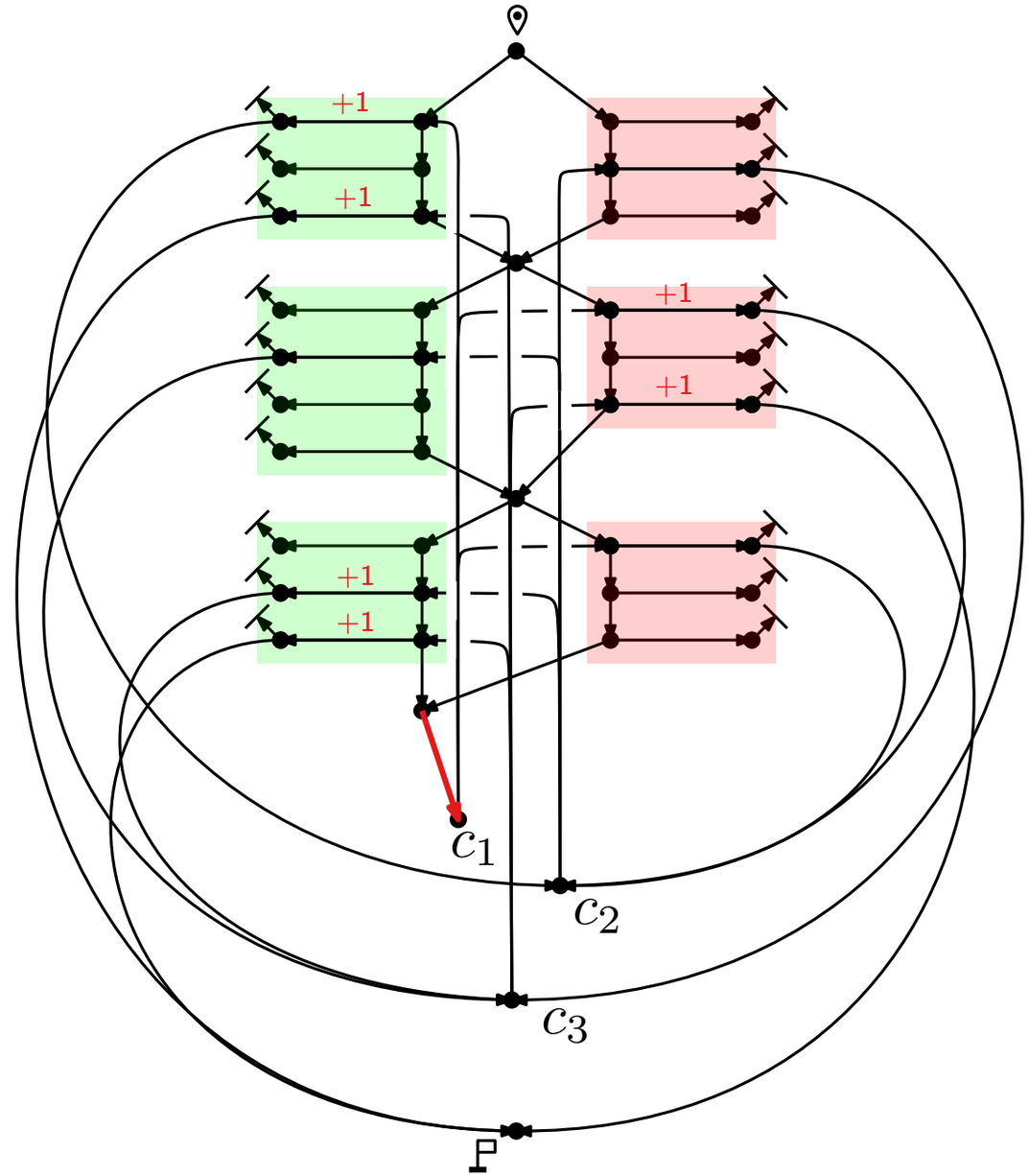
- no budget left



Clause Gadget

$$\phi = \exists x_1 \forall x_2 \exists x_3 \; ((x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3))$$

- no budget left



Clause Gadget

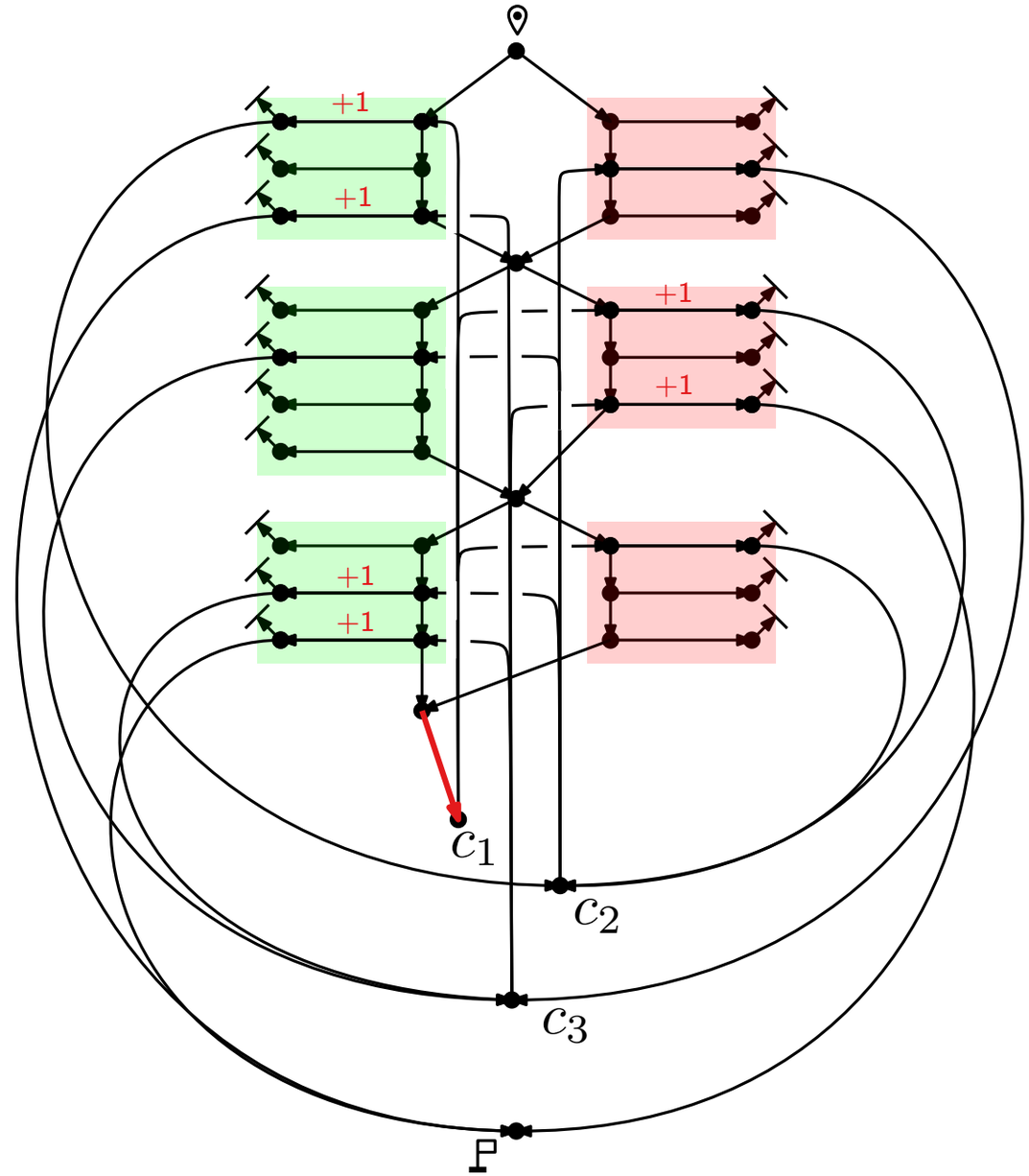
$$\phi = \exists x_1 \forall x_2 \exists x_3 \; ((x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3))$$

- no budget left

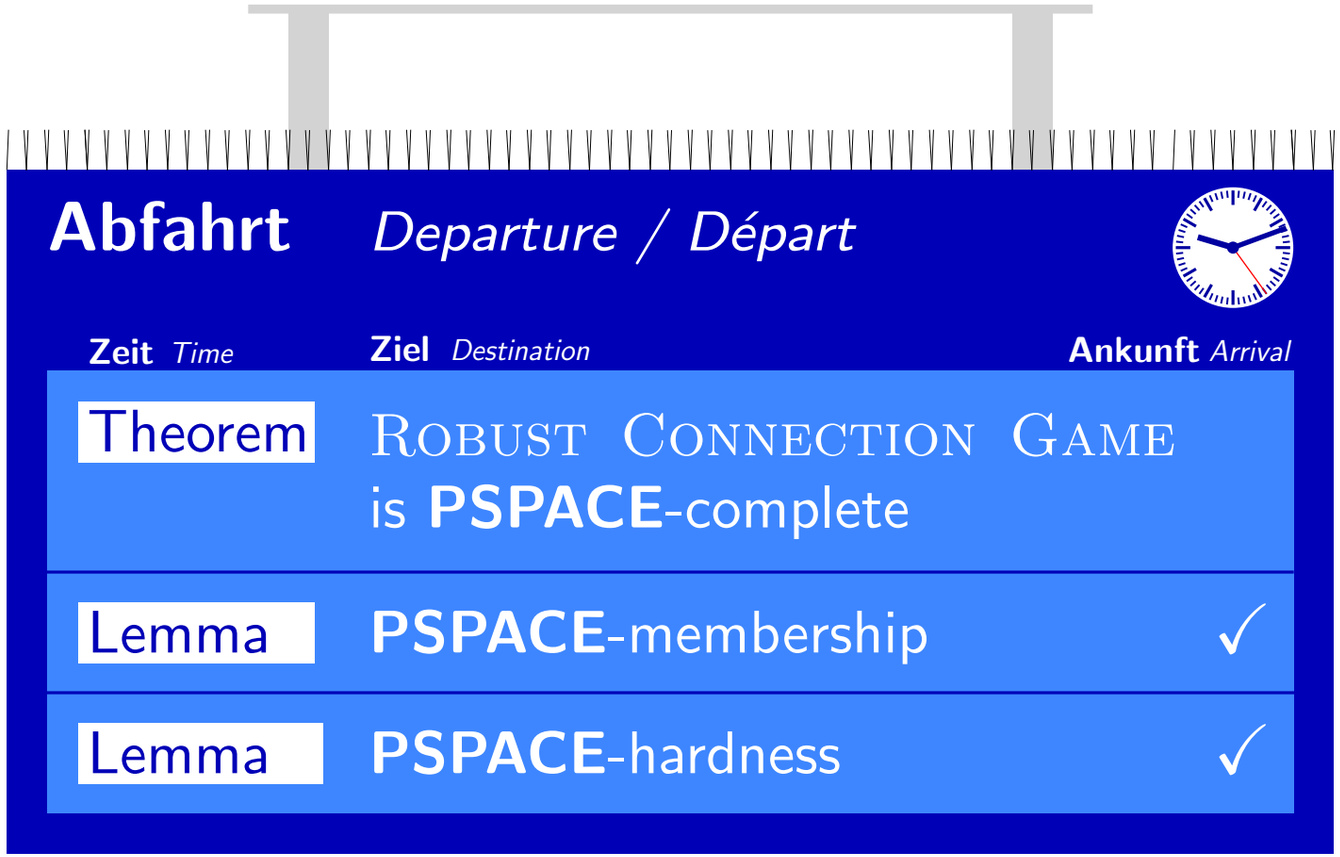
ϕ valid

$\Leftrightarrow$

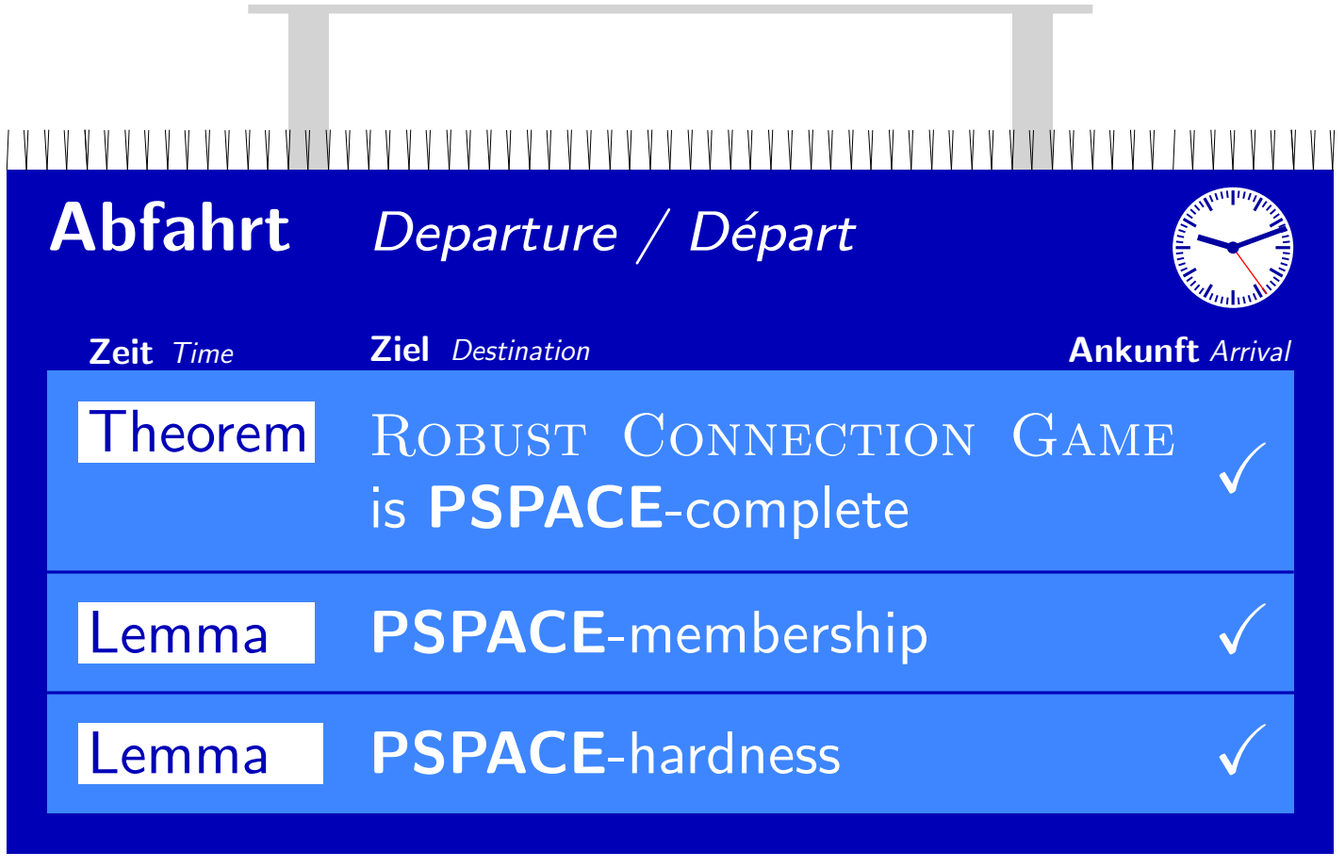
traveler has winning strategy



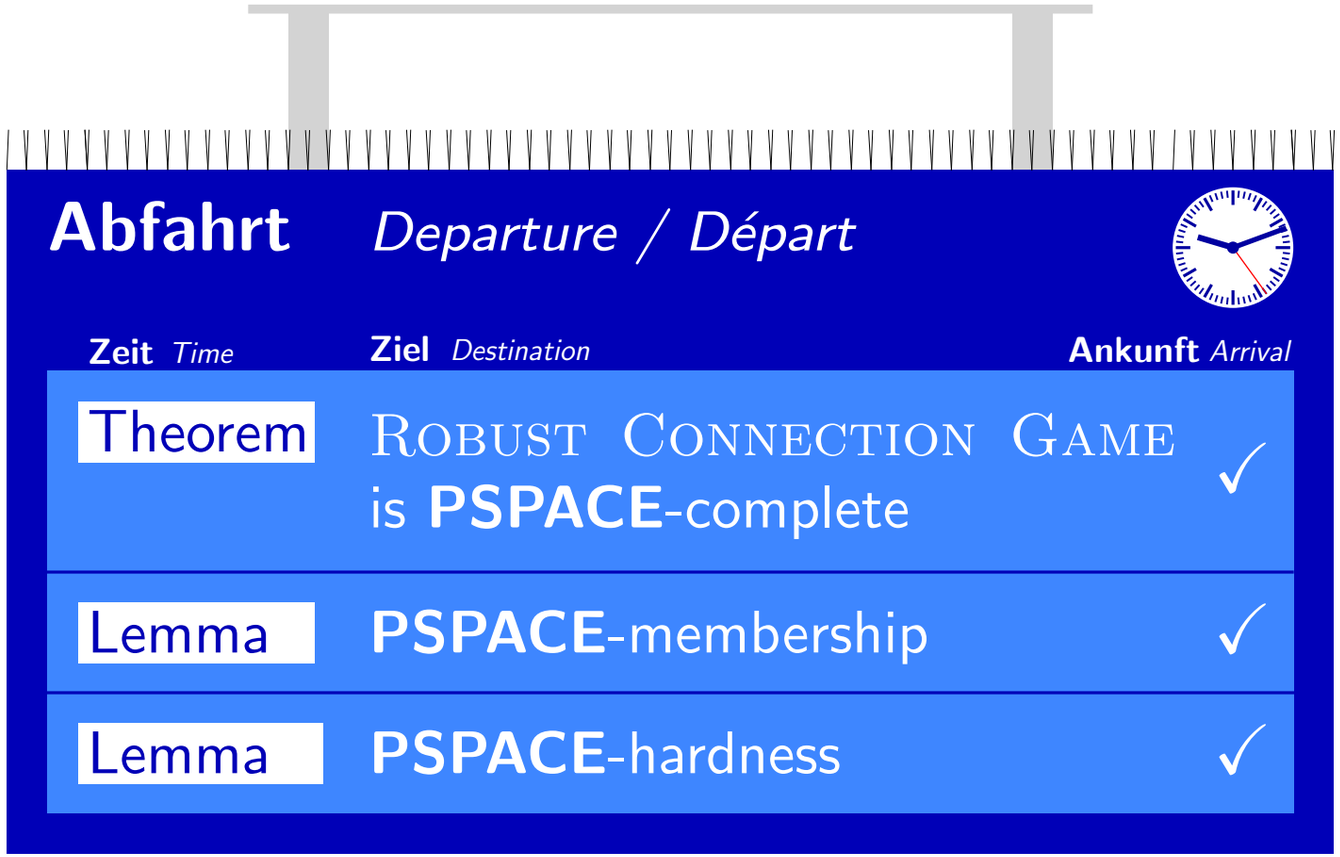
Summary


Abfahrt
Departure / Départ
Zeit Time
Ziel Destination
Ankunft Arrival
Theorem
Robust Connection Game is PSPACE-complete
Lemma
PSPACE-membership
✓
Lemma
PSPACE-hardness
✓

Summary

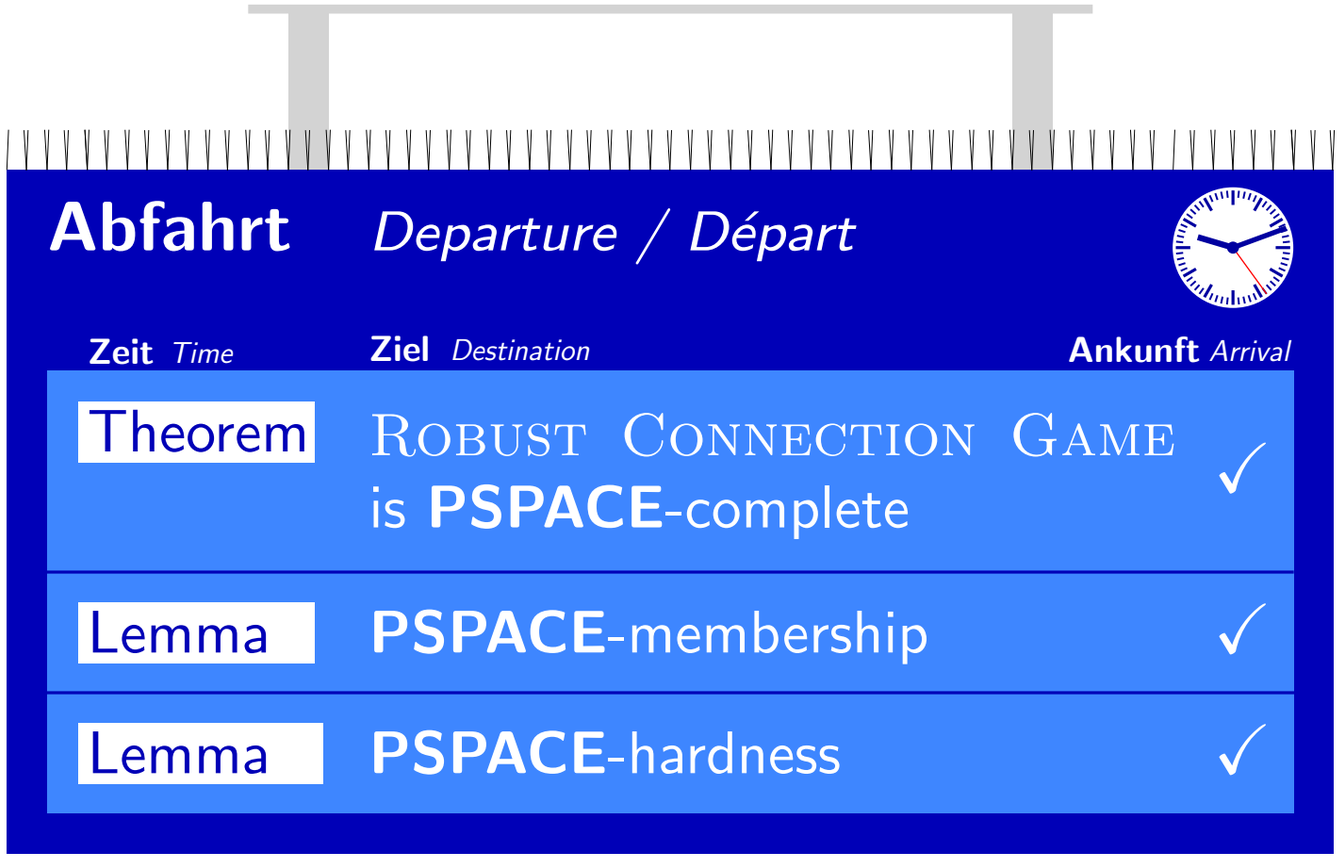

Abfahrt
Departure / Départ
Zeit Time
Ziel Destination
Ankunft Arrival
Theorem
Robust Connection Game is PSPACE-complete
✓
Lemma
PSPACE-membership
✓
Lemma
PSPACE-hardness
✓

Summary



- Tractability
- Heuristics

Summary



- Tractability
- Heuristics
- Robustness of network