Computer Lab 3
Minimum Violation Planning

Nok Wongpiromsarn
UT Austin/Iowa State

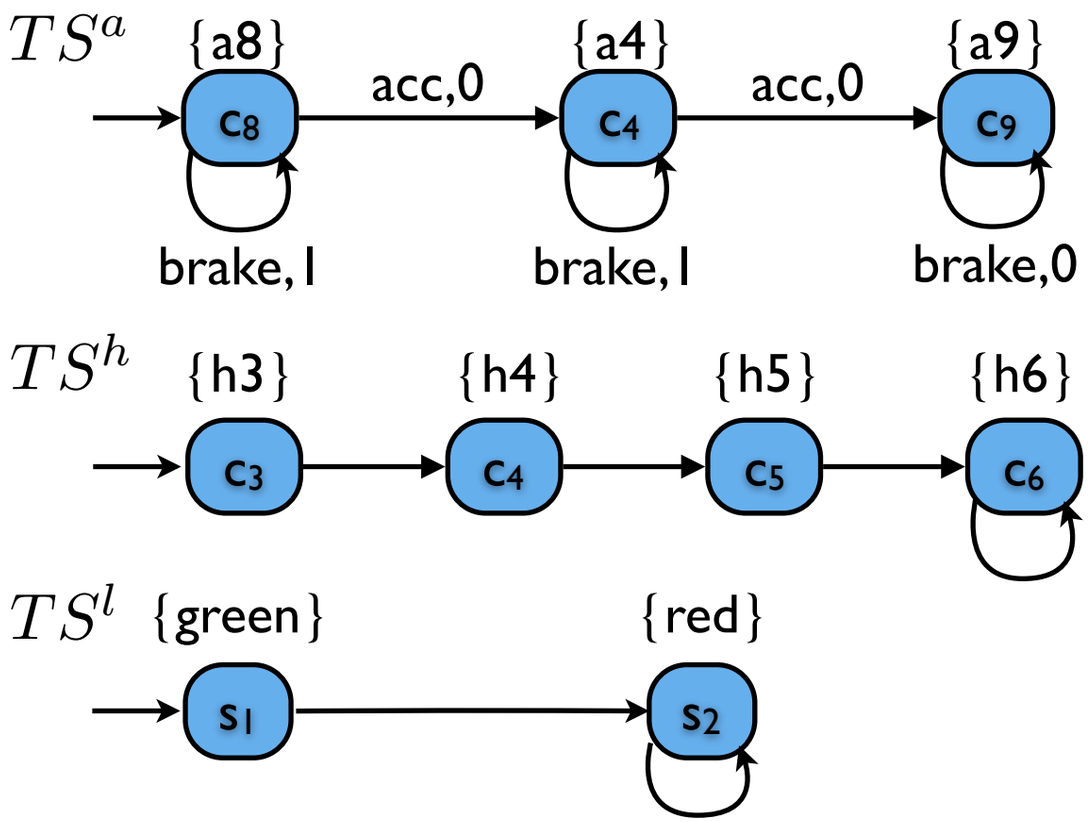
Richard M. Murray
Caltech

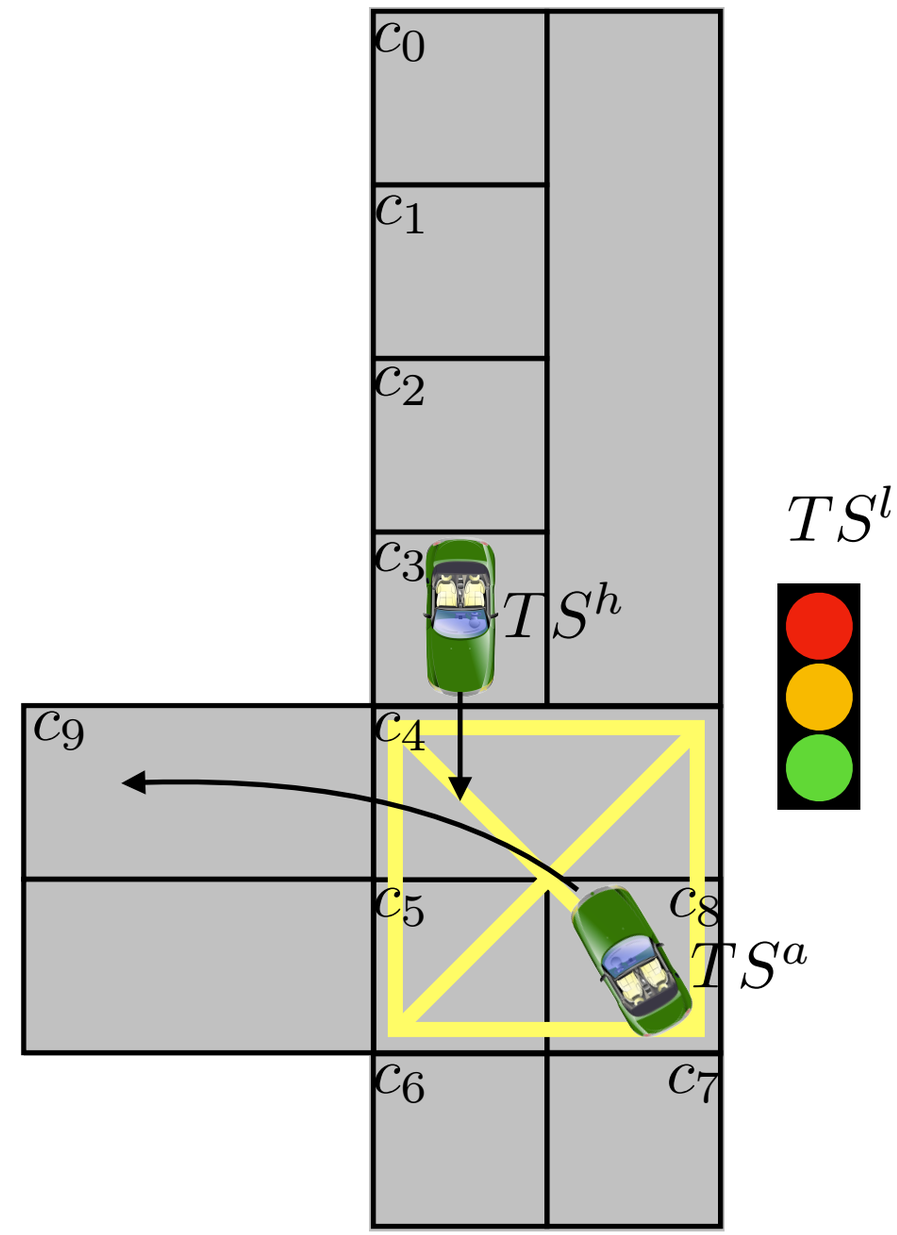
EECI-IGSC, 12 March 2020

Outline

- Defining systems and prioritized safety specification
- Solving minimum violation planning problem with TuLiP

Unprotected left turn





$$\varphi_1 = \Box\neg(\mathrm{h4} \wedge \mathrm{a4})$$

$$\varphi_2 = \Box\neg\big(\mathrm{red} \wedge (\mathrm{a8} \vee \mathrm{a4})\big)$$

$$\Psi = (\{\varphi_1\}, \{\varphi_2\}),\ \varpi(\varphi_1) = \varpi(\varphi_2) = 1$$

Weighted finite transition systems (WeightedKripkeStructure)

$$WTS = (S, R, s_{init}, AP, L, W)$$

S →
s_{init} →
R, W →
AP →
L →

```
states_a = {"c8", "c4", "c9"}
transitions_a = {
    ("c8", "c8", 1),
    ("c8", "c4", 0),
    ("c4", "c4", 1),
    ("c4", "c9", 0),
    ("c9", "c9", 0),
}
init_a = "c8"

ts_a = WeightedKripkeStructure()
ts_a.states.add_from(states_a)
ts_a.states.initial.add(init_a)
for transition in transitions_a:
    ts_a.transitions.add(
                            transition[0], transition[1], {"cost":
                                transition[2]},
                            )
for s in states_a:
    ap = "a" + s[1:]
    ts_a.atomic_propositions.add(ap)
    ts_a.states[s]["ap"] = {ap}
```

Weighted nondeterministic finite automaton (WeightedFiniteAutomaton)

$$\mathcal{A} = (Q, \Sigma, \delta, Q_0, F, W)$$

Q →
Q_0 →
F →

AP →

δ, W →

```
fa1 = WFA()
fa1.atomic_propositions.add_from(ts.atomic_propositions)
fa1.states.add_from({"q0"})
fa1.states.initial.add("q0")
fa1.states.accepting.add("q0")

# To define the transition !(h4 & a4), we define 2 sets:
#   * ap_without_h4 contains all the atomic propositions except 'h4'
#   * ap_without_a4 contains all the atomic propositions except 'a4'
# The subset of 2^{AP} corresponding to !(h4 & a4) is the union of
# PowerSet(ap_without_h4) and PowerSet(ap_without_a4).
ap_without_h4 = copy.deepcopy(fa1.atomic_propositions)
ap_without_h4.remove("h4")
ap_without_a4 = copy.deepcopy(fa1.atomic_propositions)
ap_without_a4.remove("a4")
transition_letters = set(PowerSet(ap_without_h4)) |
    set(PowerSet(ap_without_a4))
for letter in transition_letters:
    fa1.transitions.add("q0", "q0", letter=letter)
```

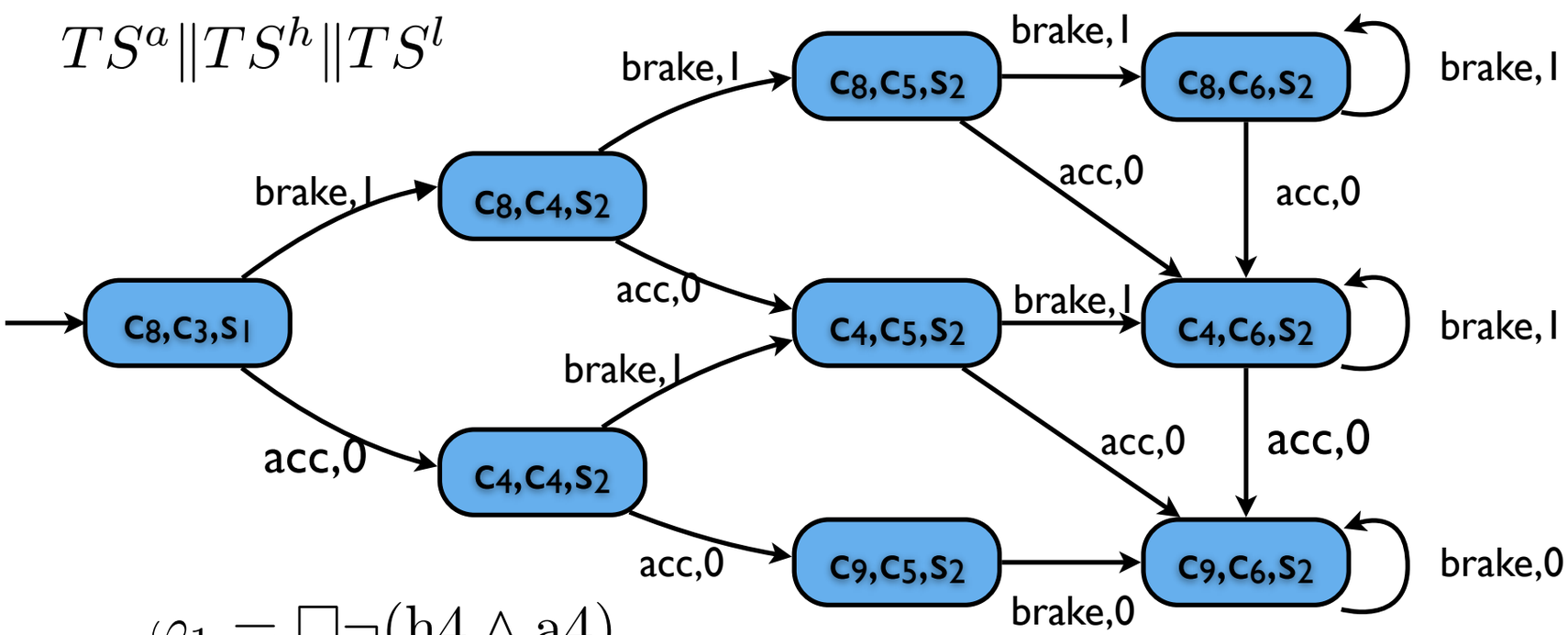
Prioritized safety specification

$$\mathcal{P} = (AP, \Phi, \Psi, \varpi)$$

```
spec = PrioritizedSpecification()
spec.add_rule(fa1, priority=1, level=0)
spec.add_rule(fa2, priority=1, level=1)
```

ϖ Ψ

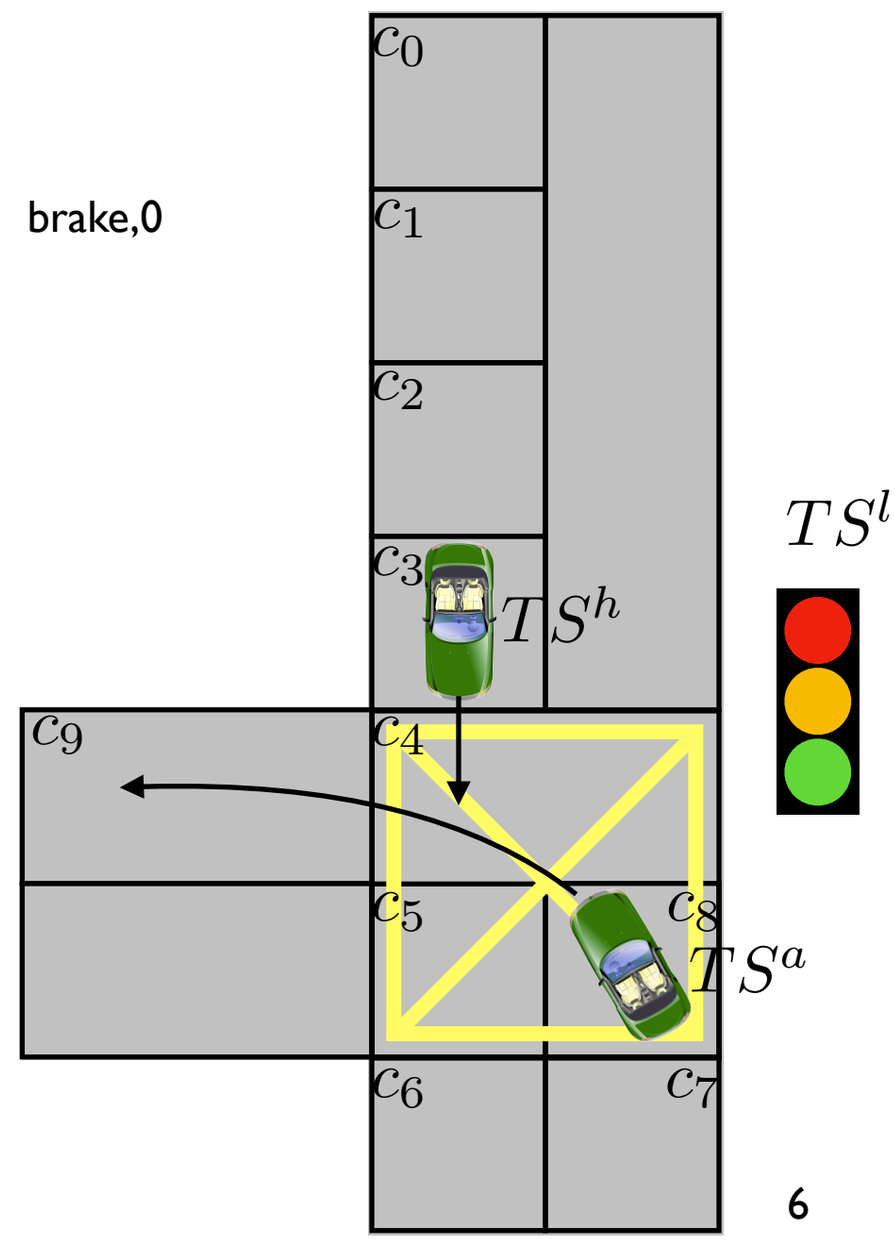
Exercise 1: Weighted Finite Automaton for φ_2



$$\varphi_1 = \Box\neg(\mathrm{h4} \wedge \mathrm{a4})$$

$$\varphi_2 = \Box\neg\big(\mathrm{red} \wedge (\mathrm{a8} \vee \mathrm{a4})\big)$$

$$\Psi = (\{\varphi_1\}, \{\varphi_2\}),\ \varpi(\varphi_1) = \varpi(\varphi_2) = 1$$



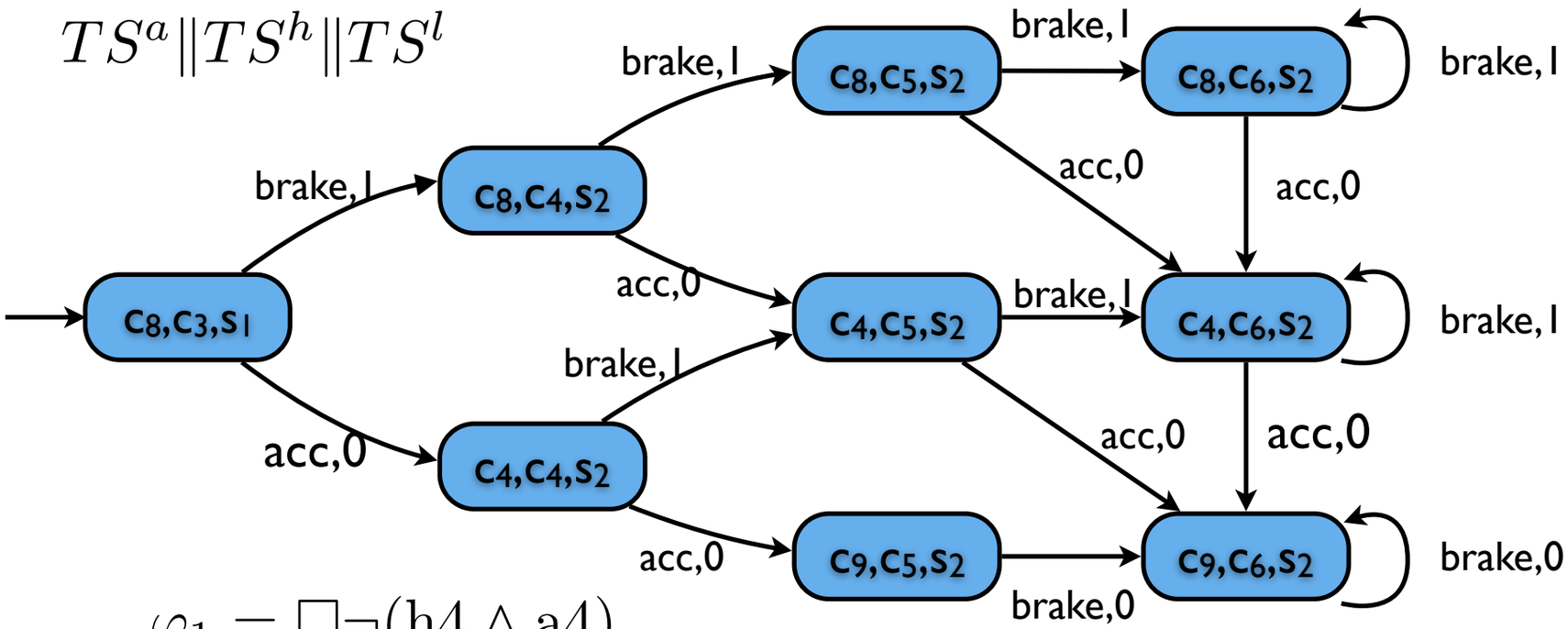
Ex: Modify

`c3-mvp/mvp.py`

to include φ_2 in the prioritized safety specification.

Exercise 2: Changing cost structure

$TS^a \| TS^h \| TS^l$



$$\varphi_1 = \Box\neg(\mathrm{h4} \wedge \mathrm{a4})$$

$$\varphi_2 = \Box\neg\big(\mathrm{red} \wedge (\mathrm{a8} \vee \mathrm{a4})\big)$$

$$\Psi = (\{\varphi_1\}, \{\varphi_2\}),\ \varpi(\varphi_1) = \varpi(\varphi_2) = 1$$

Ex: What happen if $\Psi = (\{\varphi_1, \varphi_2\})$?