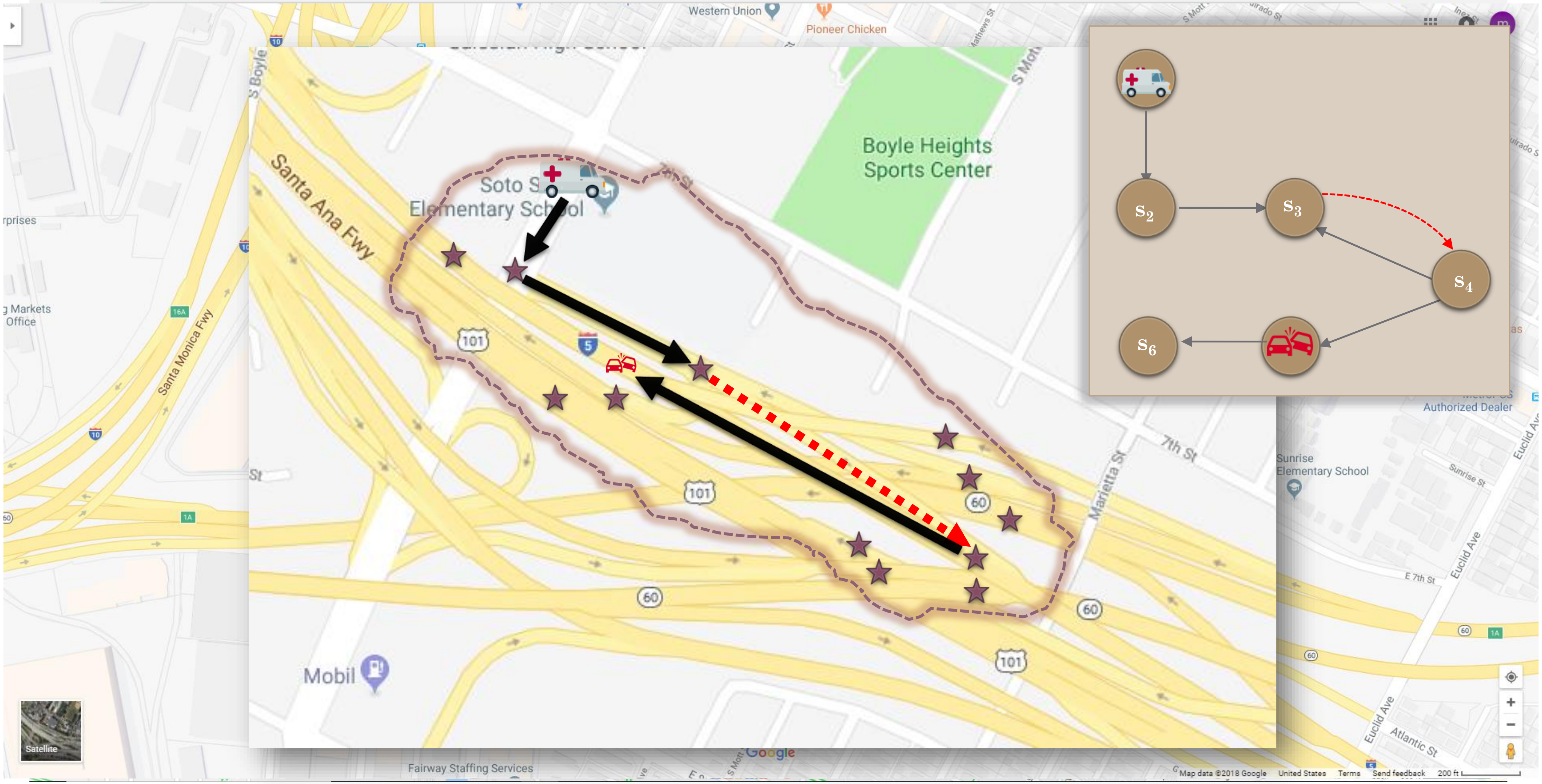
Optimal Multi-Valued LTL Planning for Systems with Access Right Levels

Mohammad Hekmatnejad, Georgios Fainekos

CPS Lab @ Arizona State University

mhekmatn at asu edu
https://www.linkedin.com/in/mohammad-hekmatnejad-54535232

*Motivation Example

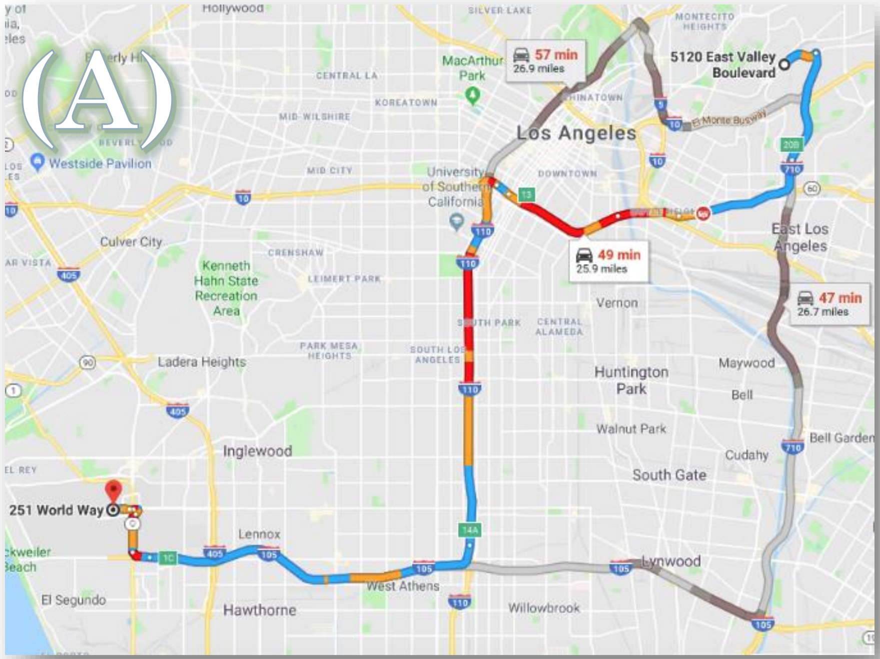


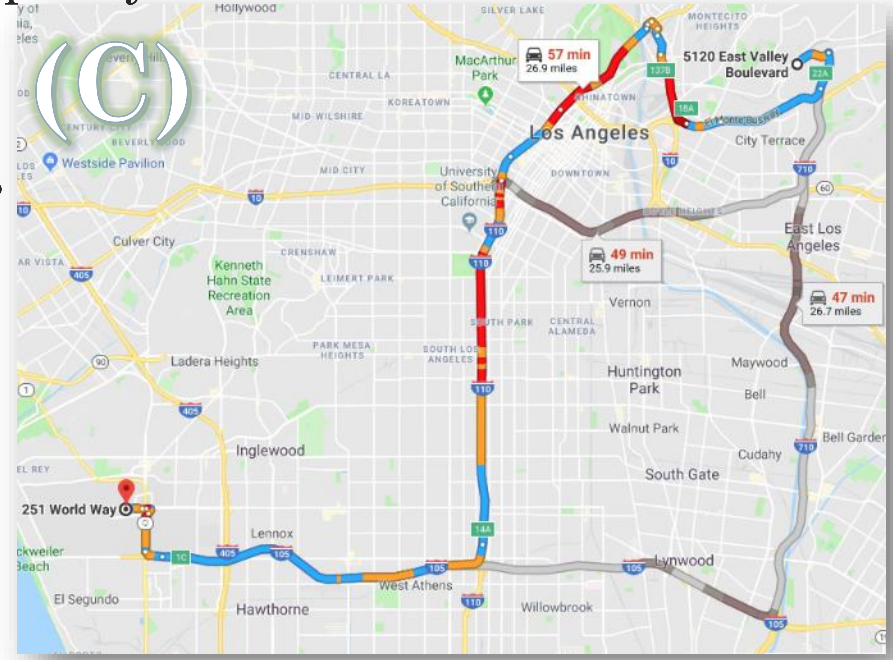
Icon made by freepik, and roundicons from www.flaticon.com

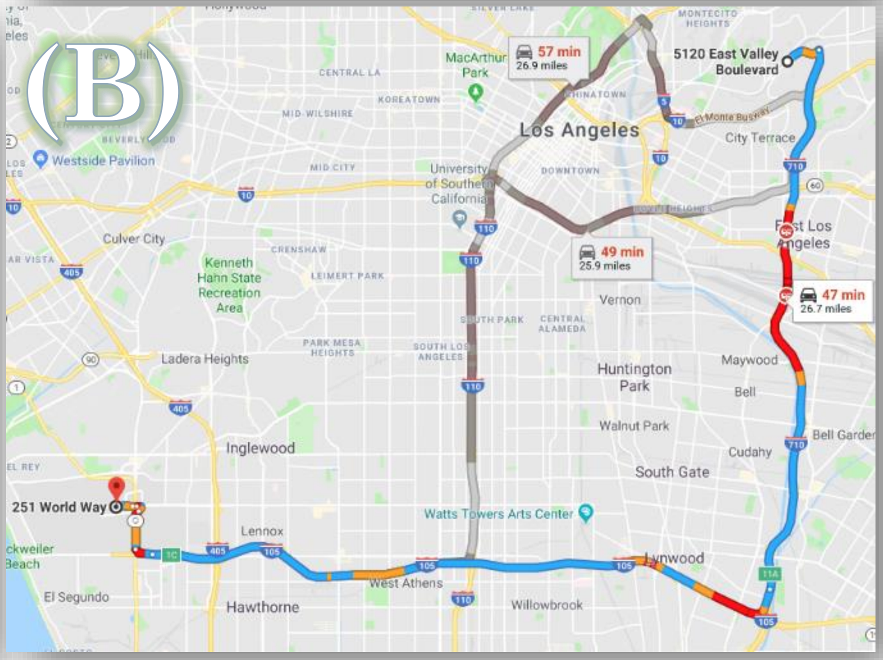
Taken from Google Maps at June 17th 1:30pm (Los Angeles)

*Motivation (cont’)

- Plan for future based on current observation (partial or incomplete information)
- Accident blockage, traffic, road work, and even accessibility are temporarily
- What is the optimal path based on:
  - Highest road capacity
  - Highest safety
  - Least traffic
  - Least red lights
  - Least turns
  - Least deadlock



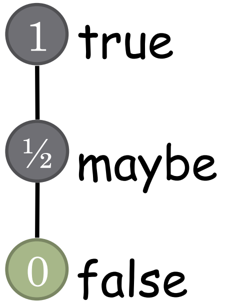




*What is insufficient?

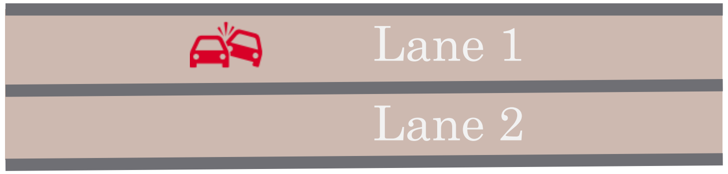
- Modeling perspective
  - More than one truth degree can be assigned to a symbol for expressing uncertainty of value assignments

$\mathcal{B}_3=(\{0,\frac{1}{2},1\},\leq)$



$\neg 1=0,\ \neg 0=1,\ \neg\frac{1}{2}=\frac{1}{2}$

What is the accessibility of the road?



- Specification perspective
  - Query the model to its highest capabilities with considering the minimum/maximum requested truth degree for the model's symbols

Never pass a road which its accessibility level is less than maybe:

$\square \boldsymbol{accessibility} \succcurlyeq \mathbf{0.5} \wedge \ \ldots$

- Theoretical perspective
  - Using multi-value sets as the range of variables and incorporate them into LTL formulas by extending the syntax and semantics of the LTL formulas.

$$\boldsymbol{\phi} ::= \boldsymbol{\pi} \preccurlyeq \boldsymbol{l} \mid \boldsymbol{\pi} \succcurlyeq \boldsymbol{l} \mid \boldsymbol{\phi} \vee \boldsymbol{\phi} \mid \boldsymbol{\phi} \wedge \boldsymbol{\phi} \mid \circ\, \boldsymbol{\phi} \mid \lozenge\, \boldsymbol{\phi} \mid \square \boldsymbol{\phi} \mid \boldsymbol{\phi} U \boldsymbol{\phi} \mid \boldsymbol{\phi} R \boldsymbol{\phi}$$

*Problem Statement

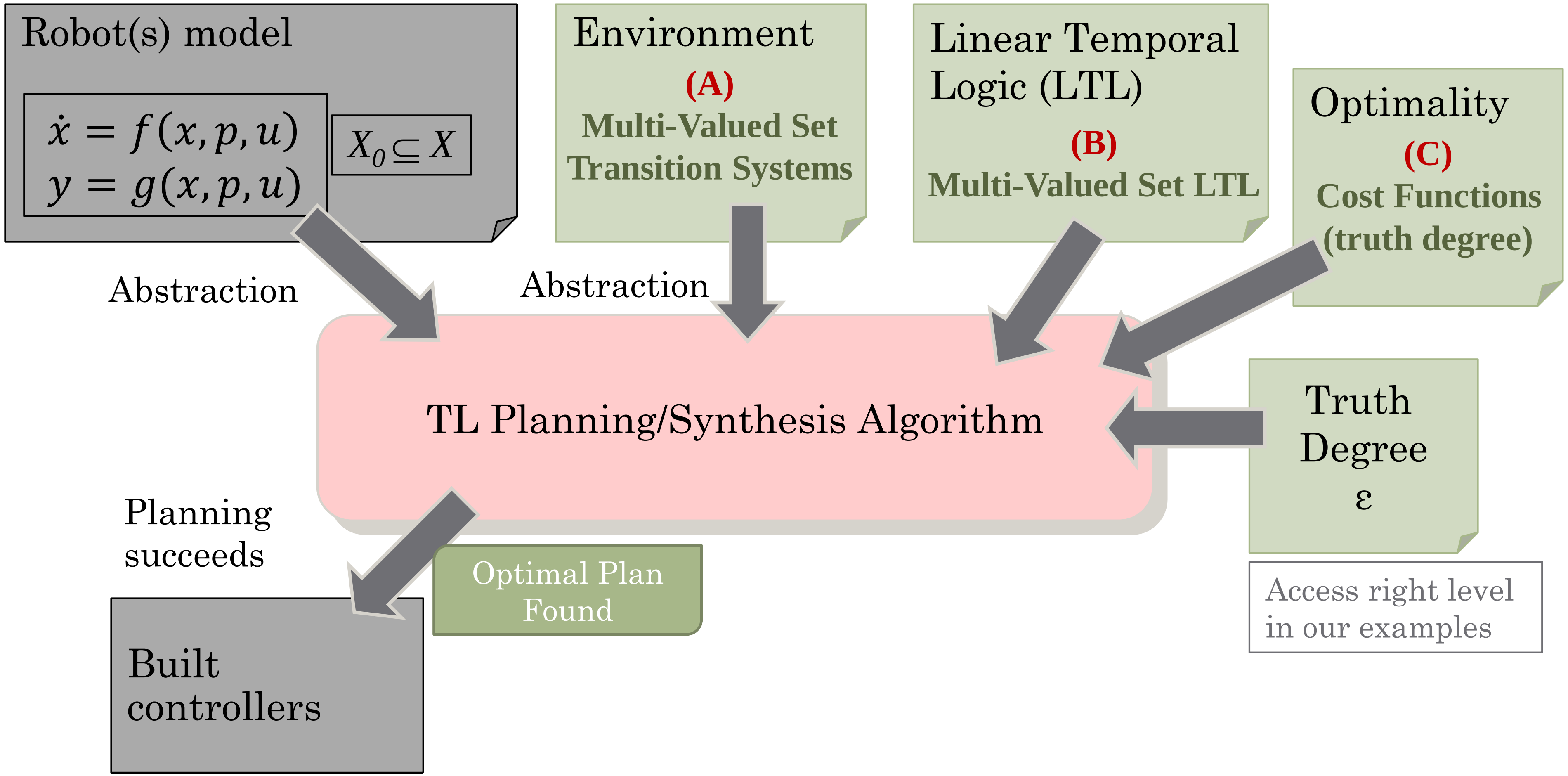
- Given
  - Some motion planning mission specifications for *n* robots or autonomous driving cars
  - Accessibility regulation of roads and access right levels of robots
  - A least required truth degree for satisfying plans (maximum violation degree tolerance)

- Global high-level path planner
  - Plan for *n* robots/cars located at *n* different locations to reach their destinations with least total cost while satisfying the mission specification requirements with an optimal truth degree (least possible violation of regulations)

*Summary of Our Contribution

- We proposed a framework for computing optimal plans based on different access right levels.
- We made new contribution to the automata theoretic multi-valued temporal logic planning problem.
- We provided modeling of uncertainty even in the truth degrees of the models themselves.
- We modeled and solved an optimal motion planning example for autonomous driving cars.

*Solution Architecture

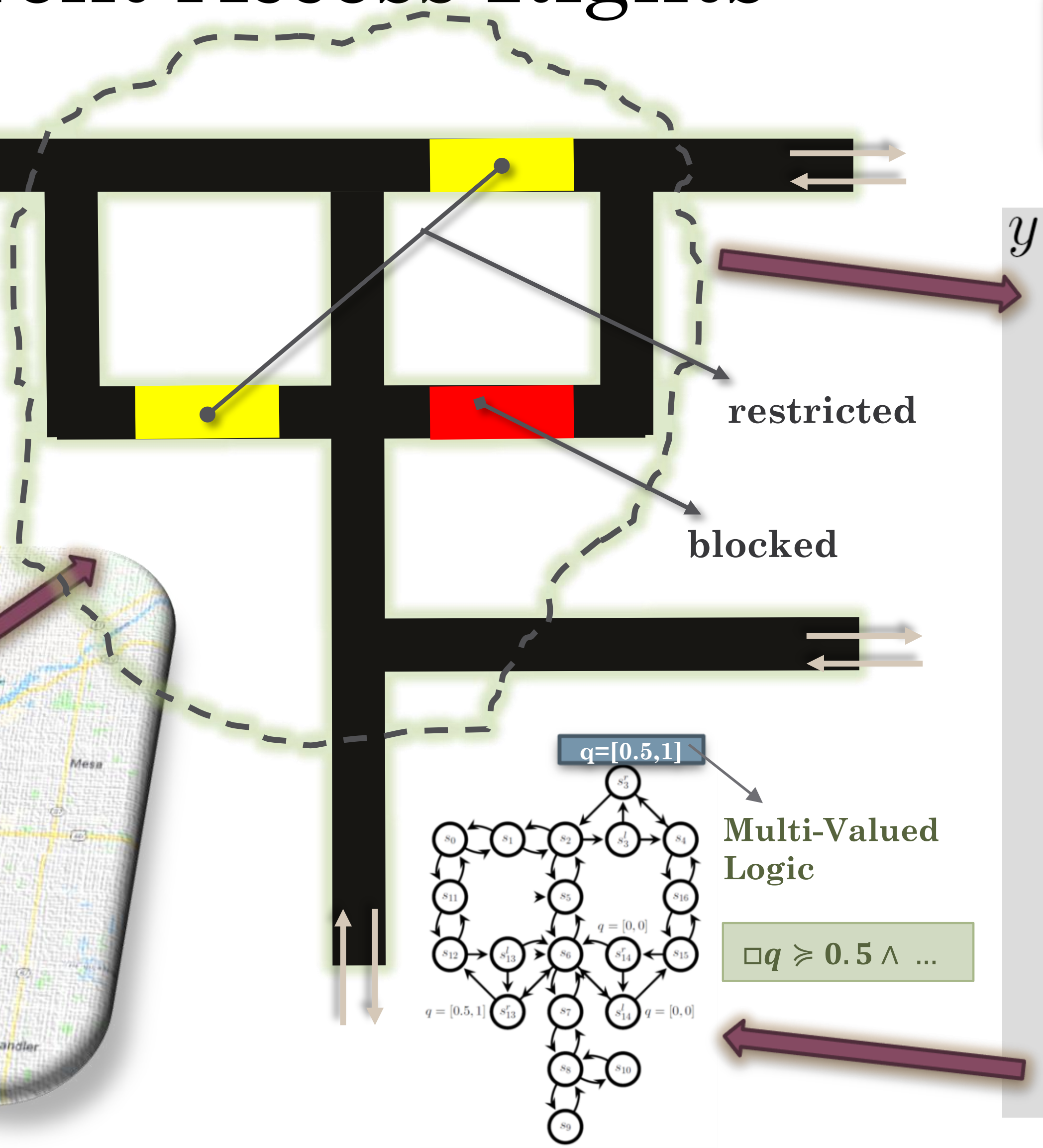


*Motivating Example: Path Planning for Vehicles with Different Access Rights

Access Right Level

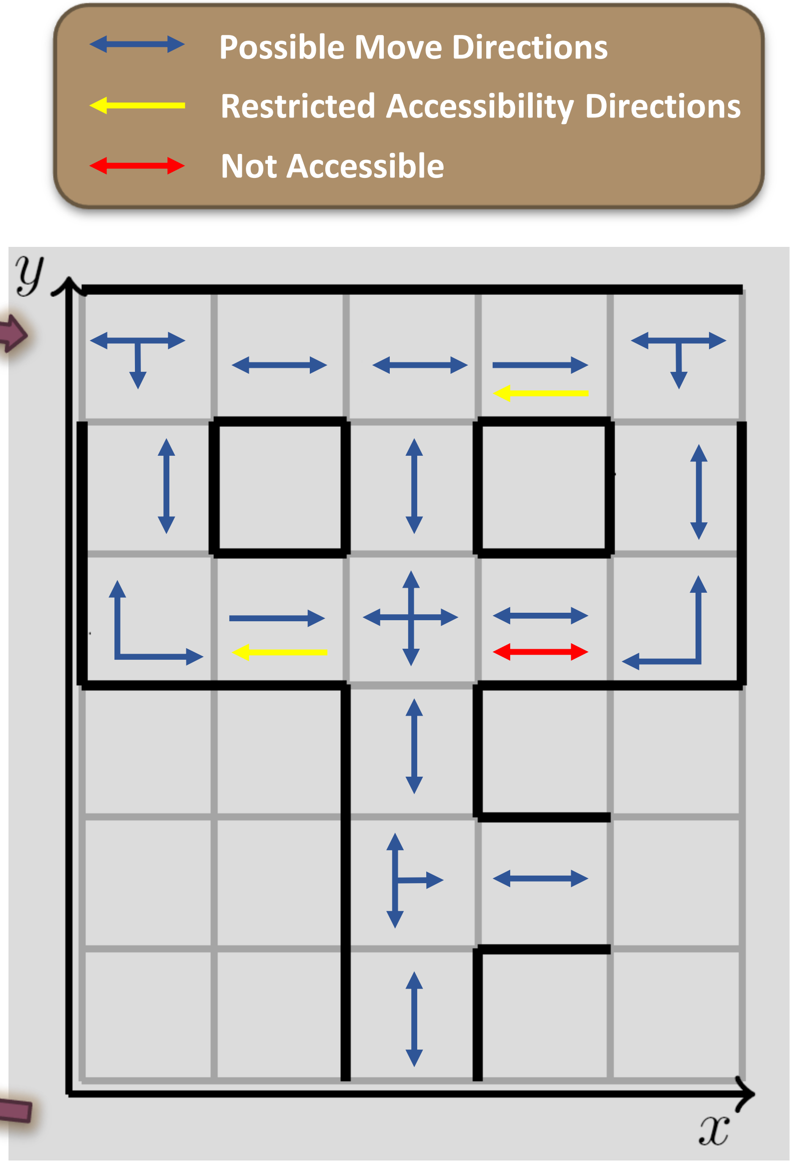
l_1
l_2
l_3

Roads are shared among:
- Citizen Vehicles
- Ambulances
- Firefighter Vehicles
- Police Vehicles
- Special Force Vehicles
- …



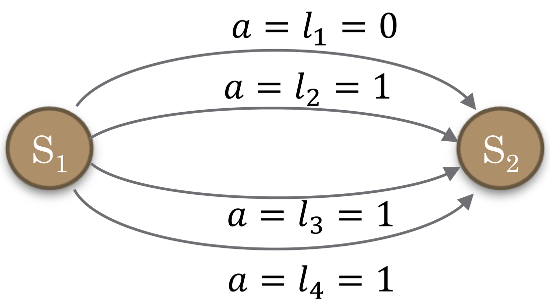
Multi-Valued Logic

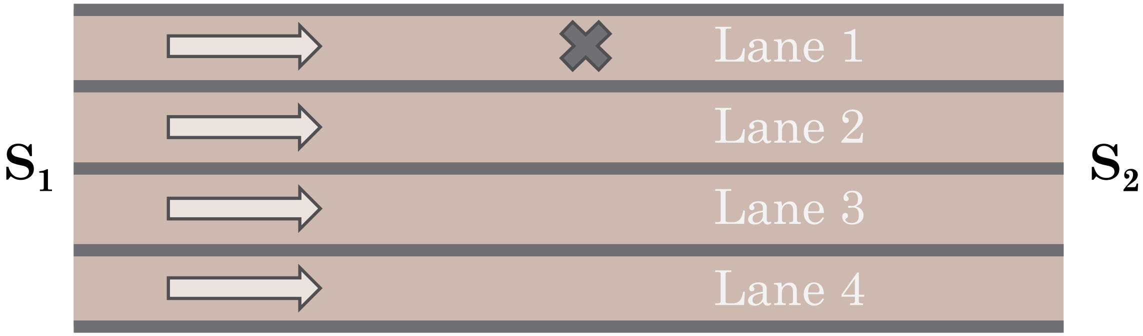
$\square q \geqslant 0.5 \wedge \ \ldots$



*Using Multi-Valued Logics (A)

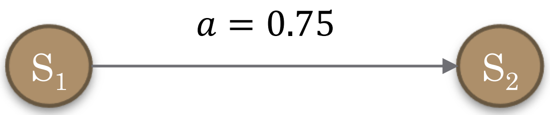
- a := Level of accessibility for a road

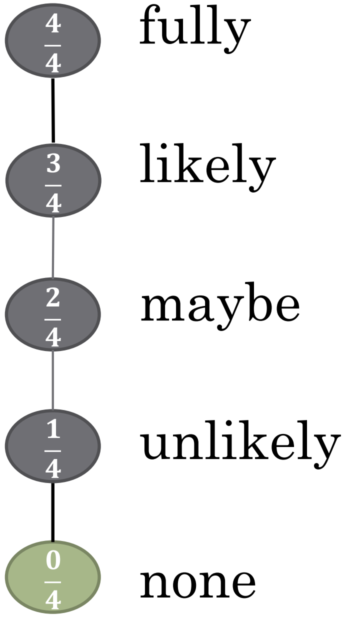




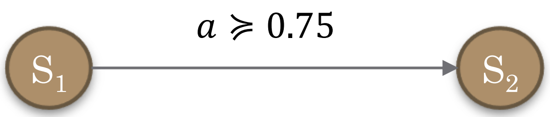
- This graph has too much information

- Abstract a 4-lane road as a single road



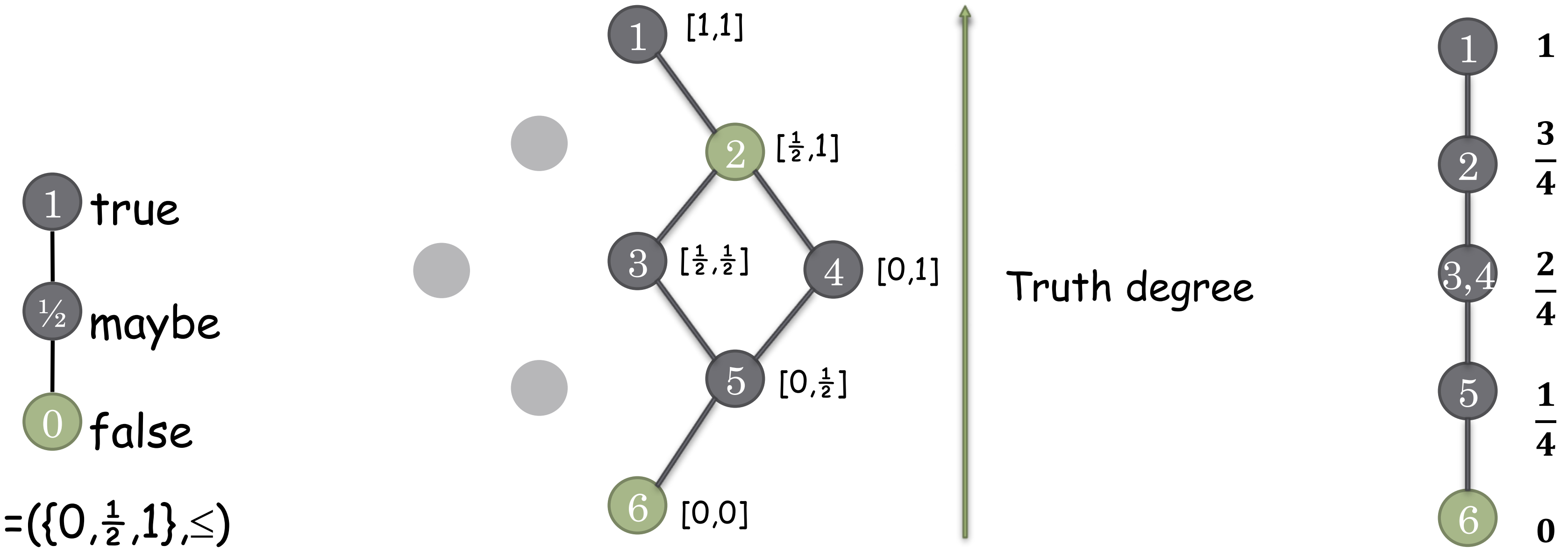


- Lane 1 is temporarily not accessible (uncertainty included)



The road's accessibility is **at least 0.75**
or equivalently
At least 3 out of four observations
reported the road is completely accessible

*Lattice of Interval Set (A)



$\mathcal{L}_3 = (\{0, \frac{1}{2}, 1\}, \leq)$

$\neg 1 = 0,\ \neg 0 = 1,\ \neg \frac{1}{2} = \frac{1}{2}$

$\mathcal{L}_3^I$

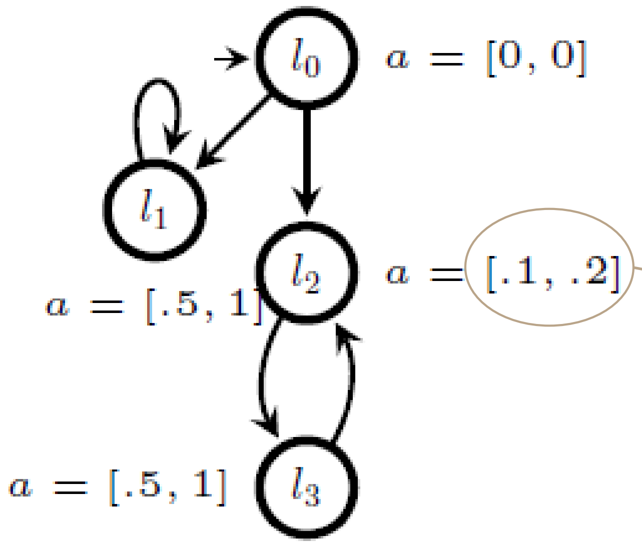
$\mathcal{L}_5 = (\{0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1\}, \leq)$

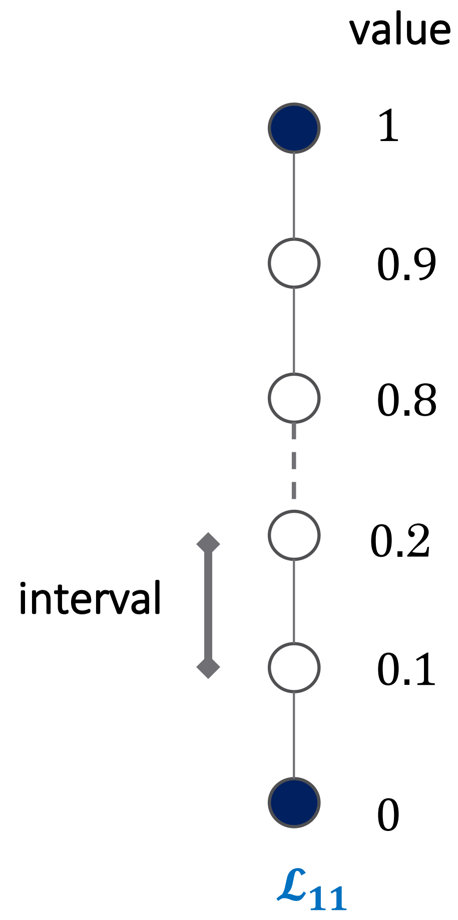
Totally Ordered quasi-Boolean Lattice

quasi-Boolean Lattice

Totally Ordered quasi-Boolean Lattice

MVS-Transition Systems (A)





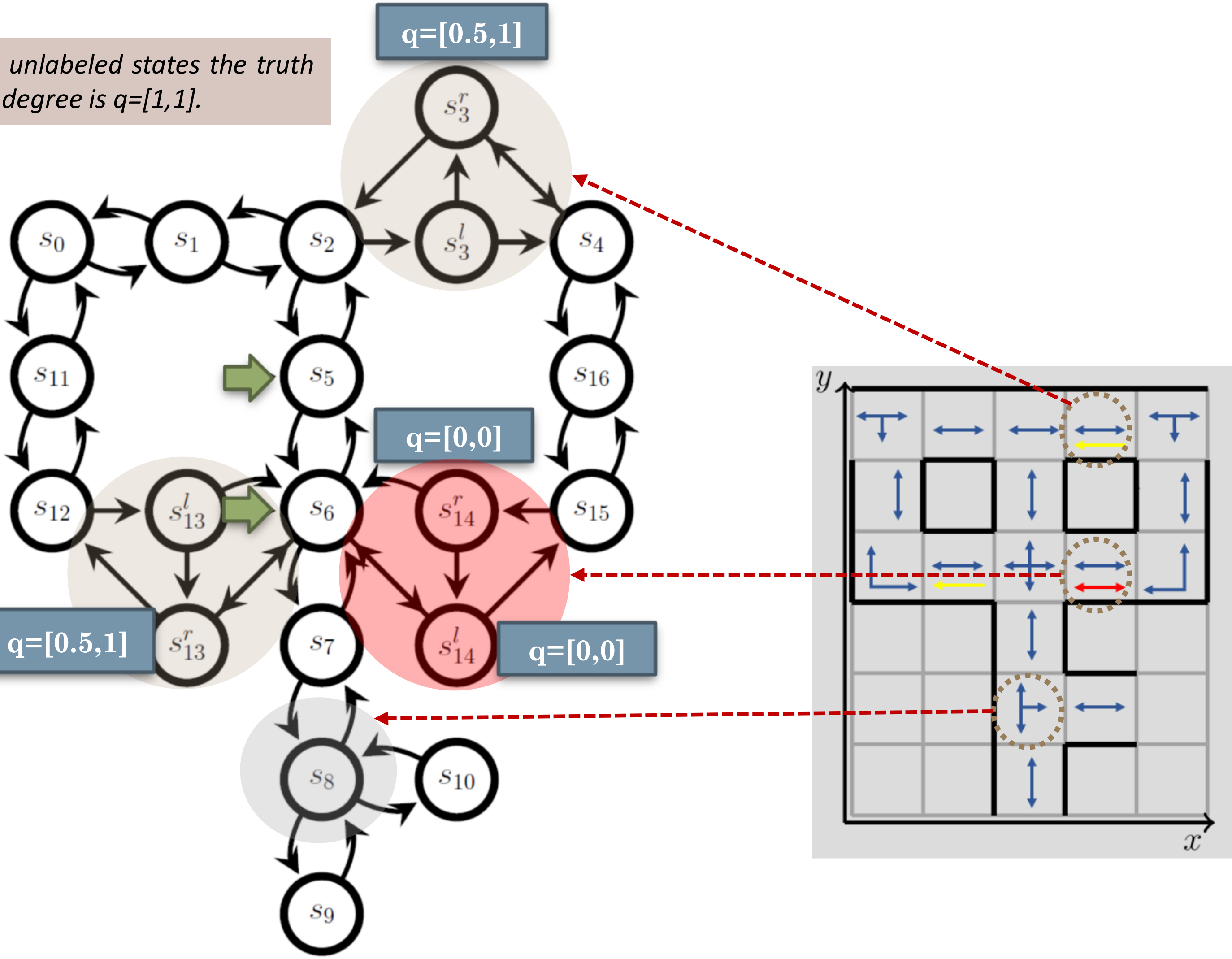
A trace $s: V \times \mathbb{N} \rightarrow L^I$

Path: $l_0, l_2, l_3, l_2, \ldots$

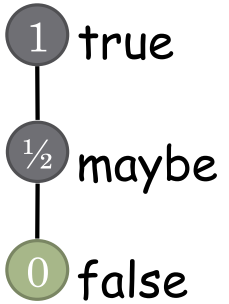
Trace: [0,0], [0.1,0.2], [0.5,1], [0.1,0.2], …

*Example-Modeling (A)

For all unlabeled states the truth degree is q=[1,1].



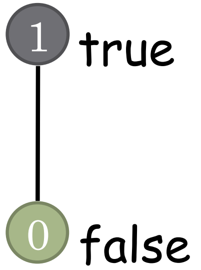
$\mathcal{B}_3$=({0,½,1},≤)



¬1=0, ¬0=1, ¬½=½

Used for accessibility of road sections

$\mathcal{B}_2$=({0,1},≤)



Used for locations of robots

mvs-LTL Syntax and Semantics (B)

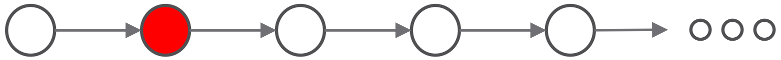
LTL = Linear(-time) Temporal Logic
Assume some interval variables
Syntax of LTL formulas ϕ:

Interval Expressions

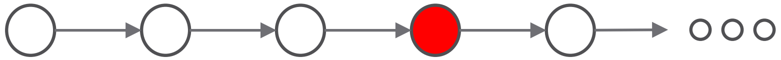
$$\boldsymbol{\phi} \ ::= \boxed{\boldsymbol{\pi \preccurlyeq l \mid \pi \succcurlyeq l}} \mid \boldsymbol{\phi \vee \phi \mid \phi \wedge \phi \mid \circ\, \phi \mid \lozenge\, \phi \mid \square \phi \mid \phi U \phi \mid \phi R \phi}$$

where $\pi \in Variables$ (no negation is allowed)

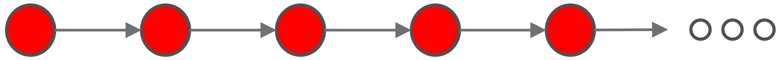
- $\circ\ \phi$ — holds if ϕ holds at the next position;



- $\lozenge\ \phi$ — holds if there exists a future position where ϕ holds;

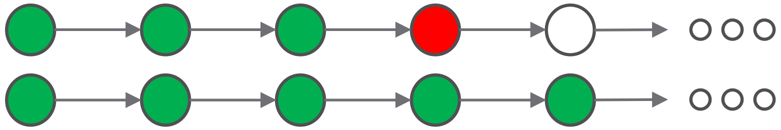


- $\square\ \phi$ — holds if, for all future positions, ϕ holds;



- ϕ U Ψ — holds if there is a future position where Ψ holds, and ϕ holds for all positions prior to that;

- ϕ R Ψ — holds if there is a future position where ϕ holds, or Ψ holds for all positions prior to that.



*mvs-LTL Syntax and Semantics (B)

Syntax of LTL formulas ϕ:

Interval Expressions

$$\boldsymbol{\phi} ::= \boxed{\pi \preccurlyeq l \mid \pi \succcurlyeq l} \mid \phi \vee \phi \mid \phi \wedge \phi \mid \circ\, \phi \mid \lozenge\, \phi \mid \square \phi \mid \phi U \phi \mid \phi R \phi$$

where $\pi \in Variables$ (no negation is allowed)

$$\neg(\pi \preccurlyeq l) = \pi \succ l$$
$$\neg(\pi \succcurlyeq l) = \pi \prec l$$

$$
\begin{aligned}
[\![\pi \preceq l]\!](\mu, i) &:= \Psi([\bot, l], \mu(i, \pi)) \quad \pi \in V,\ l \in L \\
[\![\pi \succeq l]\!](\mu, i) &:= \Psi([l, \top], \mu(i, \pi)) \quad \pi \in V,\ l \in L \\
[\![\phi_1 \vee \phi_2]\!](\mu, i) &:= [\![\phi_1]\!](\mu, i) \sqcup [\![\phi_2]\!](\mu, i) \\
[\![\phi_1 \wedge \phi_2]\!](\mu, i) &:= [\![\phi_1]\!](\mu, i) \sqcap [\![\phi_2]\!](\mu, i) \\
[\![\bigcirc \psi]\!](\mu, i) &:= [\![\psi]\!](\mu, i+1) \\
[\![\phi_1 \mathcal{U} \phi_2]\!](\mu, i) &:= \bigsqcup_{j \geq i} \left([\![\phi_2]\!](\mu, j) \sqcap \bigsqcap_{i \leq k < j} [\![\phi_1]\!](\mu, k)\right) \\
[\![\phi_1 \mathcal{R} \phi_2]\!](\mu, i) &:= \bigsqcap_{j \geq i} \left([\![\phi_1]\!](\mu, j) \sqcup \bigsqcup_{i \leq k < j} [\![\phi_2]\!](\mu, k)\right)
\end{aligned}
$$

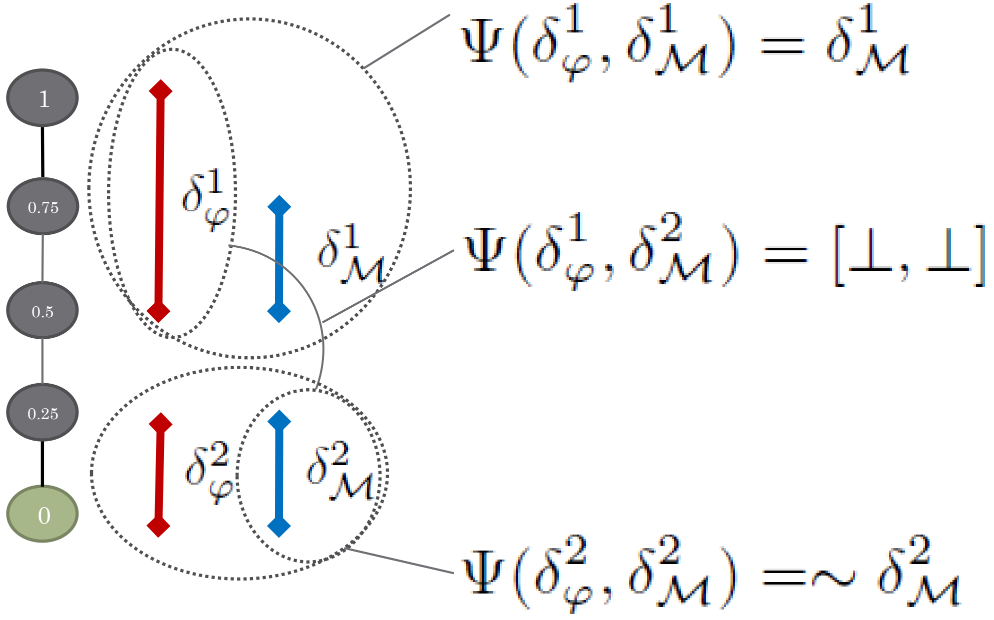
- Intuitively by $[\![\phi]\!](\mu, 0) = [l, h], l, h \in L$, we mean
  - The formula ϕ is satisfiable under trace μ with truth degree at least l, and at most h.
- Based on the above semantics, we define
  - $(eventually)\ \diamond\, \phi\ as\ (\top\ \mathcal{U}\ \phi), and\ (globally)\ \square\phi\ as\ (\bot\ \mathcal{R}\ \phi)$

*Satisfaction Evaluator Function (B)

$$\Psi : L^I \times L^I \rightarrow L^I$$

$$\Psi(\underbrace{[l_{base}, h_{base}]}_{\text{Base/Reference interval}}, \underbrace{[l, h]}_{\text{Source interval}}) = \begin{cases} [l, h] & \text{if } l_{base} \preceq l \text{ and } h_{base} = \top \\ \sim [l, h] & \text{if } l_{base} = \bot \text{ and } h \preceq h_{base} \prec \top \\ [\bot, \bot] & \text{otherwise} \end{cases}$$

- Ψ checks if the source interval is within the reference interval
  - If it was, the result is either the source interval or its negation
- This function is not symmetric!



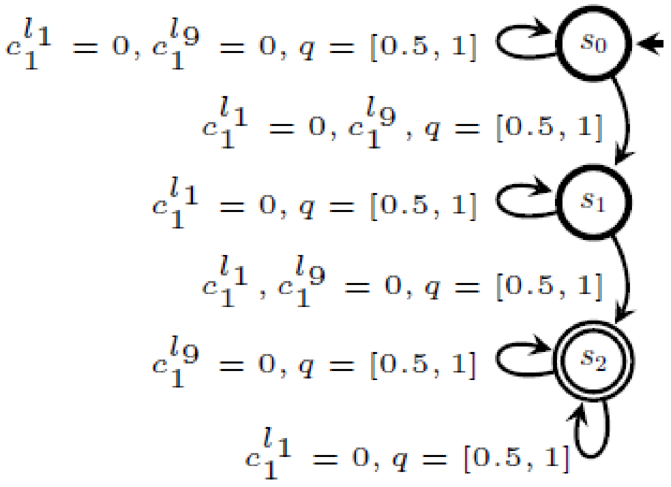
MVS-Automaton (B)

$$\phi_1 = \Box(q \succeq 0.5) \wedge \Diamond c_1^{l_1} \wedge \Diamond c_1^{l_9} \wedge (c_1^{l_1} \preceq 0)\,\mathcal{U} c_1^{l_9} \wedge \Box(c_1^{l_1} \preceq 0 \vee c_1^{l_9} \preceq 0)$$

Prepare* and Translate**

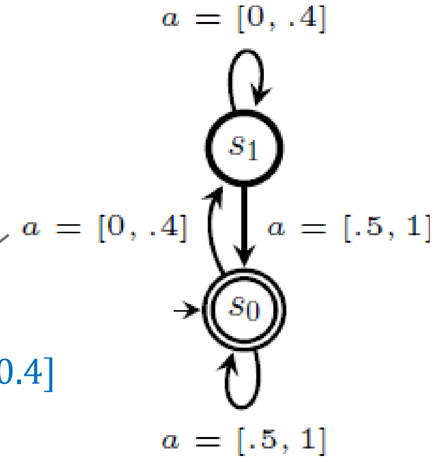
***Preprocessing**

1. Negative Normal Form
2. *$a \wedge \neg a \neq \bot$ and $a \vee \neg a \neq \top$*
3. Apply negation into interval expressions



Theorem:
The mvs-Automaton constructed for mvs-LTL formula $\boldsymbol{\phi}$*, assigns the same truth degree interval* $[l,h]$ *to a trace/word* $\boldsymbol{w}$ *that mvs-LTL semantics assigns to* $\boldsymbol{w}$ *for* $\boldsymbol{\phi}$

$$\bigsqcup_{p \in AR(\mathcal{A})} \bigsqcap_{i>0} w_i(\Delta(p_i, p_{i+1})) = [l, h] \iff [\![\phi]\!](w, 0) = [l, h].$$



$\bar{O}\,(s_0, s_1, \mathrm{a}) = [0,0.4]$

$\boldsymbol{\varphi = \Box\ \Diamond\ (a \succcurlyeq 0.5)}$

$\boldsymbol{\mathcal{L}_{11}}$

** **M. Chechik**, B. Devereux, and A. Gurfinkel, "*Model-checking infinite state-space systems with fine-grained abstractions using SPIN*," in 8th International SPIN Workshop, ser. LNCS, vol. 2057. Springer, 2001.

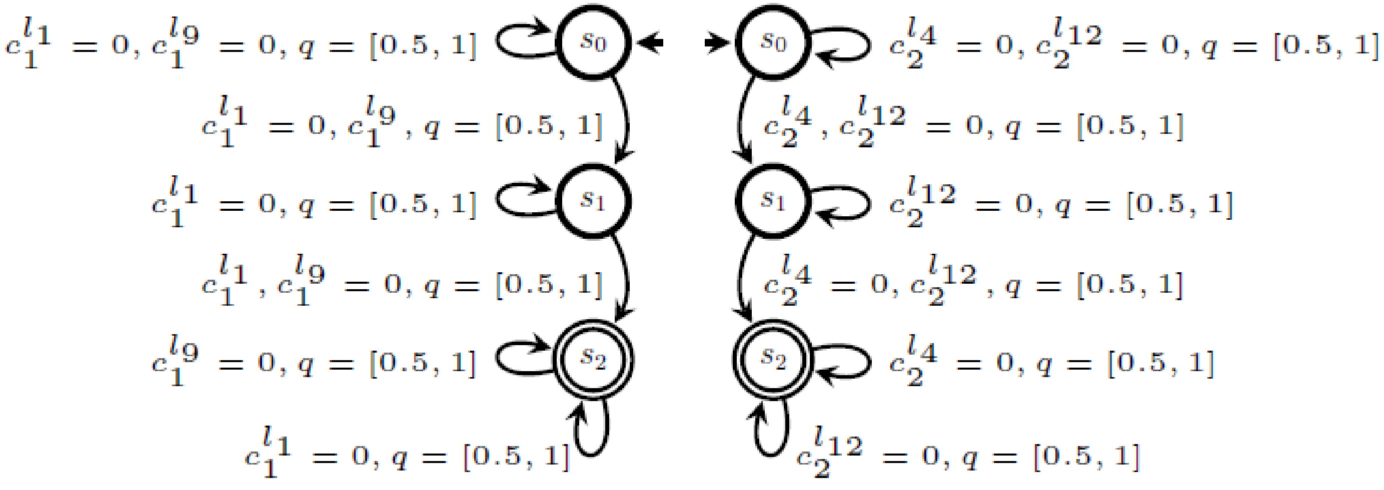
*Example-Specification Formula (B)

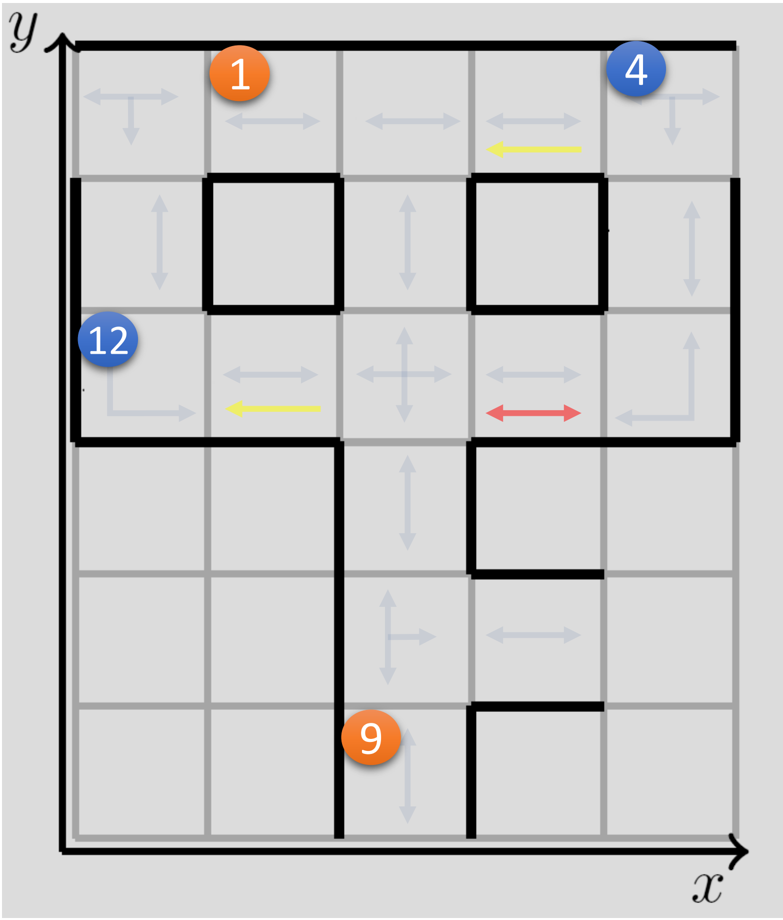
Φ_1 ≔ *always avoid entering not allowed blocks, and eventually visit 9, and then eventually visit 1 after visiting 9.*

Φ_2 ≔ *always avoid entering not allowed blocks, and eventually visit 4, and then eventually visit 12 after visiting 4.*

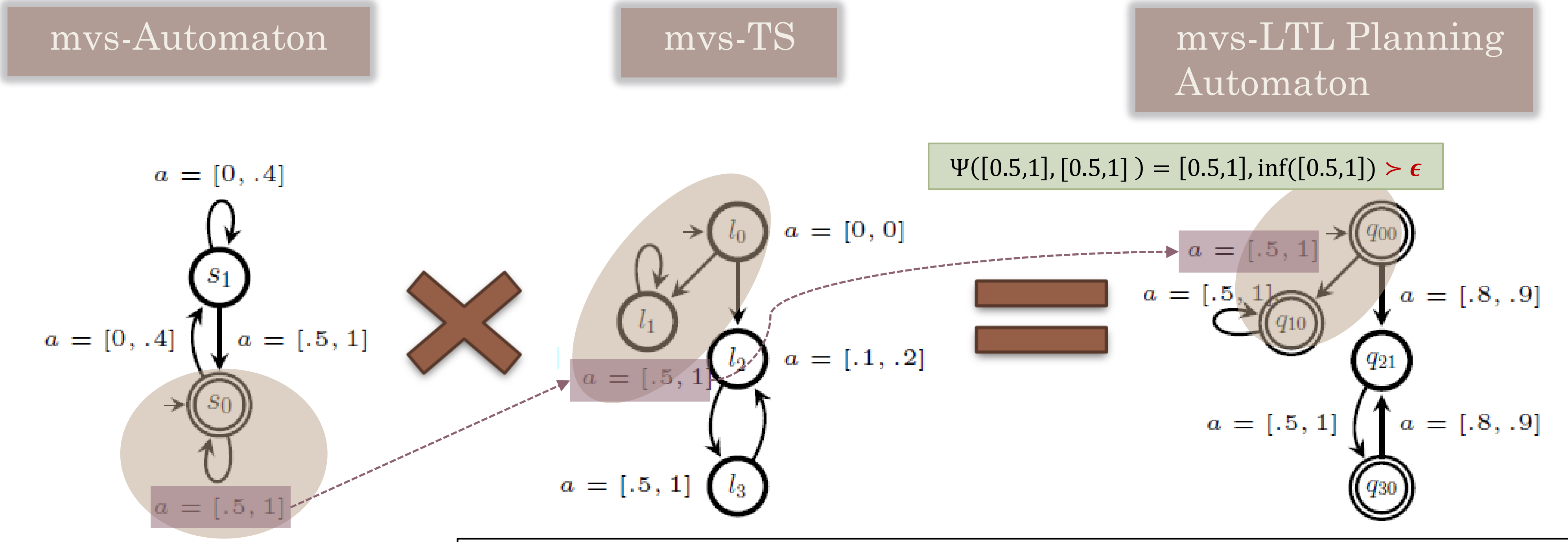
$$\phi_1 = \Box(q \succeq 0.5) \wedge \Diamond c_1^{l_1} \wedge \Diamond c_1^{l_9} \wedge (c_1^{l_1} \preceq 0)\,\mathcal{U} c_1^{l_9} \wedge \Box(c_1^{l_1} \preceq 0 \vee c_1^{l_9} \preceq 0)$$

$$\phi_2 = \Box(q \succeq 0.5) \wedge \Diamond c_2^{l_4} \wedge \Diamond c_2^{l_{12}} \wedge (c_2^{l_{12}} \preceq 0)\,\mathcal{U} c_2^{l_4} \wedge \Box(c_2^{l_4} \preceq 0 \vee c_2^{l_{12}} \preceq 0)$$





*mvs-LTL Planning Automaton (A,B) Example

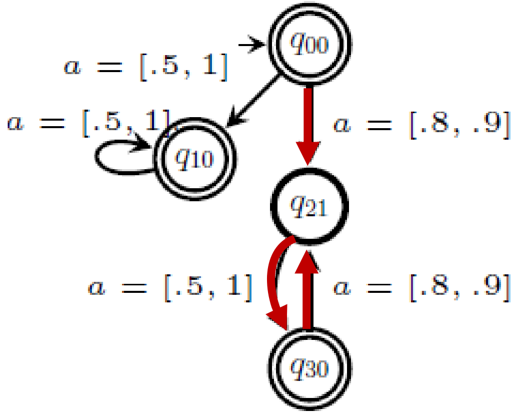


$$\varphi = \Box\Diamond(a \succeq 0.5)$$

Specification

- **Mvs-LTL planning automaton**
  - Can be computed **on-the-fly**
  - Has all the possible satisfactory plans with **a least truth degree ϵ**
  - The interval sets on the transitions can be converted to **costs/weights** for the purpose of finding optimal plans
  - Modified version of ***Dijkstra's Algorithm***, or ***A**** can be used for finding optimal plans

*Cost of Plans on mvs-LTL Planning Automaton (C)



$$p^2 \; : \; q_{00}, q_{21}, q_{30}, (q_{21}, q_{30})^{\omega}$$

traces

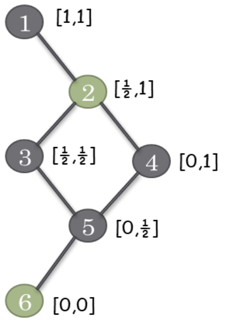
cost of a transition = cost of traverse + cost of violation

cost of violation = 1 − desirability of the minimum satisfaction degree (minimum capacity)

$$\boxed{\begin{aligned} \gamma_1(s_i, s_j) &= c_{i,j} + 1 - \lambda\big(\inf(\theta(s_i, s_j))\big), & \Delta(s_i, s_j) \neq \emptyset \\ \gamma_2(s_i, s_j) &= c_{i,j} + 1 - \lambda\big(\sup(\theta(s_i, s_j))\big), & \Delta(s_i, s_j) \neq \emptyset \end{aligned}}$$

$$\left.\begin{aligned} \boldsymbol{cost_{inf}(q_{00}, q_{21}) = 1 + 1 - \lambda(\inf([0.8, 0.9])) = 1.2} \\ \boldsymbol{cost_{inf}(q_{21}, q_{30}) = 1 + 1 - \lambda(\inf([0.5, 1])) = 1.5} \end{aligned}\right\} \text{Finite prefix}$$

$$\left.\begin{aligned} \boldsymbol{cost_{inf}(q_{30}, q_{21}) = 1 + 1 - \lambda(\inf([0.8, 0.9])) = 1.2} \\ \boldsymbol{cost_{inf}(q_{21}, q_{30}) = 1 + 1 - \lambda(\inf([0.5, 1])) = 1.5} \end{aligned}\right\} \text{Infinite executing cycle}$$



cost of violation increases

$$\boxed{\Gamma_1(p) = \sum_{0 \leq i < n} \gamma_1(\mu_i, \mu_{i+1}), \quad \Gamma_2(p) = \sum_{0 \leq i < n} \gamma_2(\mu_i, \mu_{i+1})}$$

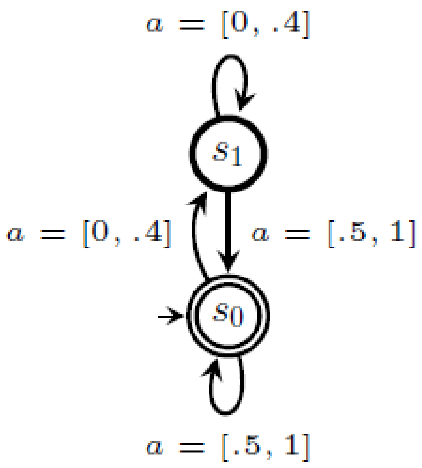
$$\boldsymbol{cost_{inf}(p^2) = 5.4}$$

Example (planning automaton + transition cost) (C)

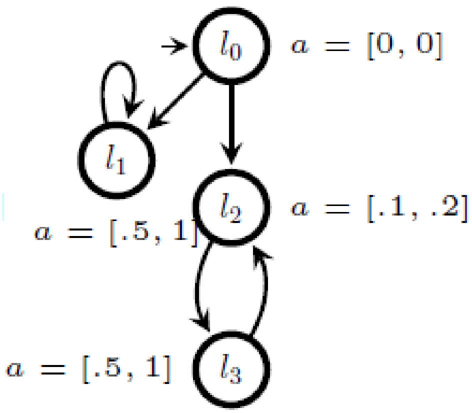
mvs-Automaton

mvs-TS

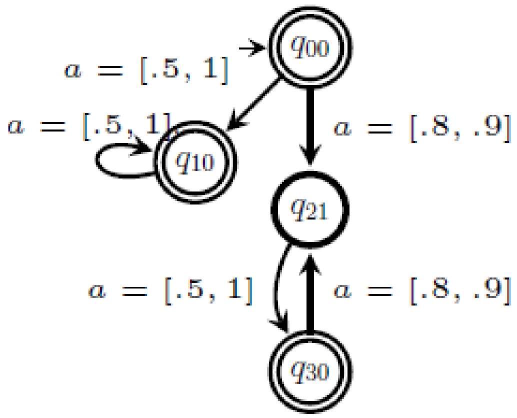
mvs-LTL Planning Automaton



×



=



$\varphi = \Box\Diamond(a \succeq 0.5)$

Specification

$p^1 : q_{00}, q_{10}, (q_{10})^\omega$

$p^2 : q_{00}, q_{21}, q_{30}, (q_{21}, q_{30})^\omega$

traces

$\Gamma_1(p^1) = 3.0,\ \Gamma_2(p^1) = 2$

$\Gamma_1(p^2) = 5.4,\ \Gamma_2(p^2) = 4.2$

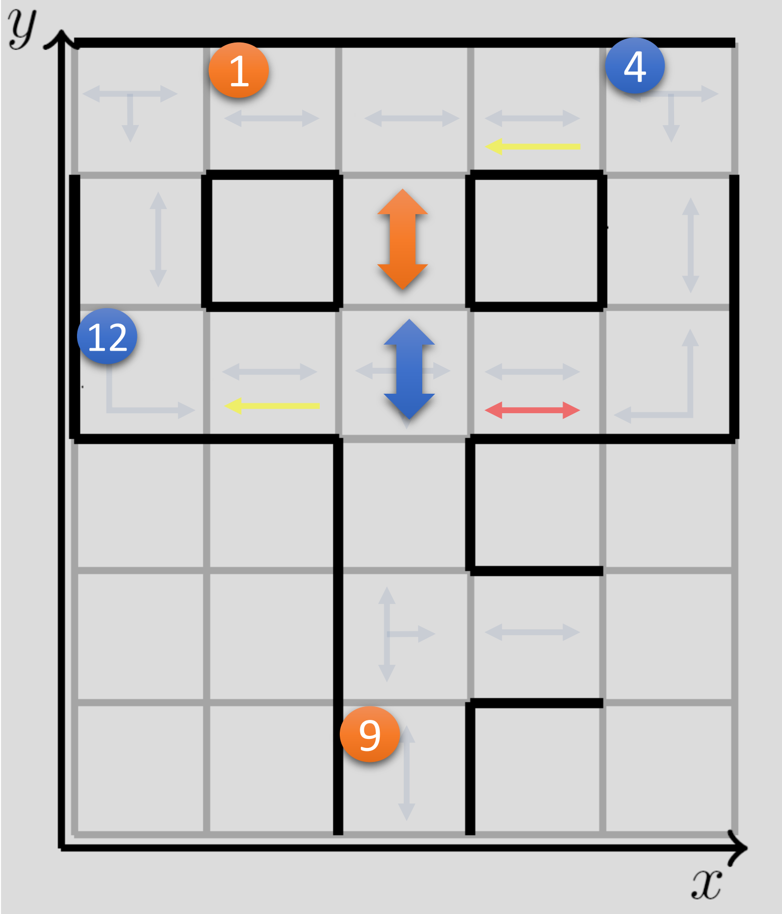
Primary and secondary costs

Running Example (A,B,C)

- *Modified version of Dijkstra's Algorithm is used for finding the optimal plans.*
- *The mvs-LTL Planning Automaton is constructed on-the-fly*

Cost Per Step, Per Car = *1*
Cost of Entering Restricted Blocks =
$$1 - \lambda(inf([0.5, 1])) = 0.5$$

Total Cost = 25+2
Steps = 12+1
Entering Restricted Blocks =
2 times (2*0.5 = 1 extra cost)



Conclusion

- New contribution to the automata theoretic multi-valued temporal logic planning problem.
  - We defined threshold based multi-valued temporal logic formulas interpreted over interval based multi-valued traces
  - The newly introduced threshold allows one to specify lower bound and higher bound truth degrees in the spirit of Signal Temporal Logic (SNL) [1].
- Modeling of uncertainty even in the truth degrees of the models themselves.
  - The interval based semantics for the execution traces of multi-valued Transition Systems (mv-TS)
- Use multi-valued logics to capture the difference degrees of permissibility for robot motions or actions in more general term.
- Define and solve an optimal temporal logic planning problem in the spirit of [2].

[1] E. Bartocci, J. Deshmukh, A. Donz´e, G. Fainekos, O. Maler, D. Niˇckovi´c, and S. Sankaranarayanan, "*Specification-based monitoring of cyber-physical systems: a survey on theory, tools and applications*," in Lectures on Runtime Verification. Springer, 2018, pp. 135–175.

[2] S. L. Smith, J. Tumova, C. Belta, and D. Rus, "*Optimal path planning for surveillance with temporal-logic constraints,*" The International Journal of Robotics Research, vol. 30, pp. 1695–1708, 2011.

Thank You!

Acknowledgement:
This work was partially supported by the NSF award CPS 1446730, the NSF I/UCRC Center for Embedded Systems, and the NSF grant 1361926.