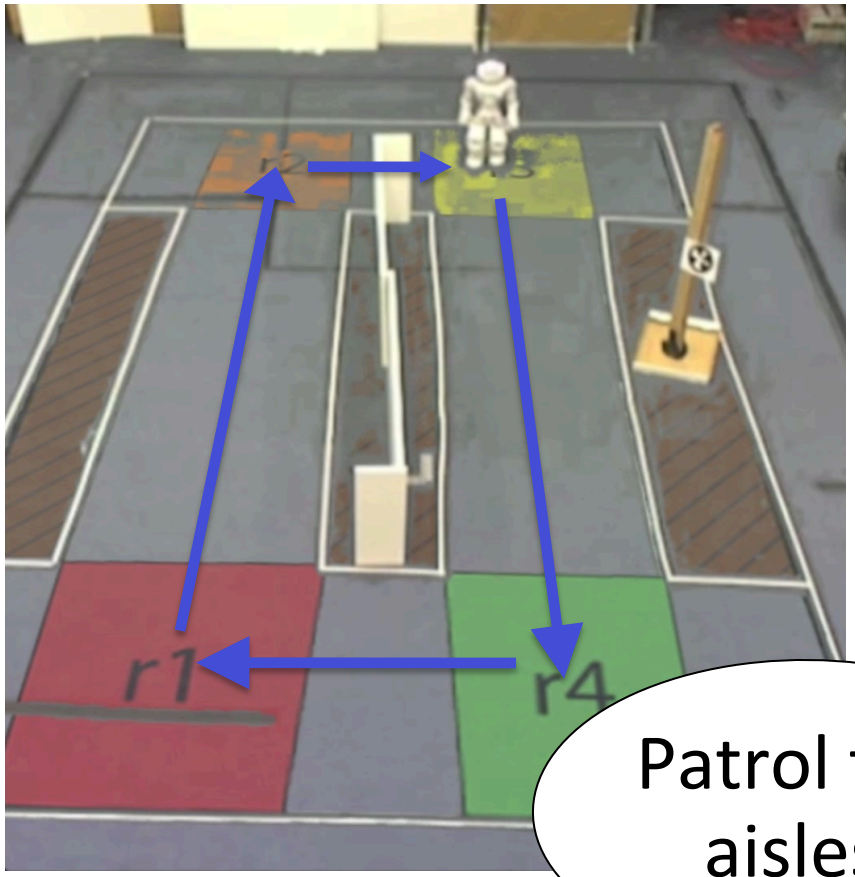
Avoiding Forgetfulness: Structured English Specifications for High-Level Robot Control with Implicit Memory

Vasu Raman
Bingxin Xu
Hadas Kress-Gazit
(presented by Cameron Finucane)

Cornell University

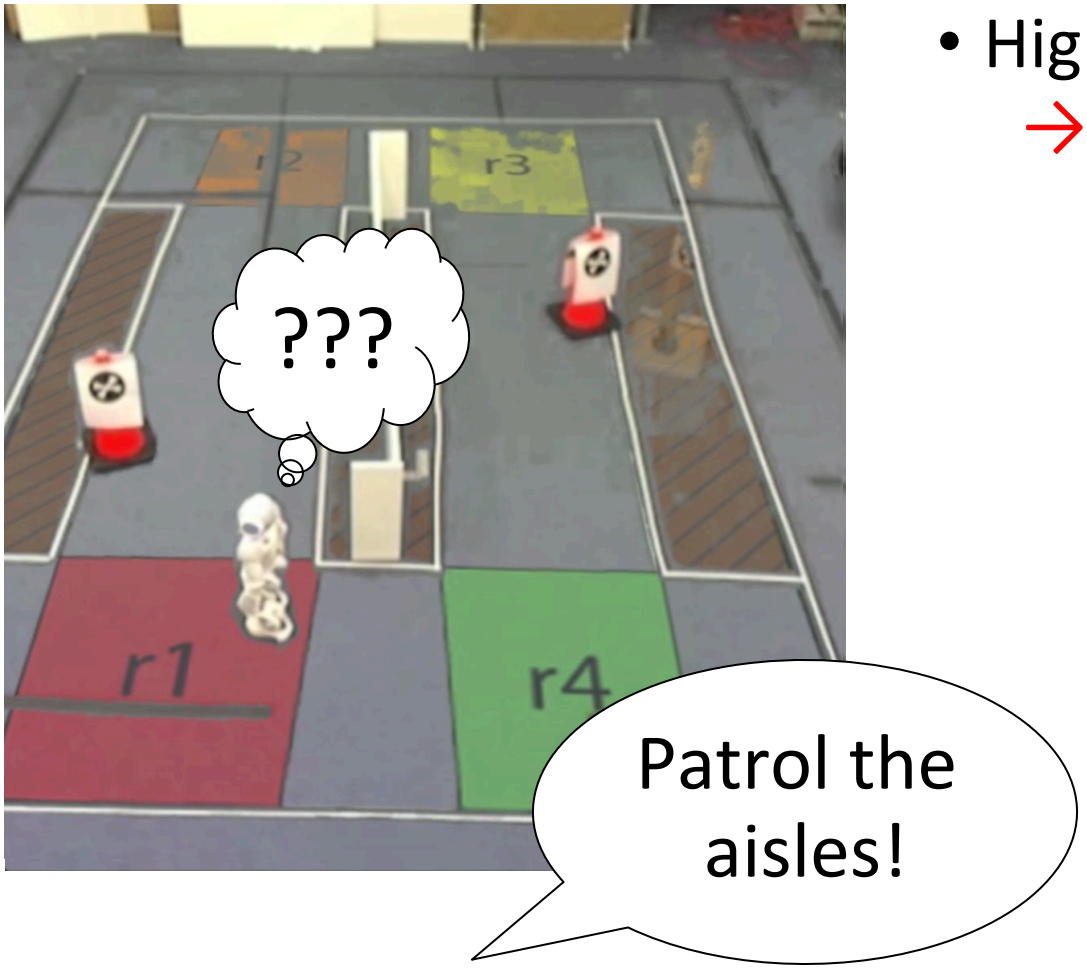
Motivation



Patrol the aisles!

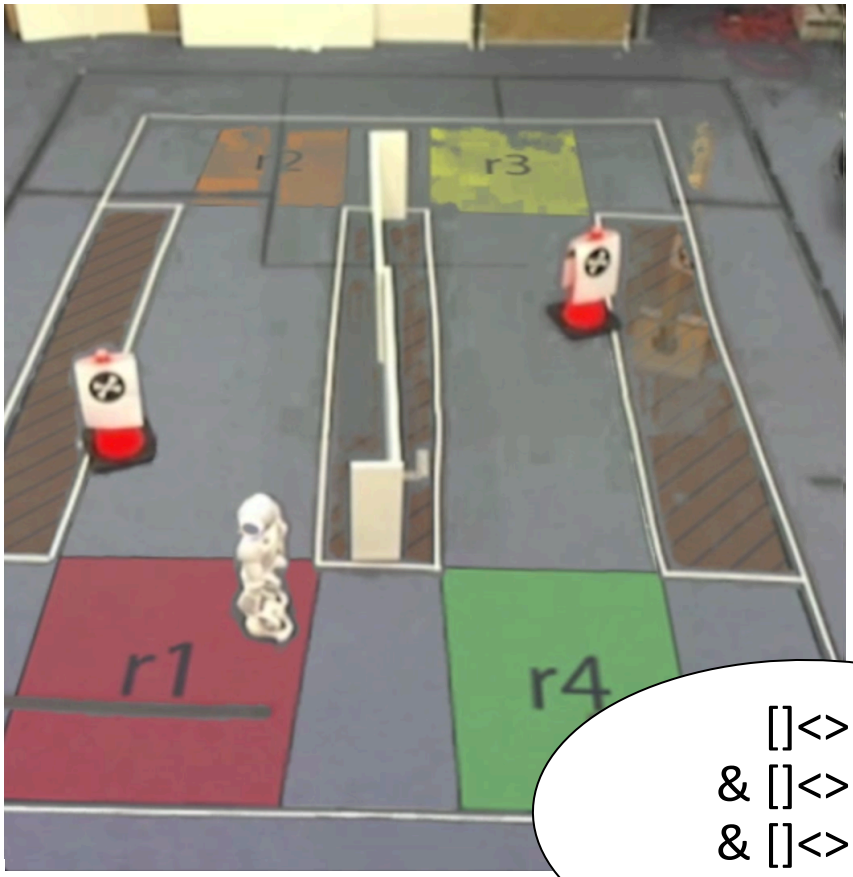
- High-level robot tasks

Motivation



- High-level robot tasks
  → No guarantees!

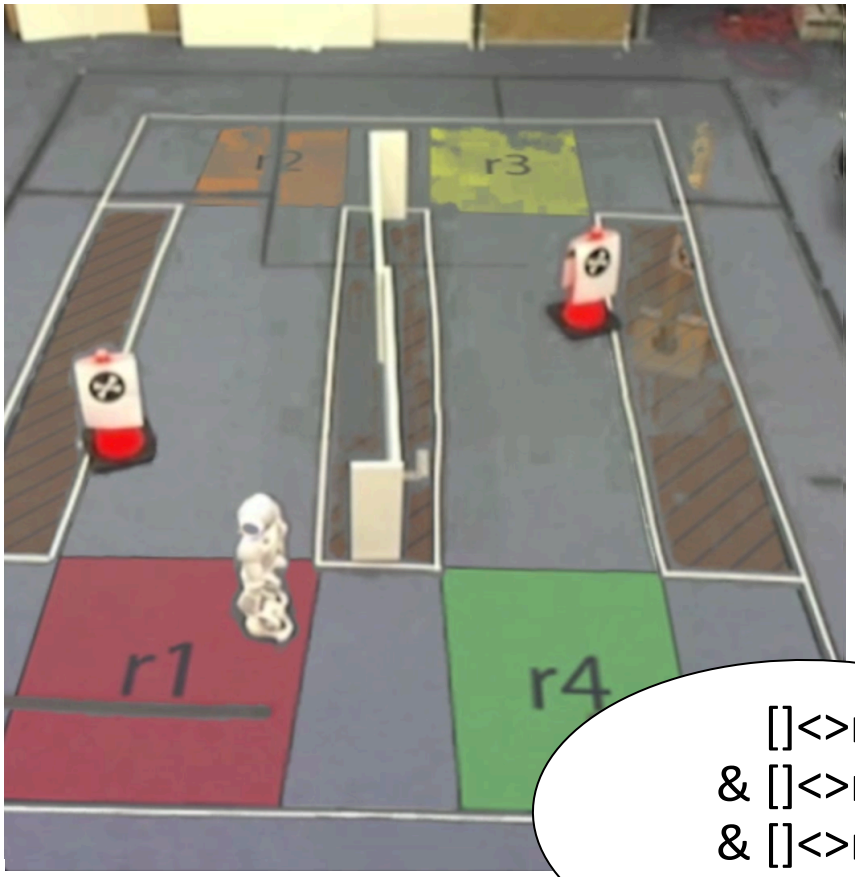
Motivation



[]<>r1
& []<>r2
& []<>r3
& []<>r4!

- High-level robot tasks
  → No guarantees!
- Synthesis from formal specifications

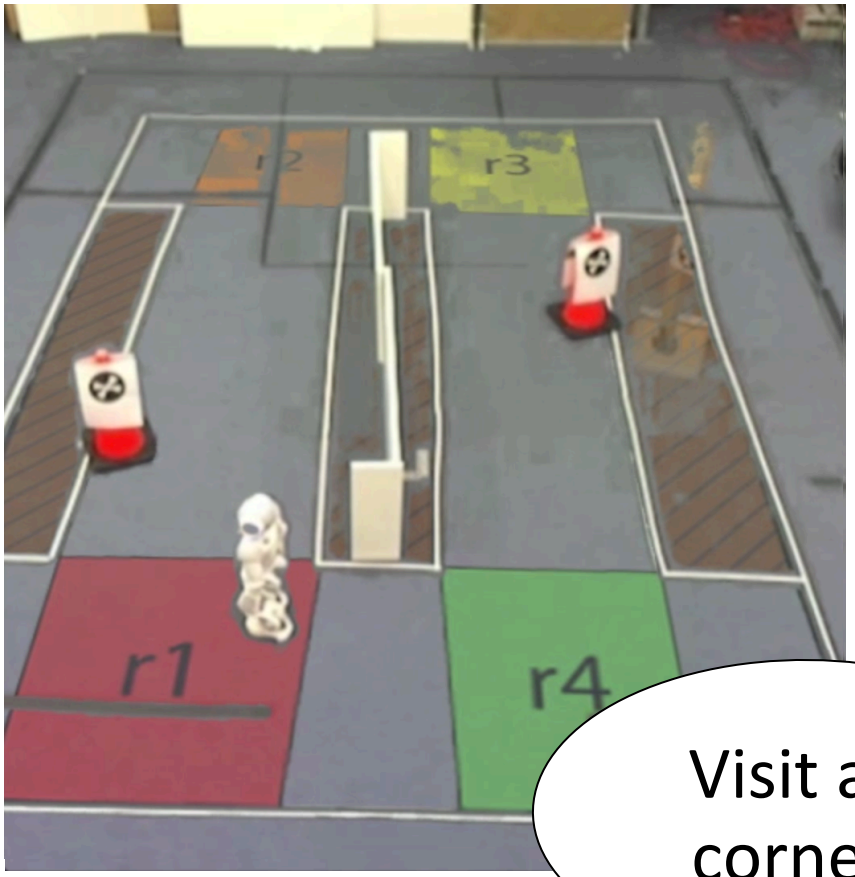
Motivation



[]<>r1
& []<>r2
& []<>r3
& []<>r4!

- High-level robot tasks
  → No guarantees!
- Synthesis from formal specifications
  → Unintuitive!

Motivation



- High-level robot tasks
  → No guarantees!
- Synthesis from formal specifications
  → Unintuitive!
- Intuitive human interface

Linear Temporal Logic Mission Planning Toolkit (LTLMoP)

- Structured English input → Correct robot control
- Grammar allows:
  - Conditionals (“*if*”, “*if and only if*”)
  - Locative prepositions (“*between*”, “*near*”)
  - Region quantifiers (“any”, “all”)
  - Goals (“*visit all checkpoints*”)
  - Safety requirements (“*avoid the kitchen*”)

Cameron Finucane, Gangyuan Jing, and Hadas Kress-Gazit. LTLMoP: Experimenting with language, temporal logic and robot control, IROS 2010.

LTLMoP – behind the scenes

- Structured English input
  → Linear Temporal Logic formulas

- Tied tightly to underlying formalism (LTL fragment)

Linear Temporal Logic

- Syntax:

$$\varphi ::= \pi \mid \neg\varphi \mid \varphi \vee \varphi \mid \bigcirc \varphi \mid \Box \varphi \mid \Diamond \varphi$$

- Temporal operators:

$\bigcirc \varphi$ next step

$\Box \varphi$ always

$\Diamond \varphi$ eventually

Example

“If you are given an order then go to the kitchen”

Example

"If you are given an order then go to the kitchen"

What is the right LTL formula to capture this?

Example

"If you are given an order then go to the kitchen"

What is the right LTL formula to capture this?

- Initial guess (direct translation):

If you are sensing order then visit kitchen

$$\Box\Diamond(\pi_{order} \Rightarrow \pi_{kitchen})$$

Example

"If you are given an order then go to the kitchen"

Implicit memory:

- Need to remember an order was received

Example

"If you are given an order then go to the kitchen"

What is the right LTL formula to capture this?

- Specification for desired behavior

m_order is set on order and never reset

$$\Box(\bigcirc m_{order} \Leftrightarrow (\bigcirc \pi_{order} \vee m_{order}))$$

If you are activating m_order then visit kitchen

$$\Box\Diamond(m_{order} \Rightarrow \pi_{kitchen})$$

This paper

- Allow users to specify tasks that include event memory
- Automatically define “memory propositions”

Memory Propositions

- Implicit
  - Not defined by user
  - Only appear in LTL, not structured English
- Respond to the explicitly specified event
  - m_ϕ responds to event ϕ
  - Example:

$$\square(\bigcirc m\_\phi \Leftrightarrow (\bigcirc \phi \vee m\_\phi))$$

Grammar for implicit memory

Type	What to remember?	Structured English ($\mathcal{S}$)	LTL ($\mathcal{M}, \Phi$)
1	Condition has happened	Once Θ_{cond} then Θ_{req_safe} from now on After Θ_{cond} then Θ_{req_live} repeatedly	$\square(m\_\phi_{cond} \Rightarrow \phi_{req\_safe}) \wedge$ $\square(\bigcirc m\_\phi_{cond} \Leftrightarrow (\bigcirc \phi_{cond} \vee m\_\phi_{cond})$ $\square \diamond (m\_\phi_{cond} \Rightarrow \phi_{req\_live}) \wedge$ $\square(\bigcirc m\_\phi_{cond} \Leftrightarrow (\bigcirc \phi_{cond} \vee m\_\phi_{cond})$
2	Requirement has happened	Θ_{req} (at least once)	$\square \diamond (m\_\phi_{req})$ $\square(\bigcirc m\_\phi_{req} \Leftrightarrow (\bigcirc \phi_{req} \vee m\_\phi_{req})$
3	Requirement has happened under certain condition	While Θ_{cond} then Θ_{req} (at least once)	$\Delta(\phi_{cond} \Rightarrow m\_\phi_{cond} \phi_{req})$ $\square(\bigcirc m\_\phi_{cond}\_\phi_{req} \Leftrightarrow (\bigcirc \phi_{req} \wedge \phi_{cond} \vee m\_\phi_{req}))$
4	Memo is set on Θ_1 and reset on Θ_2	After/once Θ_1 then Θ_{req} until Θ_2	$\Delta(m\_\phi_1 \phi_2 \Rightarrow \phi_{req})$ $\square(\bigcirc m\_\phi_1 \phi_2 \Leftrightarrow ((\bigcirc \phi_1 \vee m\_\phi_1 \phi_2) \wedge \neg \bigcirc \phi_2)$
*1	'Only'+cond	Only once Θ_{cond} then Θ_{req_safe} from now on Only after Θ_{cond} then Θ_{req_live} repeatedly	LTL in Type 1 + $\square((\neg \bigcirc m\_\phi_{cond}) \Rightarrow (\neg \bigcirc \phi_{req}))$
*2	requirement + 'only once'	Eventually Θ_{req_live} only once	LTL in Type 2 + $\square(m\_\phi_{req} \Rightarrow (\neg \bigcirc \phi_{req}))$
*3	requirement under condition + 'only once'	If Θ_{cond} then eventually Θ_{req_live} only once If Θ_{cond} then Θ_{req_safe} only once	LTL in Type 3 + $\square(m\_\phi_{cond}\_\phi_{req} \Rightarrow \neg \bigcirc \phi_{req})$
*4	Memo is self-reset when the requirement is met	After each time Θ_{cond}, Θ_{req} (at least once)	$\Delta(m\_\phi_{cond} \phi_{req} \Rightarrow \phi_{req})$ $\square(\bigcirc m\_\phi_{cond} \phi_{req} \Leftrightarrow ((\bigcirc \phi_{cond} \vee m\_\phi_{cond} \phi_{req}) \wedge$ $\neg \bigcirc \phi_{req})$
*5	Condition-Requirement memos on both sides	After the first time Θ_{cond}, Θ_{req} (at least once)	$\square(\bigcirc m\_\phi_{cond} \Leftrightarrow (\bigcirc \phi_{cond} \vee m\_\phi_{cond}))$ $\square(\bigcirc m\_\phi_{cond}\_\phi_{req} \Leftrightarrow ((\bigcirc \phi_{req} \wedge \bigcirc m\_\phi_{cond})$ $\vee m\_\phi_{req})$ $\Delta(m\_\phi_{cond} \Rightarrow m\_\phi_{cond}\_\phi_{req})$

Grammar for implicit memory

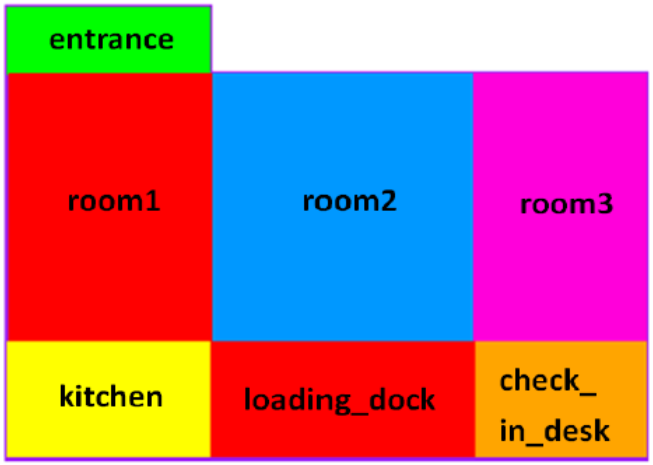
Type	What to		
1	Condit		
2	Requir		
3	Requir under		
4	Memo and re		
*1	'Only'		
*2	requirement		
*3	requirement und + 'only once'		
*4	Memo is self-reset wh the requirement is met		
*5	Condition-Requirement memos on both sides	After the first time Θ_{cond}, Θ_{req} (at least once)	$\square(\bigcirc m\_\phi_{cond} \Leftrightarrow (\bigcirc \phi_{cond} \vee m\_\phi_{cond}))$ $\square(\bigcirc m\_\phi_{cond}\_\phi_{req} \Leftrightarrow ((\bigcirc \phi_{req} \wedge \bigcirc m\_\phi_{cond}) \vee m\_\phi_{req})$ $\Delta(m\_\phi_{cond} \Rightarrow m\_\phi_{cond}\_\phi_{req})$

After the first time Θ_{cond},
Θ_{req} (at least once)

$$\square(\bigcirc m\_\phi_{cond} \Leftrightarrow (\bigcirc \phi_{cond} \vee m\_\phi_{cond}))$$
$$\square(\bigcirc m\_\phi_{cond}\_\phi_{req} \Leftrightarrow ((\bigcirc \phi_{req} \wedge \bigcirc m\_\phi_{cond}) \vee m\_\phi_{req})$$
$$\Delta(m\_\phi_{cond} \Rightarrow m\_\phi_{cond}\_\phi_{req})$$

Example: Robot Waiter

- First go to the check-in desk
- Meet the first truck at the loading dock, but ignore all following trucks



- When customers arrive, move between the three dining rooms until accepting an order
- Each time an order is made, go to the kitchen

Task 2

Meet the first truck at the loading dock, but ignore all following trucks

- Structured English:

  After the first time you have sensed truck, go to loading_dock

- Grammar: After the first time Θ_{cond}, Θ_{req} (at least once)

- LTL: $\Box(\bigcirc m_{truck} \Leftrightarrow (\bigcirc \pi_{truck} \vee m_{truck}))$

  $\wedge \quad \Box(\bigcirc m_{truck\_dock} \Leftrightarrow ((\bigcirc \pi_{dock} \wedge \bigcirc m_{truck}) \vee m_{truck\_dock})$

  $\wedge \quad \Box\Diamond(m_{truck} \Rightarrow m_{truck\_dock})$

Task 3

When customers arrive, move between the three dining rooms until accepting an order.

- Structured English:

  After you have sensed customer then visit all dining_rooms until you are sensing order.

- Grammar: After/once Θ_1 then Θ_{req} until Θ_2

- LTL: $\Box(\bigcirc m_{cust\_no\_order} \Leftrightarrow ((\bigcirc \pi_{cust} \vee m_{cust\_no\_order}) \wedge \neg \bigcirc \pi_{order}))$

  $\wedge \quad \wedge_{i=1,2,3} \Box \Diamond (m_{cust\_no\_order} \Rightarrow \pi_{r_i})$

Complete Specification

Old Grammar

- Do memo_check_in if and only if you are in check_in_desk or you were activating memo_check_in
- Repeatedly visit memo_check_in
- Do memo_truck if and only if you are sensing truck or you were activating memo_truck
- Do memo_dock if and only if you are in loading_dock or you were activating memo_dock
- If you are activating memo_truck then visit memo_dock
- Do memo_customer if and only if (you are sensing customer or you were activating memo_customer) and you are not sensing order
- Group dining_rooms is room1, room2, room3
- If you are activating memo_dock then visit all dining_rooms
- Do memo_order if and only if (you are sensing order or you were activating memo_order) and you are not in kitchen
- If you are activating memo_order then visit kitchen

Word Count: 121

New Grammar

- Go to check_in_desk.
- After the first time you have sensed truck, go to loading_dock.
- Group dining_rooms is room1, room2, room3.
- After you have sensed customer then visit all dining_rooms until you are sensing order.
- After each time you have sensed order, go to kitchen.

Word Count: 44

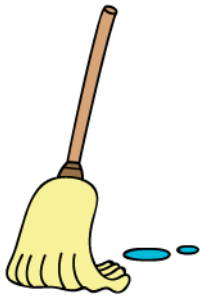
Conclusions

- Enriched Structured English grammar for memory
  - Specifications translate to LTL
  - Automatic creation of memory propositions
- Accommodates several types of events
- More concise, intuitive specifications

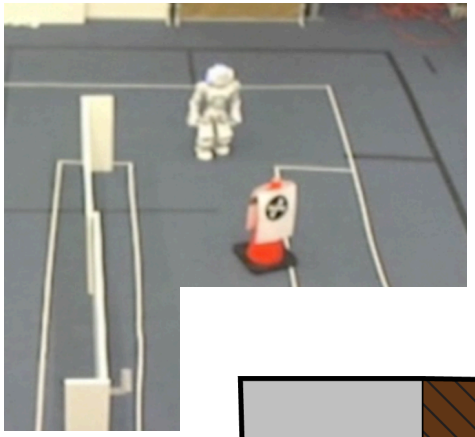
Avoiding Forgetfulness: Structured English Specifications for High-Level Robot Control with Implicit Memory

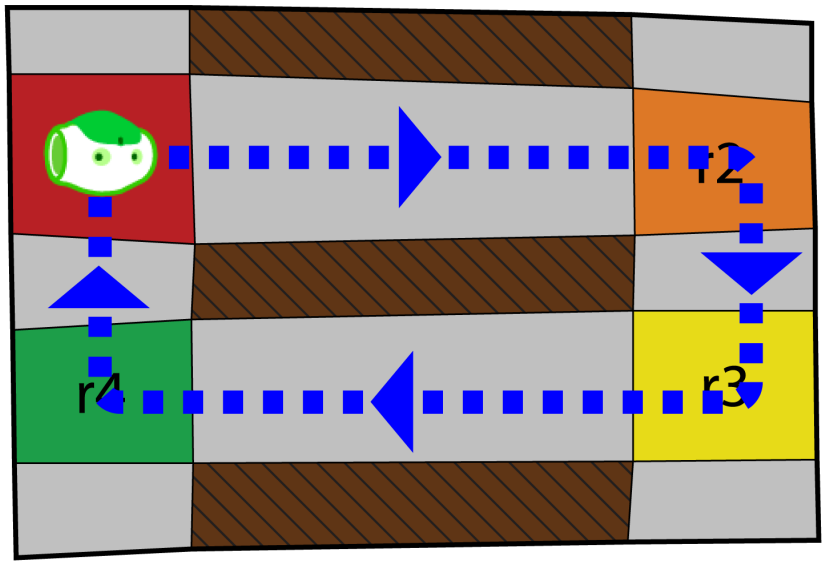
Vasu Raman (vraman@cs.cornell.edu)
Bingxin Xu (bx38@cornell.edu)
Hadas Kress-Gazit (hadaskg@cornell.edu)
Cameron Finucane (cpf37@cornell.edu)

Cornell University



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




r2
r4
r3