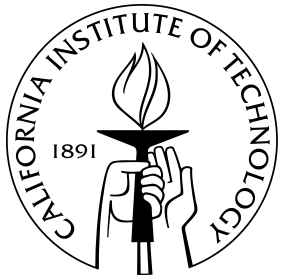
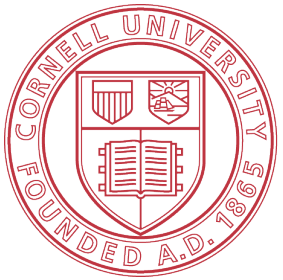
Towards Minimal Explanations of Unsynthesizability for High-Level Robot Behaviors

Vasu Raman
Department of Computing and Mathematical Sciences
California Institute of Technology

Hadas Kress-Gazit
Sibley School of Mechanical and Aerospace Engineering
Cornell University





High-Level Tasks:

Carrying meals to patients

Delivering medical records

Patrolling patient rooms

.

.

.

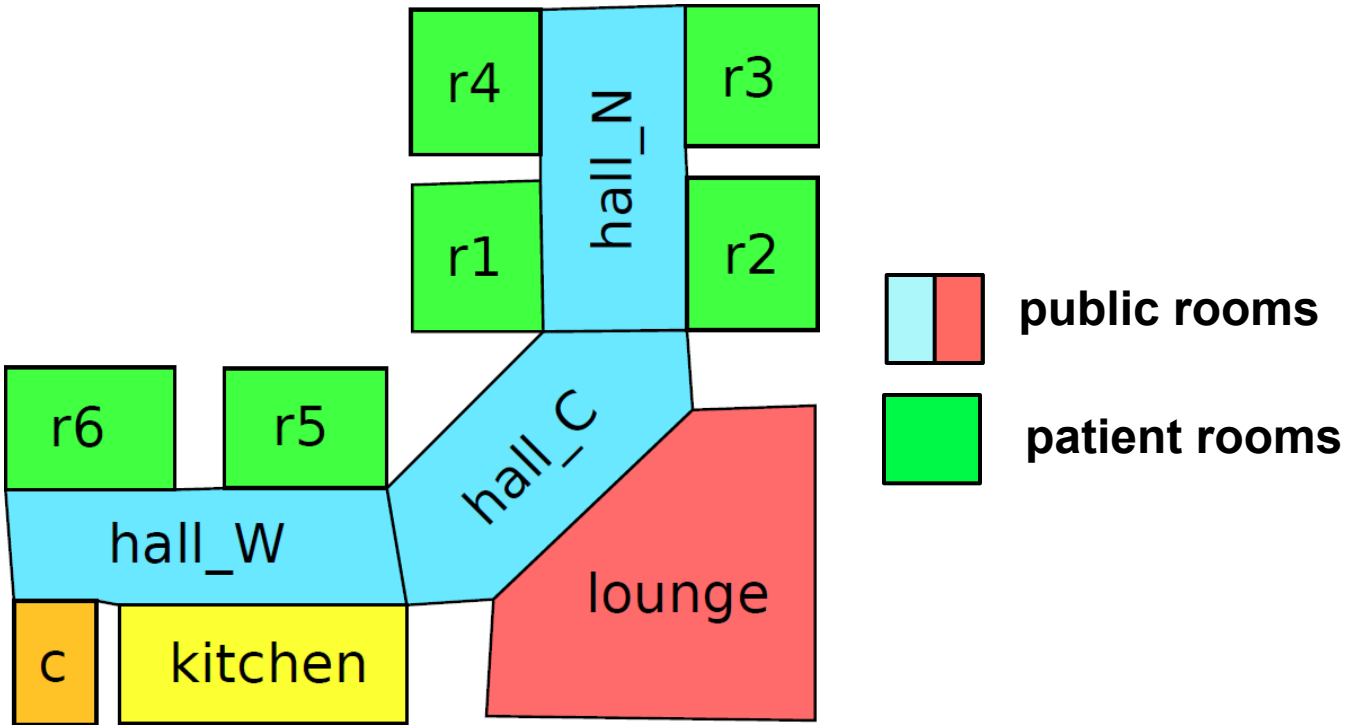
http://newsroom.ucla.edu/portal/ucla/artwork/4/5/7/9/4/245794/EVA_Robot_3-c.jpg

High-Level Tasks:
Carrying meals to patients
Delivering medical records
Patrolling patient rooms

Challenges:
Easy to instruct
Does as it is told*

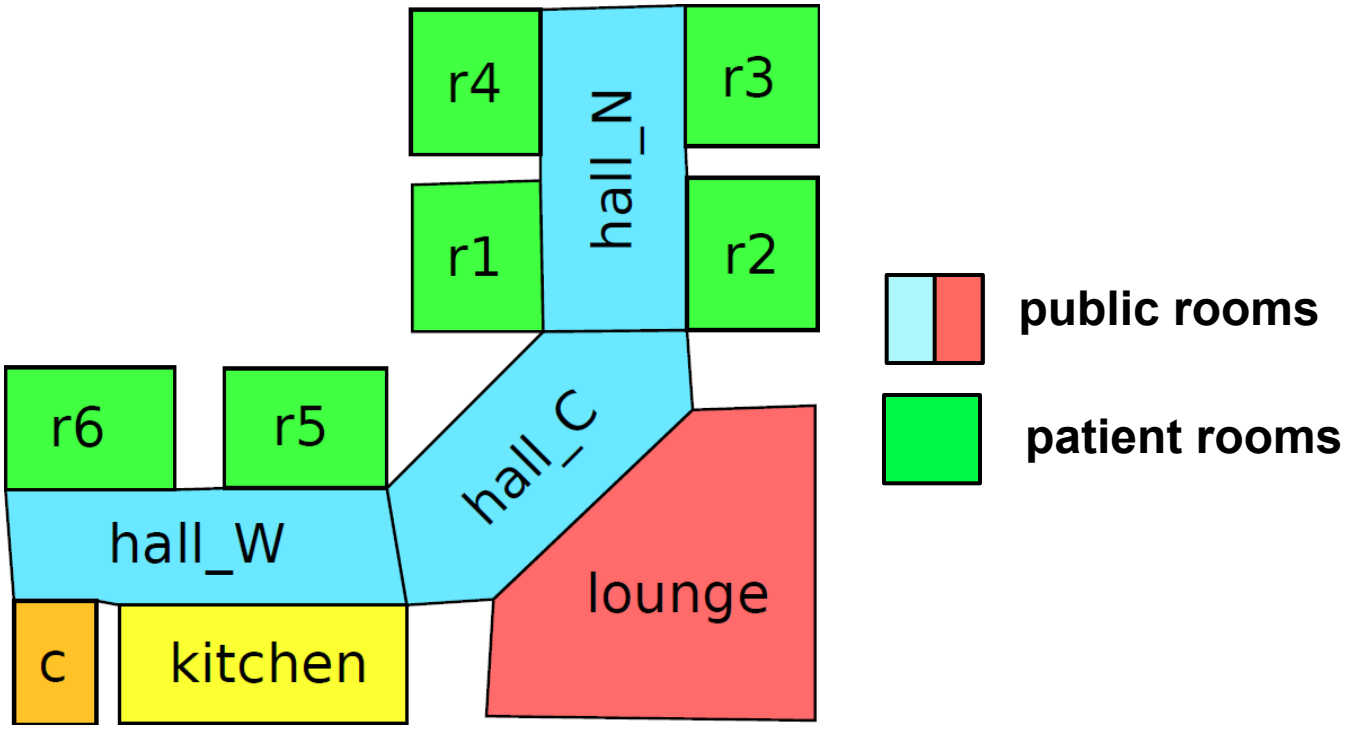
http://newsroom.ucla.edu/portal/ucla/artwork/4/5/7/9/4/245794/EVA_Robot_3-c.jpg

Example



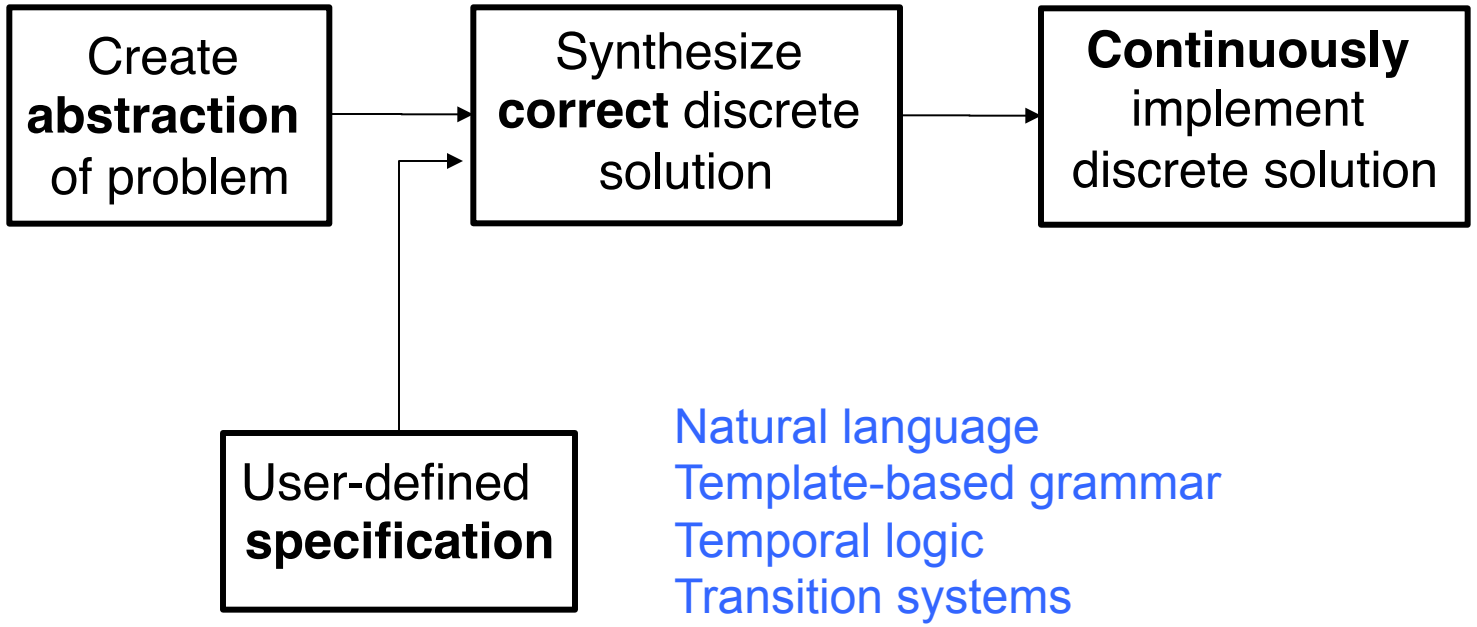
Carry meals from the kitchen to all patient rooms.

Example

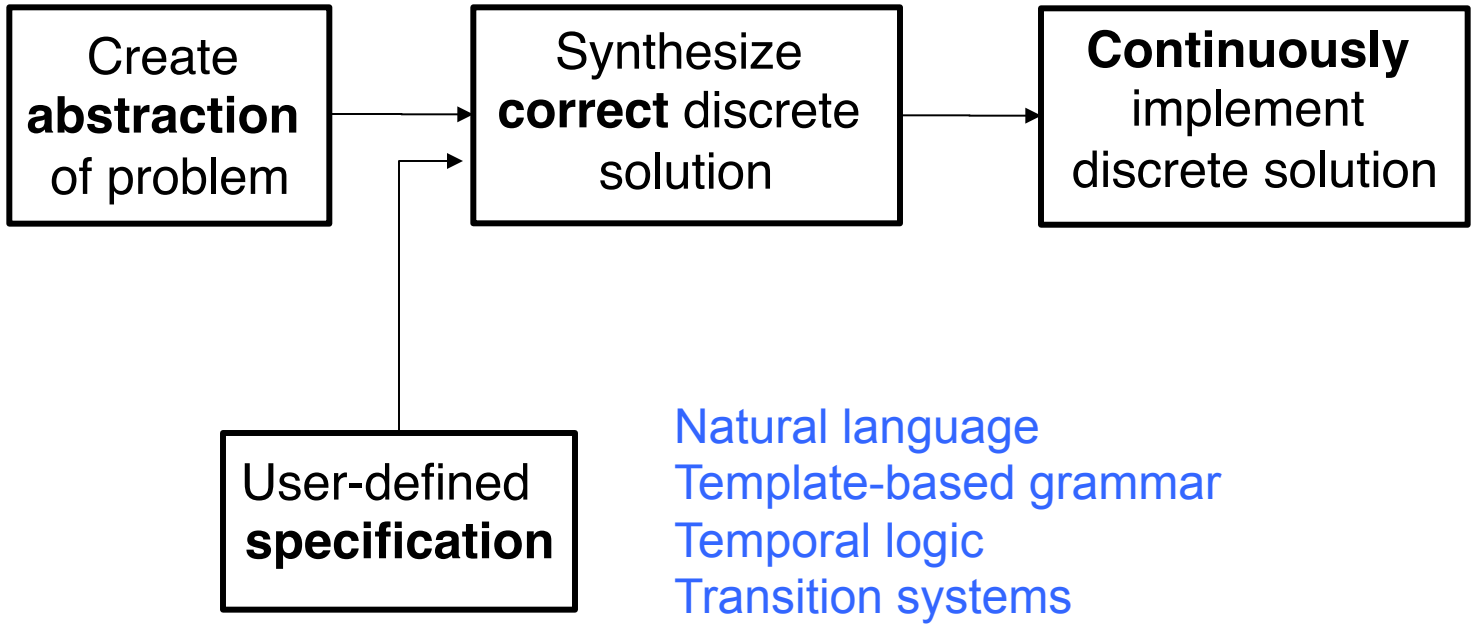


Start in the closet. Carry meals from the kitchen to all patient rooms. Don't go into any public rooms.

Formal Methods for High-Level Control



Formal Methods for High-Level Control



- Fainekos, Kress-Gazit and Pappas, ICRA 2005
- Kress-Gazit, Fainekos and Pappas, ICRA 2007
- Kloetzer and Belta, TAC 2008
- Karaman and Frazzoli, CDC 2009
- Bhatia, Kavraki and Vardi, ICRA 2010
- Wongpiromsarn, Topcu and Murray, HSCC 2010

Controller Synthesis Overview (LTLMoP* Toolkit)

Specification

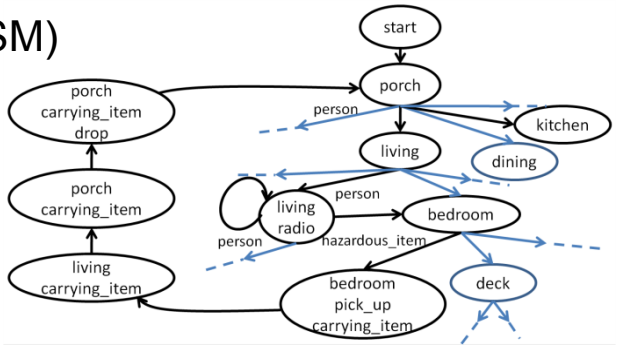
Synthesis

Synthesizable

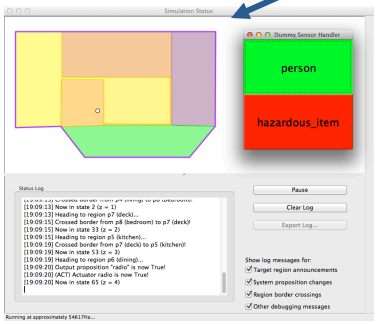
Unsynthesizable

?

(FSM)



Hybrid Controller

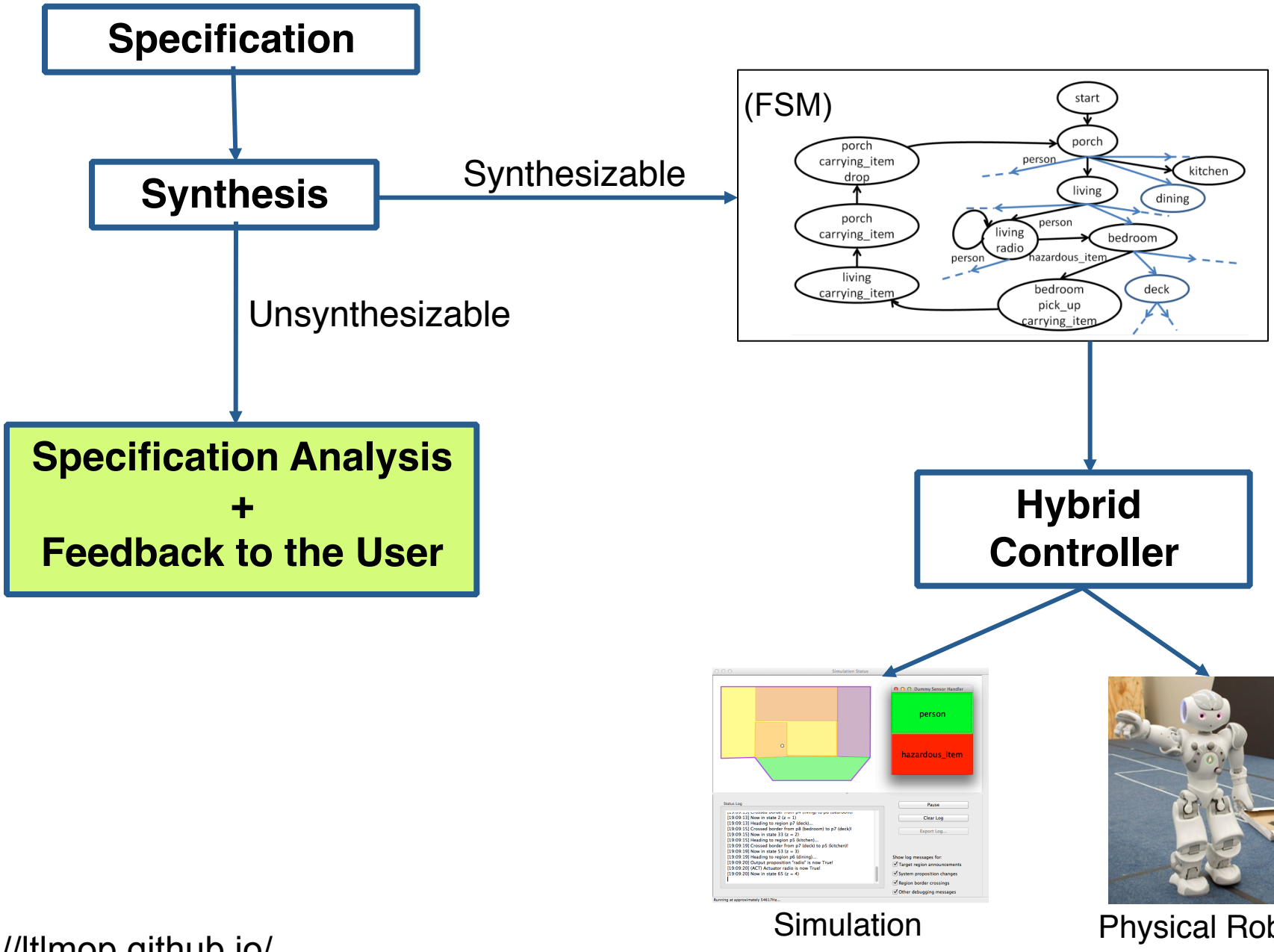


Simulation

Physical Robot

*http://ltlmop.github.io/

Controller Synthesis Overview (LTLMoP* Toolkit)



*http://ltlmop.github.io/

Controller Synthesis Overview (LTLMoP* Toolkit)

Specification

Synthesis

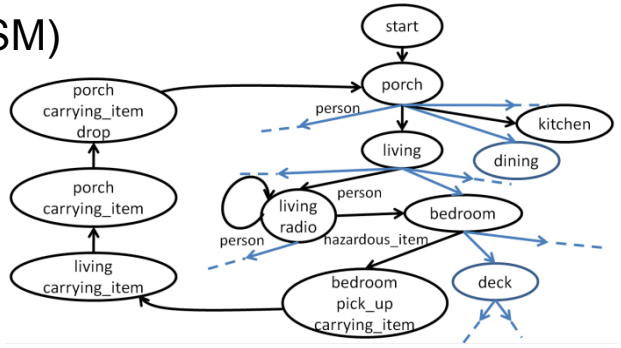
Synthesizable

Unsynthesizable

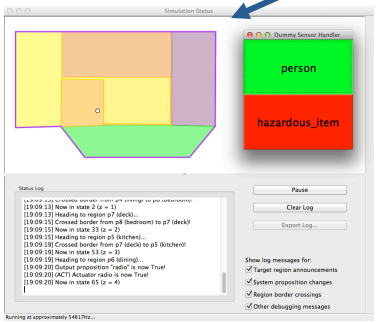
Specification Analysis
+
Feedback to the User

Change the Specification

(FSM)



Hybrid Controller



Simulation

Physical Robot

*http://ltlmop.github.io/

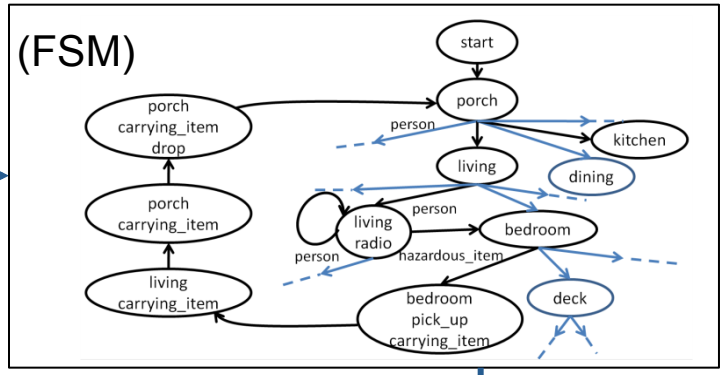
Controller Synthesis Overview (LTLMoP* Toolkit)

Specification

Synthesis

Synthesizable

Unsynthesizable

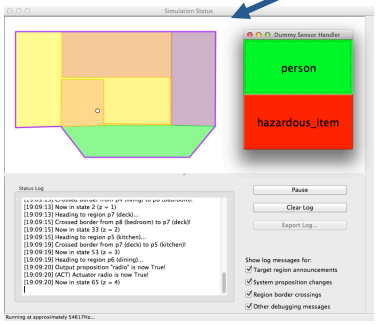


Specification Analysis + Feedback to the User

This talk

Hybrid Controller

Change the Specification



Simulation

Physical Robot

*http://ltlmop.github.io/

Form of Specification

$$\varphi = (\varphi_e \Rightarrow \varphi_s)$$

$$= (\varphi^i{}_e \wedge \varphi^t{}_e \wedge \varphi^g{}_e \Rightarrow \varphi^i{}_s \wedge \varphi^t{}_s \wedge \varphi^g{}_s)$$

Environment assumptions

Desired guarantees

Unsatisfiable

UNSYNTHESIZABLE

Unrealizable

Two levels of analysis:

- identify subformulas that contribute
- compute minimal subformula causing failure

Form of Specification

$$\varphi = (\varphi_e \Rightarrow \varphi_s)$$

$$= (\varphi^i{}_e \wedge \varphi^t{}_e \wedge \varphi^g{}_e \Rightarrow \varphi^i{}_s \wedge \varphi^t{}_s \wedge \varphi^g{}_s)$$

Environment assumptions

Desired guarantees

UNSYNTHESIZABLE

Unsatisfiable

Unrealizable

Two levels of analysis:

- identify subformulas that contribute
- compute minimal subformula causing failure

Problem Statement

Highlight a MINIMAL cause of unsynthesizability

- Find a small subformula φ' of φ such that:

→ φ' is by itself unsynthesizable
(an unsynthesizable "core")

→ every proper subformula of φ' is synthesizable

Linear Temporal Logic (LTL)

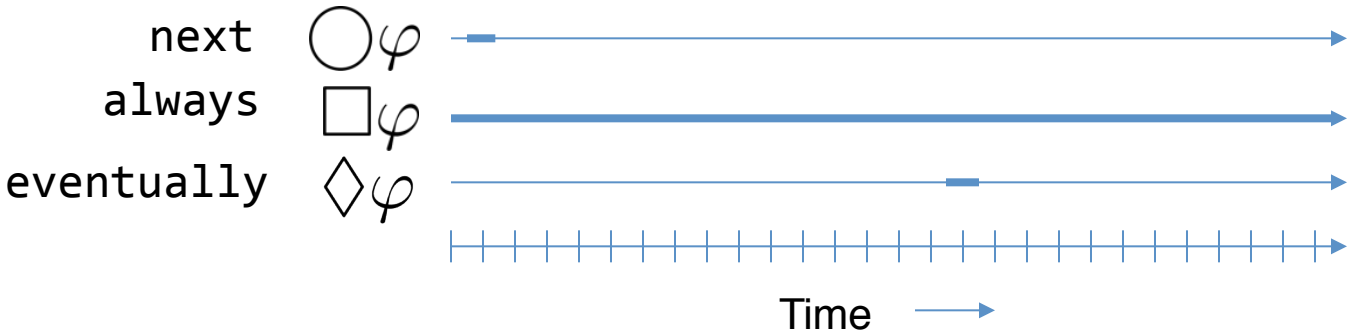
Syntax

AP is a set of atomic propositions, $\pi \in AP$

$$\varphi ::= \pi \mid \neg\varphi \mid \varphi \vee \varphi \mid \bigcirc\varphi \mid \Box\varphi \mid \Diamond\varphi \mid \varphi \mathcal{U} \varphi$$

Semantics

$\sigma \models \varphi$: infinite sequence of truth assignments σ satisfies φ



Unsatisfiable Cores

via propositional satisfiability (SAT)

General Idea:

1. "Unroll" LTL specification to some depth
   (encode as a propositional SAT problem)
2. Use off-the-shelf SAT solver to find MUS
3. Given a MUS, map it back to the LTL

Unsatisfiable Cores

via propositional satisfiability (SAT)

Unrolling the LTL specification:

- Fix unroll depth *d*
- Construct SAT instance
  - instantiate each atomic proposition for each time step from *0* to *d*
  - restrict the propositions at each time step

e.g. $\Box\varphi \rightarrow \varphi_0 \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \ldots \wedge \varphi_d$

$\texttt{always wave} \rightarrow \texttt{wave}_1 \wedge \texttt{wave}_2 \wedge \ldots \wedge \texttt{wave}_d$

Unsatisfiable Cores

via propositional satisfiability (SAT)

Unrolling the LTL specification:

- Fix unroll depth *d*
- Construct SAT instance

$$\Box\varphi \rightarrow \varphi_0 \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \ldots \wedge \varphi_d$$

e.g. `always wave` $\rightarrow \texttt{wave}_1 \wedge \texttt{wave}_2 \wedge \ldots \wedge \texttt{wave}_d$

Unsatisfiable Cores

via propositional satisfiability (SAT)

Use off-the-shelf SAT solver to find MUS:

- Input – unrolled specification in CNF form
- Output – Subset of CNF clauses

e.g. PicoSAT*

*http://fmv.jku.at/picosat/

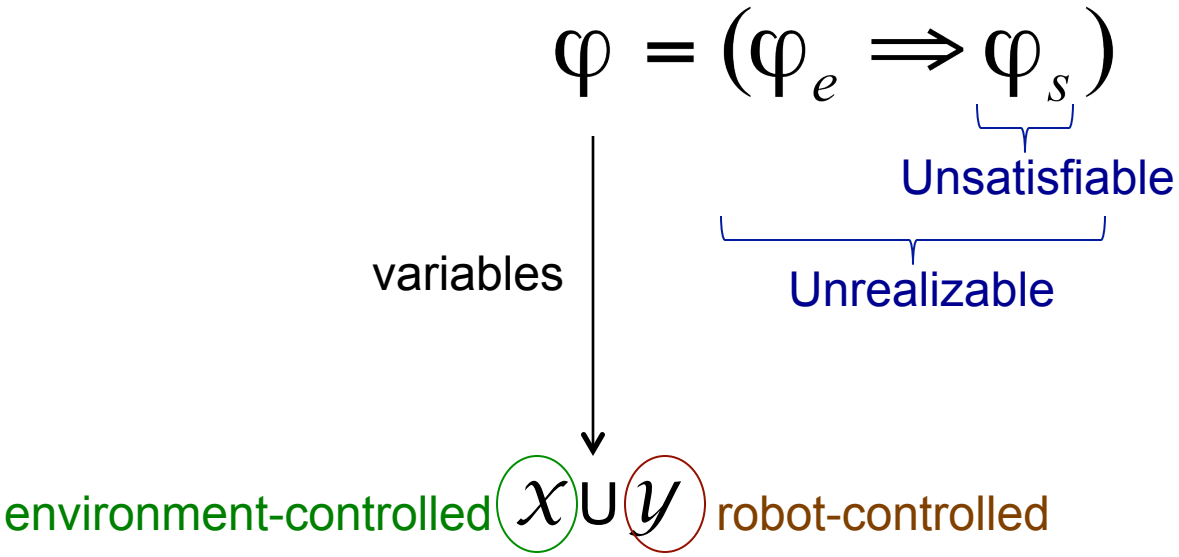
Unsatisfiable Cores

via propositional satisfiability (SAT)

Given an MUS, map it back to the LTL:

1. Track origin of each CNF clause
2. Depth of unrolling determines “core” found

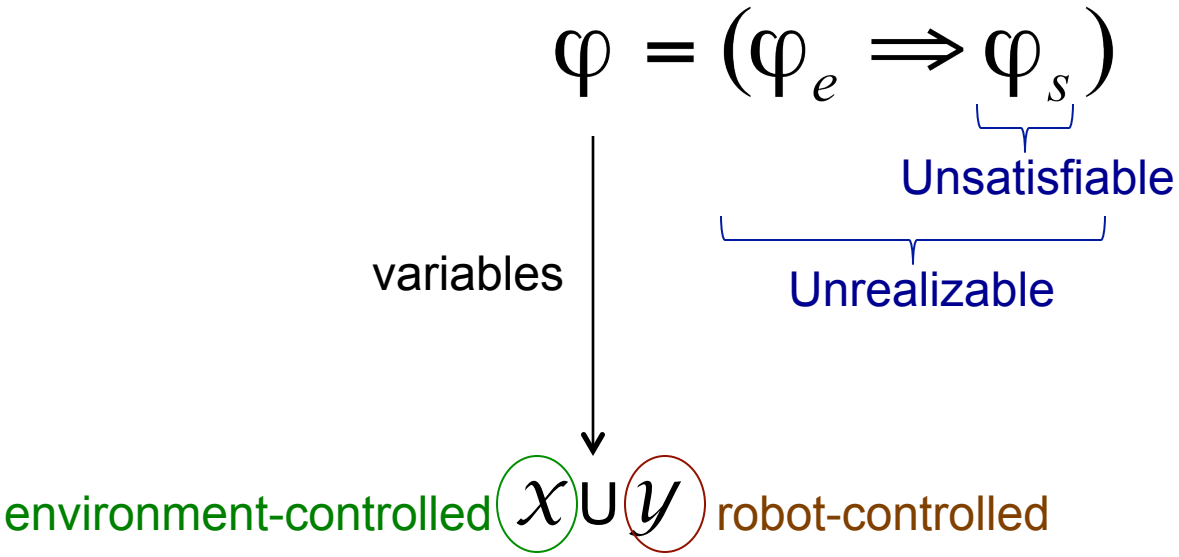
Unrealizable "Cores"



Unsatisfiability: no assignment to $\mathcal{X}$ U $\mathcal{Y}$ satisfies φ

Unrealizability: exists assignment to $\mathcal{X}$ such that no assignment to $\mathcal{Y}$ satisfies φ

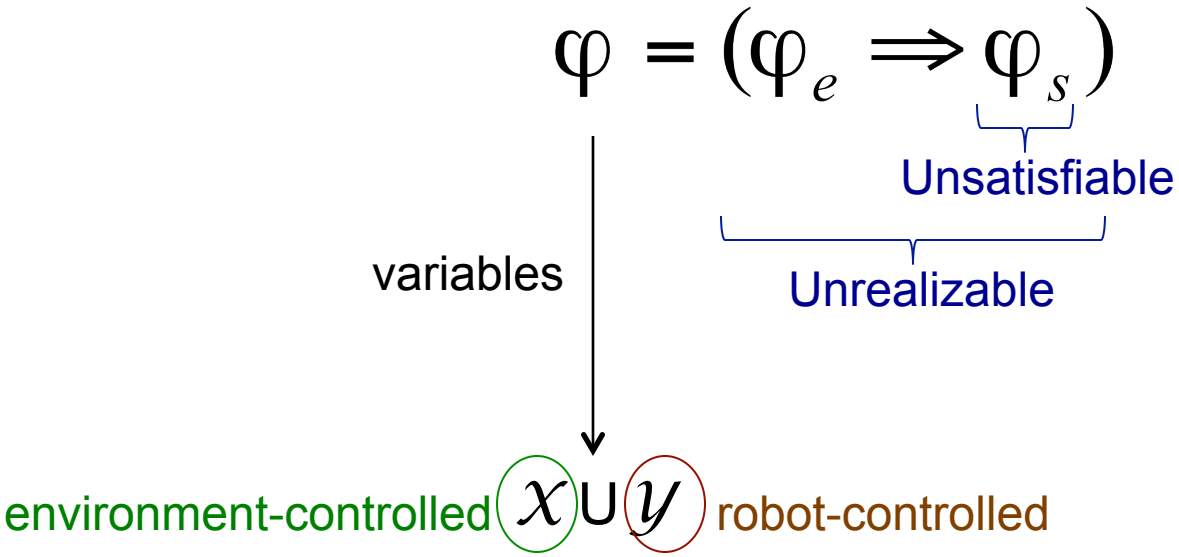
Unrealizable "Cores"



Can we still use SAT-based techniques?

Yes, but we need to restrict the environment variables $\mathcal{X}$ in the "right" way

Unrealizable "Cores"



Counterstrategy + SAT-based techniques:

1. Unroll the specification as before
2. Restrict inputs according to environment counterstrategy
3. Compute MUS of resulting SAT formula

Unrealizable “Cores”

Two types of unrealizability

- *Deadlock*: a state with no next robot move
  Blocking states in the counterstrategy graph
  (states with no out-transitions)

- *Livelock*: cycle of states that do not satisfy the goals
  Cycles in the counterstrategy graph
  Unroll to a certain depth to restrict environment

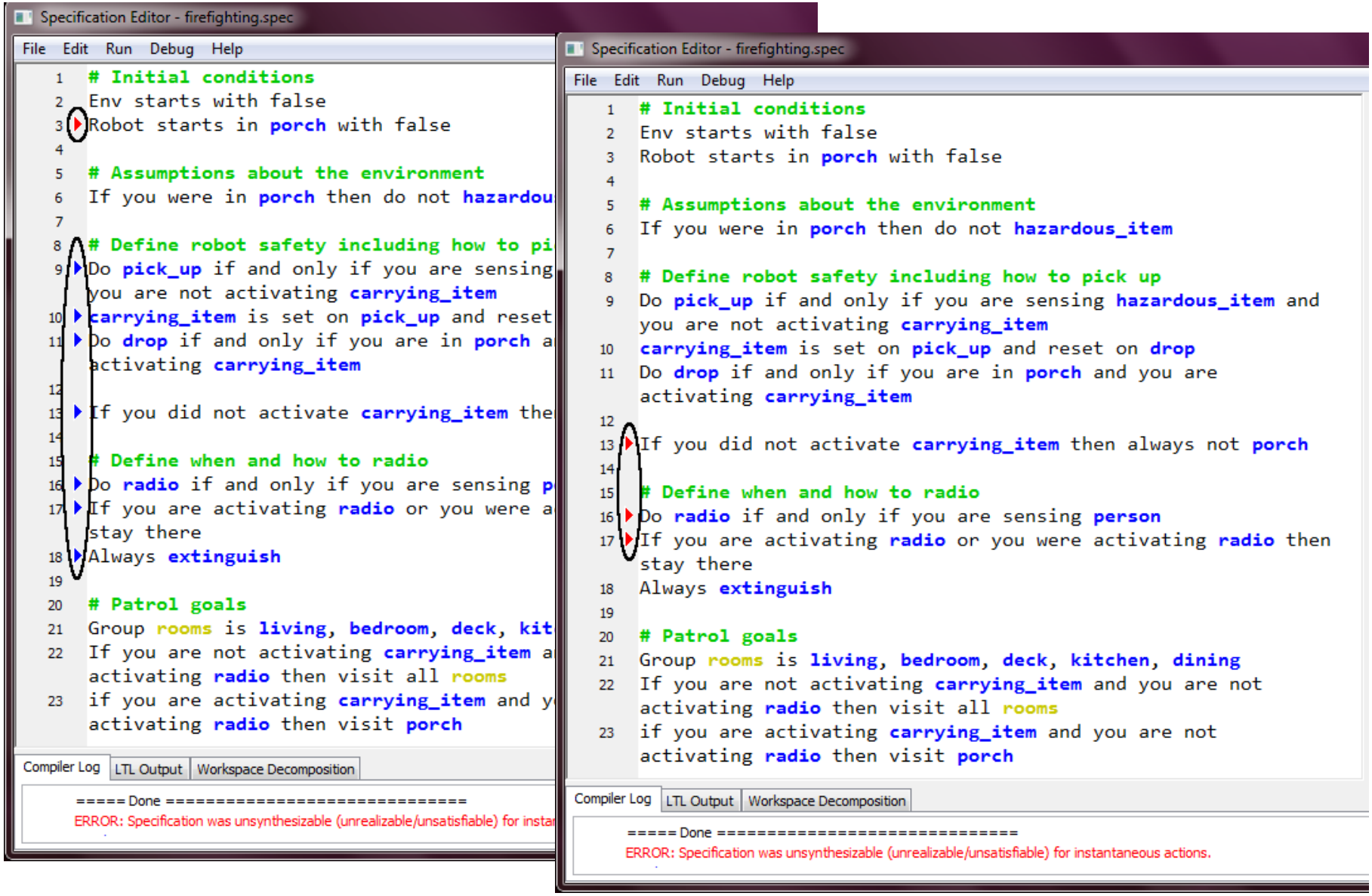
Unrealizable “Cores”

Iterated realizability tests:

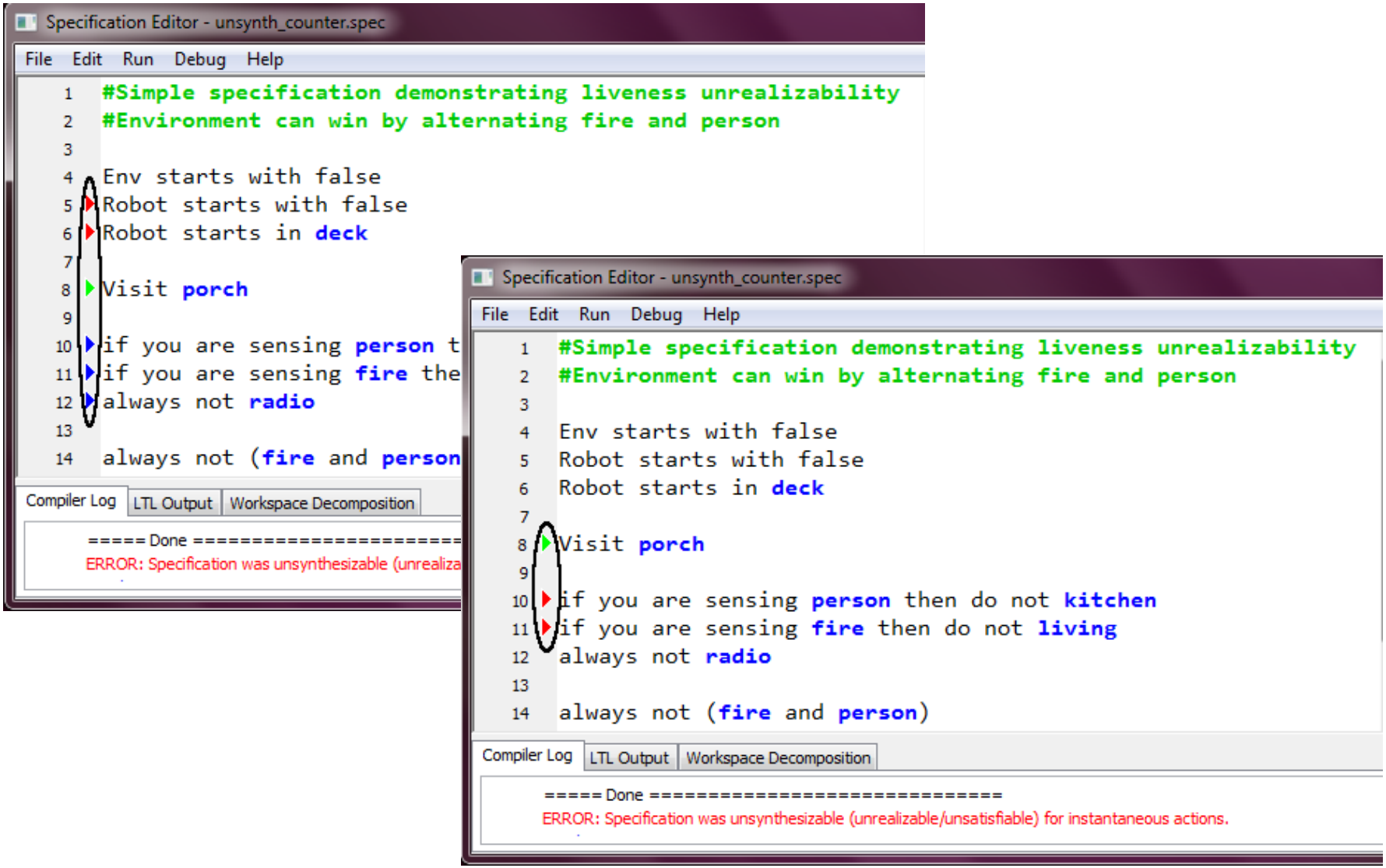
1. Remove conjunct *i* and test realizability
2. If the specification is still unrealizable, leave it out, otherwise add it back in
3. Repeat for *i++*

Remaining conjuncts form an unrealizable core

Unrealizable “Cores”



Unrealizable “Cores”



Analysis Output:

The problematic goal is 'Carry meals from the kitchen to all patient rooms.'. The system cannot achieve the sub-goal "Deliver 'meal' to 'r1'.".

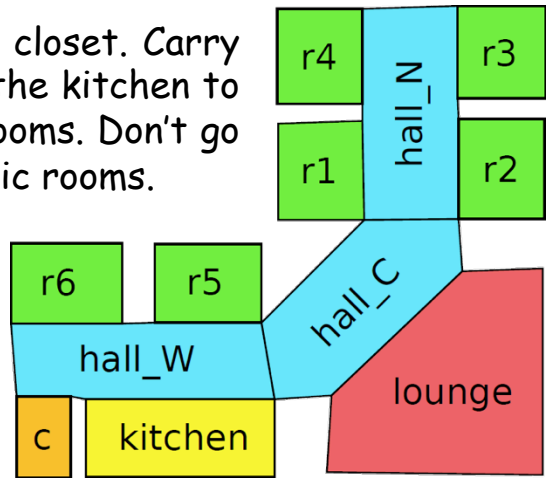
The statements that cause the problem are:
'Carry meals from the kitchen to all patient rooms.' because of item(s): "Deliver 'meal' to 'r1'.".
"Don't go into any public rooms." because of item(s): "Do not go to 'hall_c'.".

No further analysis available.

SLURP Traceback:

- Start in the closet.
- Carry meals from the kitchen to all patient rooms.
  - Action: 'carry', Argument: 'meal', Source: 'kitchen', Destination: 'rooms'
    - Nothing is carried or delivered at the start.
    - Only pick up if you can carry more.
    - Only drop if you are carrying something.
    - Stay where you are when picking up and dropping.
    - Pick up 'meal' in 'kitchen'.
    - Deliver 'meal' to 'r1'.
      - ([](next(s.mem_deliver_r1) <-> (s.mem_deliver_r1 | (next(s.r1) & next(s.drop))))))
      - ([]<>(s.mem_deliver_r1))
    - Deliver 'meal' to 'r2'.
    - Deliver 'meal' to 'r3'.
    - Deliver 'meal' to 'r4'.
    - Deliver 'meal' to 'r5'.
    - Deliver 'meal' to 'r6'.
- Don't go into any public rooms.
  - Action: do not 'go', Location: 'rooms'
    - Do not go to 'hall_c'.
      - ([](!s.hall_c))
    - The robot does not begin in 'hall_c'.
    - Do not go to 'lounge'.
    - The robot does not begin in 'lounge'.

Start in the closet. Carry meals from the kitchen to all patient rooms. Don't go into any public rooms.



Towards Minimal Explanations of Unsynthesizability for High-Level Robot Behaviors

Vasu Raman
vasu@caltech.edu

Hadas Kress-Gazit
hadaskg@cornell.edu