Synthesis for Multi-Robot Controllers with Interleaved Motion

Vasu Raman[1]
Hadas Kress-Gazit[2]

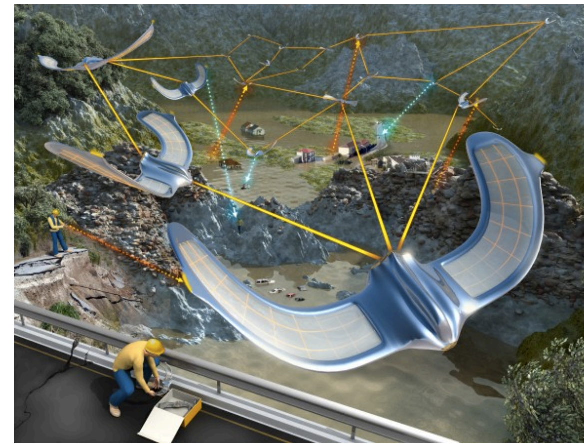
[1]California Institute of Technology
[2]Cornell University

ICRA
3 June 2014

Multi-Robot High-Level Tasks

Kiva Systems

SARTRE Road Train Project



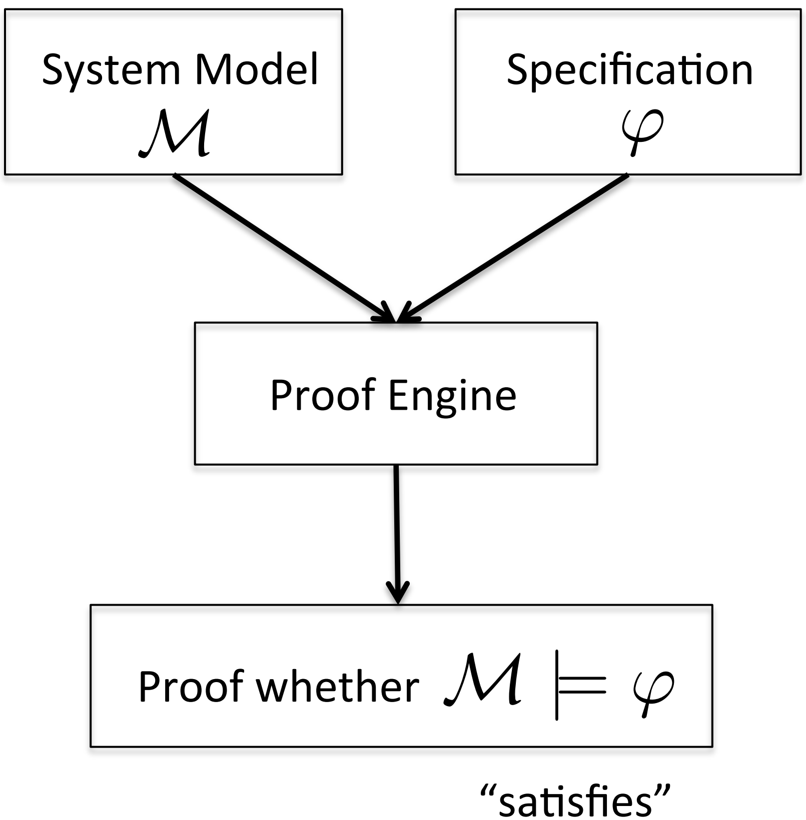
Disaster-Response UAVs (EPFL)

- Teams that operate **autonomously**
- Fulfill **complex** requirements
- Easy to **specify** and **enforce** guarantees

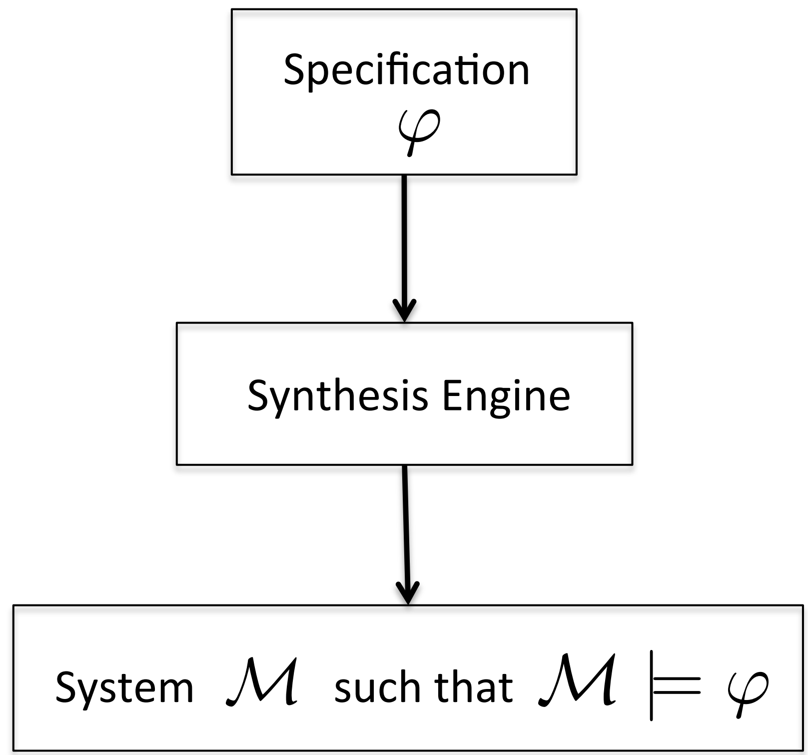
Vasu Raman
(vasu@caltech.edu)

Formal Methods: Two Perspectives

Verification

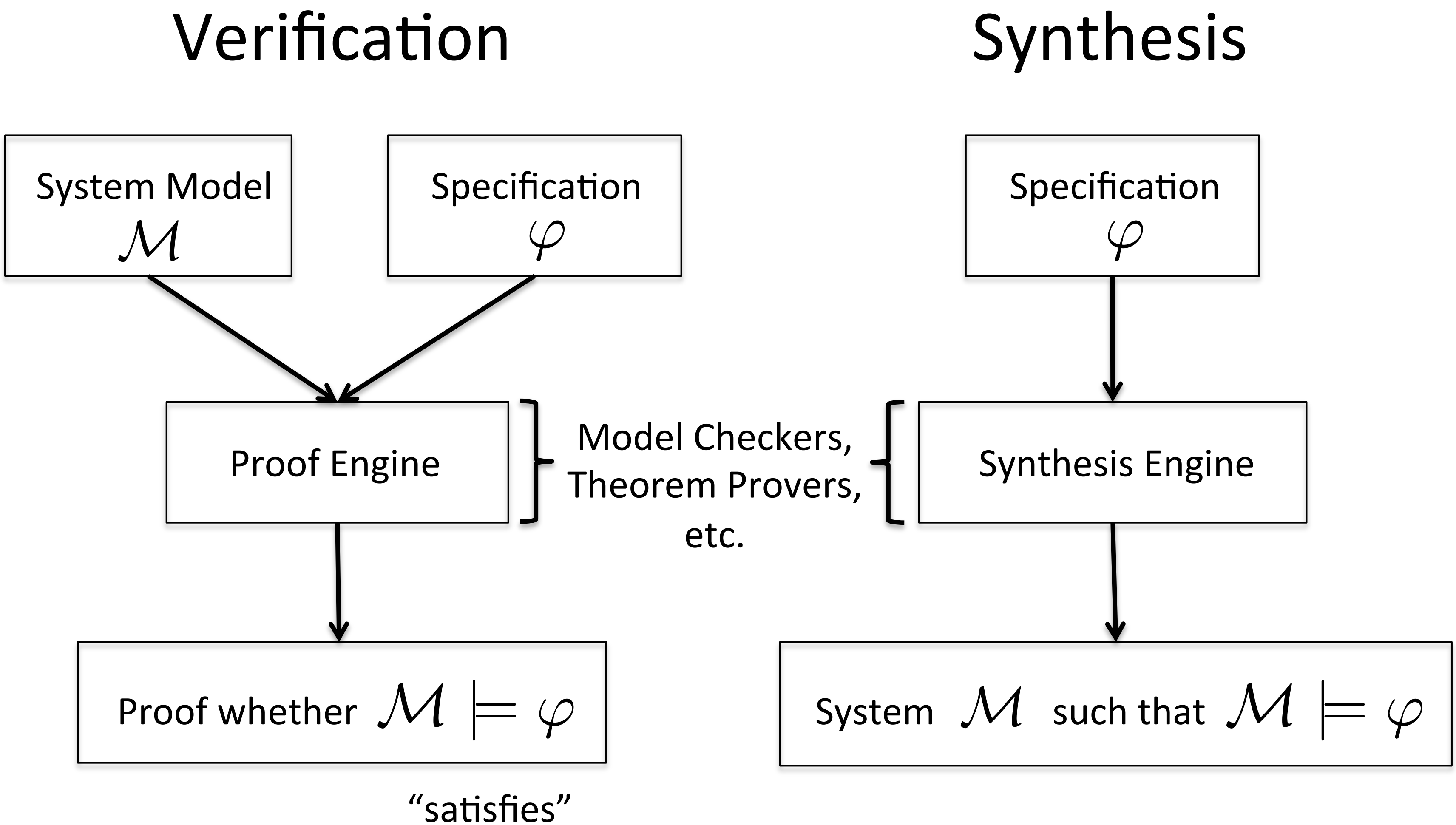


Synthesis



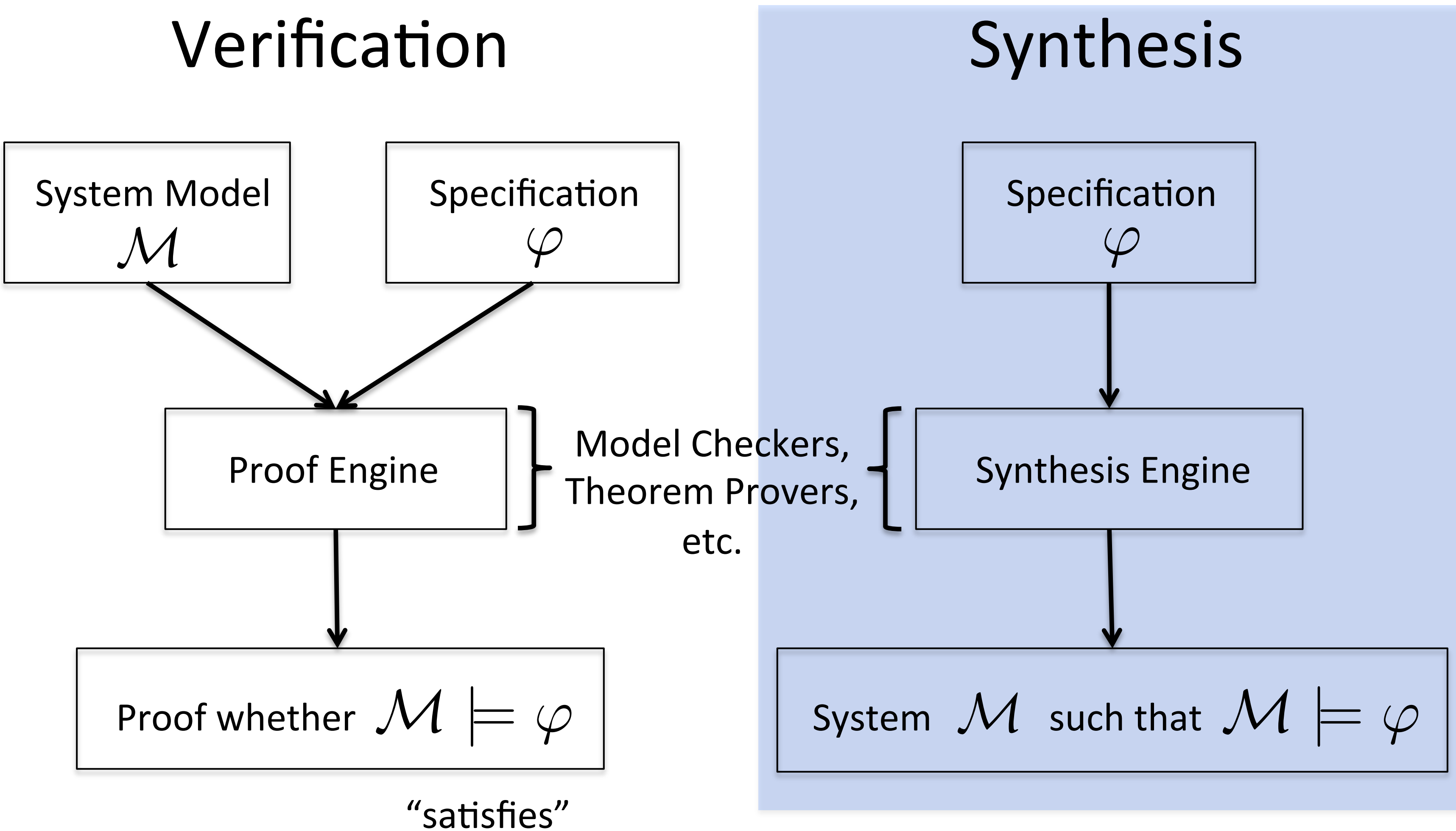
Vasu Raman
(vasu@caltech.edu)

Formal Methods: Two Perspectives



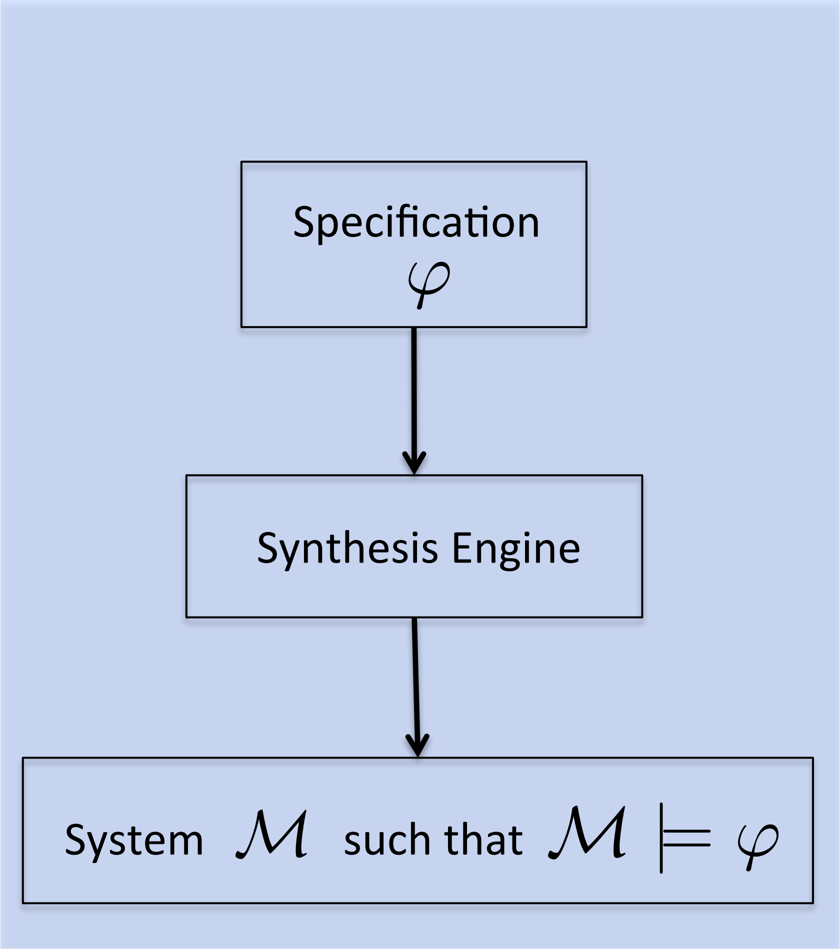
Vasu Raman
(vasu@caltech.edu)

Formal Methods: Two Perspectives



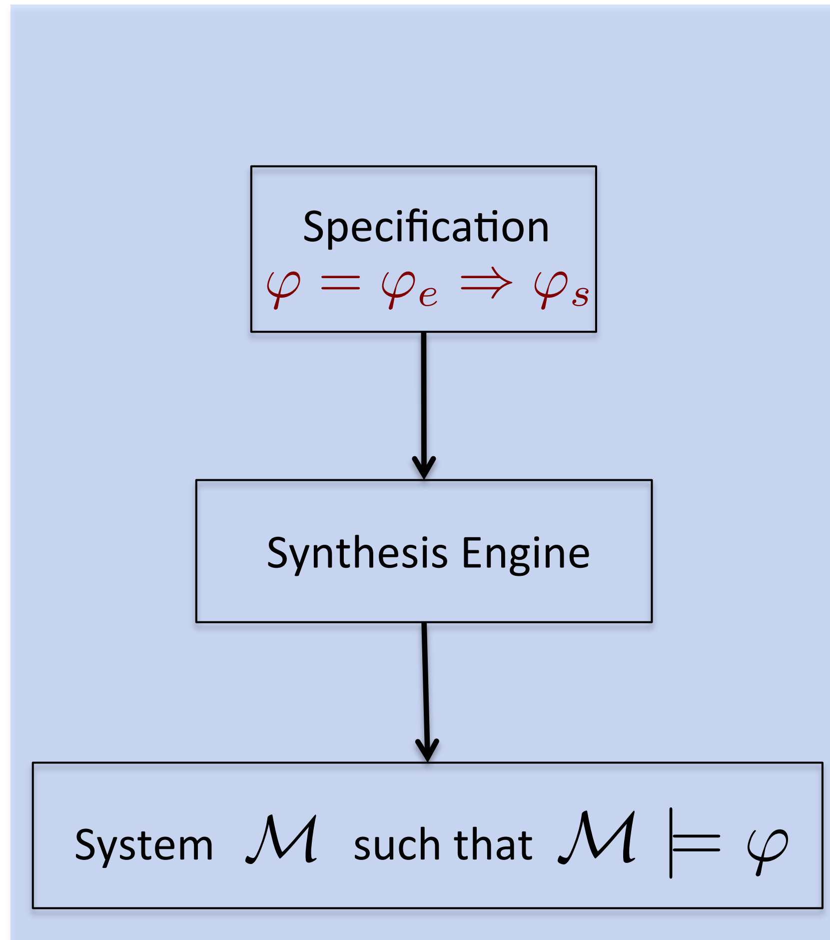
Vasu Raman
(vasu@caltech.edu)

Reactive Synthesis



Vasu Raman
(vasu@caltech.edu)

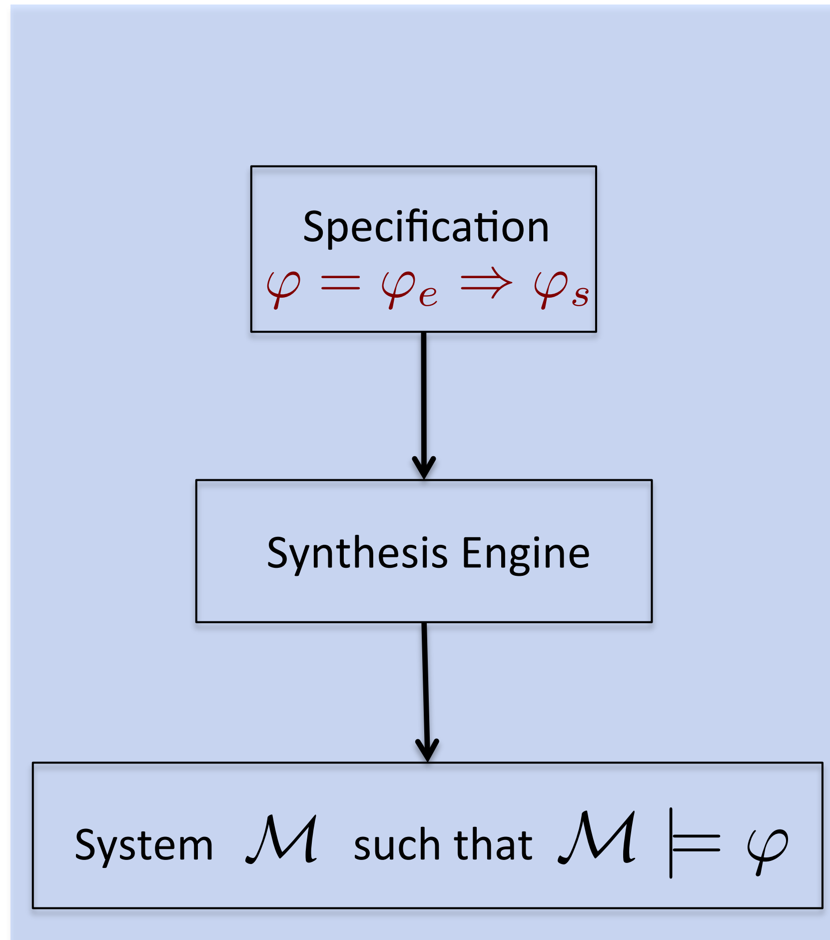
Reactive Synthesis



Vasu Raman
(vasu@caltech.edu)

Reactive Synthesis

- If the operating environment obeys φ_e, the system satisfies φ_s.

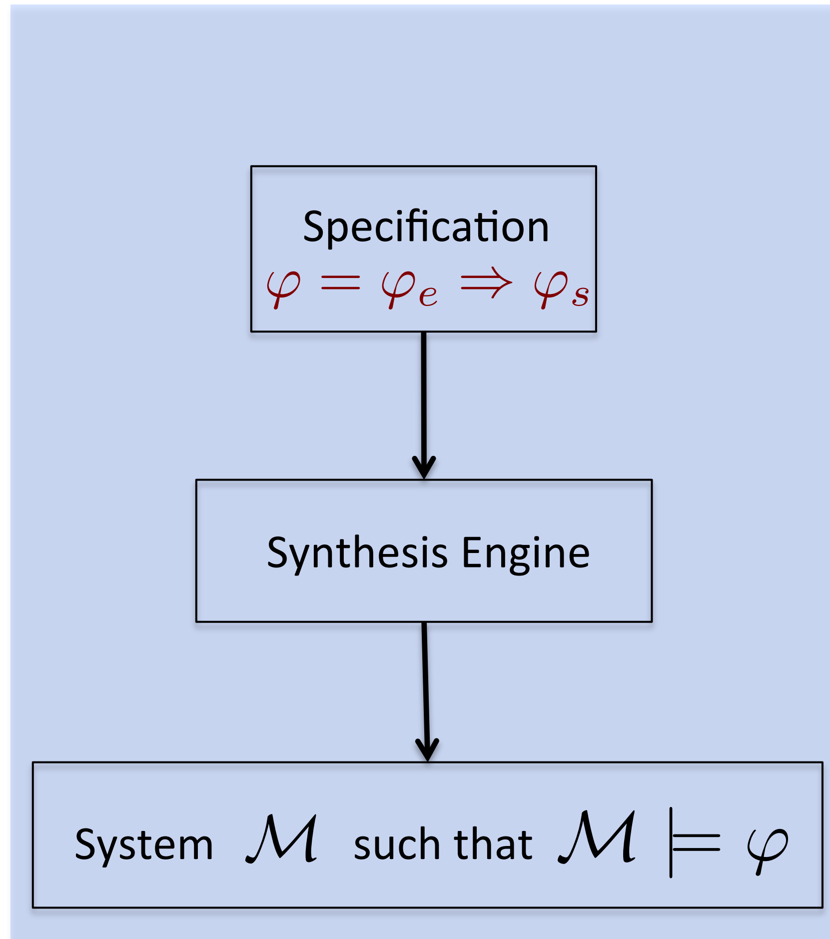


Vasu Raman
(vasu@caltech.edu)

Reactive Synthesis

- If the operating environment obeys φ_e, the system satisfies φ_s.

multi-robot



Vasu Raman
(vasu@caltech.edu)

Reactive Synthesis

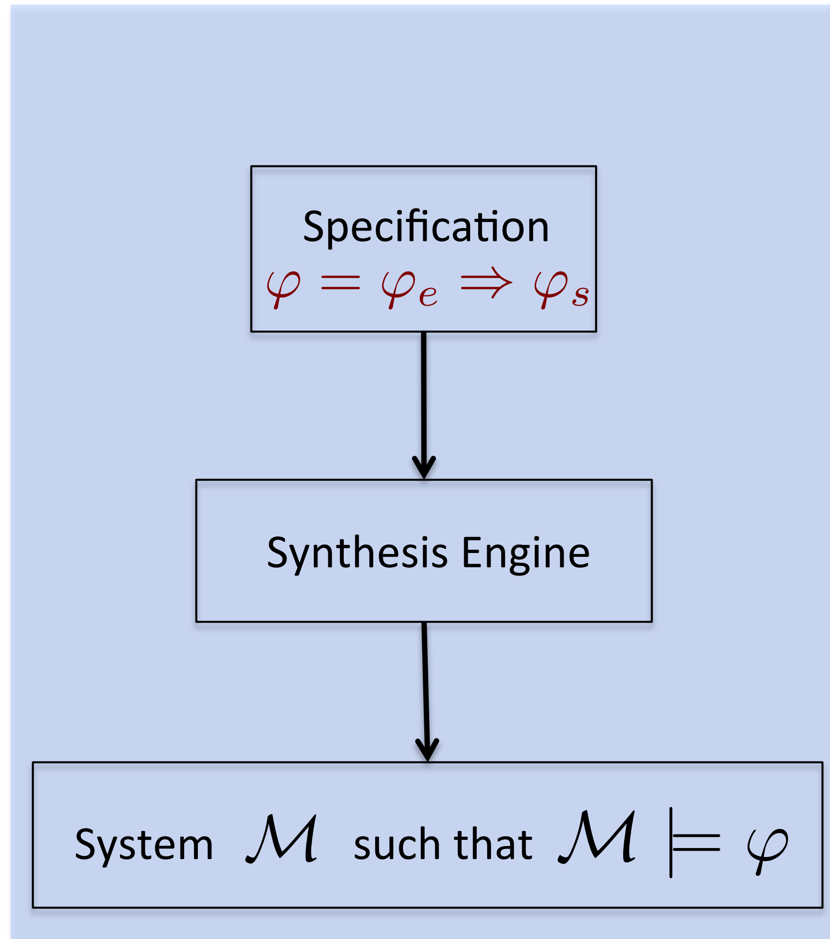
- If the operating environment obeys φ_e, the system satisfies φ_s.

multi-robot

- Today: specifications in Linear Temporal Logic (LTL)*
  - Propositional logic +

    $\mathcal{U}$ (until) $\bigcirc$ (next)
    $\square$ (always) $\diamondsuit$ (eventually)

*actually, GR(1)



Vasu Raman
(vasu@caltech.edu)

LTL Synthesis for Multi-Robot Systems (Related Work)

- Kloetzer and Belta, T-Ro 2007, 2010 — not reactive
- Chen & Belta, T-Ro 2010 — smaller class of specifications
- Loizou and Kyriakopoulo, CDC 2004 — restrict non-motion actions to be continuous
- Kress-Gazit, Ayanian, Pappas & Kumar, CASE 2008 — restrict robot motion to one at a time

LTL Synthesis for Robot Control

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

Vasu Raman
(vasu@caltech.edu)

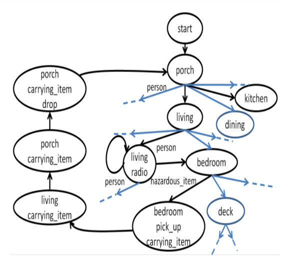
LTL Synthesis for Robot Control

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

Boolean propositions representing sensing & actuation events
+
LTL specification

Finite State Machine such that

$$\mathcal{M} \models \varphi$$



Hybrid controller executing low-level actions according to FSM

Vasu Raman
(vasu@caltech.edu)

Example: Recycling Robots

- Robot 1 starts in r1, robot 2 in r4. Neither is carrying anything.
- Robot 1 activates `carrying_glass` if sensing glass, `carrying_metal` if sensing metal.
- If Robot 1 is `carrying_glass`, it should visit r4 and if `carrying_metal`, it should visit r2.
- Robot 1 always eventually goes to r4. Robot 2 always eventually goes to r1.

Vasu Raman
(vasu@caltech.edu)

Example: Recycling Robots

- Robot 1 starts in r1, robot 2 in r4. Neither is carrying anything.
- Robot 1 activates `carrying_glass` if sensing glass, `carrying_metal` if sensing metal.
- If Robot 1 is `carrying_glass`, it should visit r4 and if `carrying_metal`, it should visit r2.
- Robot 1 always eventually goes to r4.
  Robot 2 always eventually goes to r1.

worker

freeloader

Vasu Raman
(vasu@caltech.edu)

Example: Recycling Robots

- Robot 1 starts in r1, robot 2 in r4. Neither is carrying anything.
- Robot 1 activates `carrying_glass` if sensing glass, `carrying_metal` if sensing metal.
- If Robot 1 is `carrying_glass`, it should visit r4 and if `carrying_metal`, it should visit r2.
- Robot 1 always eventually goes to r4. Robot 2 always eventually goes to r1.

For each robot *i*:

<u>Actions</u>:

Motion to region r1-4 $\pi_{i_{r_j}}$

`carrying_glass`, `carrying_metal`

$\pi_{i_{carrying\_glass}}$ $\pi_{i_{carrying\_metal}}$

<u>Sensors</u>:

glass $\pi_{i_{glass}}$

metal $\pi_{i_{metal}}$

Vasu Raman
(vasu@caltech.edu)

Example: Recycling Robots

- Robot 1 starts in r1, robot 2 in r4. Neither is carrying anything.

$$\begin{aligned} & (\pi 1_{r_1} \wedge \pi 2_{r_4} \\ \wedge \quad & \neg\pi 1_{carrying\_metal} \wedge \neg\pi 1_{carrying\_glass} \\ \wedge \quad & (\neg\pi 2_{carrying\_metal} \wedge \neg\pi 2_{carrying\_glass}) \end{aligned}$$

- Robot 1 activates `carrying_glass` if sensing glass, `carrying_metal` if sensing metal.

$$\begin{aligned} & \Box(\bigcirc \pi 1_{glass} \Rightarrow \bigcirc \pi 1_{carrying\_glass}) \\ \wedge \quad & \Box(\bigcirc \pi 1_{metal} \Rightarrow \bigcirc \pi 1_{carrying\_metal}) \end{aligned}$$

- If Robot 1 is `carrying_glass`, it should visit r4 and if `carrying_metal`, it should visit r2.

$$\begin{aligned} & \Box \Diamond(\pi 1_{carrying\_glass} \rightarrow \bigcirc \pi 1_{r_4}) \\ \wedge \quad & \Box \Diamond(\pi 1_{carrying\_metal} \rightarrow \bigcirc \pi 1_{r_2}) \end{aligned}$$

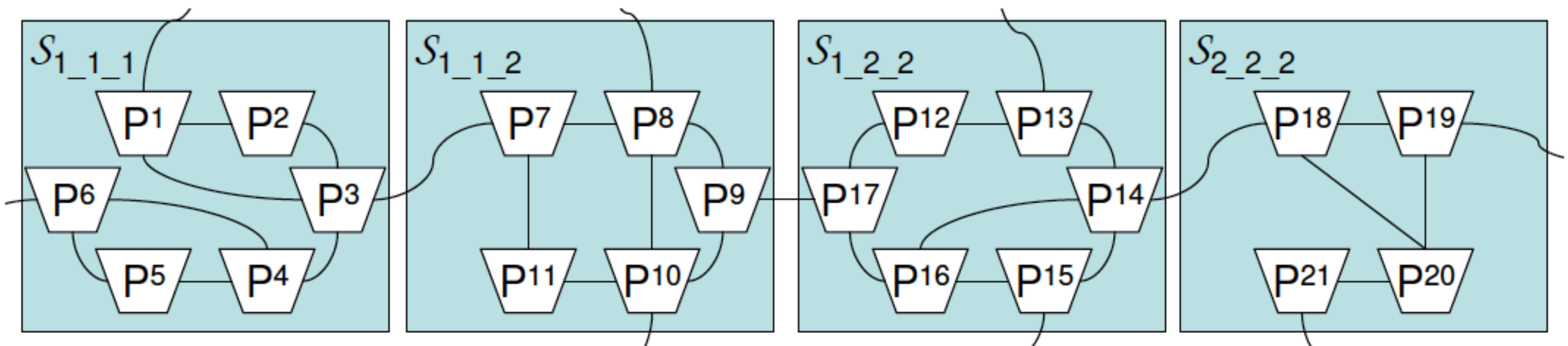
- Robot 1 always eventually goes to r4. Robot 2 always eventually goes to r1.

$$\Box \Diamond(\pi 1_{r_4}) \wedge \Box \Diamond(\pi 2_{r_1})$$

Vasu Raman
(vasu@caltech.edu)

Motion Control

- Atomic motion controllers [Ayanian & Kumar, T-Ro 2010]
  - Polytope graph on team configuration space
    - (includes proximity constraints)

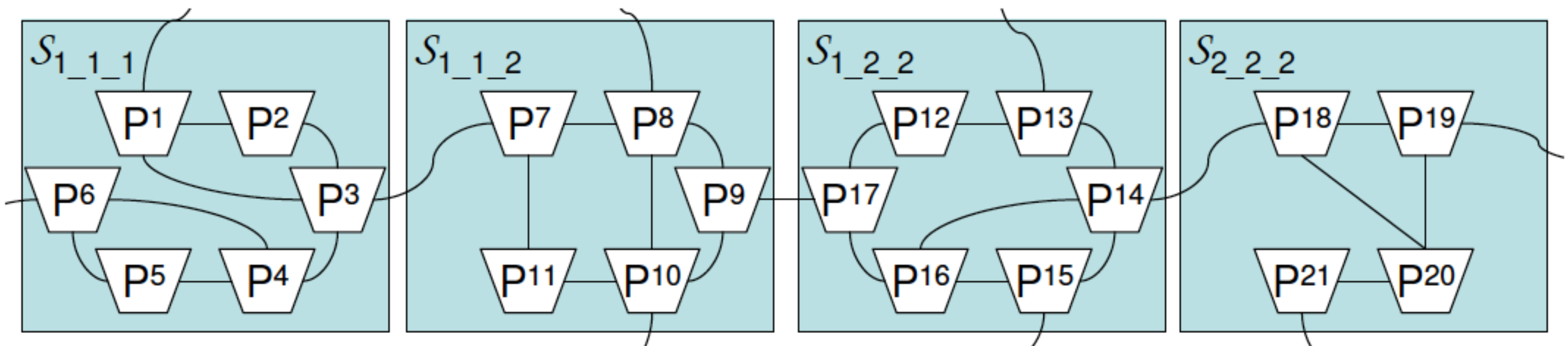


[Kress-Gazit, Ayanian, Pappas & Kumar, CASE 2008]

Vasu Raman
(vasu@caltech.edu)

Motion Control

- Atomic motion controllers [Ayanian & Kumar, T-Ro 2010]
  - Polytope graph on team configuration space
    - (includes proximity constraints)



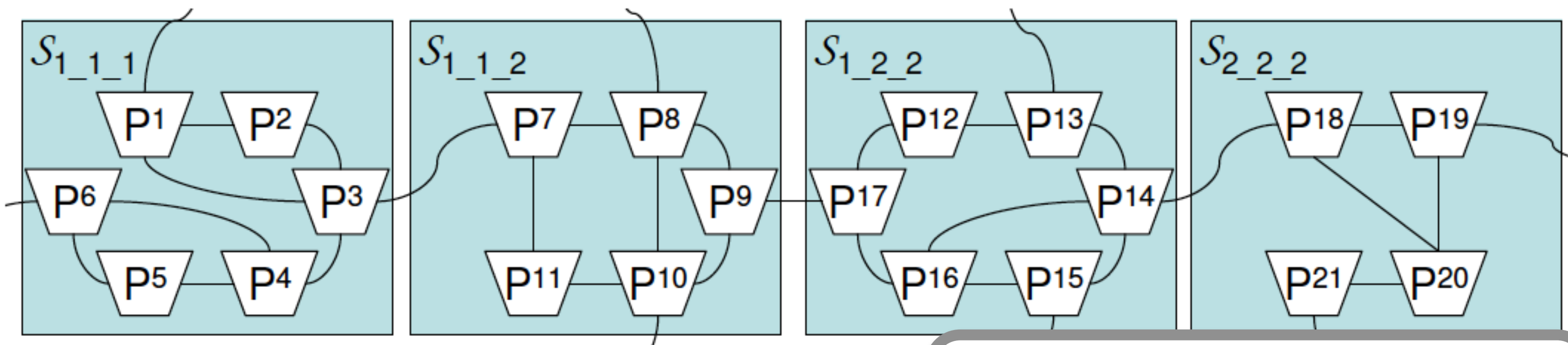
[Kress-Gazit, Ayanian, Pappas & Kumar, CASE 2008]

- Bisimulation Condition
  - controller exists for every admissible discrete transition

Vasu Raman
(vasu@caltech.edu)

Motion Control

- Atomic motion controllers [Ayanian & Kumar, T-Ro 2010]
  - Polytope graph on team configuration space
    - (includes proximity constraints)



[Kress-Gazit, Ayanian, Pappas & Kumar, CASE 2008]

Only one robot crosses a room threshold at a time

- Bisimulation Condition
  - controller exists for every admissible discrete transition

Vasu Raman
(vasu@caltech.edu)

Naïve Approach

- Restrict discrete transition system
  - If robot 1 is moving, robot 2 stays in place

Spec summary:

- Robot 1 starts in r1, robot 2 in r4.
- Robot 1 does a bunch of recycling tasks and also always eventually goes to r4.
- Robot 2 always eventually goes to r1.

Vasu Raman
(vasu@caltech.edu)

Naïve Approach

- Restrict discrete transition system
  - If robot 1 is moving, robot 2 stays in place

Spec summary:

- Robot 1 starts in r1, robot 2 in r4.
- Robot 1 does a bunch of recycling tasks and also always eventually goes to r4.
- Robot 2 always eventually goes to r1.

- Too conservative
  - If robot 1 (worker) is always moving, robot 2 (freeloader) can never reach r1 → synthesis fails

Vasu Raman
(vasu@caltech.edu)

Solution Overview

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

- Explicitly model motion initiation/completion

Actions: $\pi_{i_{r_j}}$, $\pi_{i_{carrying\_glass}}$, $\pi_{i_{carrying\_metal}}$

Sensors: $\pi_{i_{glass}}$, $\pi_{i_{metal}}$, $\pi^{c}_{i_{r_j}}$, $\pi^{c}_{i_{carrying\_glass}}$, $\pi^{c}_{i_{carrying\_metal}}$

"Robot 1 starts in r1, robot 2 in r4."

$\pi^{c}_{1_{r_1}} \wedge \pi^{c}_{2_{r_4}}$ (instead of $\pi_{1_{r_1}} \wedge \pi_{2_{r_4}}$)

"Robot 1 always eventually goes to r4."

$\Diamond \Box \pi^{c}_{1_{r_4}}$ (instead of $\Diamond \Box \pi_{1_{r_4}}$)

Solution Overview

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

Enhanced GR(1) Synthesis Algorithm

[Raman, Piterman and Kress-Gazit, ICRA 13]

- Accommodates "transition" goals

  $\Box\Diamond(\varphi_1 \Rightarrow \bigcirc\varphi_2)$ in addition to $\Box\Diamond\varphi$

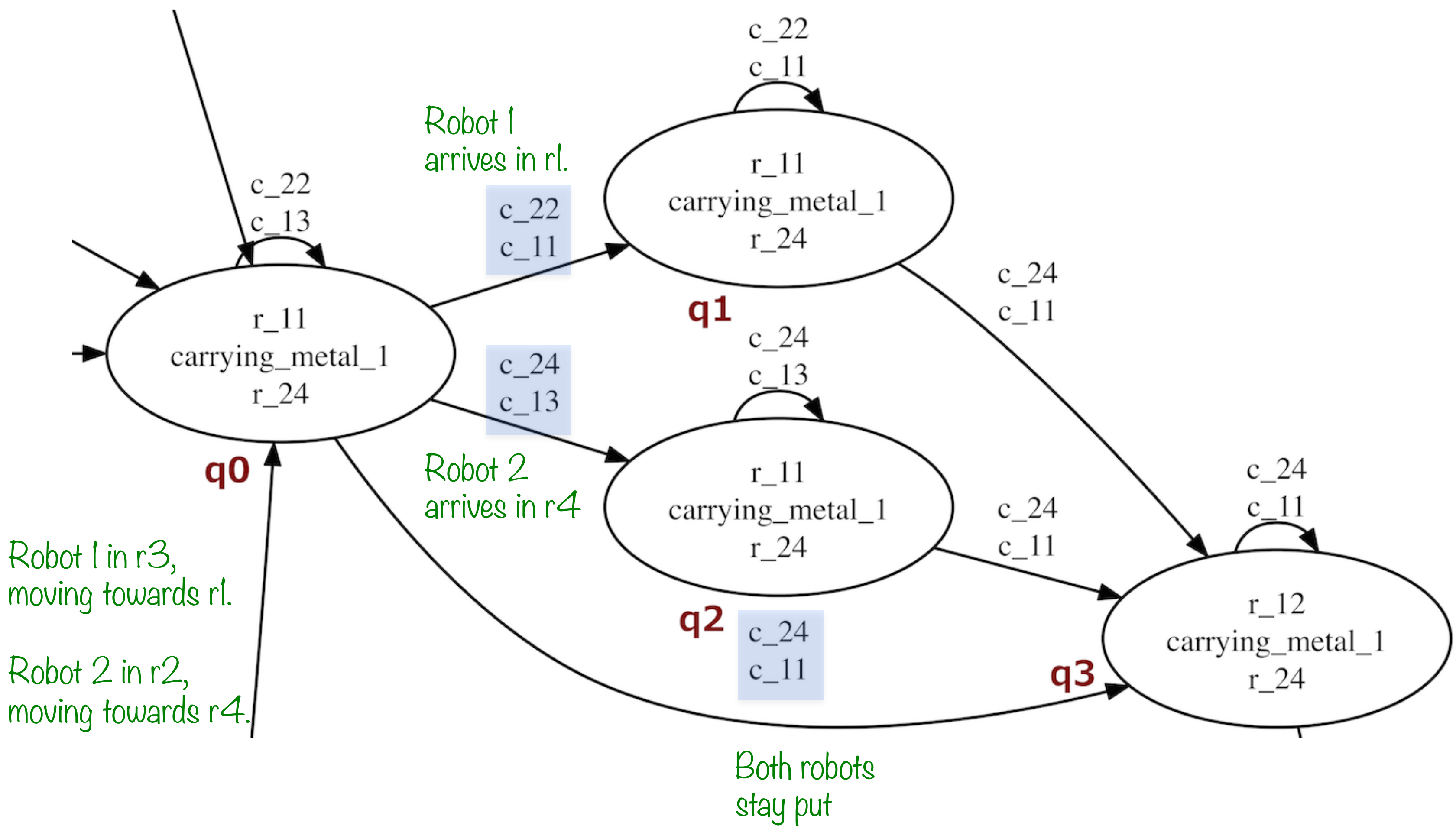
- Efficient for new specifications

Solution Overview

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

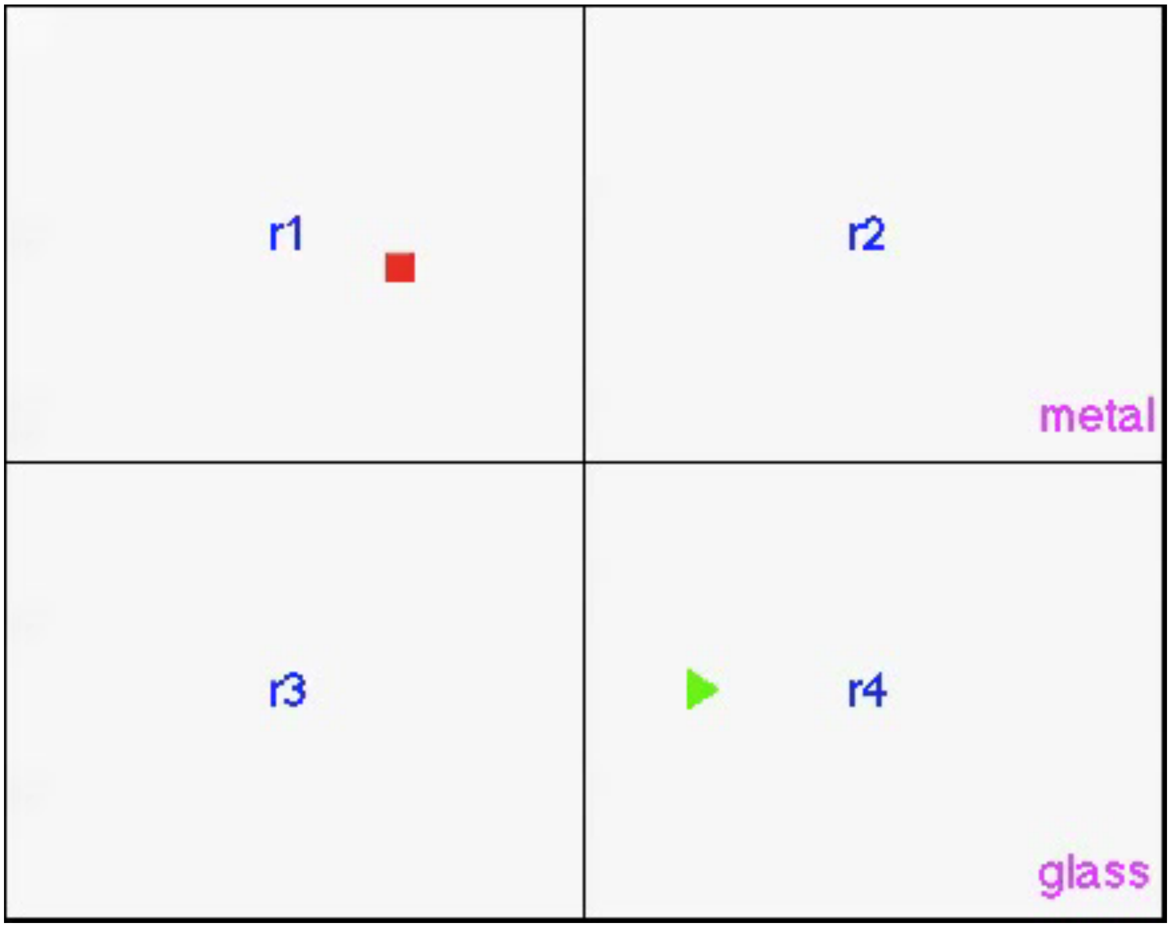
- Simultaneous motion **initiation** for all robots
- At most one robot **completes** region change at a time

Example (finite state machine)



Vasu Raman
(vasu@caltech.edu)

Example (simulation)



Vasu Raman
(vasu@caltech.edu)

Next Steps

- Experimental validation on hardware

- More concise discrete abstraction
  - Currently adding one sensor proposition per action

Vasu Raman
(vasu@caltech.edu)

Thanks!

Synthesis for Multi-Robot Controllers with Interleaved Motion

Vasu Raman[1]
Hadas Kress-Gazit[2]

[1]California Institute of Technology
[2]Cornell University

Contact: vasu@caltech.edu

ICRA
3 June 2014