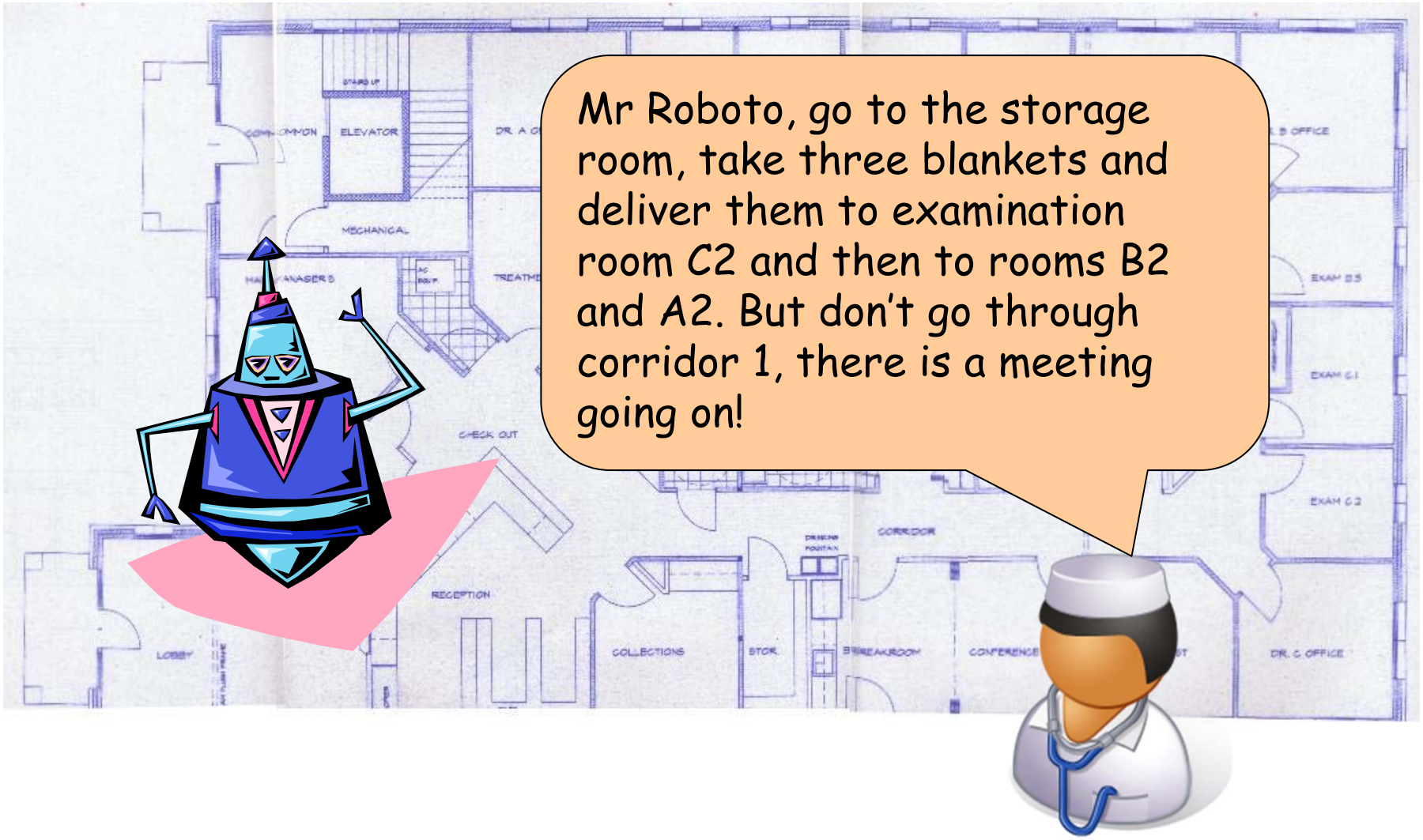
Automatically Modifying Conflicting Specifications

Georgios E. Fainekos
Arizona State University

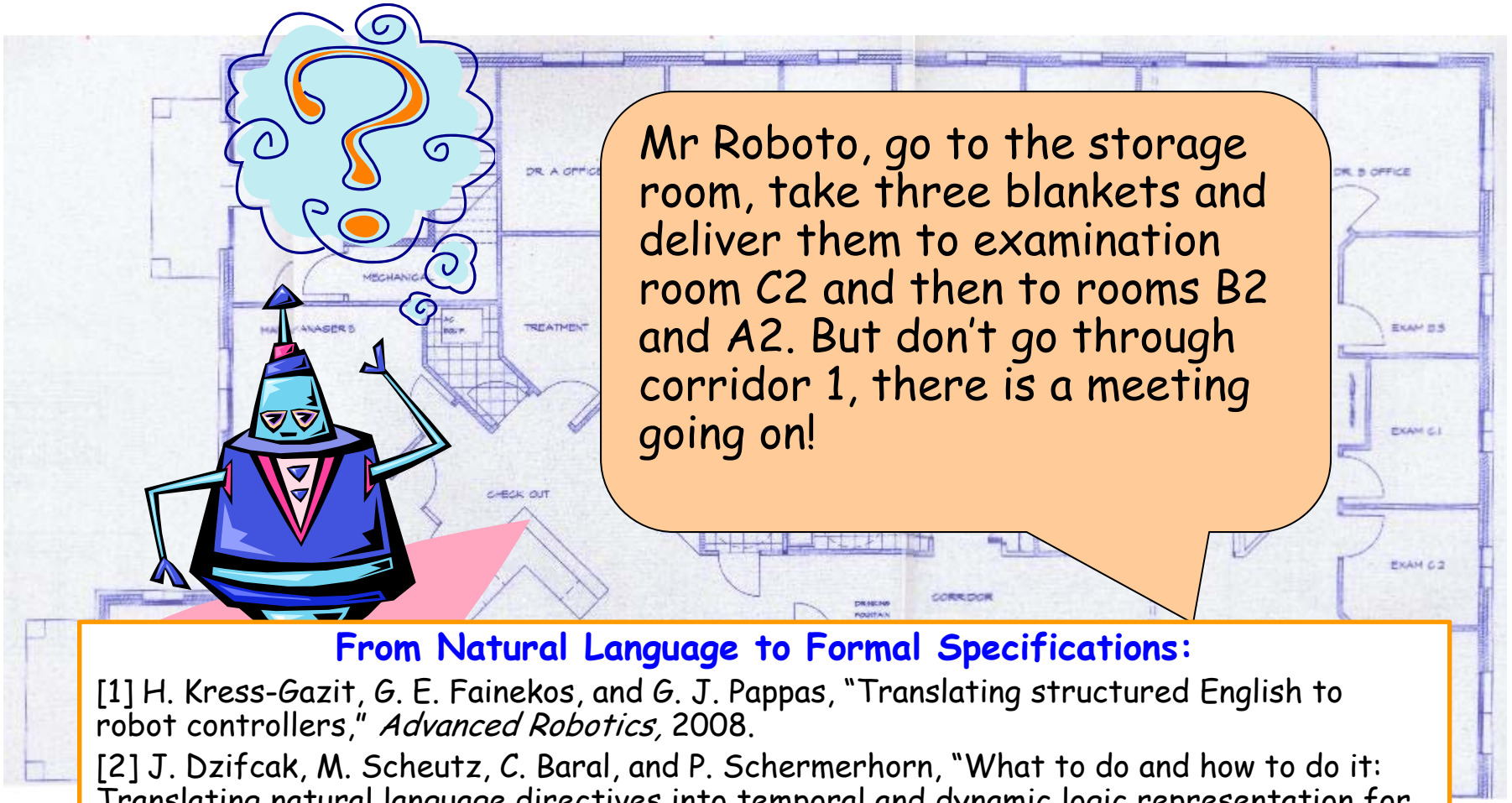
fainekos at asu edu

http://www.public.asu.edu/~gfaineko

Humans & Robots in the near future



Humans & Robots in the near future



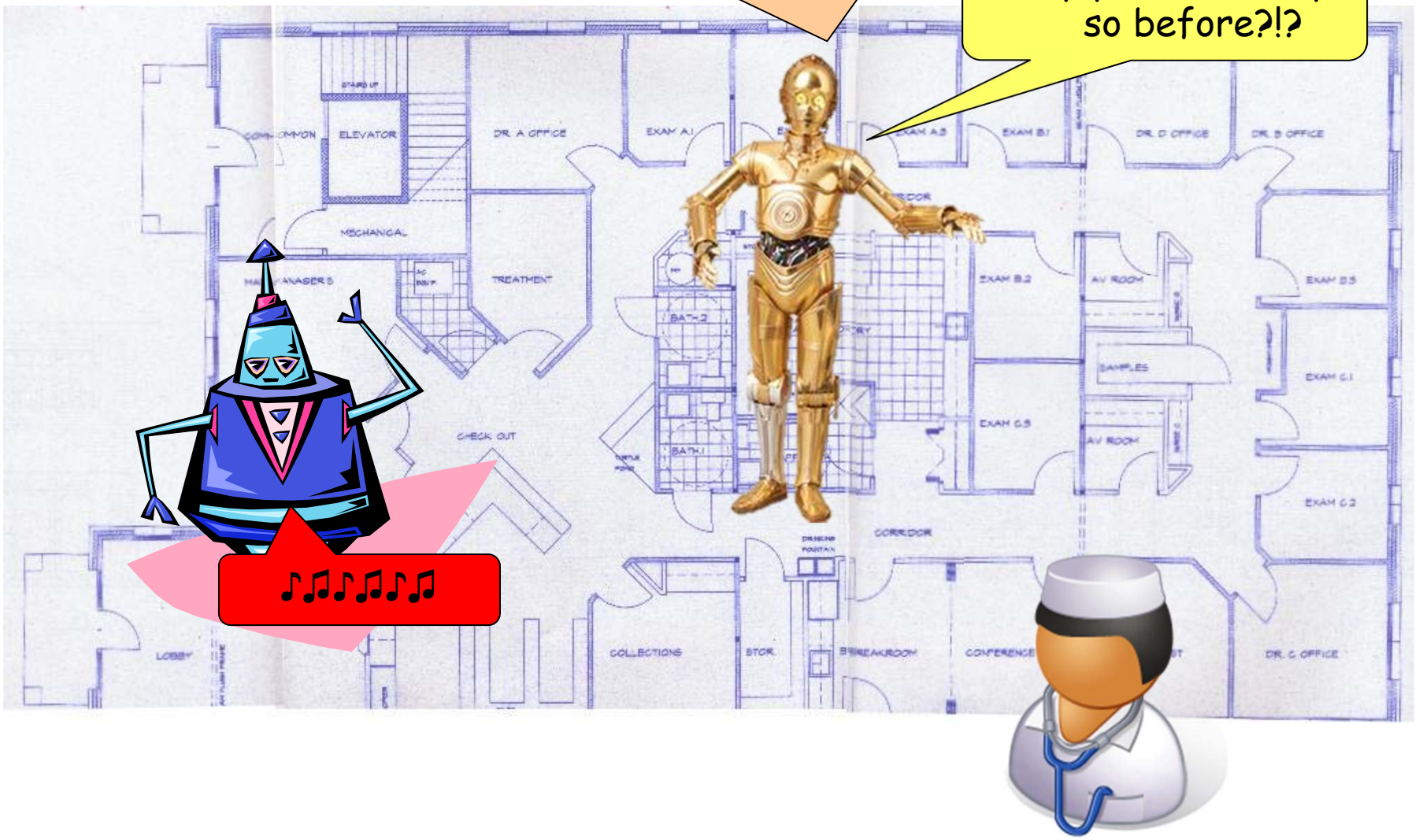
From Natural Language to Formal Specifications:

[1] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Translating structured English to robot controllers," *Advanced Robotics,* 2008.

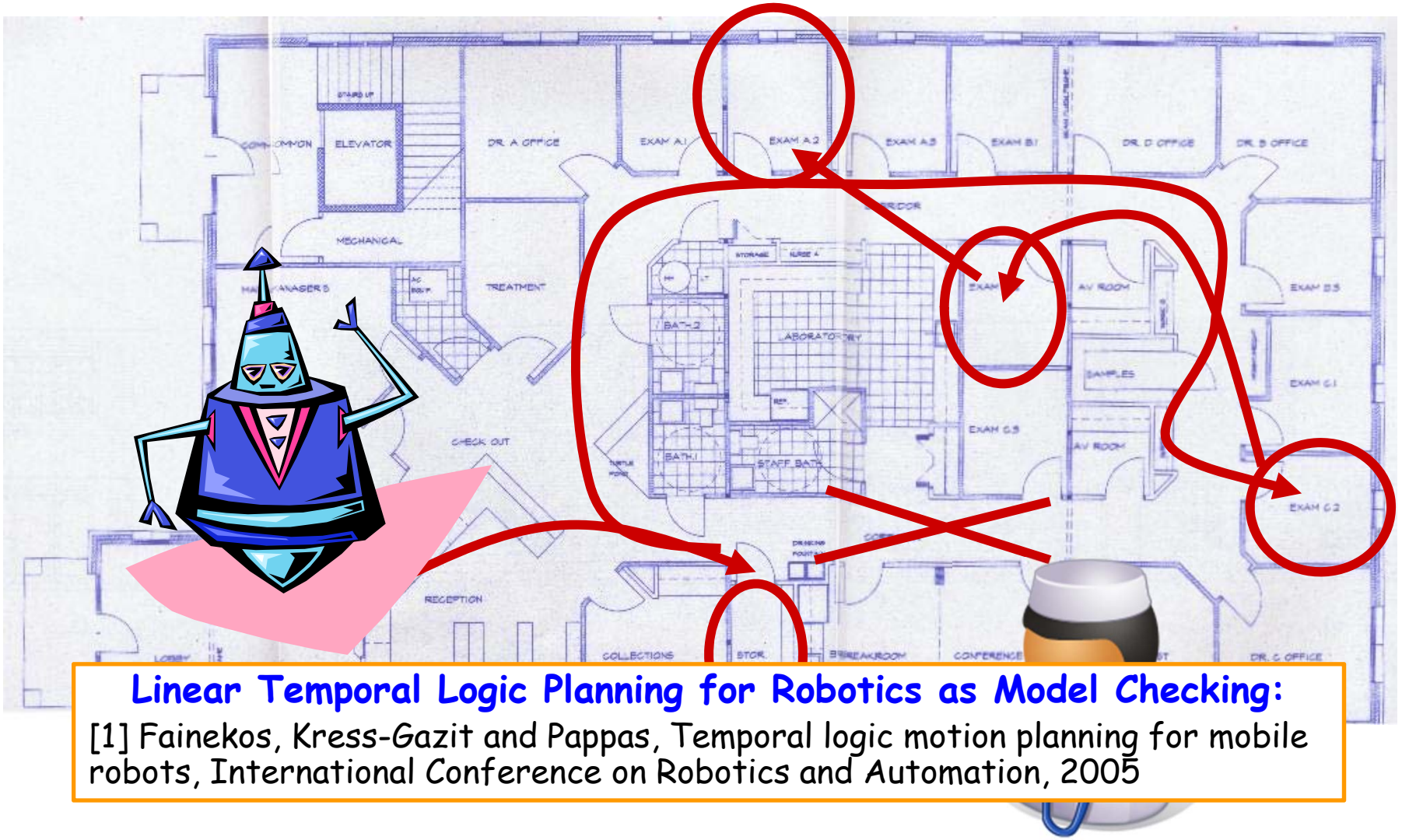
[2] J. Dzifcak, M. Scheutz, C. Baral, and P. Schermerhorn, "What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution," ICRA 2009.

Hu ... r future

$G(\neg corr\_1) \wedge F(stor \wedge F(exam\_C2 \wedge F\ exam\_A2 \wedge F\ exam\_B2\ )$



Humans & Robots in the near future



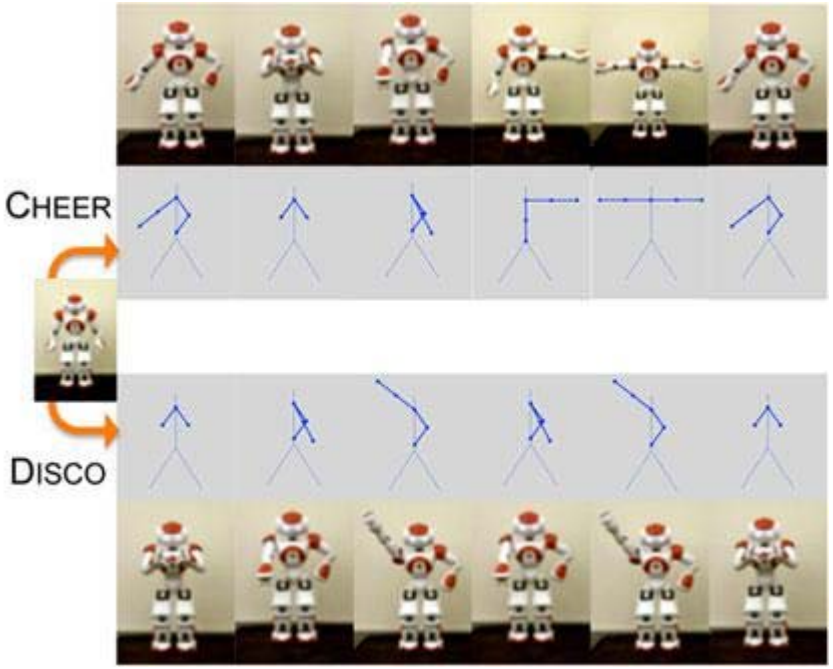
Linear Temporal Logic Planning for Robotics as Model Checking:

[1] Fainekos, Kress-Gazit and Pappas, Temporal logic motion planning for mobile robots, International Conference on Robotics and Automation, 2005

Linear Temporal Logic Planning for Robotics as Model Checking

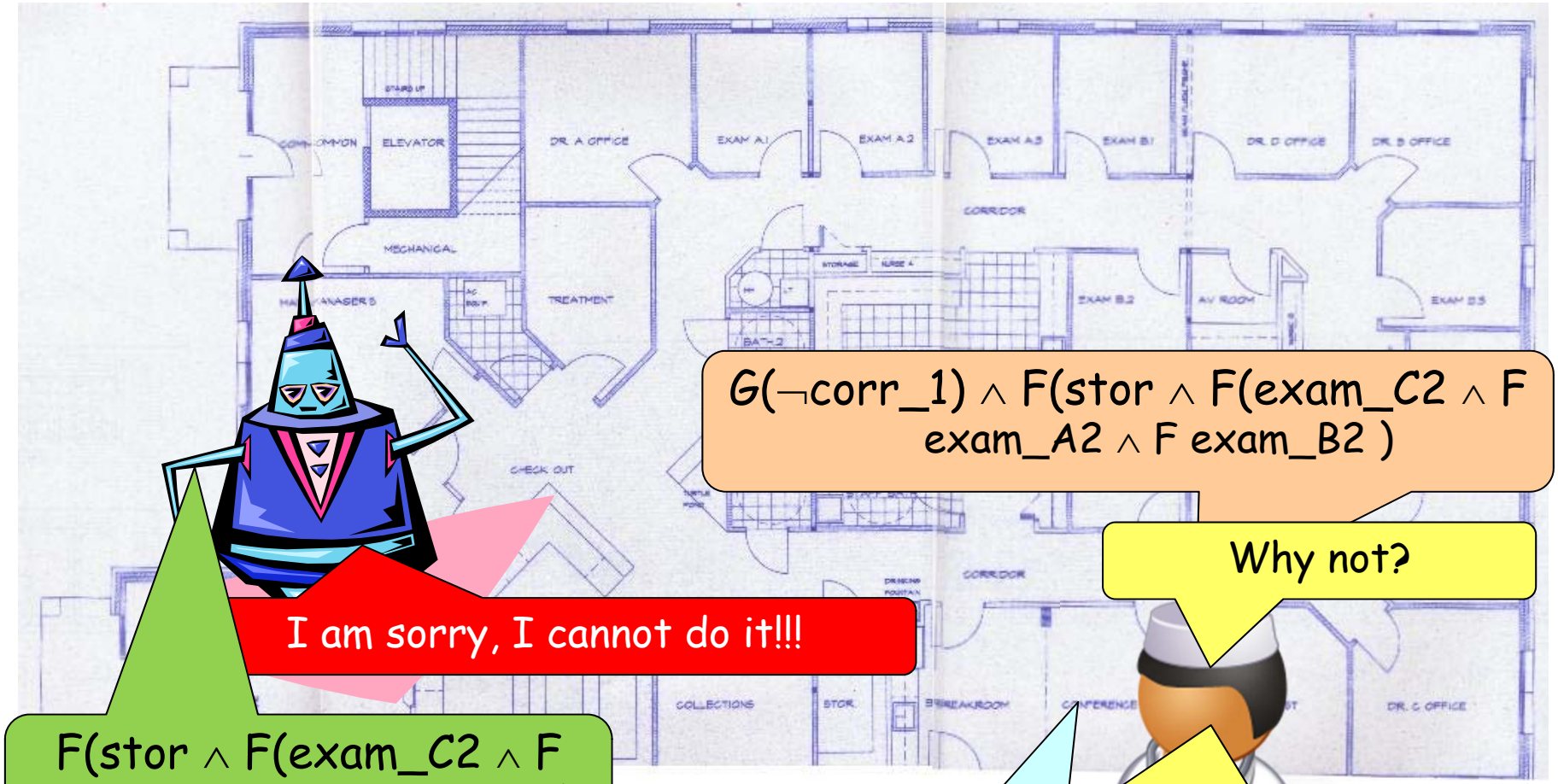
Lahijanian, Kloetzer, Itani, Belta, Andersson, ICRA 2009

Bobadilla, Sanchez, Czarnowski, Gossman, LaValle, RSS 2011



LaViers, Chen, Belta, Egerstedt, IEEE RAM 2011

Humans & Robots in the near future



Overview & Problem Statement

Environment

Specification in Natural Language

Linear Temporal Logic

Robot(s) model

$\dot{x} = f(x, p, u)$
$y = g(x, p, u)$
$X_0 \subseteq X$

TL Planning Algorithm Based on Model Checking

Planning succeeds

Built Feedback controllers

This presentation

Planning fails

Return to the user:

1. Why the specification failed
2. A specification that can be realized on the system
   - ***and it is as close as possible to the initial intention of the user***

Linear Temporal Logic: Semantic Intuition

a – a now

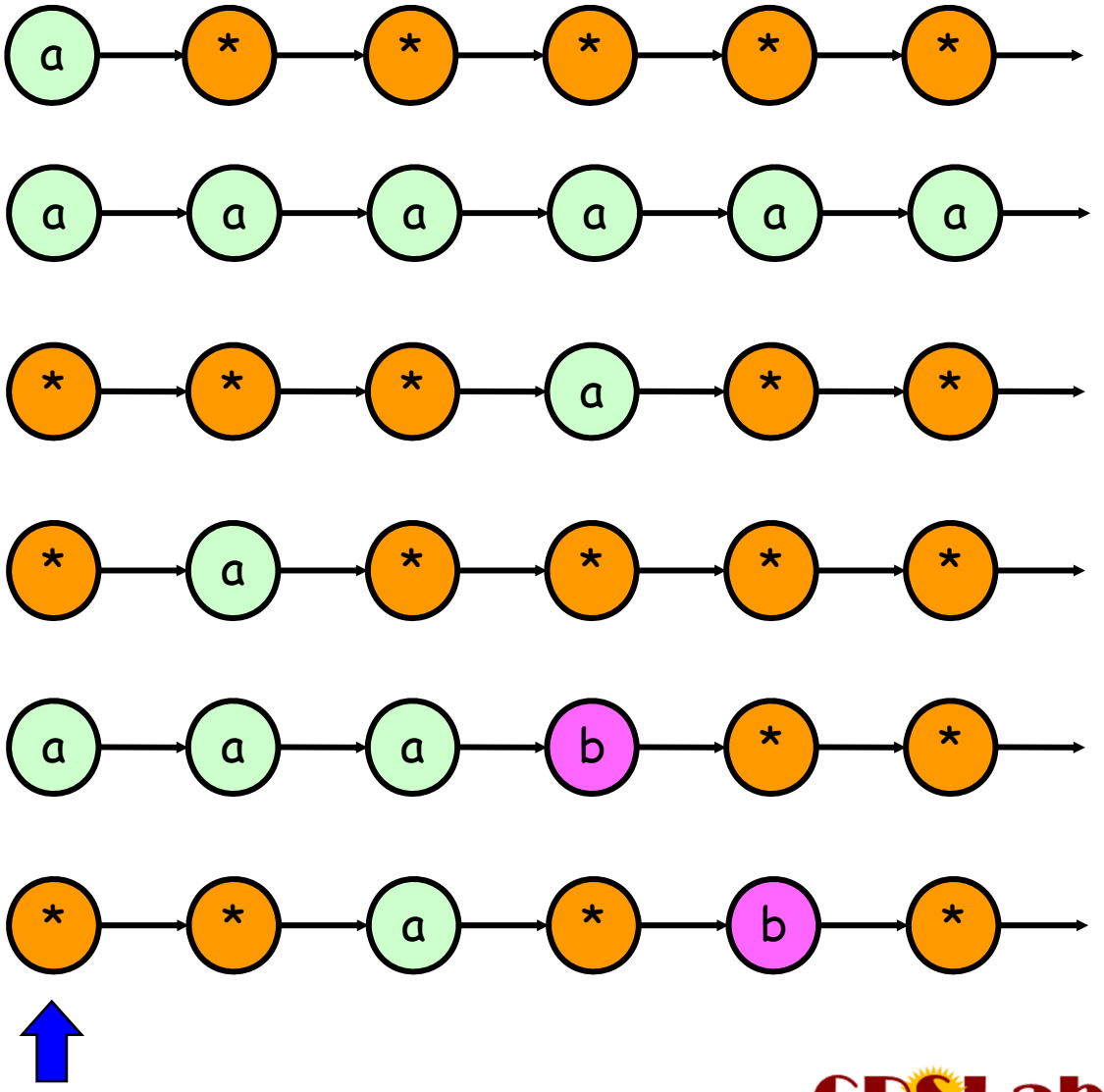
G a - always a

F a – eventually a

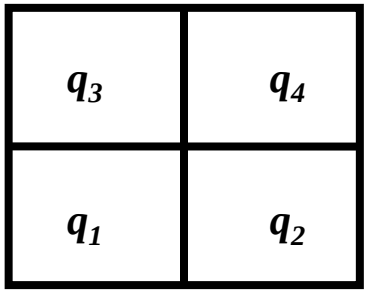
X a – next state a

a U b – a until b

a B b – a before b



Supervisory Controller



$\varphi = (\neg q_1) U (q_2 \wedge (\neg q_1) U q_3)$

Environment/ Robot Actions

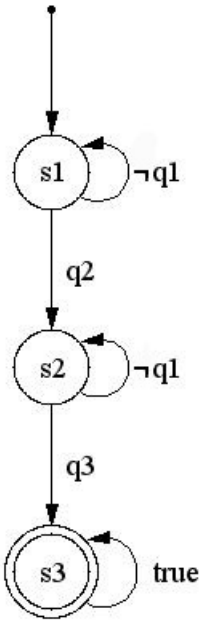
Temporal Logic Specification

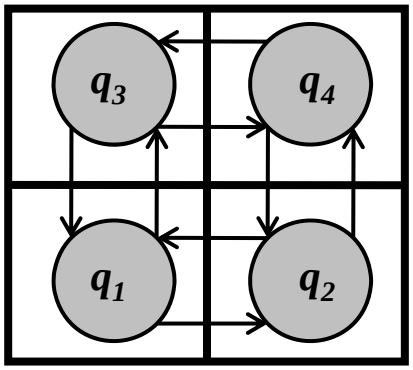
Discrete time Semantics

Discrete Abstraction

Product Automaton

Buechi Automaton



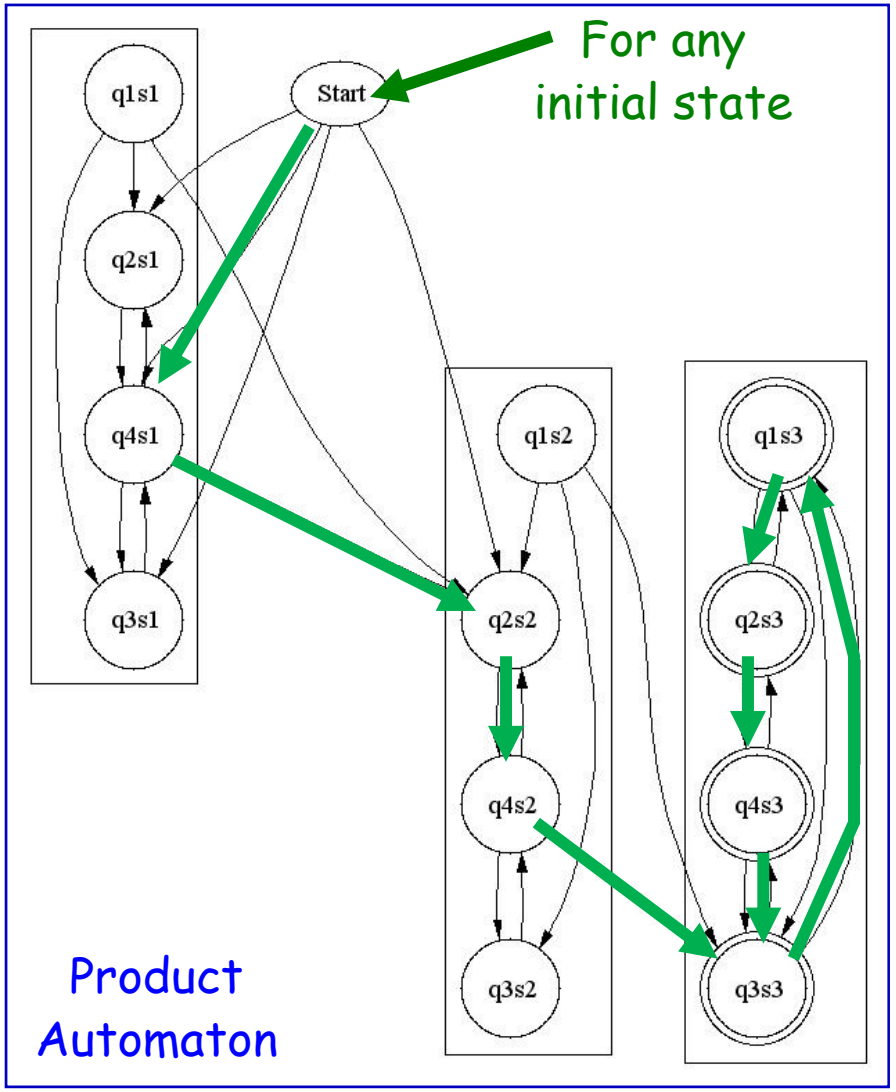


Nondeterministic Supervisor

*Simplified figure since, for instance, "q2 ∧ q3" cannot happen in this example

Supervisory Controller

$\varphi = (\neg q_1) U (q_2 \wedge (\neg q_1) U q_3)$

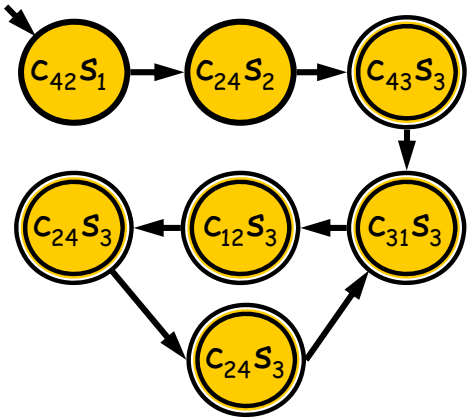


Find a path to an accepting state

and

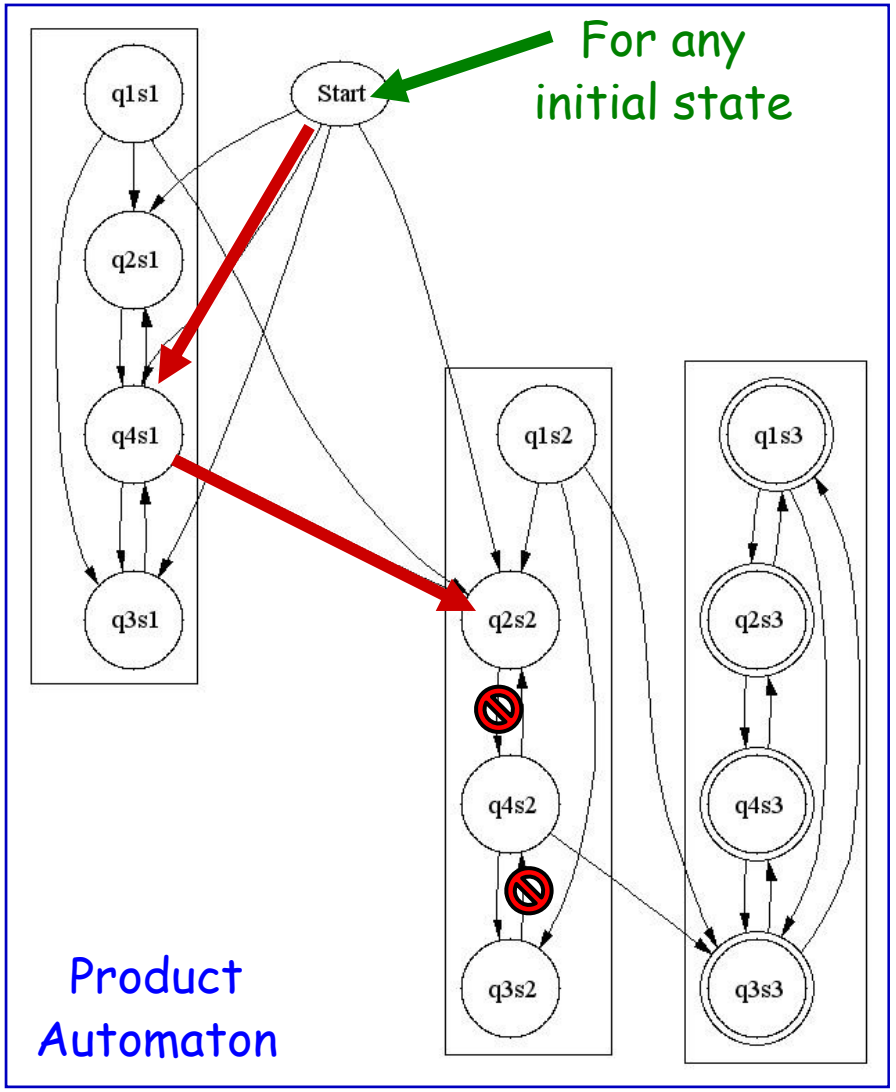
Find a path from an accepting state back to itself

Supervisor



Supervisory Controller

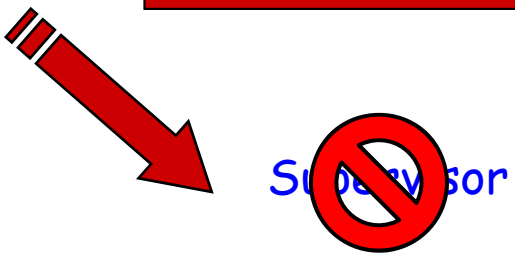
$\varphi = (\neg q_1) U (q_2 \wedge (\neg q_1 \wedge \neg q_4) U q_3)$



Cannot find a path to an accepting state

or

Cannot find a path from an accepting state back to itself



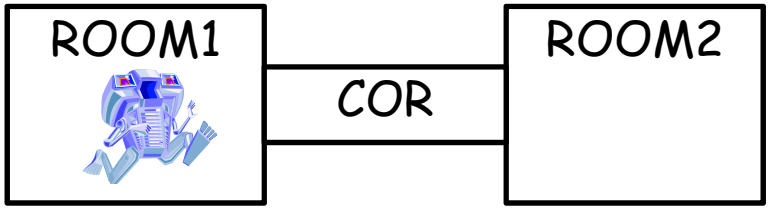
Problem 1: Why did the specification failed?

- Solution straightforward:
  - Find the set of reachable atomic propositions R_Π
  - If an atomic proposition in the specification φ is not in R_Π, then the planning failed due to an unreachable state in the system

φ = F ROOM2

  - If all the atomic propositions in the specification φ are in R_Π, then the planning failed due to "logical inconsistencies"

φ = (¬COR) U ROOM1



Revising the specification

- Ok, $\varphi = (\neg q_1)U(q_2 \wedge (\neg q_1 \wedge \neg q_4)Uq_3)$ cannot be satisfied.

What specification can be satisfied?

- We must restrict our search space
  - "Irrelevant" solutions are not an option, e.g., $\varphi = G\ q_5$
    - E.g., you ask the robot to bring oranges and the robot responds: "*Actually, I can only bring apples*"
  - Non-minimal solutions are not desirable, e.g., $\varphi =$ **true**
    - E.g., you ask the robot to bring oranges and the robot responds: *"Actually, I can only stay here and do nothing or I can visit all the rooms in the house. Please choose."*

Partial Order on the LTL formulas

- Let φ_1 and φ_2 be 2 LTL formulas, then

$$\varphi_1 \prec \varphi_2 \text{ if } \varphi_1 \Rightarrow \varphi_2$$

- Remark: In order to have a lattice, we need to consider the congruence under the equivalence relation of LTL formulas
  - e.g. $F(p_1 \vee p_2) \equiv Fp_1 \vee Fp_2$

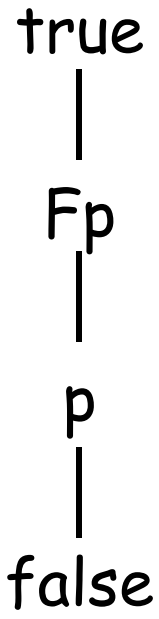
- Hence, the conjunction and disjunction become the meet and join operations over the lattice

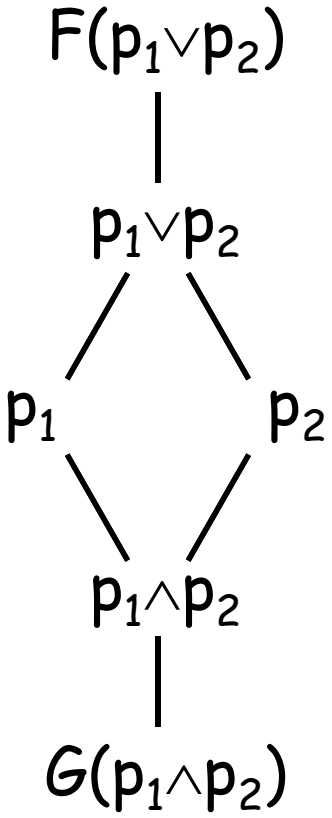
Examples

Notation

$\varphi_1 \prec \varphi_2$

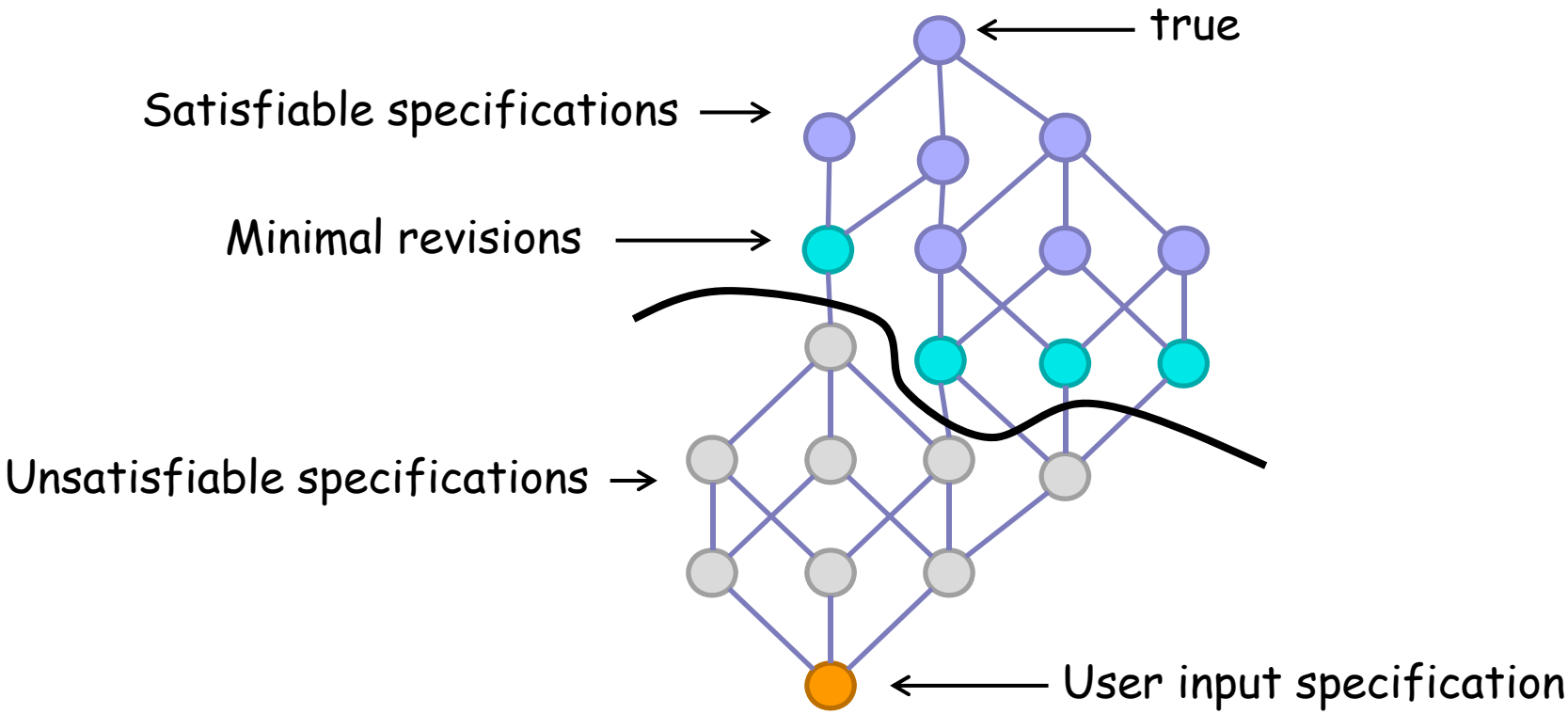
$false \prec p \prec Fp \prec true$





Possible space of solutions …

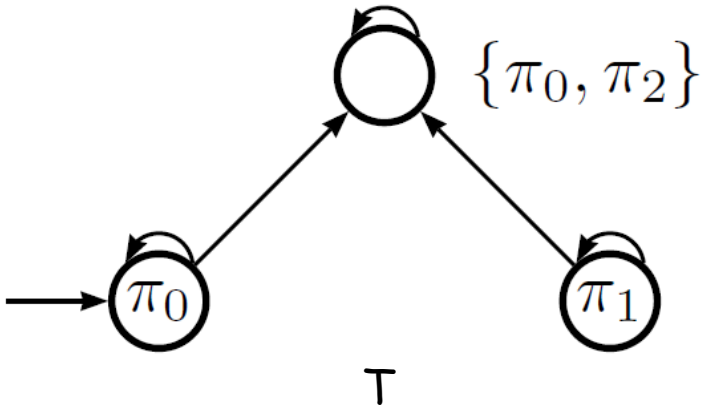
$$F(\varphi, T) = \{\varphi' \in LTL \mid T \models \varphi' \text{ and } \varphi \prec \varphi'\}$$



Possible space of solutions …

$$F(\varphi,T) = \{\varphi' \in LTL \mid T \models \varphi' \text{ and } \varphi \prec \varphi'\}$$

- However, searching for a minimal solution in F(φ,T) is not enough …



$\varphi = \pi_0 \wedge F(\pi_2 \wedge F\pi_1)$

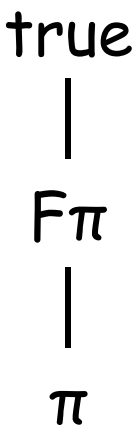
$\varphi' = (\pi_0 \wedge F(\pi_2 \wedge F(\pi_1))) \vee G(\pi_0 \wedge \neg\pi_1 \wedge \neg\pi_2)$

Modified possible space of solutions …

$$RF(\varphi,T) = \{\varphi' \in LTL \mid T \vDash \varphi' \text{ and } \varphi \prec \varphi'\} \cap rel(\varphi)$$

where rel(φ) removes some "irrelevant" solutions

- Example: rel(φ) is the set of all formulas where each atomic proposition π in φ is replaced by a formula in its upper bound
  - e.g.



Automatic Revision and Minimality due to unreachable atomic propositions

Theorem: Let LTL formula φ be unsatisfiable on system T and U be the set of unreachable atomic propositions in T. Set $\varphi' = rem_U(\varphi)$. Then,

1. we have $\varphi \prec \varphi'$.
2. φ' is minimal in $RF(\varphi, T)$

where:

1. $rem_U(\varphi)$ replaces each atomic proposition in φ that is also in U with *true*

2. φ' is minimal in $RF(\varphi, T)$ if for any other ψ in $RF(\varphi, T)$ we have $\psi \prec \varphi'$, then $\psi \equiv \varphi'$.

$$\varphi = \pi_0 \wedge F(\pi_2 \wedge F\pi_1)$$

$$\Downarrow$$

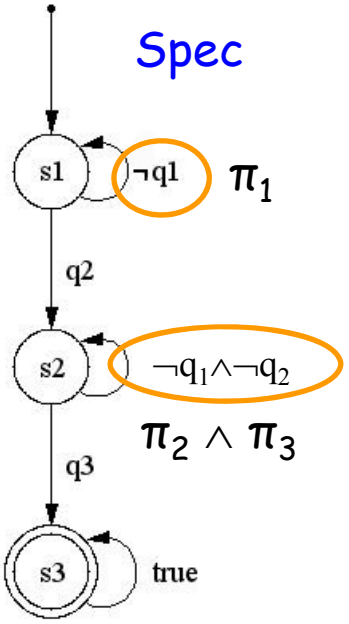
$$\varphi' = \pi_0 \wedge F\pi_2$$

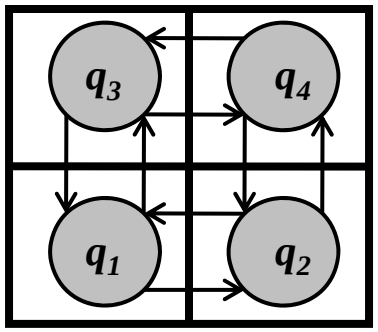
Automatic Revision and Minimality due to "logical inconsistencies"

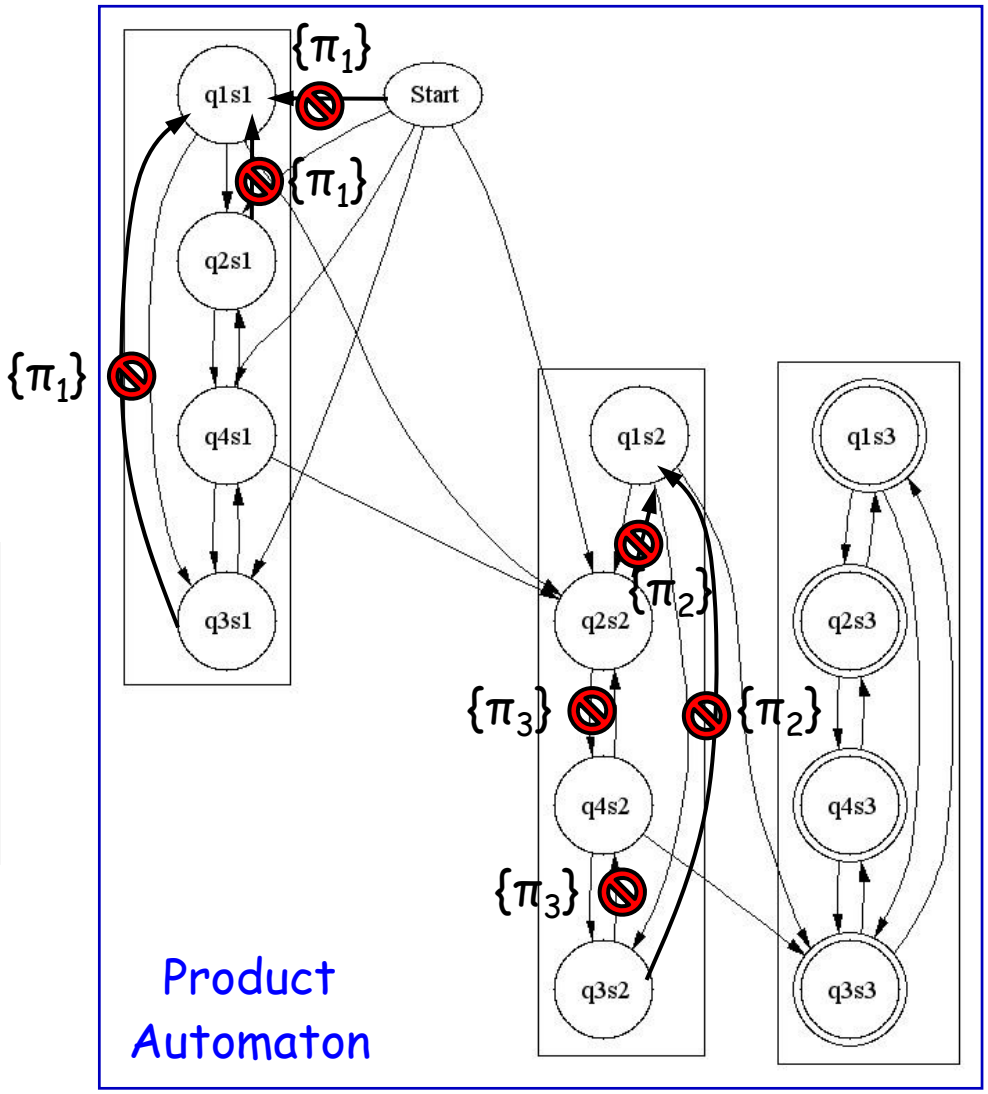
- Can we find the minimal revision in RF(φ,T)?
  - Obviously, the problem is decidable
  - Assume that we are going to maximally relax (i.e., set to true) each atomic proposition in φ
  - We need to consider $2^{|AP(\varphi)|}$ combinations of maximal relaxations
  - For each combination, we need to run the LTL planning algorithm which is of complexity $|T|2^{O(|\varphi|)}$
  - Worst case complexity $|T|2^{O(|\varphi|)}2^{|AP(\varphi)|} = |T|2^{O(|\varphi|)+|AP(\varphi)|}$

Translating the minimal specification revision problem into a graph problem …

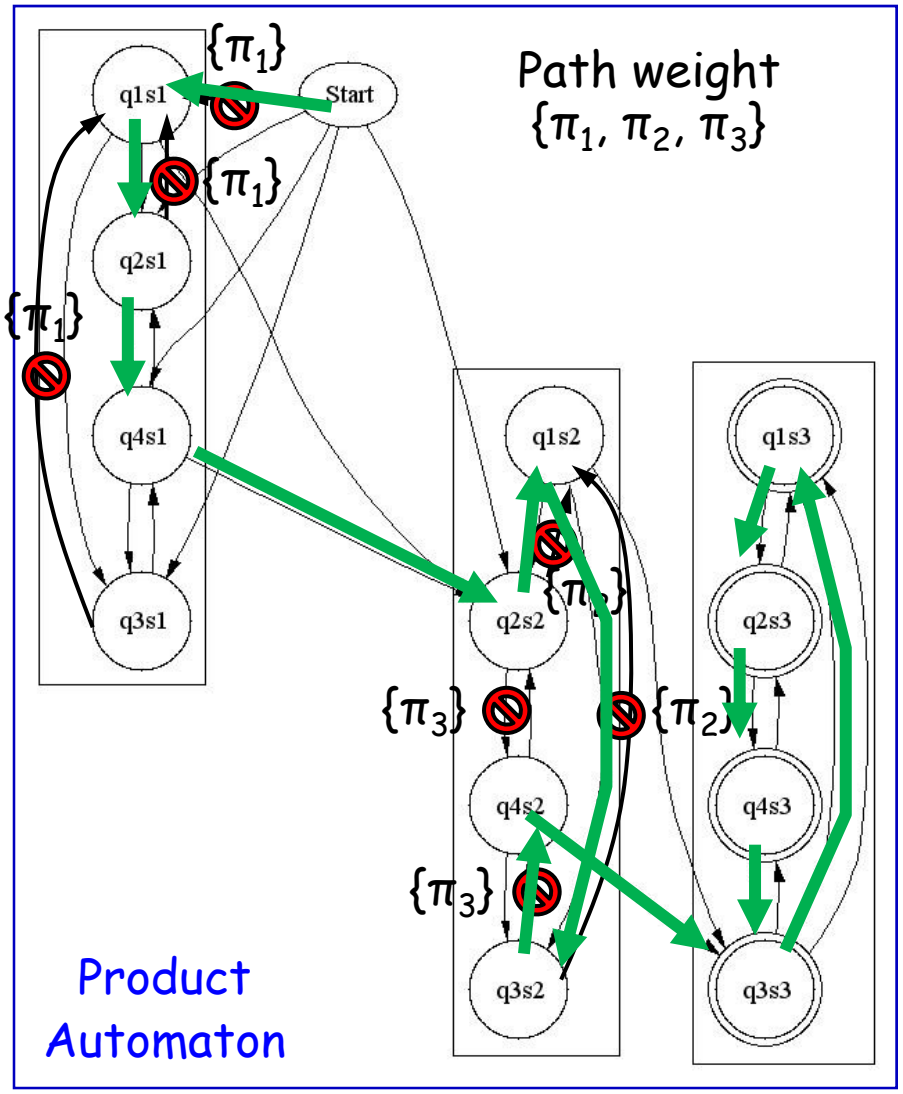
Label each edge in the product automaton with the set of atomic propositions from the spec that must be removed for edge to become enabled.

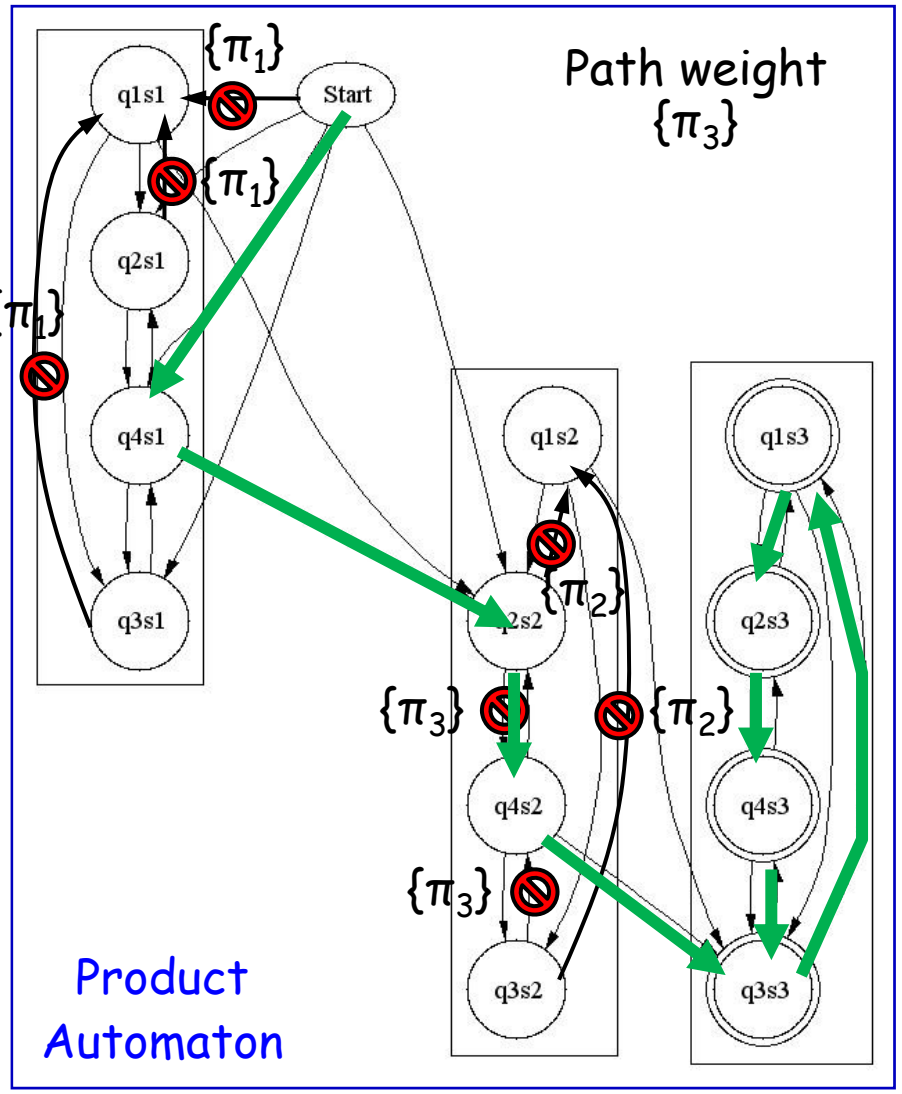






Problem: Find the path on the graph with the smallest number of AP to be removed





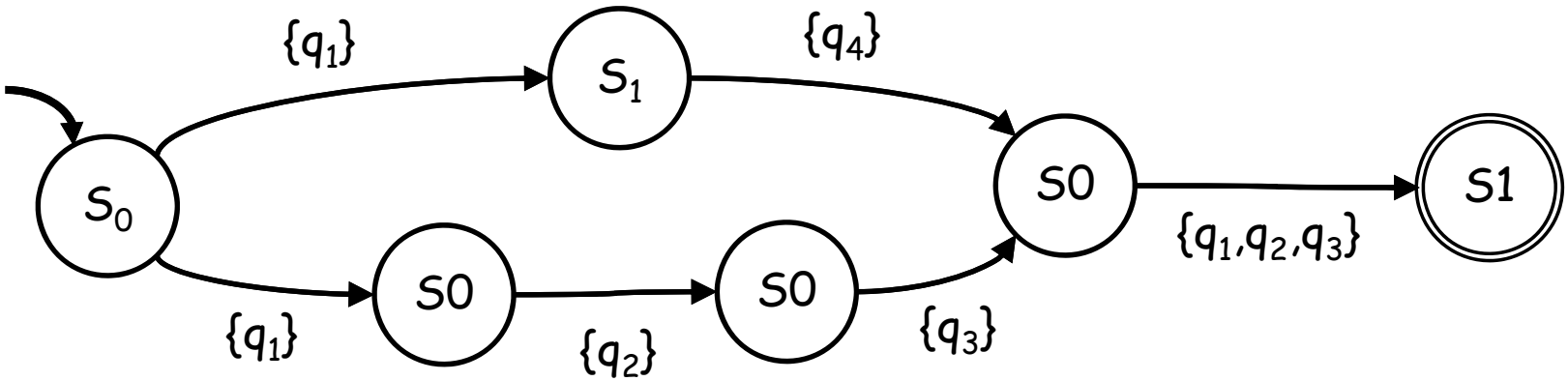
Main result: NP-Completeness

Minimal Accepting Path (MAP) Problem:
Given an instance of the minimal accepting path problem (G, Y, L, v_0, F) and a bound W, the decision of problem of whether there exists a truth assignment $Z \subseteq Y$ such that $|Z| \leq W$ is NP-Complete.

Proof* by reduction from 3-CNF-SAT.

* The proof does not appear in the ICRA 2012 paper, so please ask me for the technical report or wait for the journal version.

Intuition on why the problem is hard



Using SAT solvers* and ASP to get a solution

- We use variables of the form REACH(v_0, v) and REACH(v_f, v) for every vertex v ∈ V and v_f ∈ F.

- Reachability from v_0 and from v_f ∈ F:
  - If (v_0, v) ∈ E, then REACH(v_0, v) ⇔ $\wedge_{y \in L(v_0,v)}$ y
  - If (v_0, v) ∉ E, then REACH(v_0, v) ⇔
    $\vee_{(u,v) \in E}$ (REACH(v_0, u) ∧ $\wedge_{y \in L(u,v)}$y)
  - If (v_f, v) ∈ E, then REACH(v_f, v) ⇔ $\wedge_{y \in L(v_f,v)}$ y
  - If (v_f, v) ∉ E, then REACH(v_f, v) ⇔
    $\vee_{(u,v) \in E}$ (REACH(v_f, u) ∧ $\wedge_{y \in L(u,v)}$y)

- "Lasso" condition:
  - $\vee_{v_f \in F}$(REACH(v_0, v_f) ∧ REACH(v_f, v_f))

*Yices and/or Z3

Experimental Results: SAT problem

The experiments were run on the ASU supercomputing center* which consists of clusters of Dual 4-core processors, 16GB Intel(R) Xeon(R) CPU, X5355 @2.66 GHz.

Edges →	Sparse: $2n-2$				Medium: $3n$				Dense: n^2			
Nodes n ↓	min	avg	max	succ	min	avg	max	succ	min	avg	max	succ
10	0.0	0.1	0.2	100/100	0.0	0.0	0.1	100/100	0.0	0.1	0.9	100/100
100	0.3	0.6	1.5	100/100	0.9	41.5	1934.2	100/100	1425.1	2541.5	5970.4	67/100
200	1.8	4.7	24.1	100/100	9.5	273.4	6400.8	77/100				0/100
300	5.9	15.4	76.3	100/100	34.8	536.5	5624.3	71/100				0/100
400	14.7	58.2	244.9	100/100	87.1	1218.8	4175.3	50/100				0/100
500	33.2	125.7	473.0	100/100	176.8	1800.8	6939.2	48/100				0/100

*Our implementations do not utilize the parallel architecture.

Experimental Results:
2 Approximation Algorithm*

Nodes	ASP				AAMRP				RATIO		
	min	avg	max	succ	min	avg	max	succ	min	avg	max
9	0.003	0.0071	0.012	200/200	0.013	0.0157	0.025	200/200	1	1.00667	2
100	0.099	0.1954	1.405	200/200	0.027	0.06727	0.09	200/200	1	1.000625	1.125
196	0.335	1.25058	6.003	200/200	0.057	0.22372	0.289	200/200	1	1	1
324	0.869	5.3316	14.731	200/200	0.113	0.6601	0.912	200/200	1	1.001417	1.2
400	1.267	12.87	35.58	200/200	0.131	1.28913	1.351	200/200	1	1	1
529	3.086	34.1642	103.638	200/200	0.37	3.107	4.141	200/200	1	1	1

TABLE I

NUMERICAL EXPERIMENTS: NUMBER OF NODES VERSUS THE RESULTS OF ASP SOLVER AND AAMRP. UNDER THE ASP AND AAMRP COLUMNS THE NUMBERS INDICATE COMPUTATION TIMES IN sec. RATIO INDICATES THE EXPERIMENTALLY OBSERVED APPROXIMATION RATIO TO THE OPTIMAL SOLUTION.

Nodes	ASP				AAMRP				RATIO		
	min	avg	max	succ	min	avg	max	succ	min	avg	max
9	0.005	0.0097	0.039	200/200	0.012	0.0153	0.0449	200/200	1	1	1
100	0.378	18.4679	3502.343	200/200	0.028	0.063	0.09	200/200	1	1.001	1.2
196	3.336	31.995	685.819	167/200	0.0439	0.203	0.249	200/200	1	1	1
306	9.801	75.524	2795.337	149/200	0.101	0.5493	0.7	200/200	1	1.000839	1.125
400	21.744	124.7486	164.5459	148/200	0.134	1.124	1.2929	200/200	1	1	1
506	58.67	241.167	1054.98	152/200	0.2329	2.0795	1.821	200/200	1	1.002193	1.333333

TABLE II

NUMERICAL EXPERIMENTS: NUMBER OF NODES VERSUS THE RESULTS OF ASP SOLVER AND AAMRP. UNDER THE ASP AND AAMRP COLUMNS THE NUMBERS INDICATE COMPUTATION TIMES IN sec. RATIO INDICATES THE EXPERIMENTALLY OBSERVED APPROXIMATION RATIO TO THE OPTIMAL SOLUTION.

*Algorithm under review

Experimental Results:
2 Approximation Algorithm*

Preliminary results on scalability using a prototype Python implementation:

Nodes	ASP				AAMRP				RATIO
	min	avg	max	succ	min	avg	max	succ	
1024	24.438	168.2133	237.758	10/10	0.125	0.23	0.325	9/10	1
10000				0/10	15.723	76.164	128.471	9/10	
20164				0/10	50.325	570.737	1009.675	8/10	
50176				0/10	425.362	1993.449	4013.717	3/10	
60025				0/10	6734.133	6917.094	7100.055	2/10	

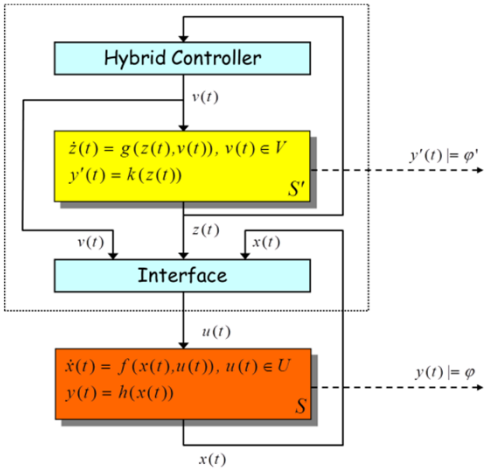
TABLE III

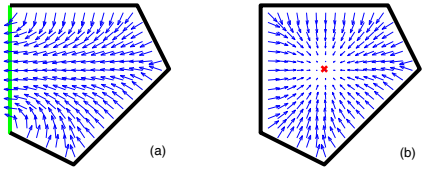
NUMERICAL EXPERIMENTS: NUMBER OF NODES VERSUS THE RESULTS OF ASP SOLVER AND AAMRP. UNDER THE ASP AND AAMRP COLUMNS THE NUMBERS INDICATE COMPUTATION TIMES IN sec. RATIO INDICATES THE EXPERIMENTALLY OBSERVED APPROXIMATION RATIO TO THE OPTIMAL SOLUTION.

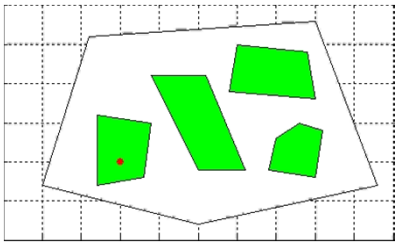
*Algorithm under review

Take home message

As **automatic synthesis methods** for software and, in particular, for **embedded control software** move from theory and prototypes into technology, we will need methods for **specification debugging** and **automatic revision**.

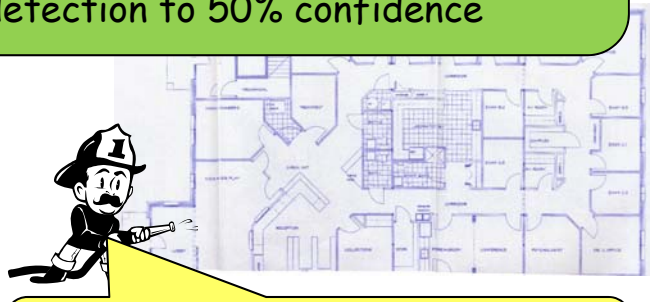






[ICRA 2012 example]

Battery power is not sufficient.
Please choose:
1. Visit only north side
2. Use some sensors and reduce detection to 50% confidence



Visit all rooms and check for people with 95% detection confidence

Conclusions

- Contributions
  1. We have defined the **Minimal Specification Revision Problem**
     - *Important for user-friendly temporal logic planning frameworks*
  2. We showed NP-completeness of MSRP even in its simplest version
  3. We have provided a SAT encoding of the MSRP
  4. We have developed an approximation algorithm

- Future work
  1. Define appropriate search space restrictions for different classes of planning problems
  2. Extend the theory to LTL games
  3. Provide direct feedback in natural language

Thank you! Questions?

- Acknowledgements:
  - Students: Kangjin Kim (PhD)
  - Collaborators: Sriram Sankaranarayanan (University of Colorado, Boulder)
  - Support: NSF CNS 1116136

- References:

1. G. Fainekos, Revising Temporal Logic Specifications for Motion Planning, IEEE International Conference on Robotics and Automation, Shanghai, May 2011
2. K. Kim, G. Fainekos and S. Sankaranarayanan, IEEE International Conference on Robotics and Automation, St. Paul, Minnesota, May 2012
3. K. Kim and G. Fainekos, Approximate Solutions for the Minimal Revision Problem of Specification Automata (Under review)