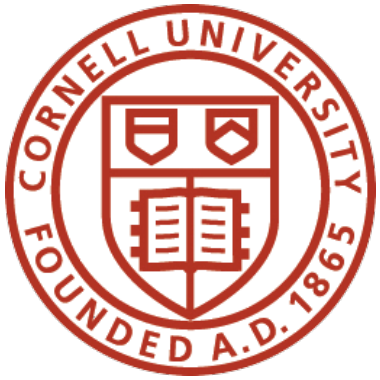
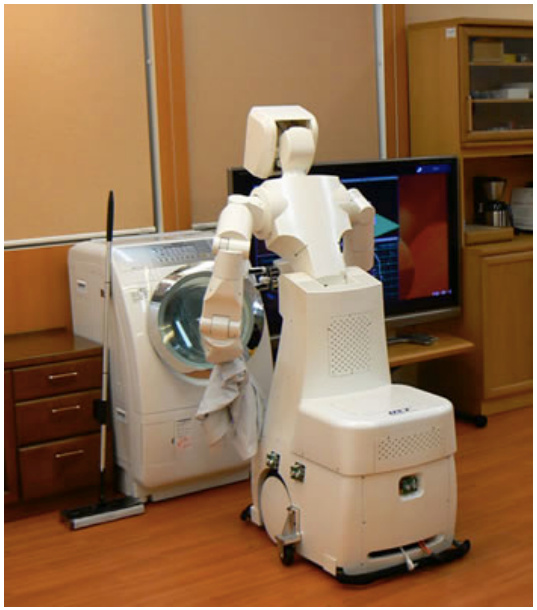
Temporal Logic Robot Mission Planning for Slow and Fast Actions



Vasu Raman
Cameron Finucane
Hadas Kress-Gazit

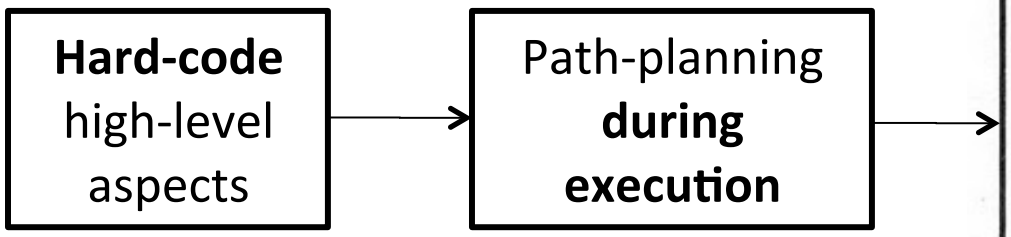
Cornell University

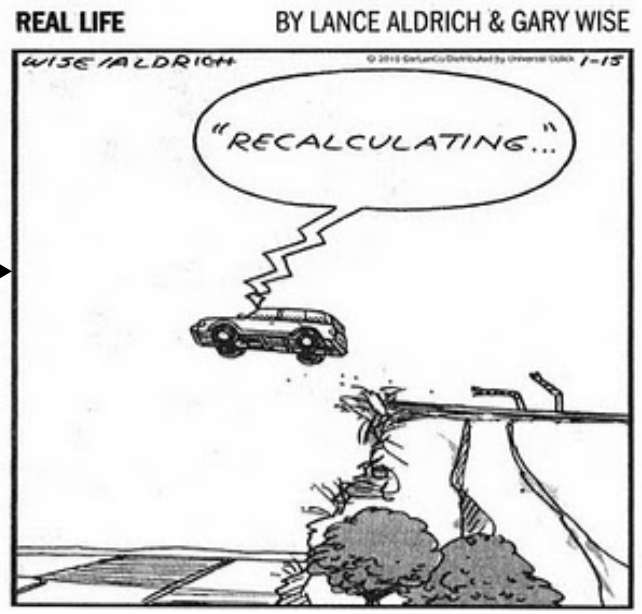
High-Level Tasks



Images: http://www.popsci.com/files/imagecache/article_image_large/articles/beerbot.jpg
http://www.news.cornell.edu/stories/Nov07/DarpaCar.jpg
http://www.technovelgy.com/graphics/content08/toyota-robot-maid-laundry.jpg

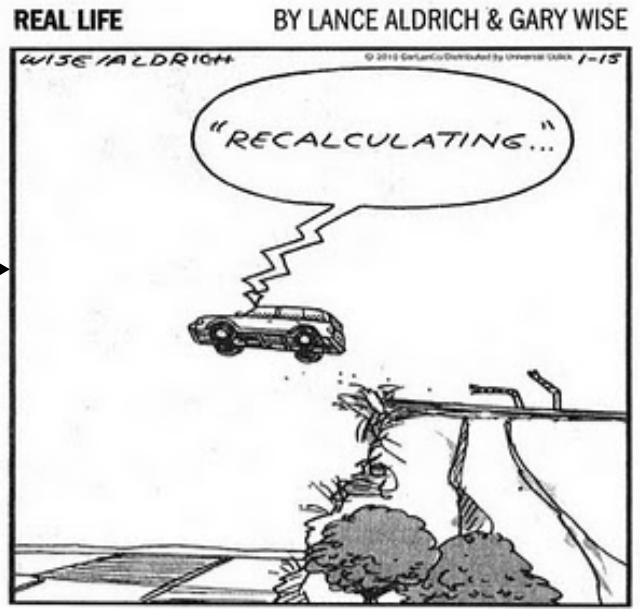
Usual Approach





Usual Approach

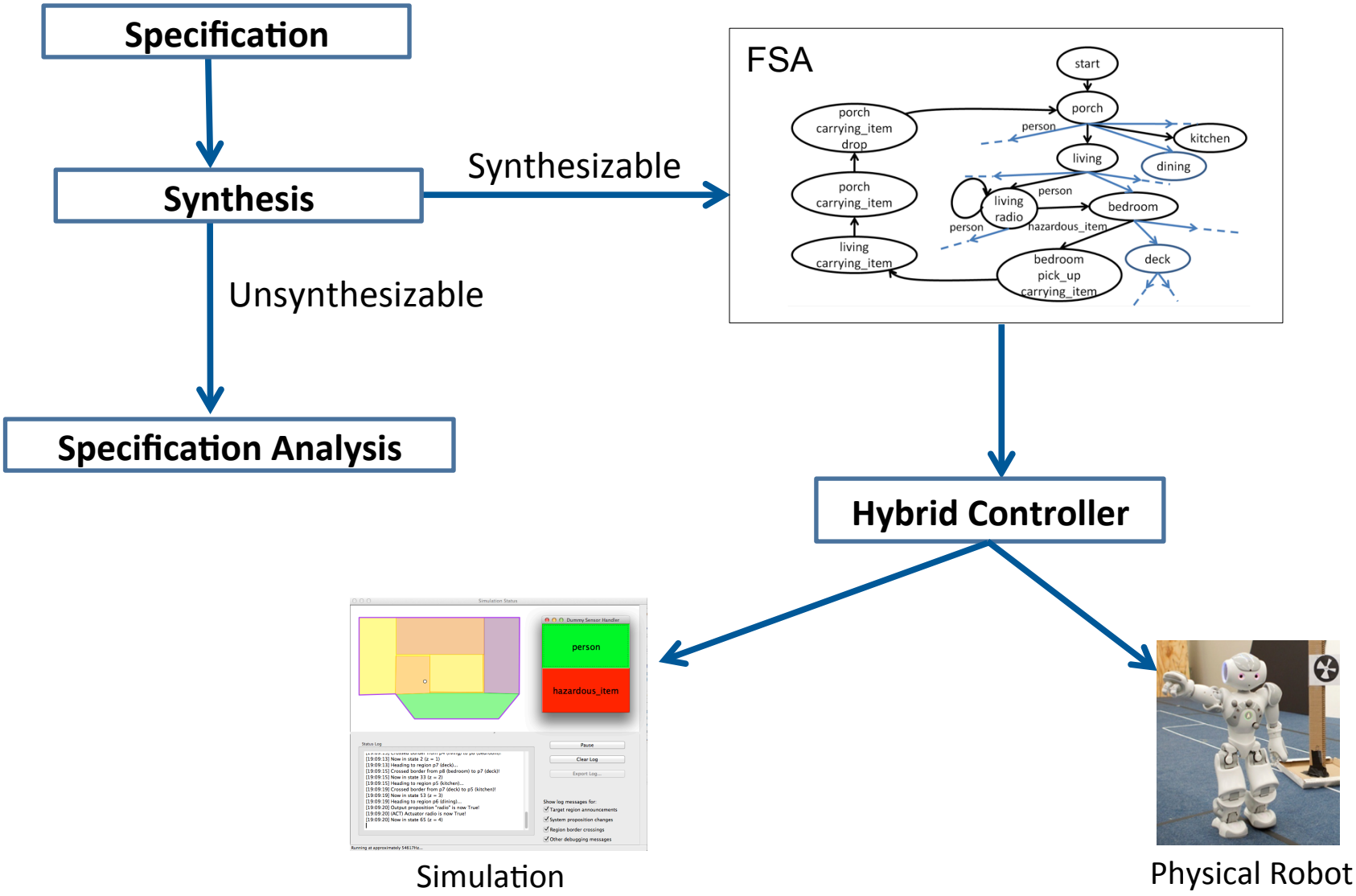
Hard-code high-level aspects → Path-planning **during execution** →



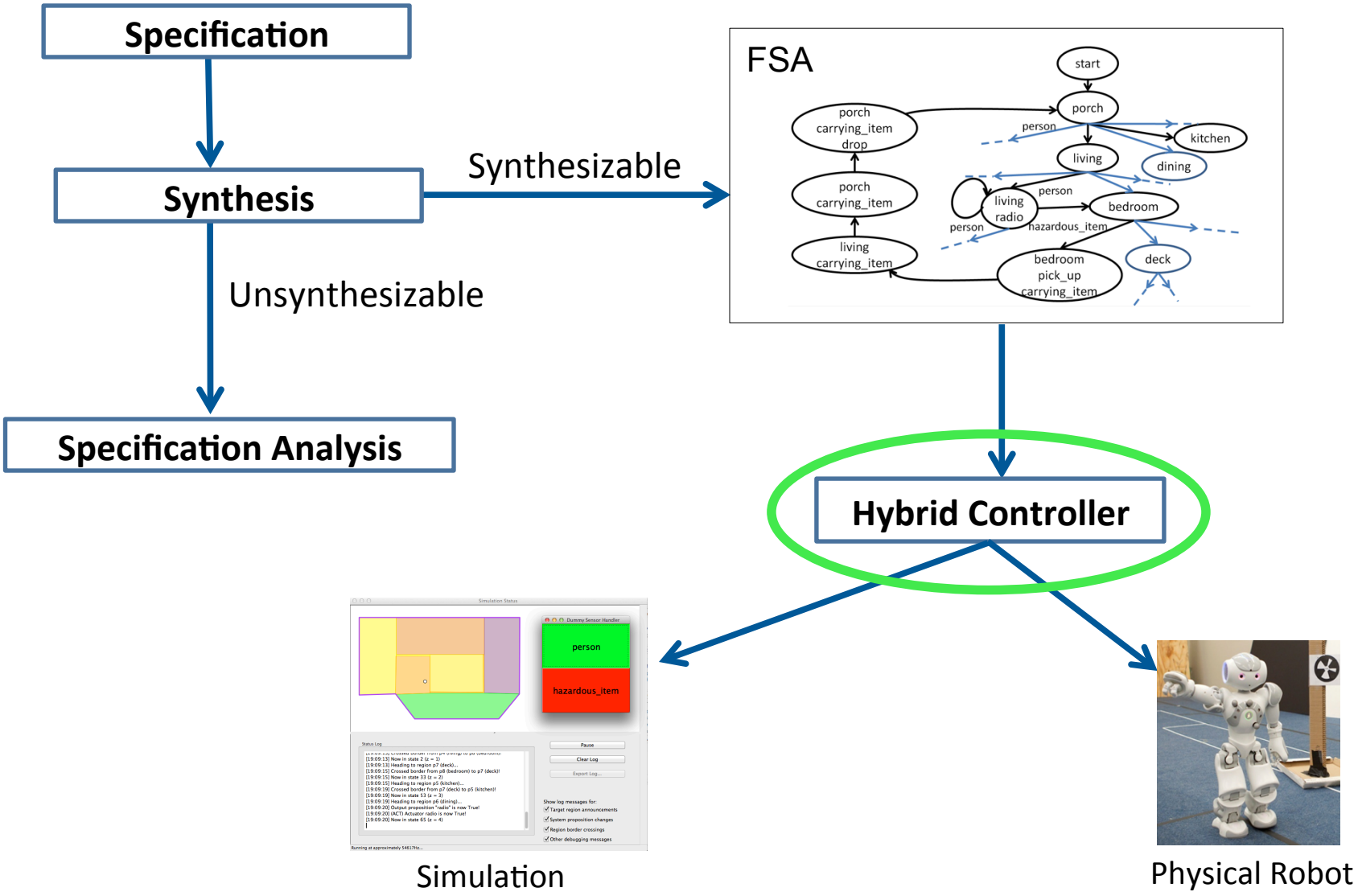
Recent Approaches – formal methods

Create **abstraction** of problem → Synthesize **correct** discrete solution → **Continuously** implement discrete solution

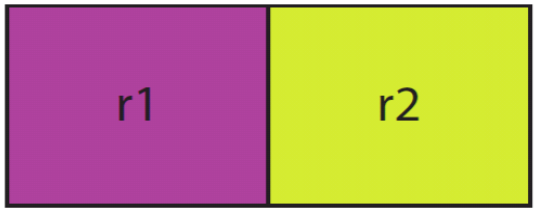
LTLMoP Toolkit Overview



LTLMoP Toolkit Overview



Example: fast camera, slow motion



- **Robot starts in region r1 with the camera off**

$$\varphi_{r_1} \wedge \neg \pi_{camera}$$

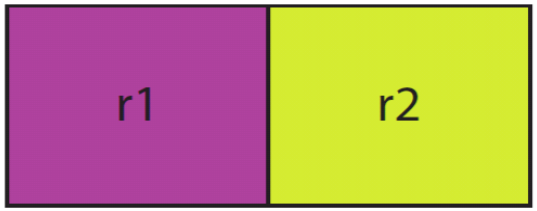
- **Activate the camera if and only if you see a person**

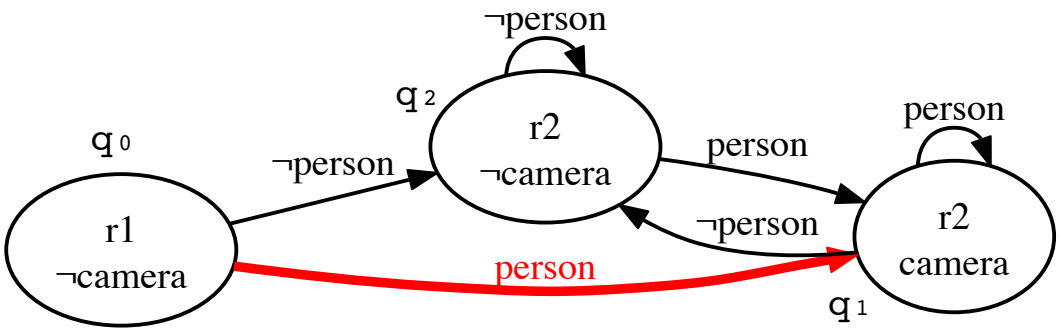
$$\wedge \, \square(\bigcirc \pi_{person} \Leftrightarrow \bigcirc \pi_{camera})$$

- **Go to r2 infinitely often**

$$\wedge \, \square \diamondsuit (\varphi_{r_2})$$

Example: fast camera, slow motion





- **Robot starts in region r1 with the camera off**

$$\varphi_{r_1} \wedge \neg \pi_{camera}$$

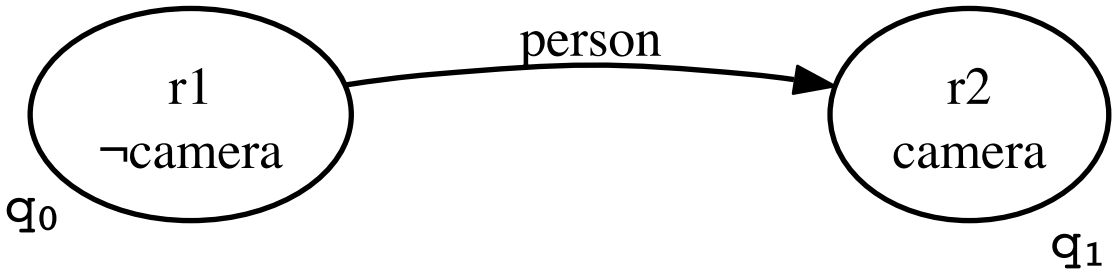
- **Activate the camera if and only if you see a person**

$$\wedge\ \square(\bigcirc \pi_{person} \Leftrightarrow \bigcirc \pi_{camera})$$

- **Go to r2 infinitely often**

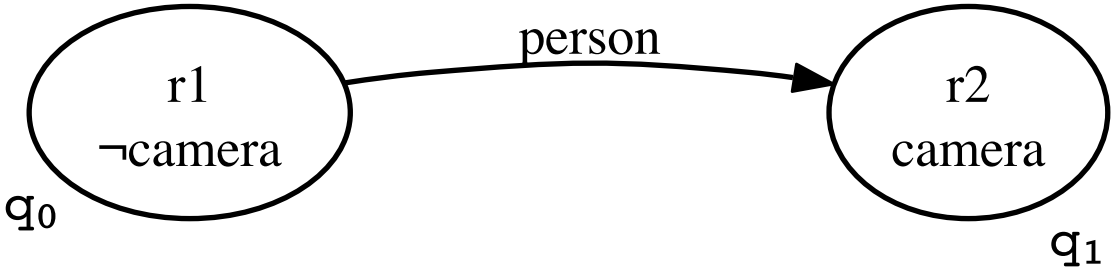
$$\wedge\ \square\diamondsuit(\varphi_{r_2})$$

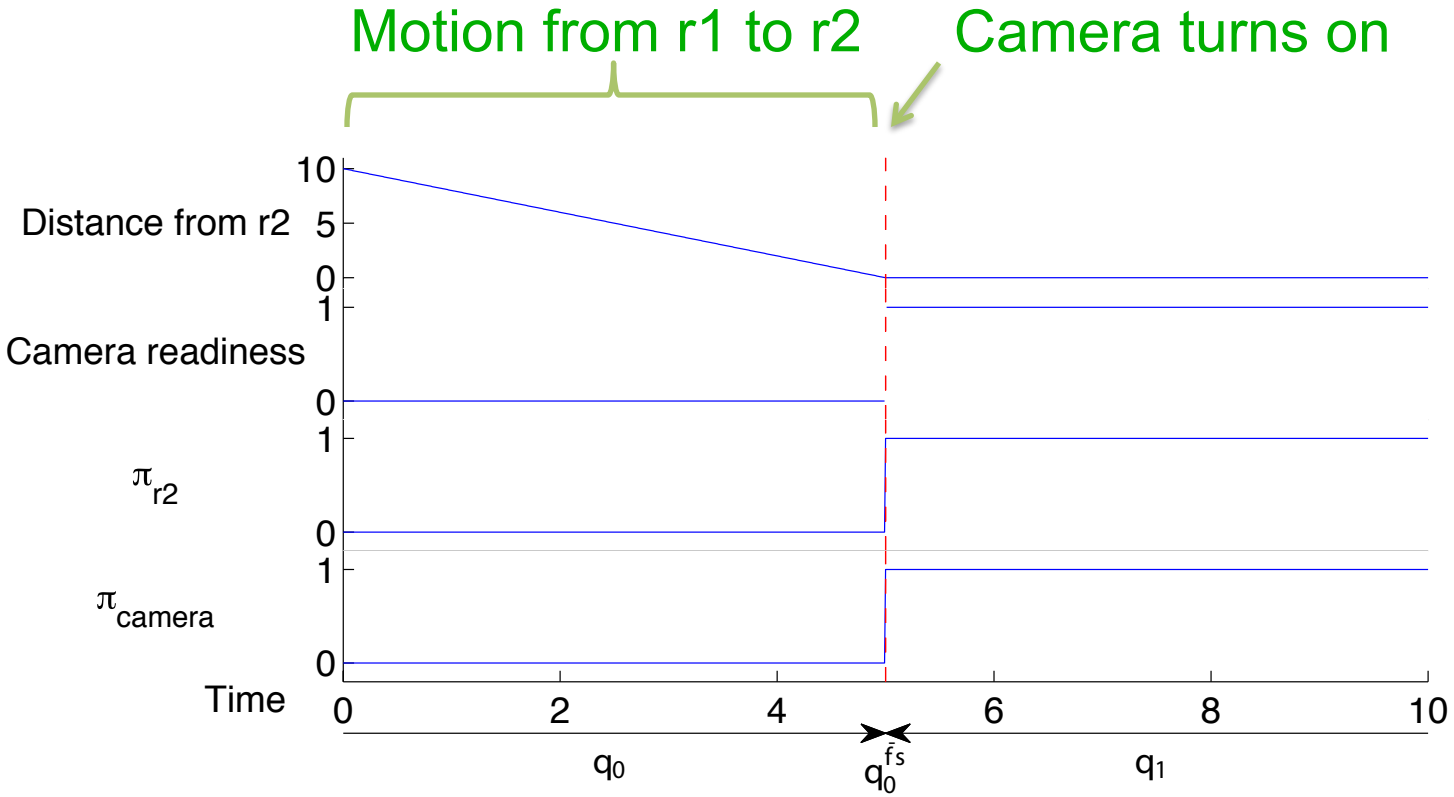
Continuous Execution



– Camera turns on
– Motion from r1 to r2

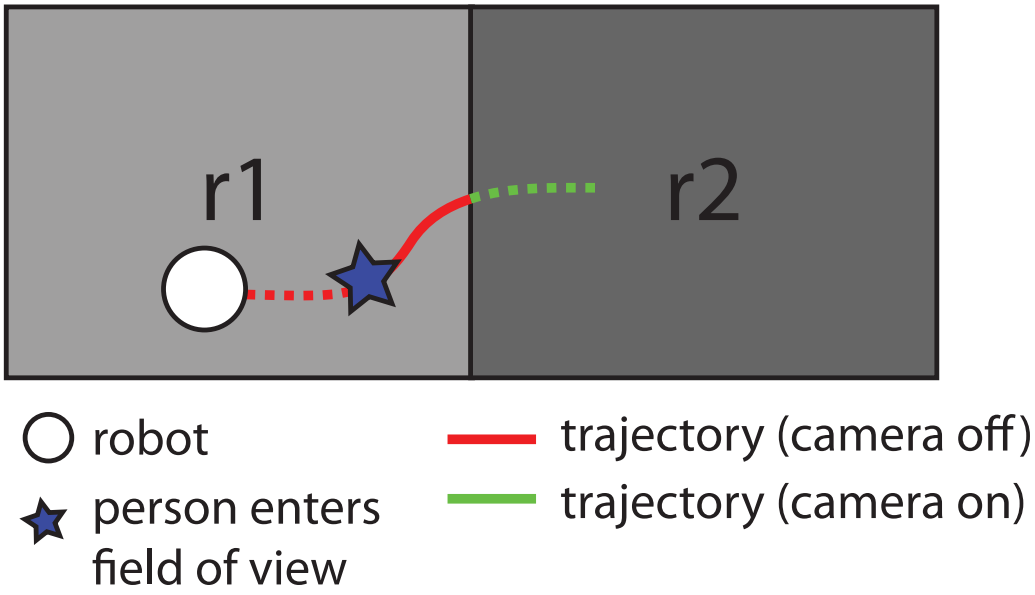
Original Approach: actions *after* motion





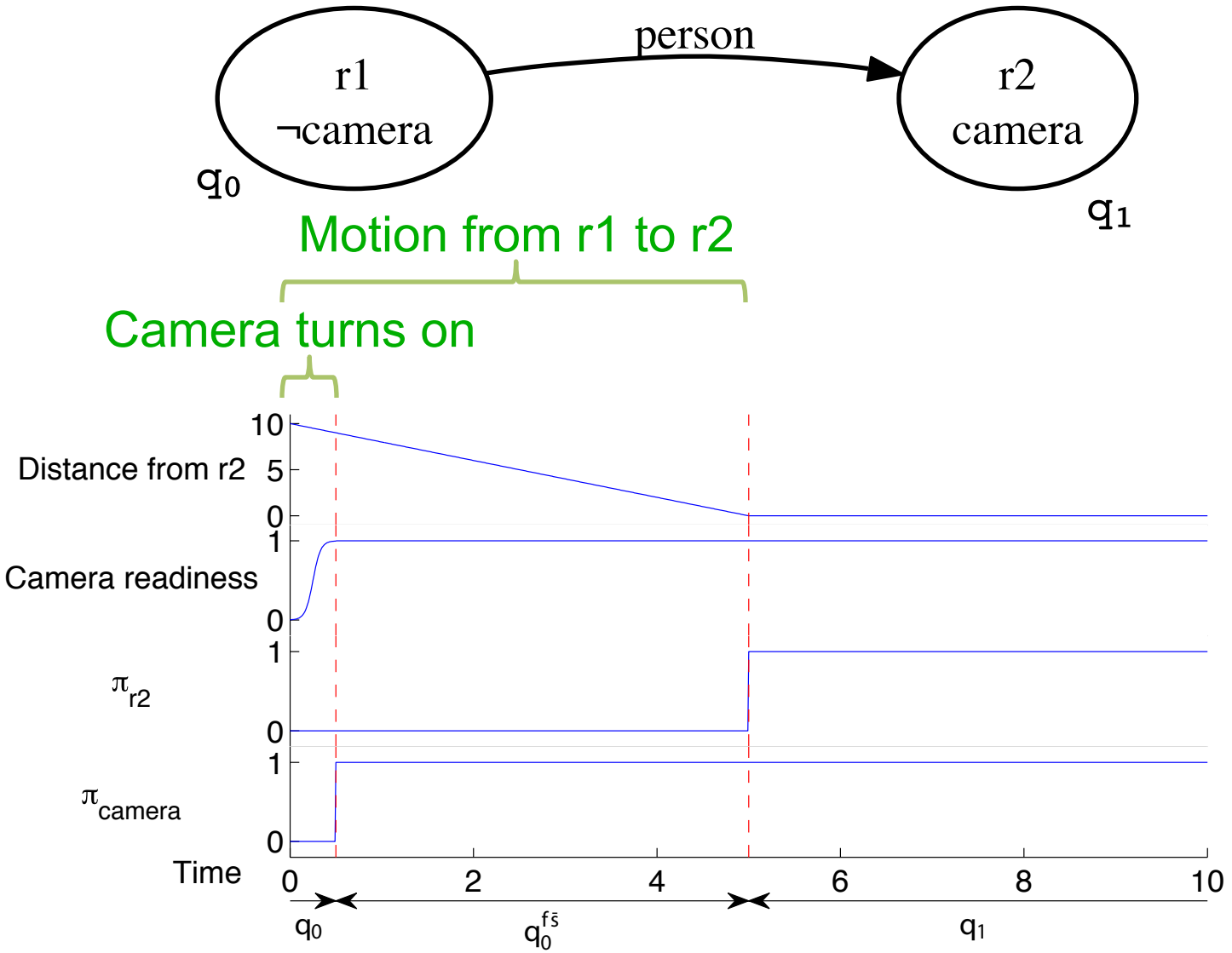
Why is this undesirable?

- Delayed reactivity



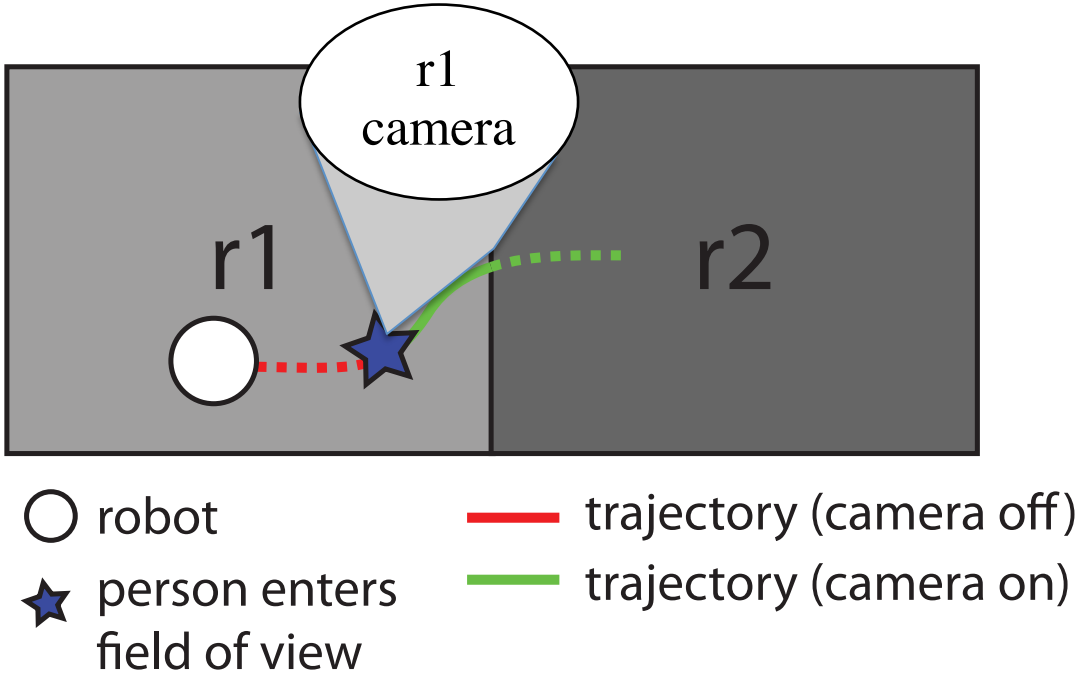
- Potentially unsafe states

Alternative Approach: *simultaneous* actions



Why is this undesirable?

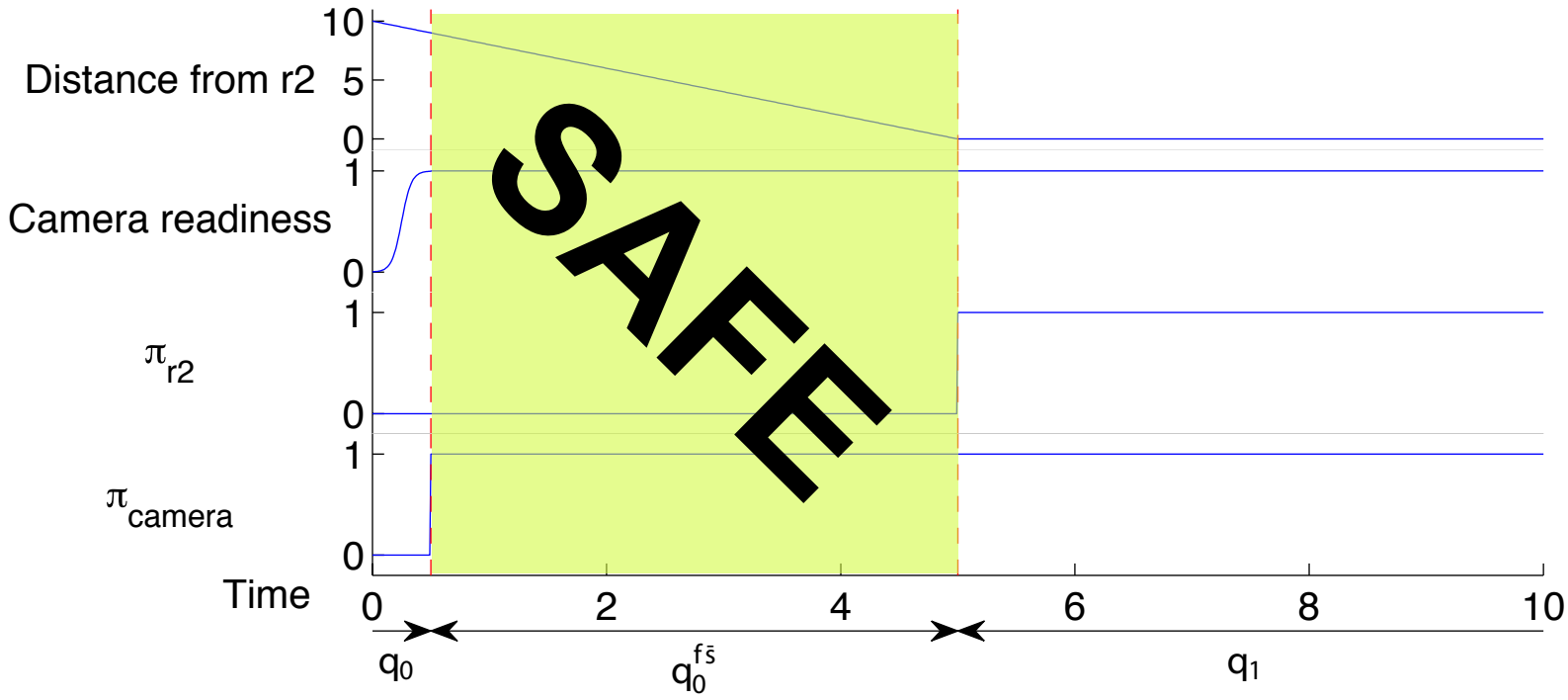
- Potentially unsafe executions



Do not activate the camera in r1

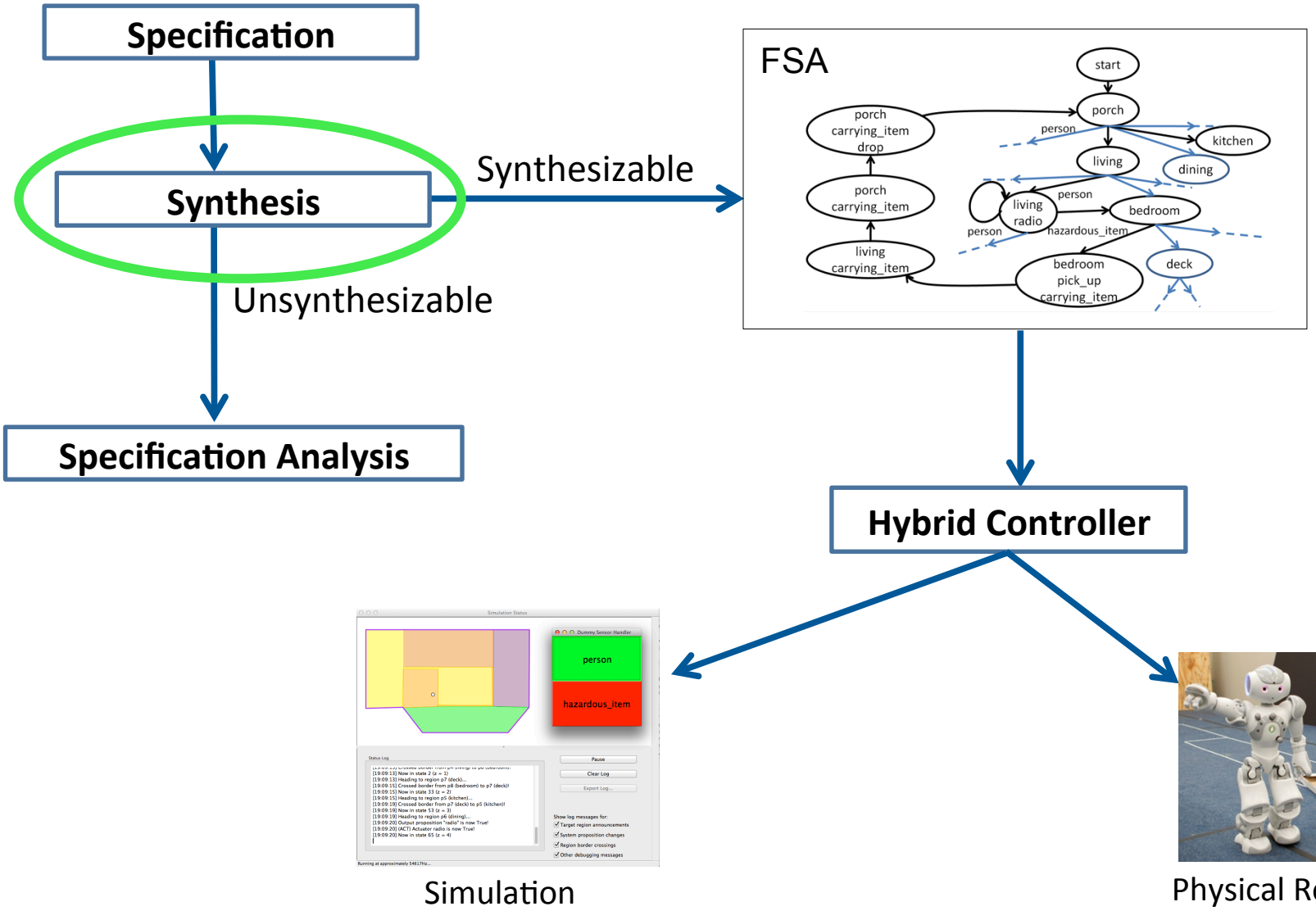
$$\square(\neg(\pi_{camera} \wedge \pi_{r_1}))$$

What we really want:



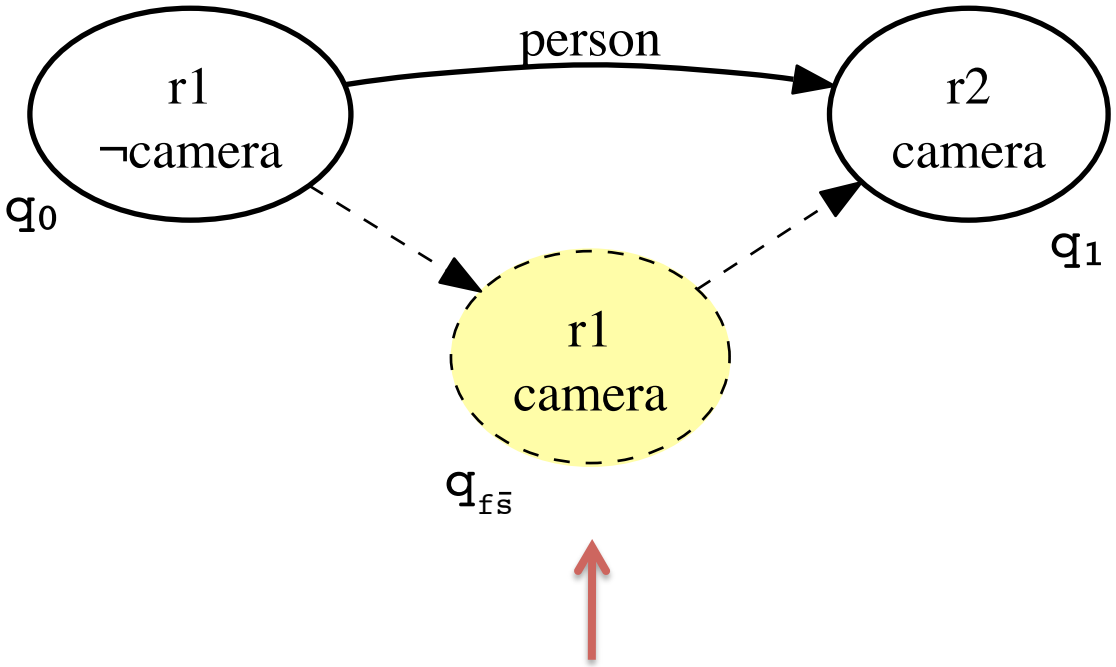
- Automatically check for safe continuous execution during synthesis

Synthesis for Fast/Slow Actions



Synthesis for Fast/Slow Actions

"Slow" = motion, "Fast" = everything else



Implicit intermediate state

Synthesis for Fast/Slow Actions

Implicit intermediate states are safe!

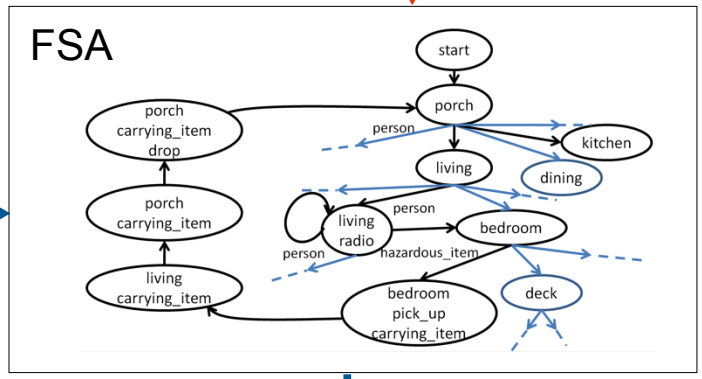
Specification

Synthesis

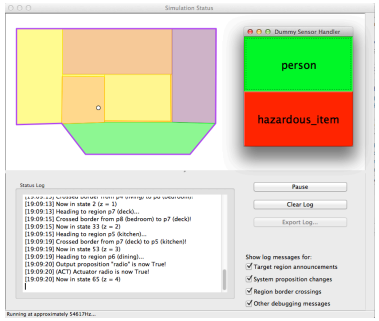
Synthesizable

Unsynthesizable

Specification Analysis



Hybrid Controller



Simulation

Physical Robot

Future Work

- More than two relative action completion durations

Robot actions (in order of duration):

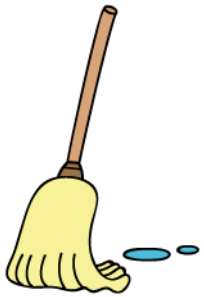
- Turning on the camera
- Waving hand
- Motion between regions

- Explaining unsynthesizability arising from different controller execution durations

Temporal Logic Robot Mission Planning for Slow and Fast Actions

Vasu Raman (vraman@cs.cornell.edu)
Cameron Finucane (cpf37@cornell.edu)
Hadas Kress-Gazit (hadaskg@cornell.edu)

Cornell University



LTLMoP: http://ltlmop.github.com/ (GPL)