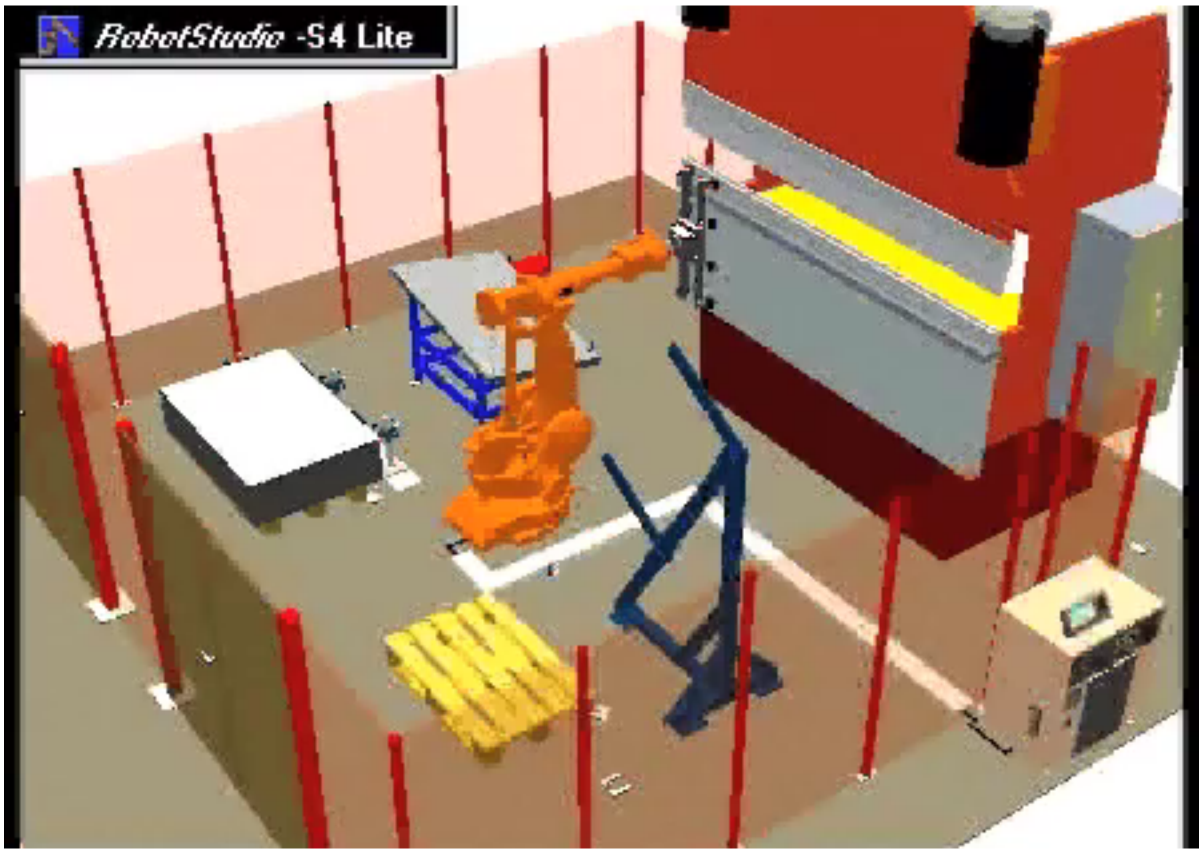
Motion Planning for Physical Systems

Lydia E. Kavraki
Department of Computer Science
Rice University

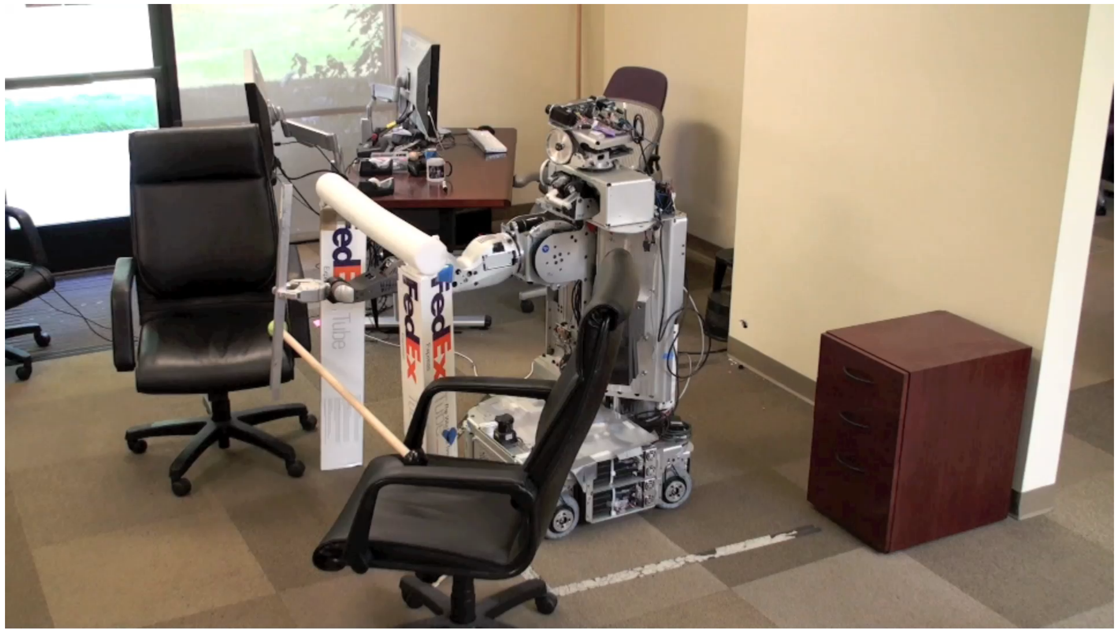
Informatics and Telematics Institute
June 6, 2011

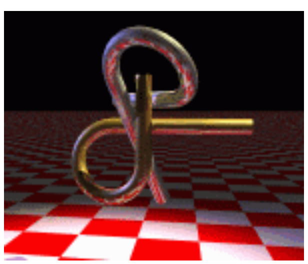
Motion Planning



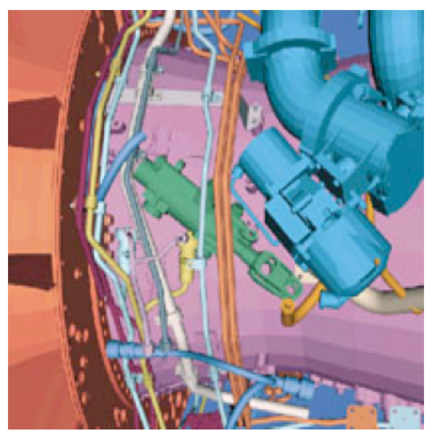
[Bohlin, Kavraki]

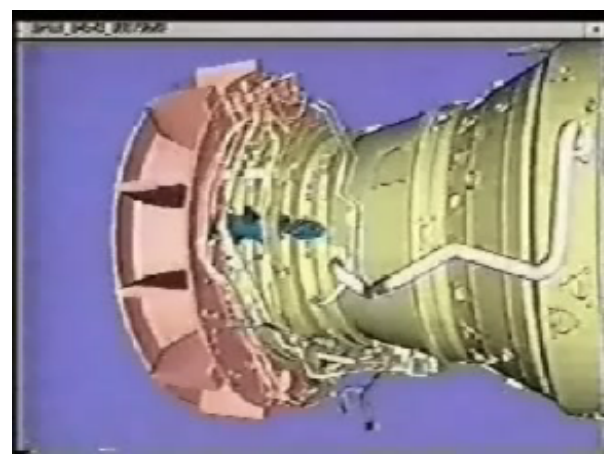
Motion Planning





Geometric Puzzles [from Kuffner]





[from Latombe]

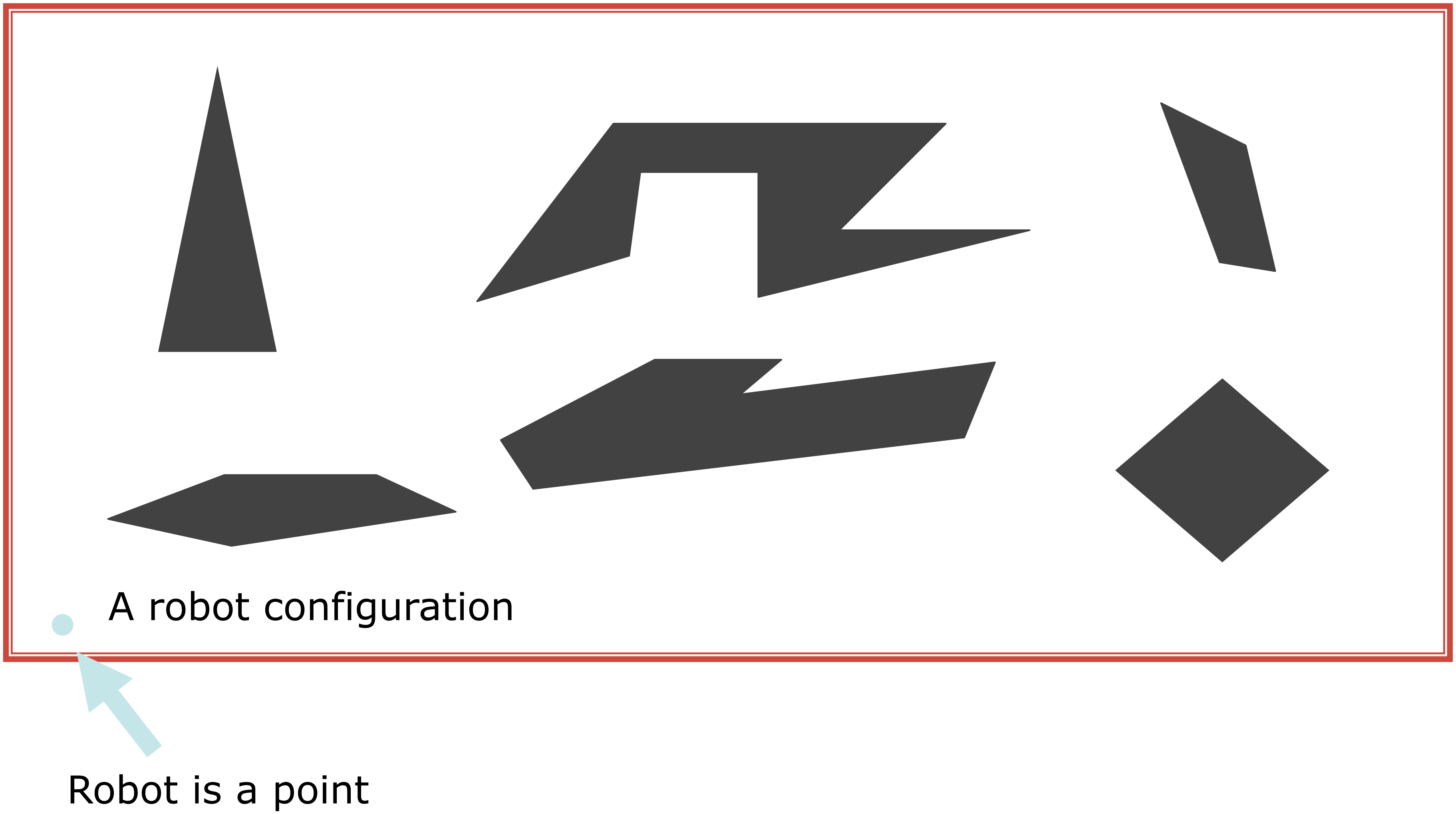
Motion-planning Problems are Hard

PROBLEM	COMPLEXITY
Geometric Constraints	
Sofa Mover (3DOF)	$O(n^{2+\varepsilon})$ - not implemented [HS96]
Piano Mover (6DOF)	Polynomial – no practical algorithm [SS83]
n Disks in the Plane	NP-Hard [SS83]
n Link Chain in 3D	PSPACE-Complete [HSS87]
Generalized Mover	PSPACE-Complete [Canny88]
Dynamics Constraints	
Point with Newtonian Dynamics	NP-Hard [DXCR93]
Polygon Dubin's Car (Linear)	Decidable [CPK08]
Nonlinear	Unknown, probably undecidable
Discrete Transitions and Dynamics Constraints	
Hybrid Systems	Undecidable [Alur et. al 95]

Exact, Approximate and Heuristic

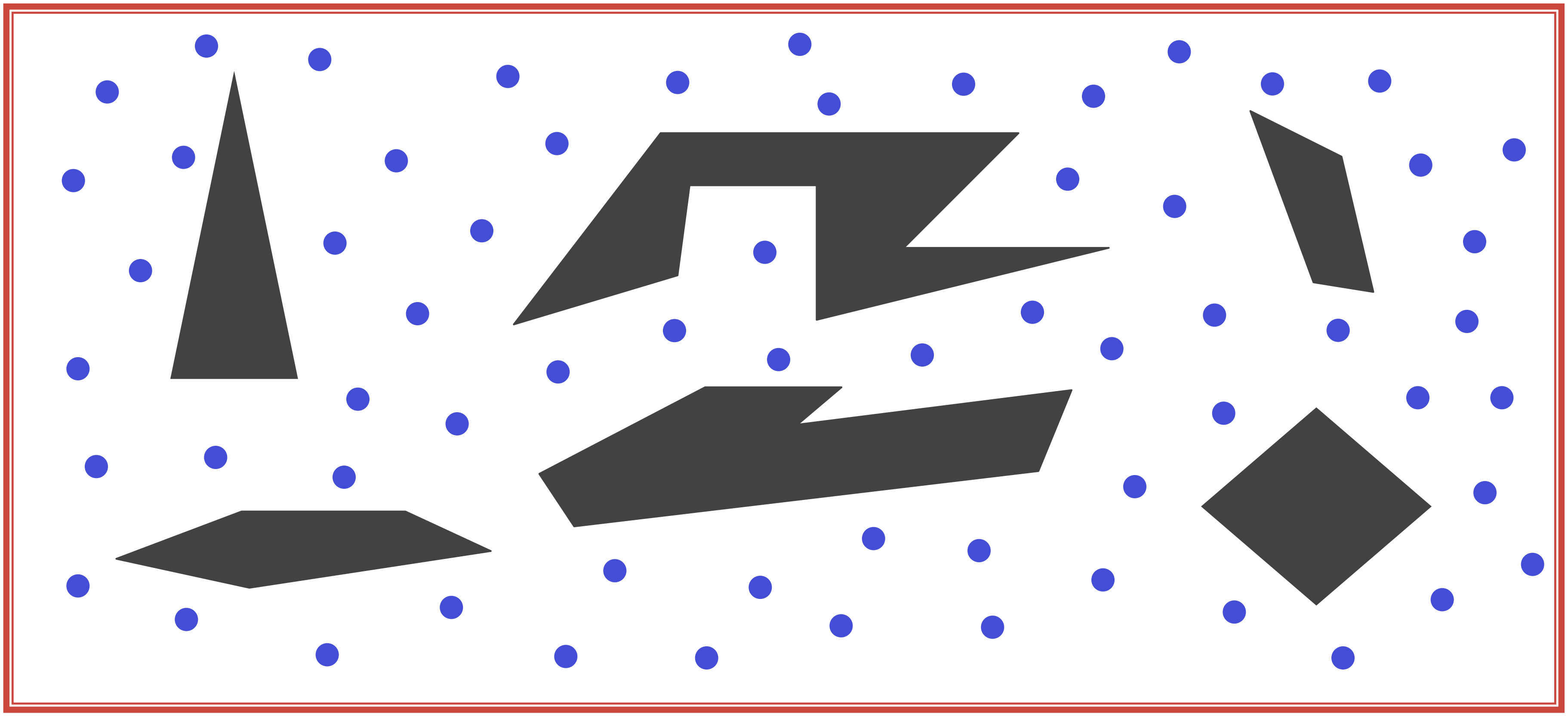
Method	Advantage	Disadvantage
Exact	theoretically insightful	impractical
Cell Decomposition	easy	does not scale
Control-Based	online, very robust	requires good trajectory
Potential Fields	online, easy	slow or fail
$\boxed{\text{Sampling-based}}$	fast and effective	cannot recognize impossible query

Probabilistic RoadMap (PRM)



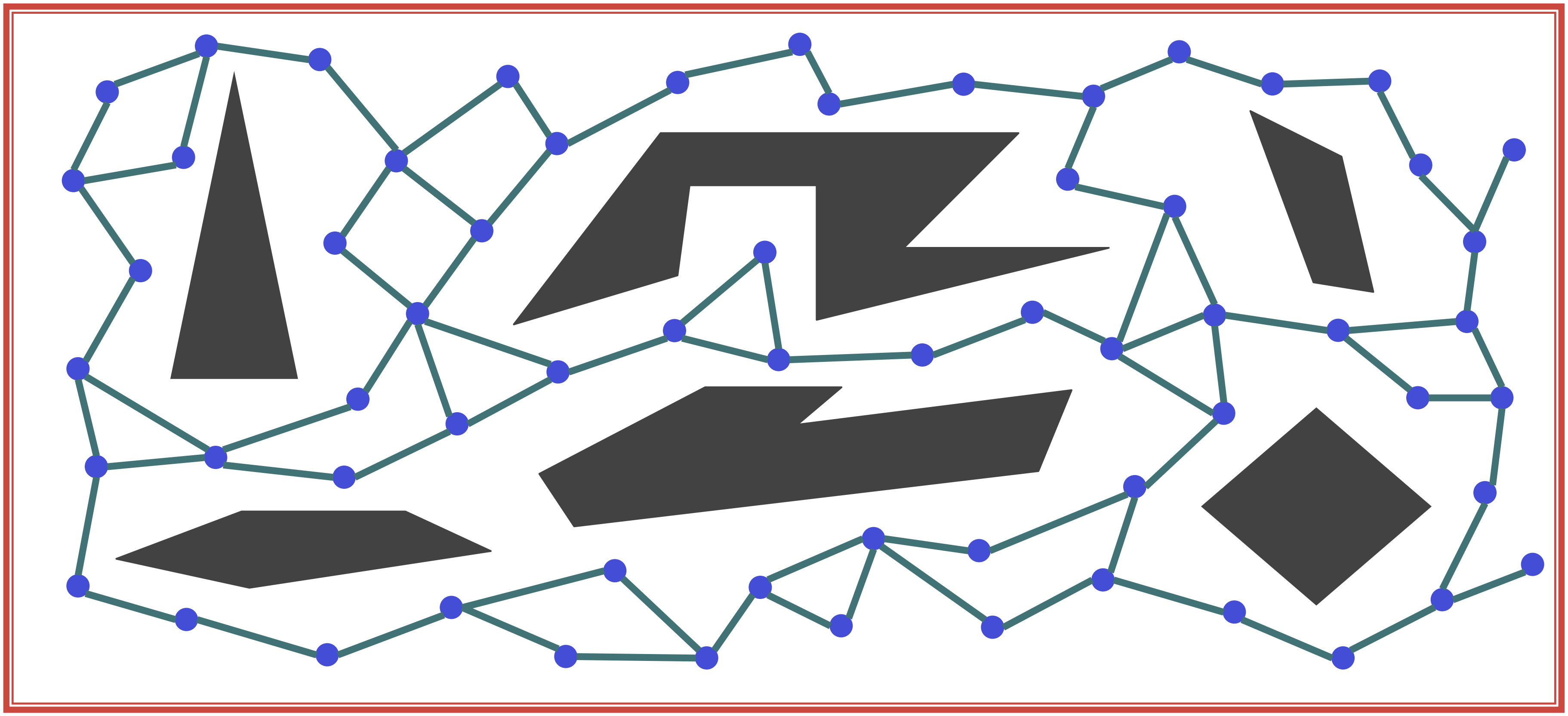
Probabilistic RoadMap (PRM)

Probabilistic RoadMap (PRM)



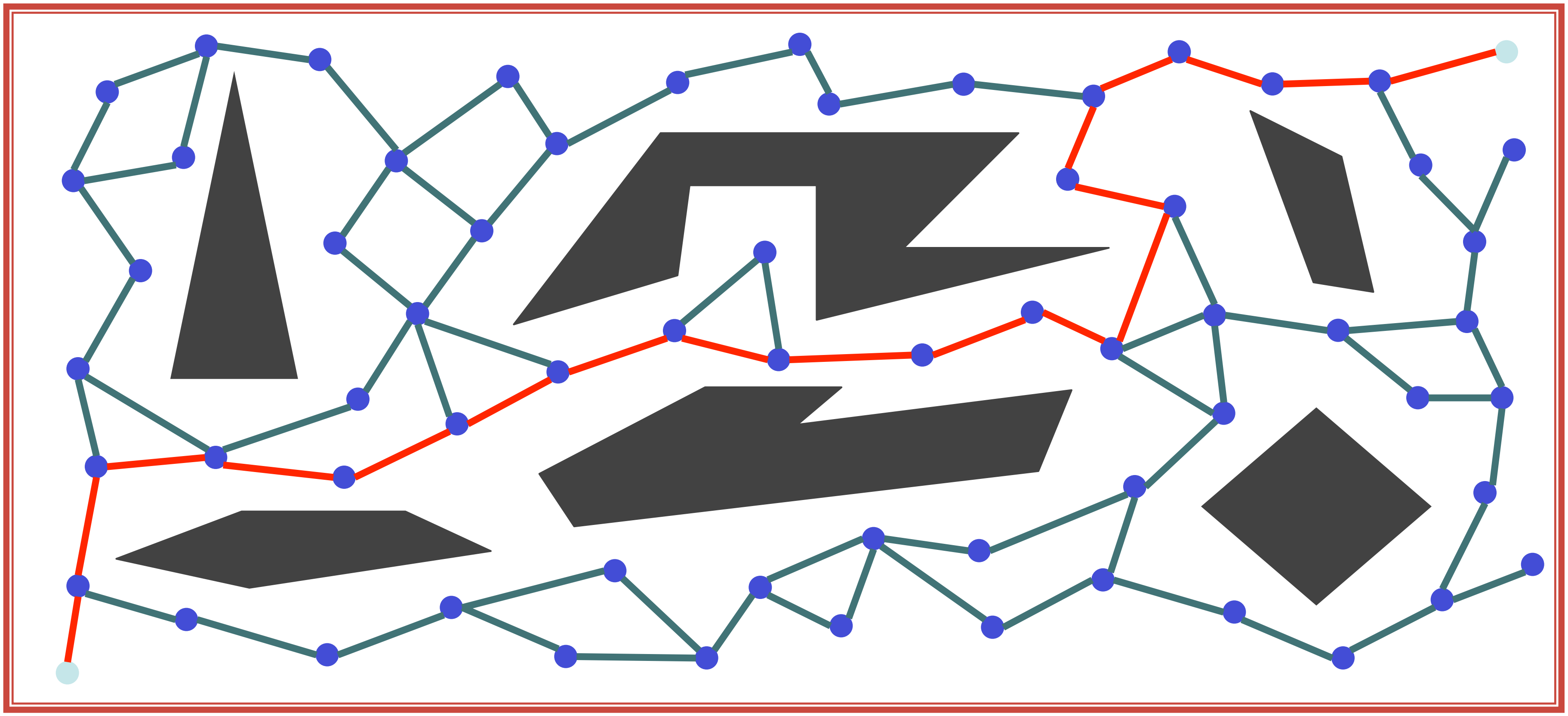
● Nodes: random configurations

Probabilistic RoadMap (PRM)



— Edges: computed by some local planner

Probabilistic RoadMap (PRM)

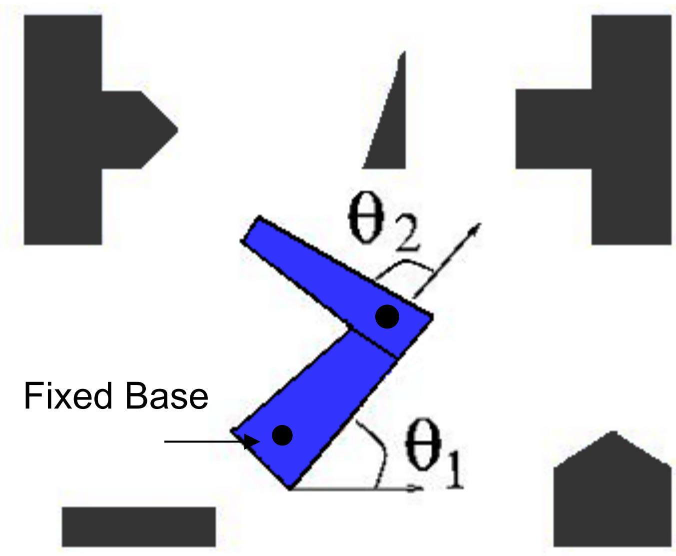


Plan a path

Connect start & goal to roadmap

Perform graph search

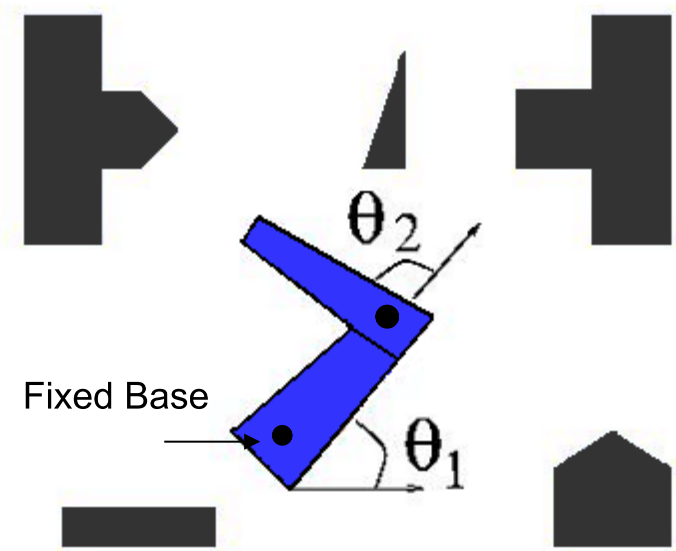
Degrees of Freedom (DOF)



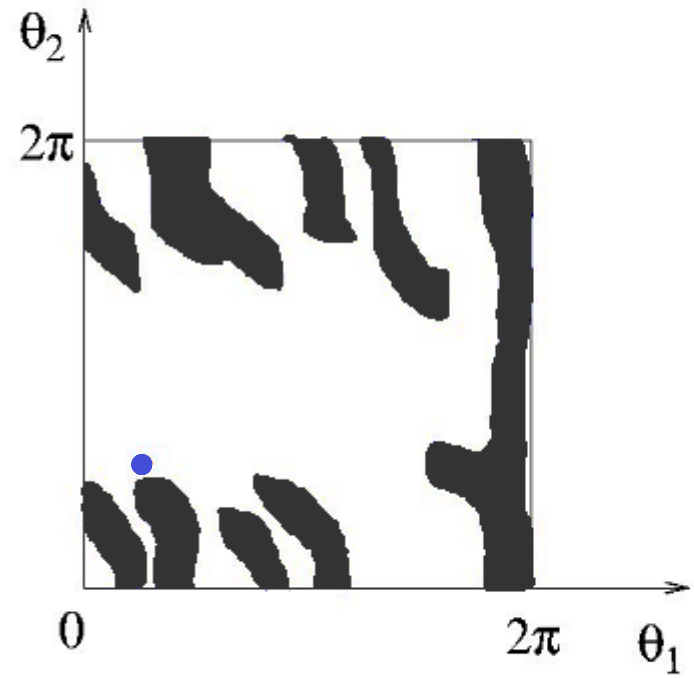
Workspace

Planar Arm: 2 Degrees of Freedom (DOF)

Configuration Space



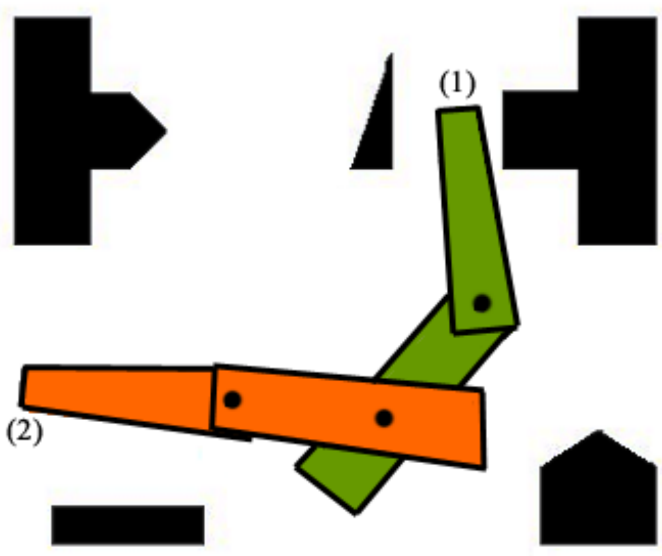
Workspace



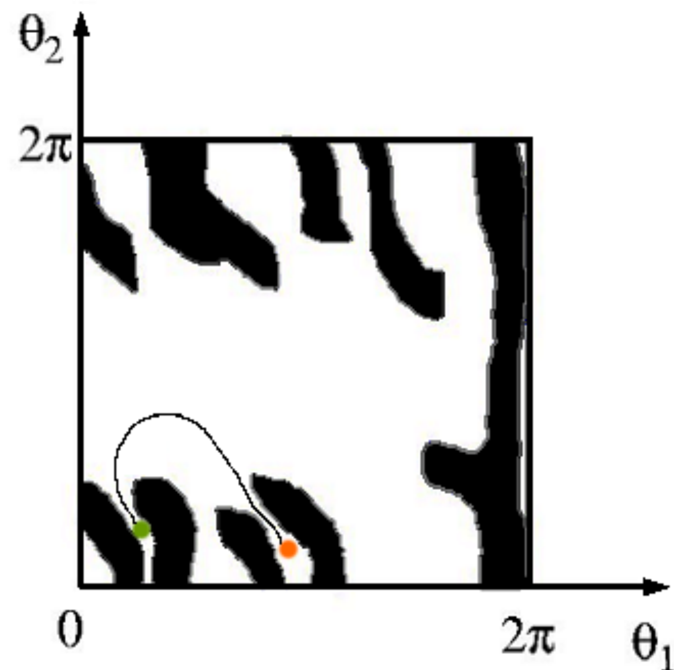
Configuration Space

Most Interesting Problems are High Dimensional

Configuration Space



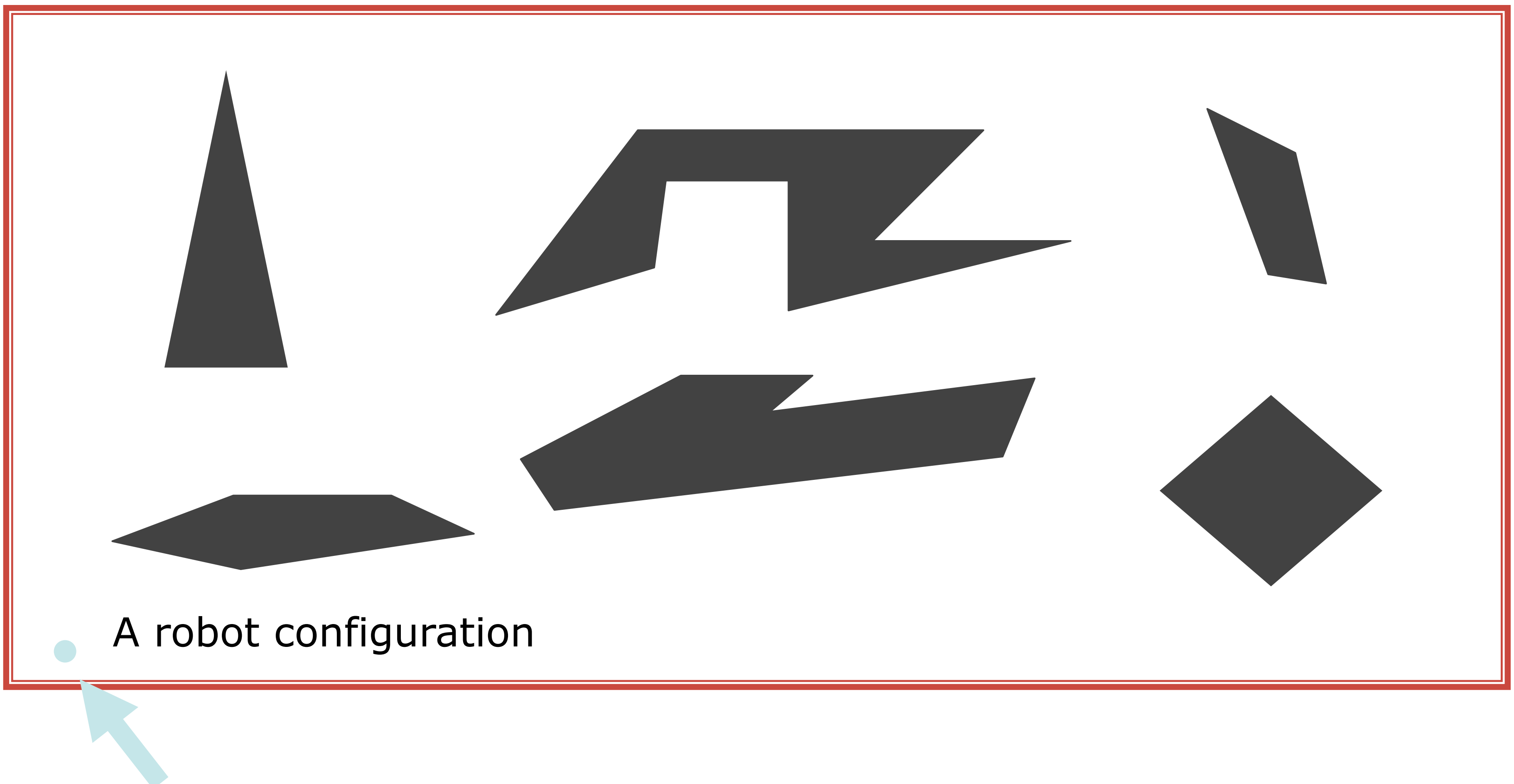
Workspace



Configuration Space

Most Interesting Problems are High Dimensional

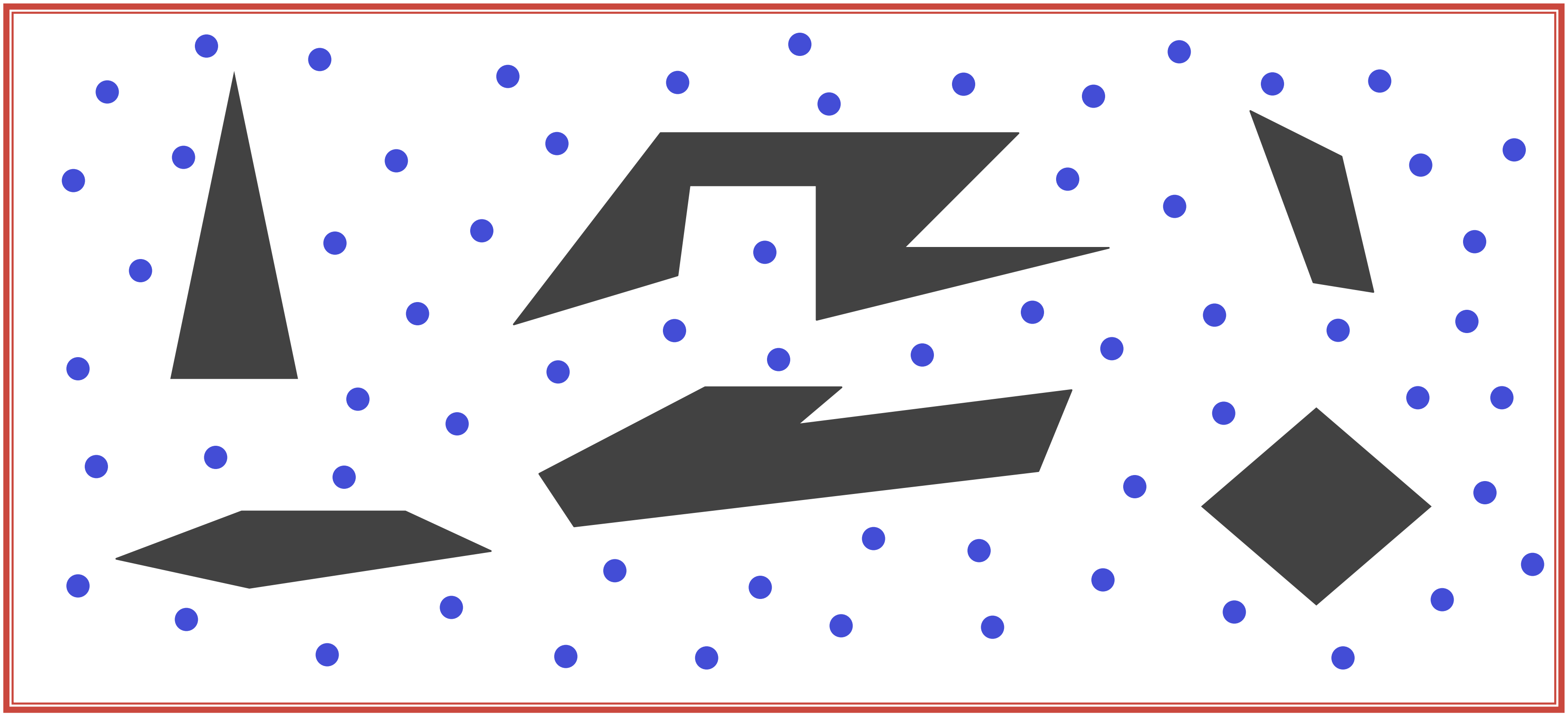
Probabilistic RoadMap (PRM)



Robot is a point

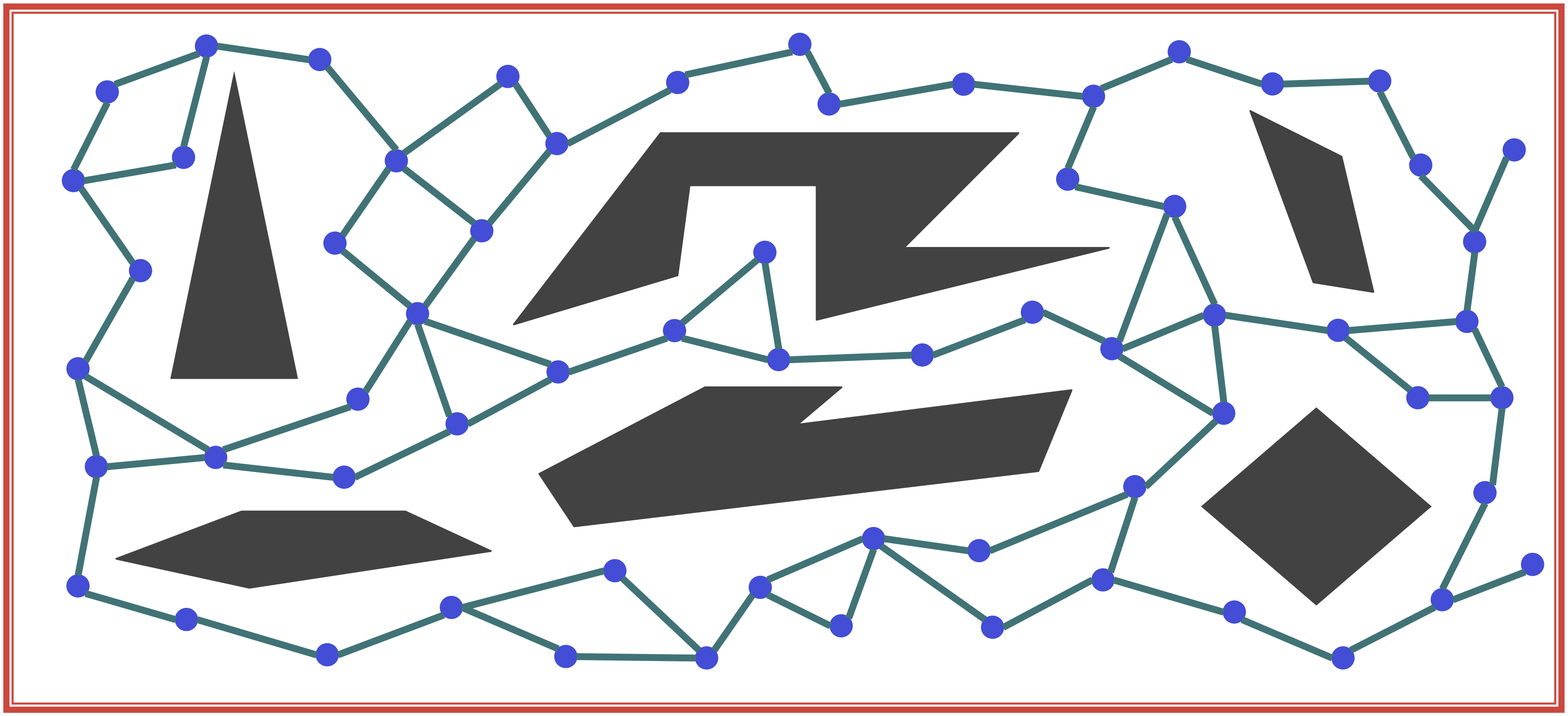
Probabilistic RoadMap (PRM)

Probabilistic RoadMap (PRM)



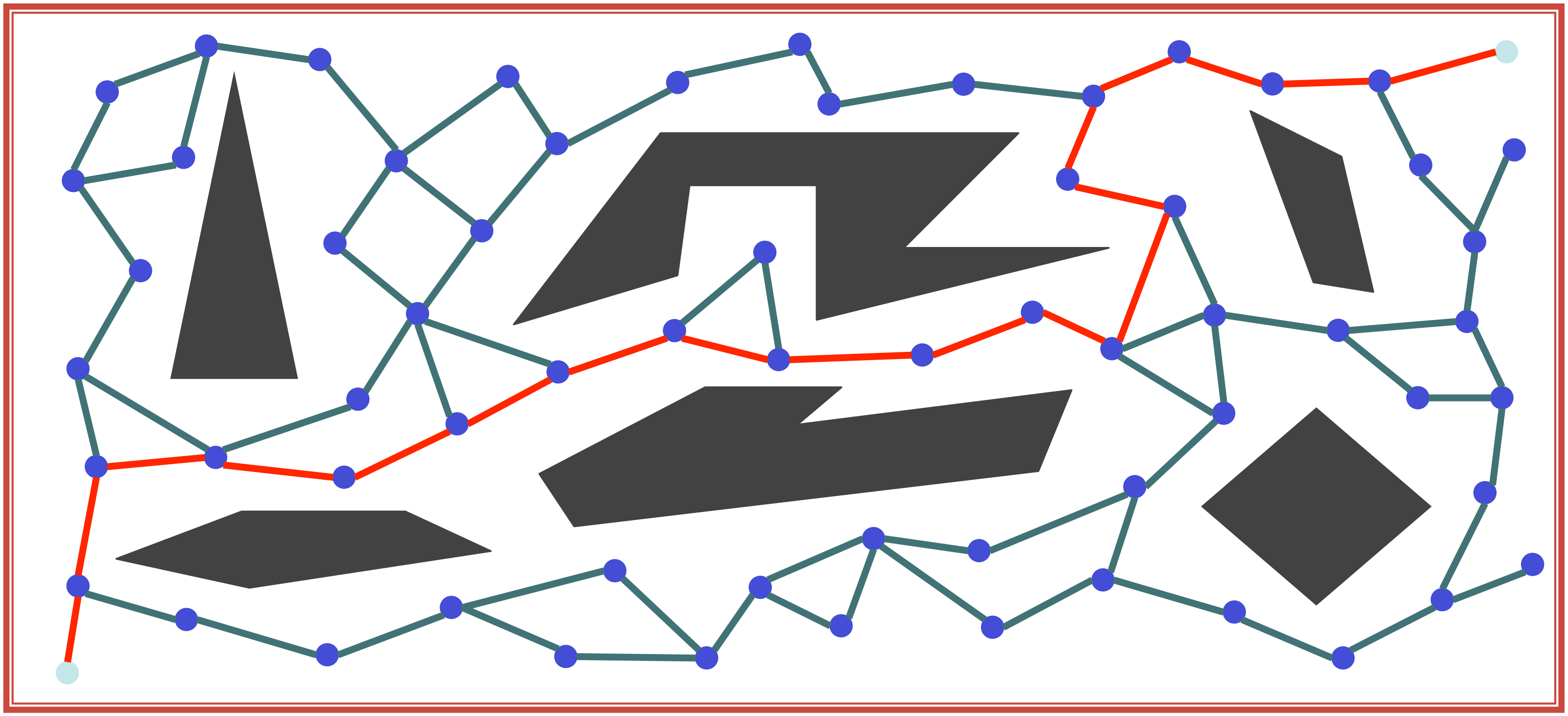
Nodes: random configurations

Probabilistic RoadMap (PRM)



— Edges: computed by some local planner

Probabilistic RoadMap (PRM)

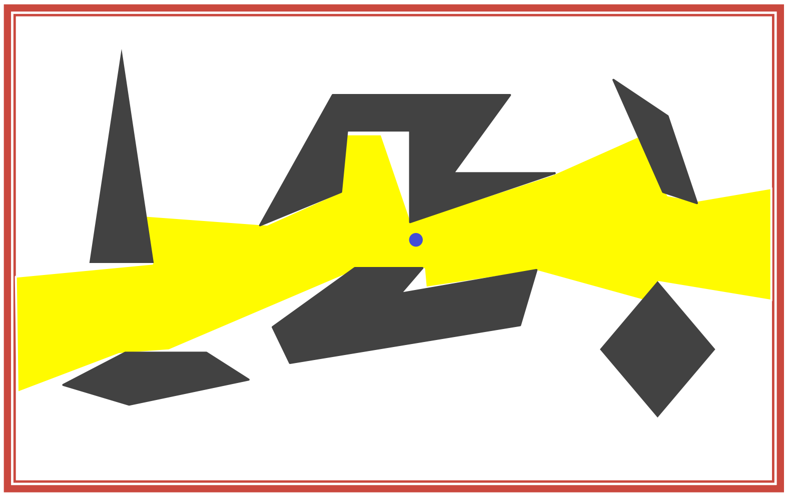


Plan a path

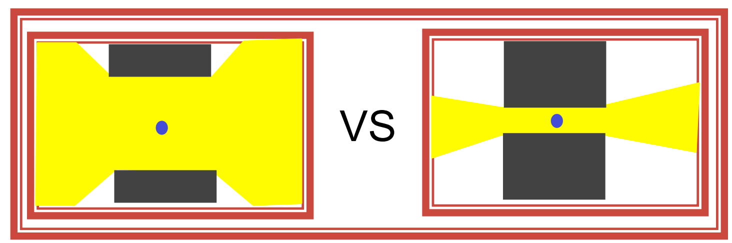
Connect start & goal to roadmap

Perform graph search

Theoretical Analysis of PRM



ε-goodness property



- Tradeoff: planner may fail with probability a
- Number of nodes

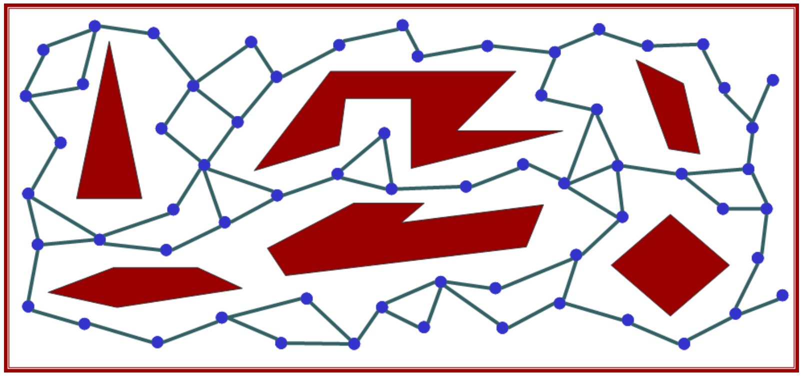
$$N \approx \frac{1}{\varepsilon}\left[\log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{4}{\alpha}\right)\right]$$

- Important: Performance related to properties of the space

Sampling-based Methods

- Roadmap-based
  - Sample valid states
  - Connect neighboring samples

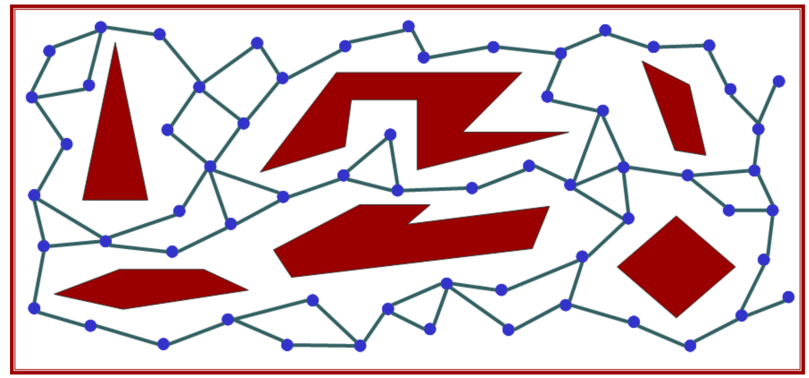
[Kavraki, Svetska, Latombe, Overmars and many others]



Sampling-based Methods

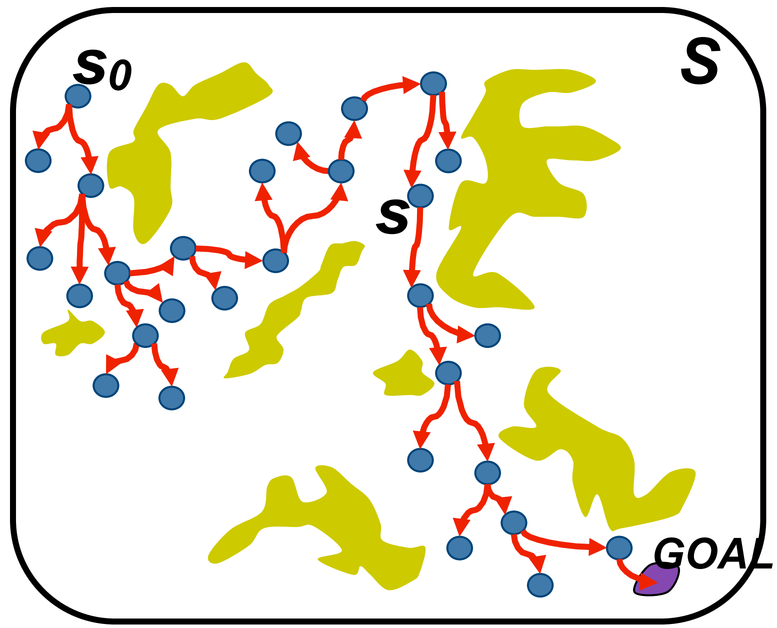
- Roadmap-based
  - Sample valid states
  - Connect neighboring samples

[Kavraki, Svetska, Latombe, Overmars and many others]



- Tree-based
  - Grow tree ***T*** rooted at initial state

[LaValle, Kuffner, Hsu, Ladd, Plaku, Bekris and many others]



A Few Sampling-Based Planners

EST [Hsu et al.'97, '00]
RRT [Kuffner, LaValle '99]
RRT-Connect [Kuffner, LaValle '00]
SBL [Sanchez, Latombe '01]
Guided EST [Phillips et al. '03]
PDRRT [Ranganathan, Koenig '04]
SRT [Plaku et al. '05]
DDRRT [Yershova et al. '05]
ADDRRT [Jaillet et al. '05]
RRT-Blossom [Kalisiak, van Panne '06]
PDST [Ladd, Kavraki '06]
Utility RRT [Burns, Brock '07]
GRIP [Bekris, Kavraki '07]
Multiparticle RRT [Zucker et al. '07]
TC-RRT [Stillman et al. '07]
RRT-JT [Vande Wege et al '07]
DSLX [Plaku, Kavraki, Vardi '08]
KPIECE [Șucan, Kavraki '09]

RPDST [Tsianos, Kavraki '08]
BiSpace [Diankov et al. '08]
GRRT [Chakravorty, Kumar '09]
IKBiRRT [Berenson et al.'09]
CBiRRT [Berenson et al.'09]
J+RRT [Vahrenkamp '09]
RRT*[Kamaran et al.11]
and many others
....

Sampling-Based Planners Today

Universities:

- Rice University
- Texas A & M University
- Stanford University
- University of Illinois, Urbana Champaign
- University of Washington
- Rensselaer Polytechnic Institute
- Simon Fraser, Canada
- Oxford, UK
- Göteborg University, Sweden
- Tel-Aviv University, Israel
- Carnegie Mellon
- University of Utrecht, The Netherlands
- National University of Singapore
- Institut Polytechnique de Toulouse, France

and others

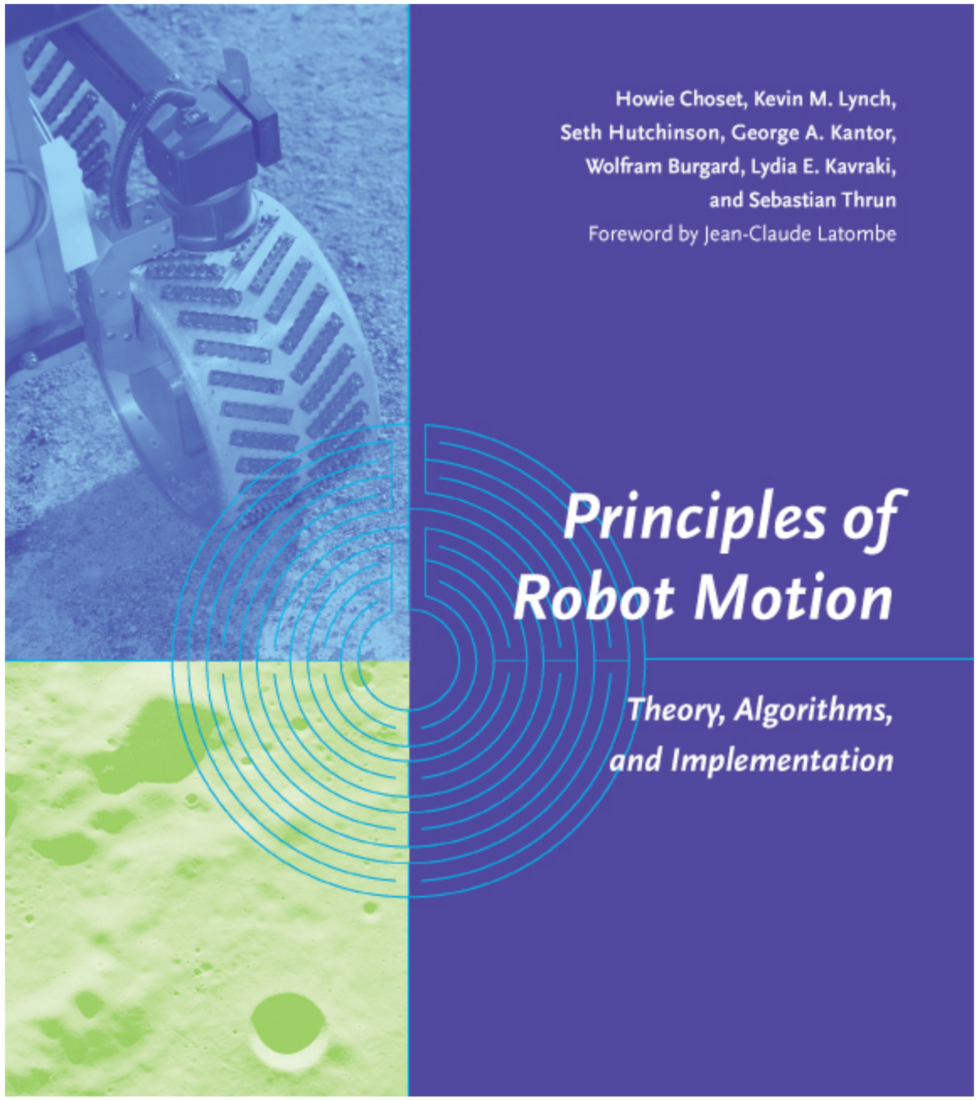
Companies:

- General Electric
- General Motors
- ABB Robotics
- Prosolvia
- Amrose Automation
- Electricité de France
- Honda
- Volvo
- Draper Laboratories

Research Laboratories

- LAAS CNRS, France
- INRIA, France
- NASA

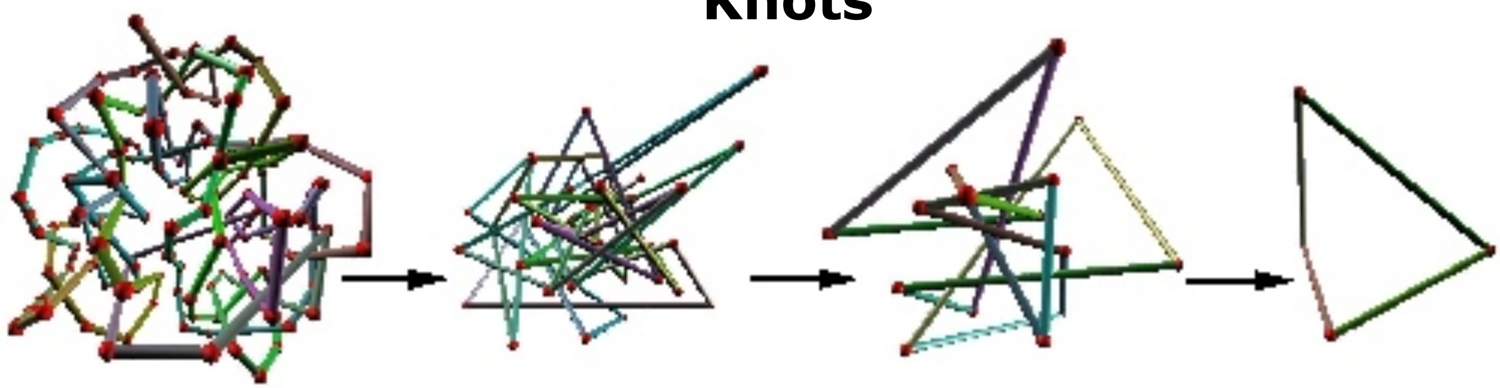
A New Textbook

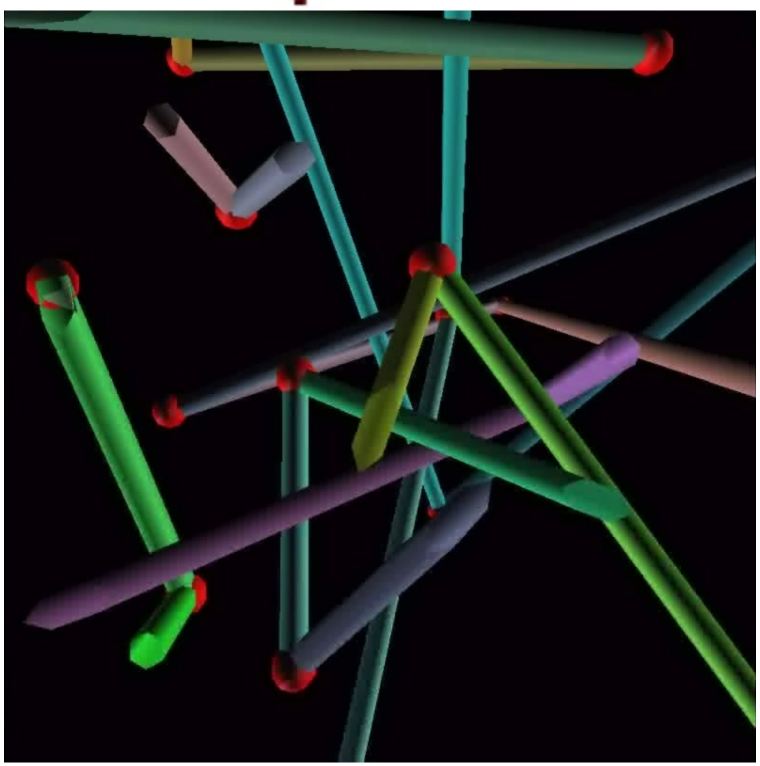


MIT Press

Many Degrees of Freedom

Knots





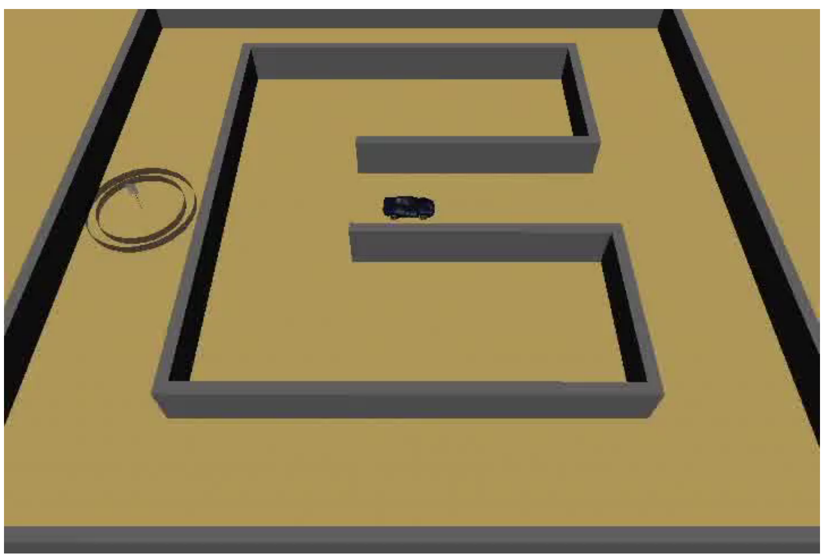
[Ladd, Kavraki]

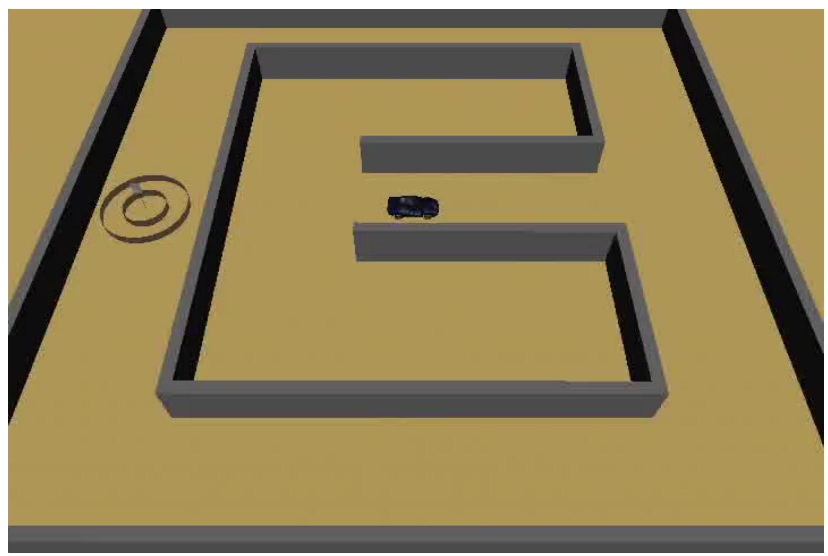
139 vertices (over 400 DOFs)

Increasing the Complexity of the Robot

- Geometry/Dimension

- Differential constraints (maneuver automata, integrators)

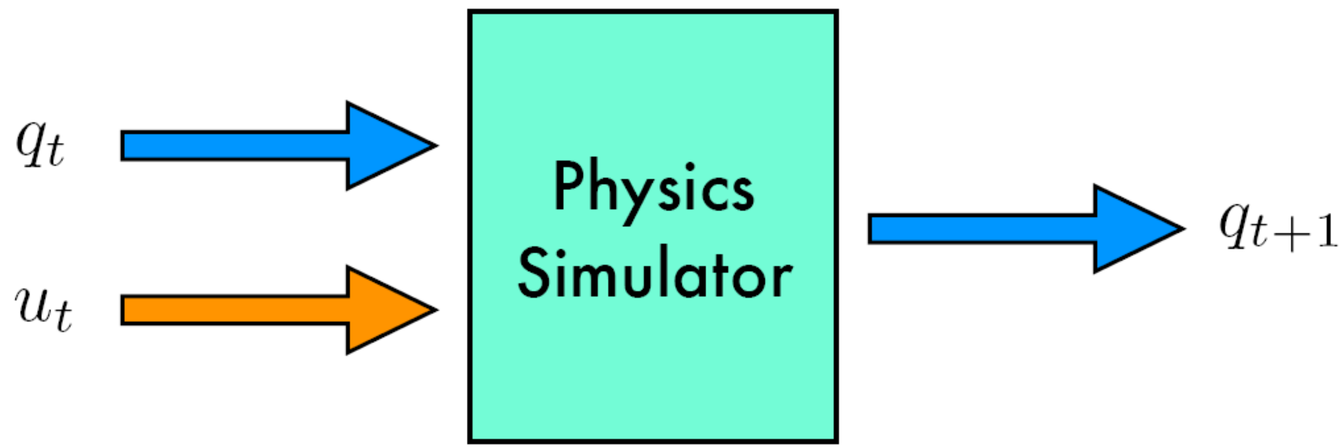




[Bekris, Kavraki]

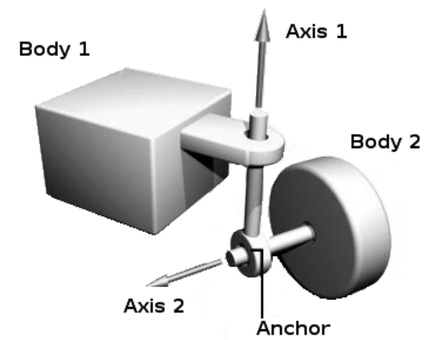
Increasing the Complexity of the Robot

- Geometry/Dimension
- Differential constraints (integrators, maneuver automata)
- Simulation

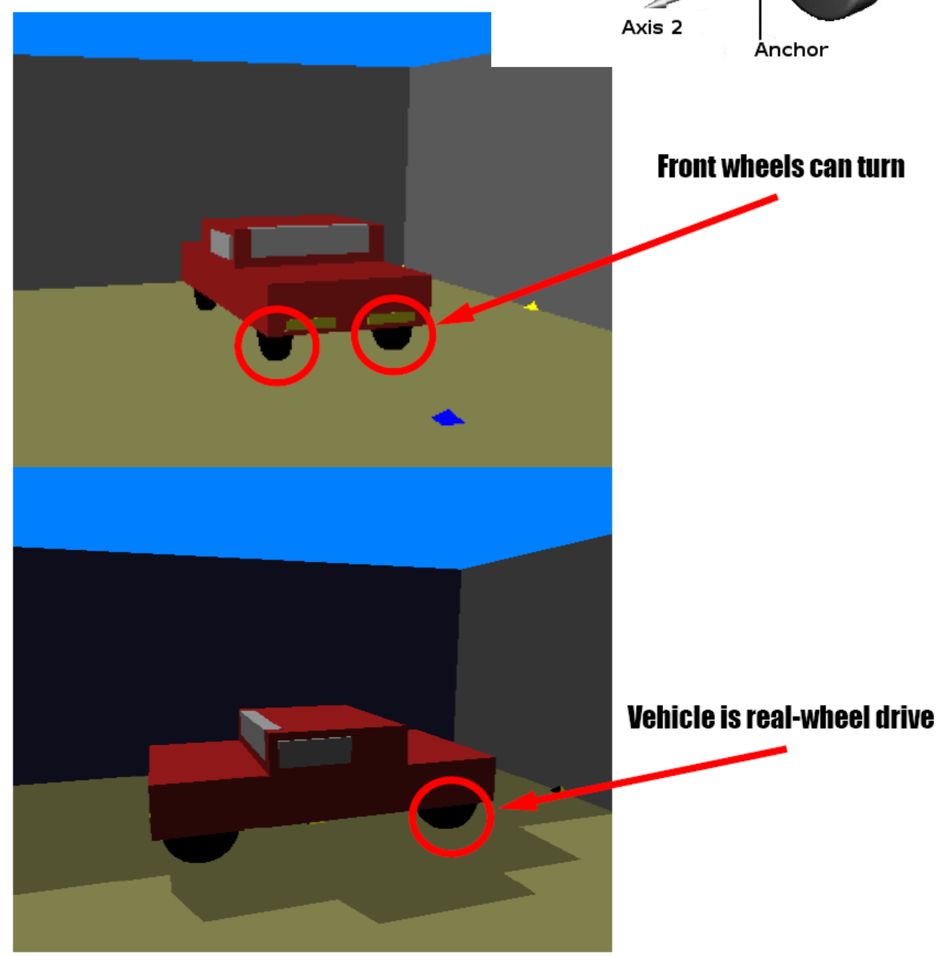


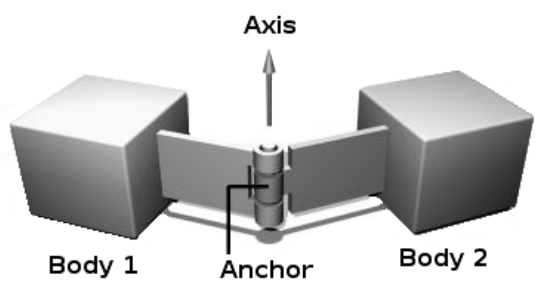
$$F(q_t, u_t) = q_{t+1}$$

Simulated Car

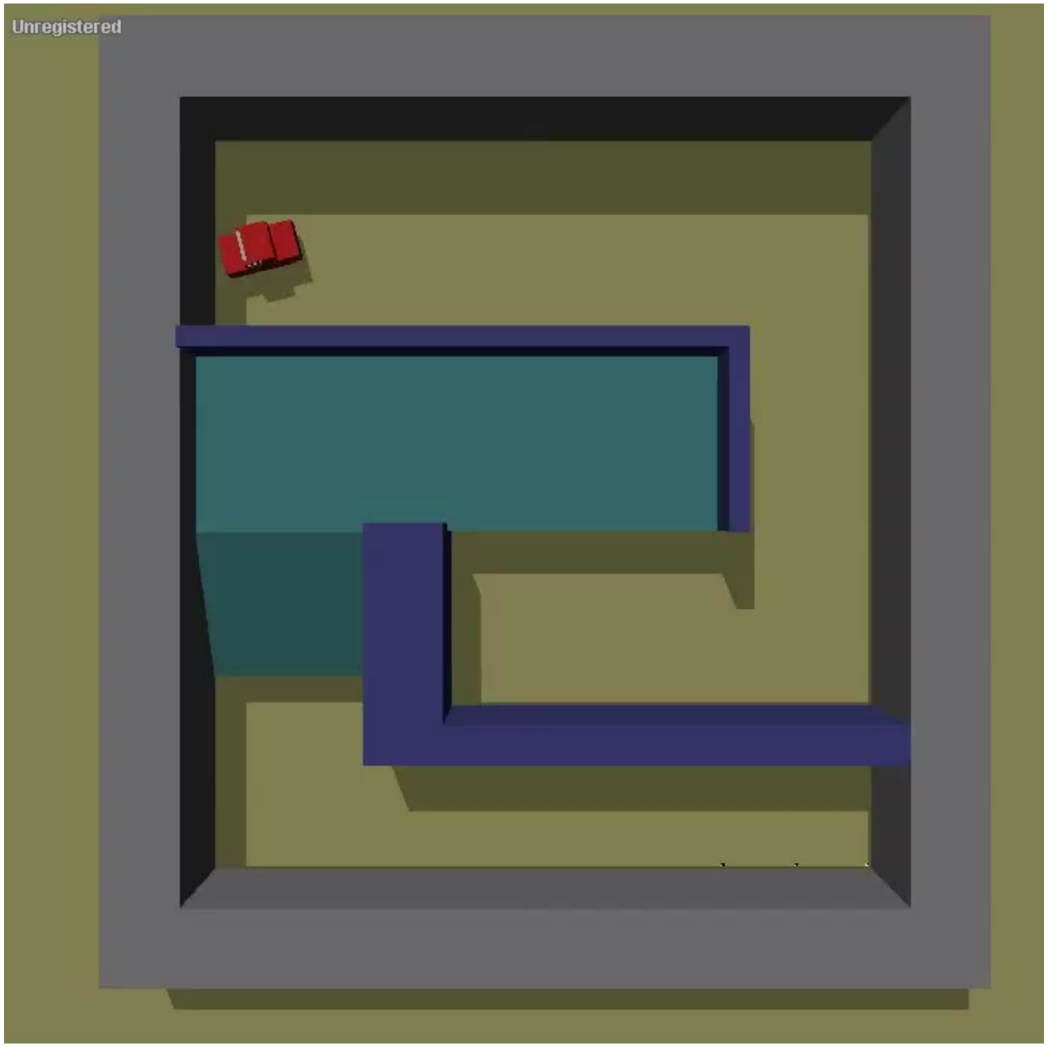


- 3D rigid body dynamics
- Car consists of the body and 4 wheels
- Wheels form friction contacts
- Wheel torques are bounded
- Physical Simulator: ODE
- Stewart-Trinkle model



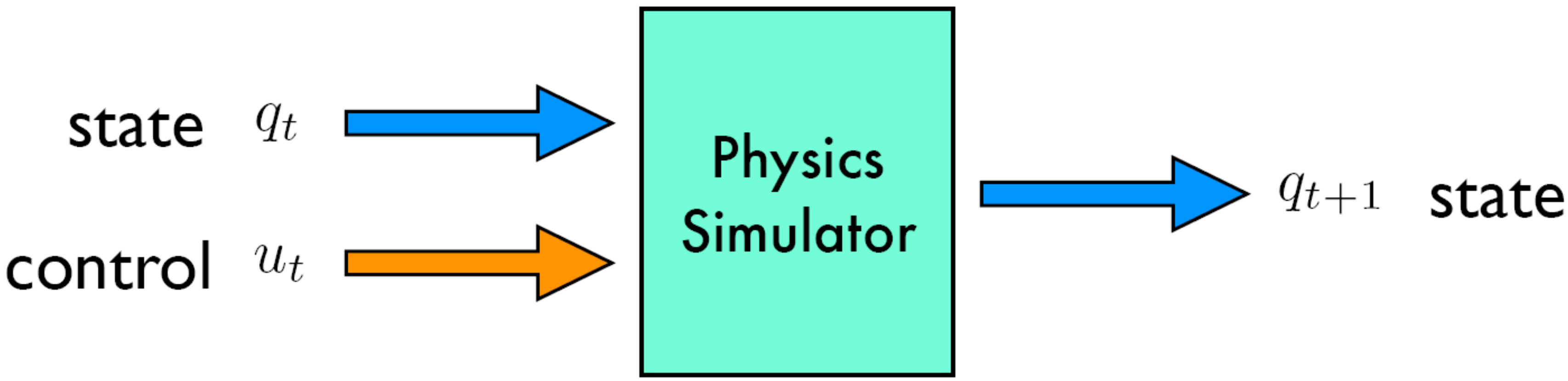


Ramp



[Ladd, Kavraki, PDST]

Physical System Model

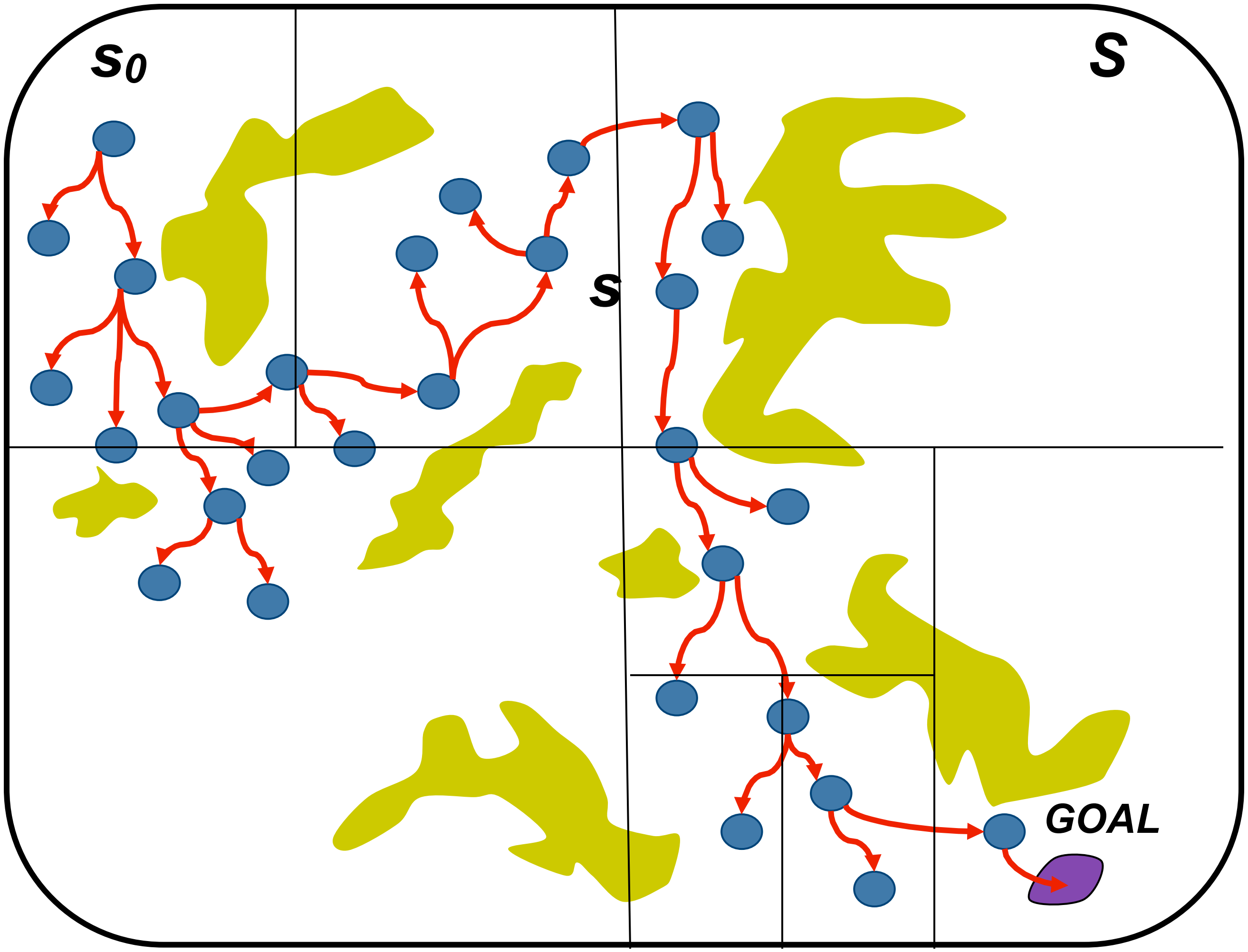


$$F(q_t, u_t) = q_{t+1}$$

q is a state in the state space *Q*

u is a control in the control space *U*

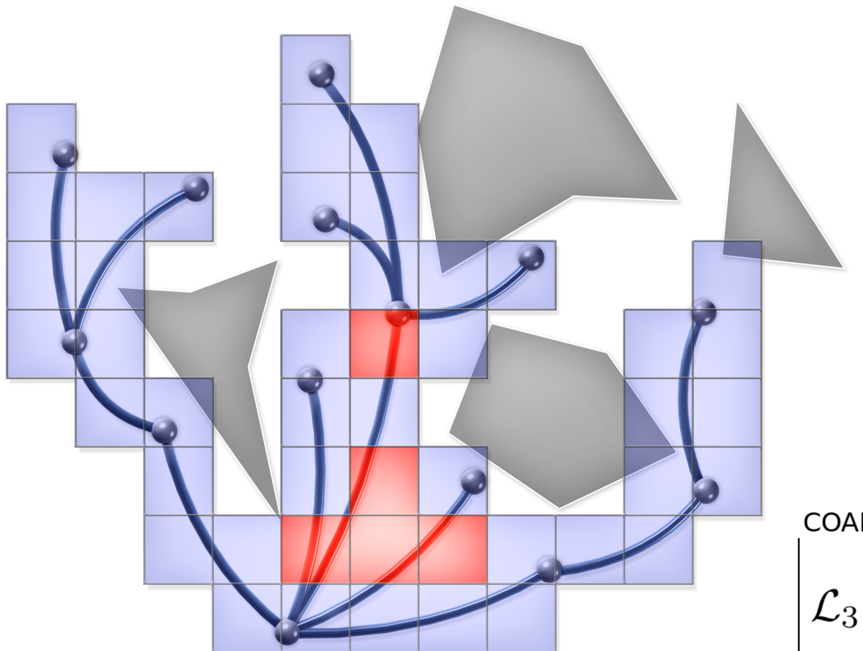
PDST-EXPLORE: Search Combined with Some Decomposition of the Space

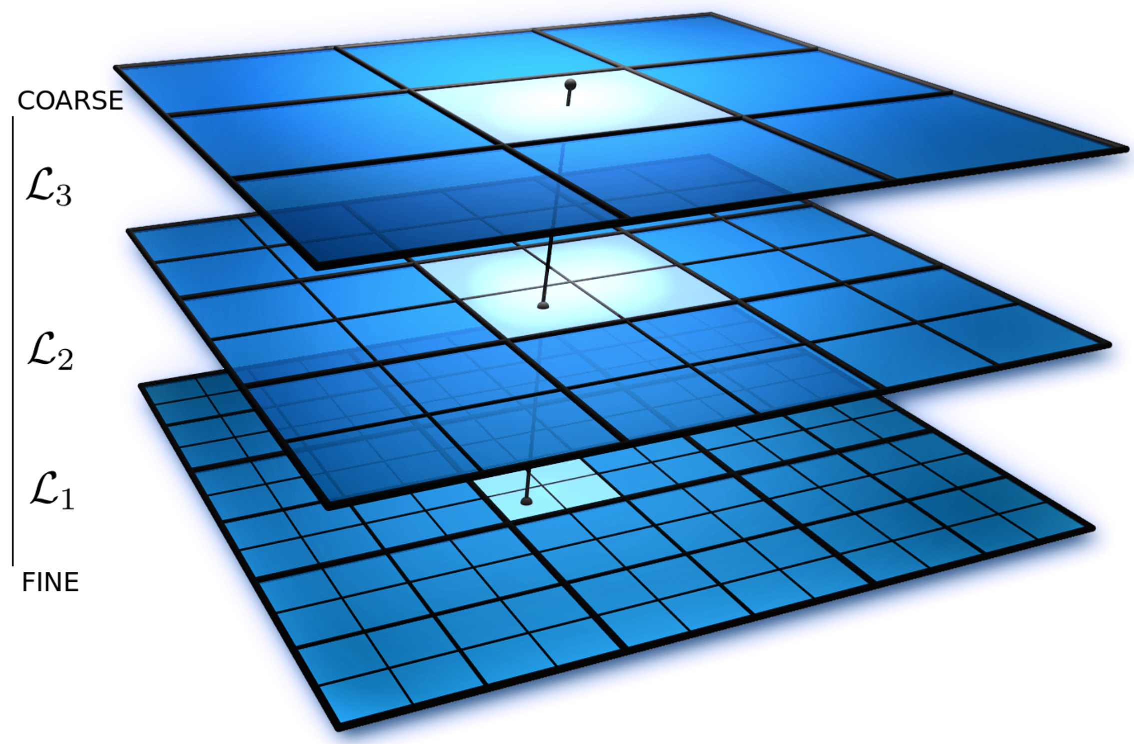


[Ladd, Kavraki]

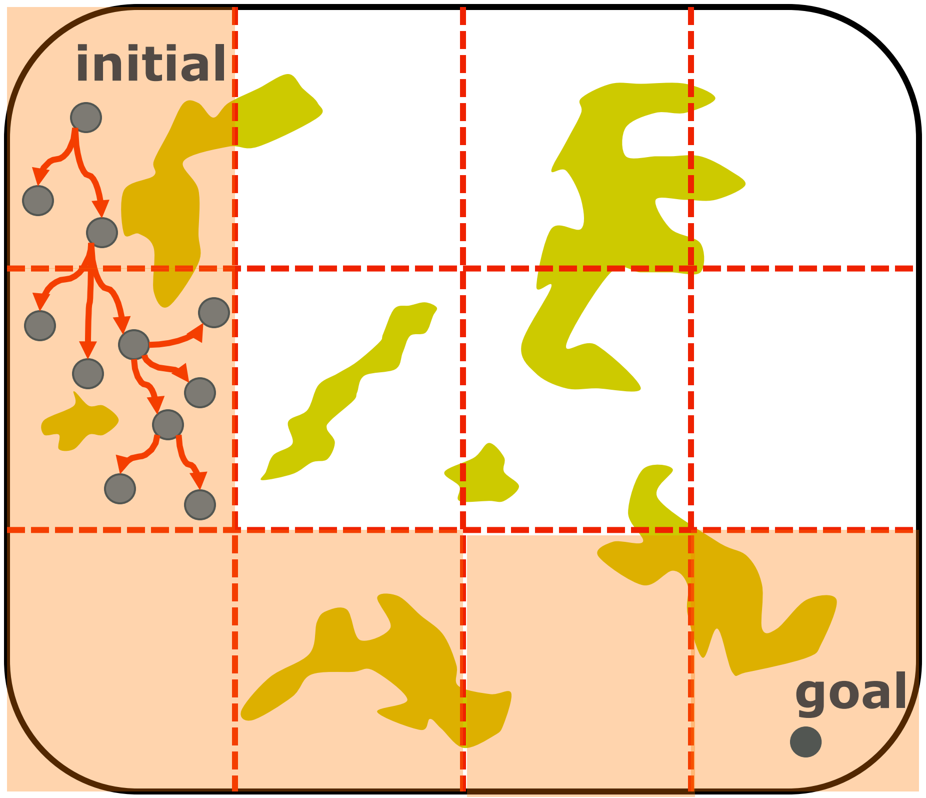
KPIECE

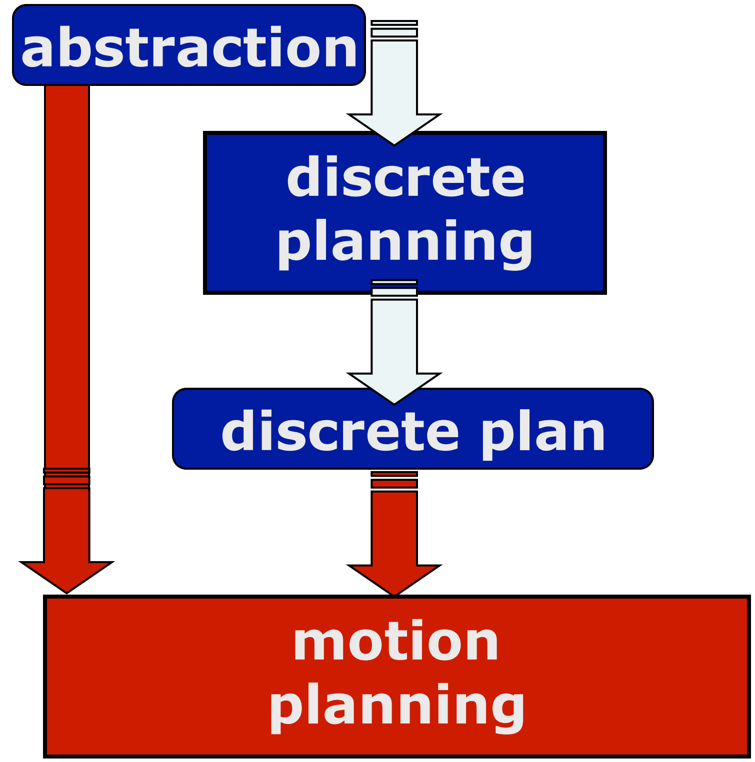
[Sucan, Kavraki]





SyCLoP: Synergistic Combination of Discrete and Continuous Search

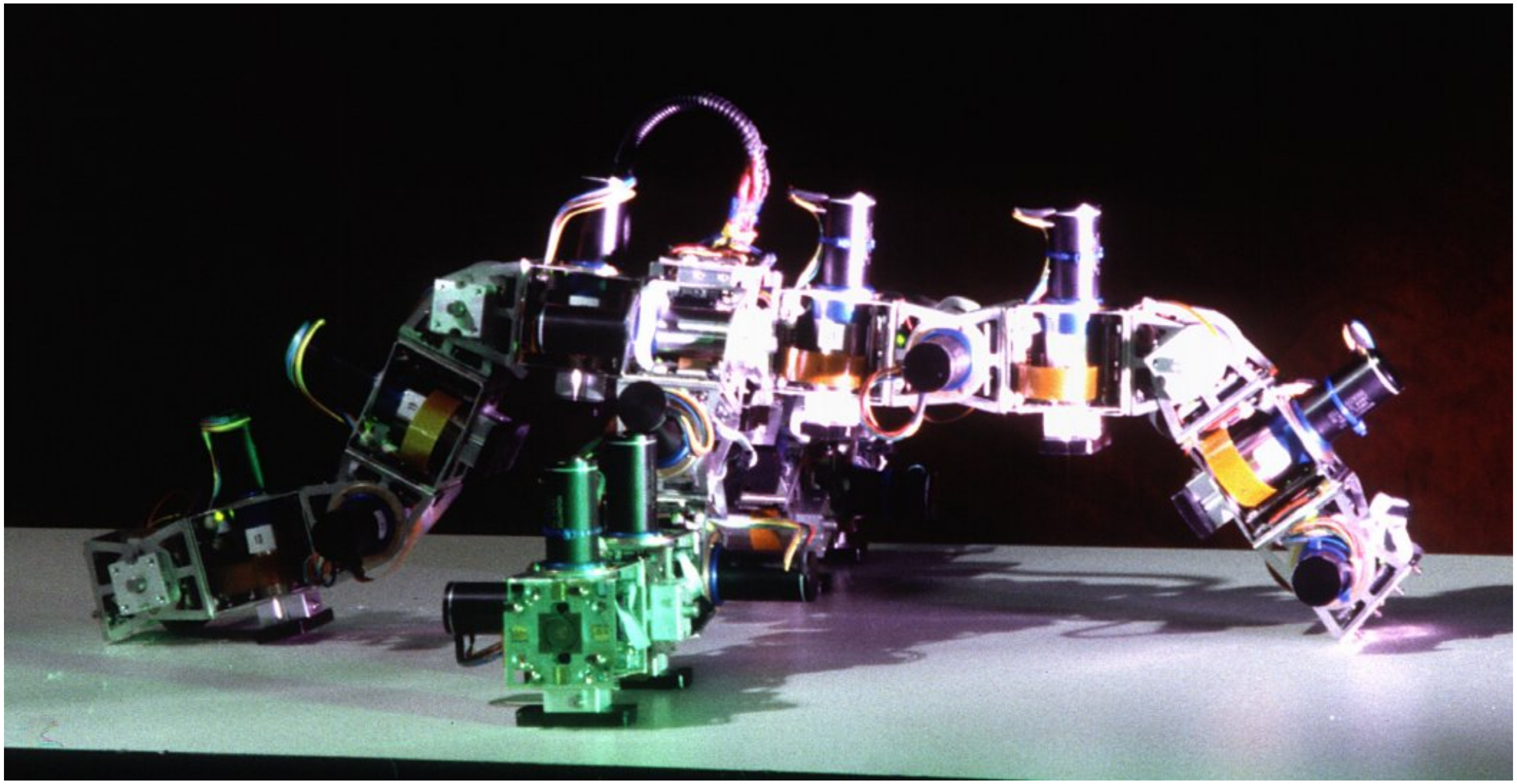




Extend tree branches along regions specified by current discrete plan

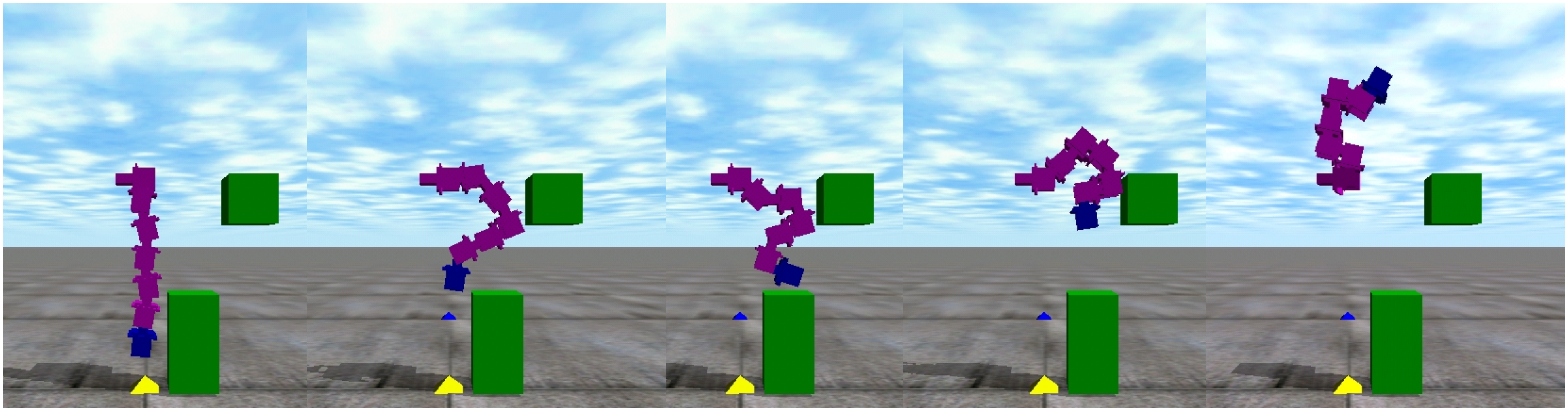
[Plaku, Kavraki]

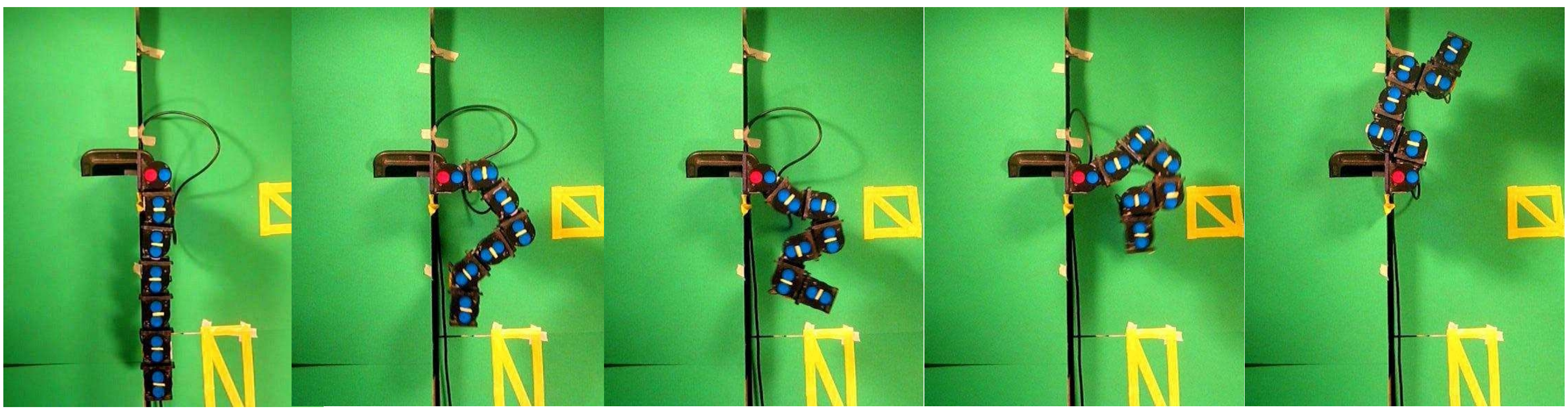
Application to Modular Systems



[with Mark Yim, UPenn]

Actual Systems

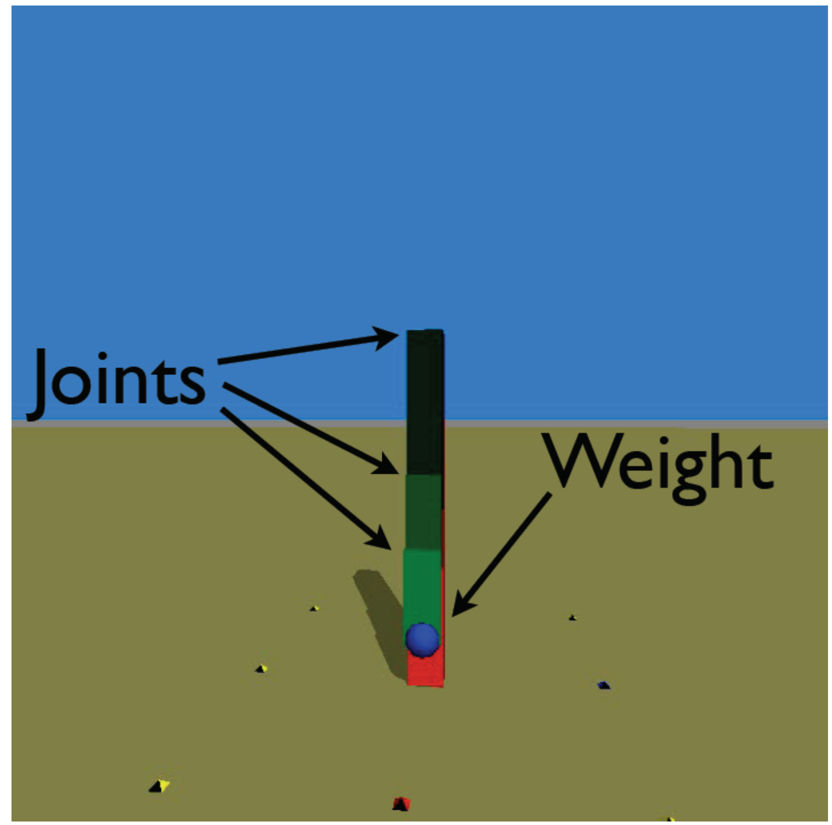




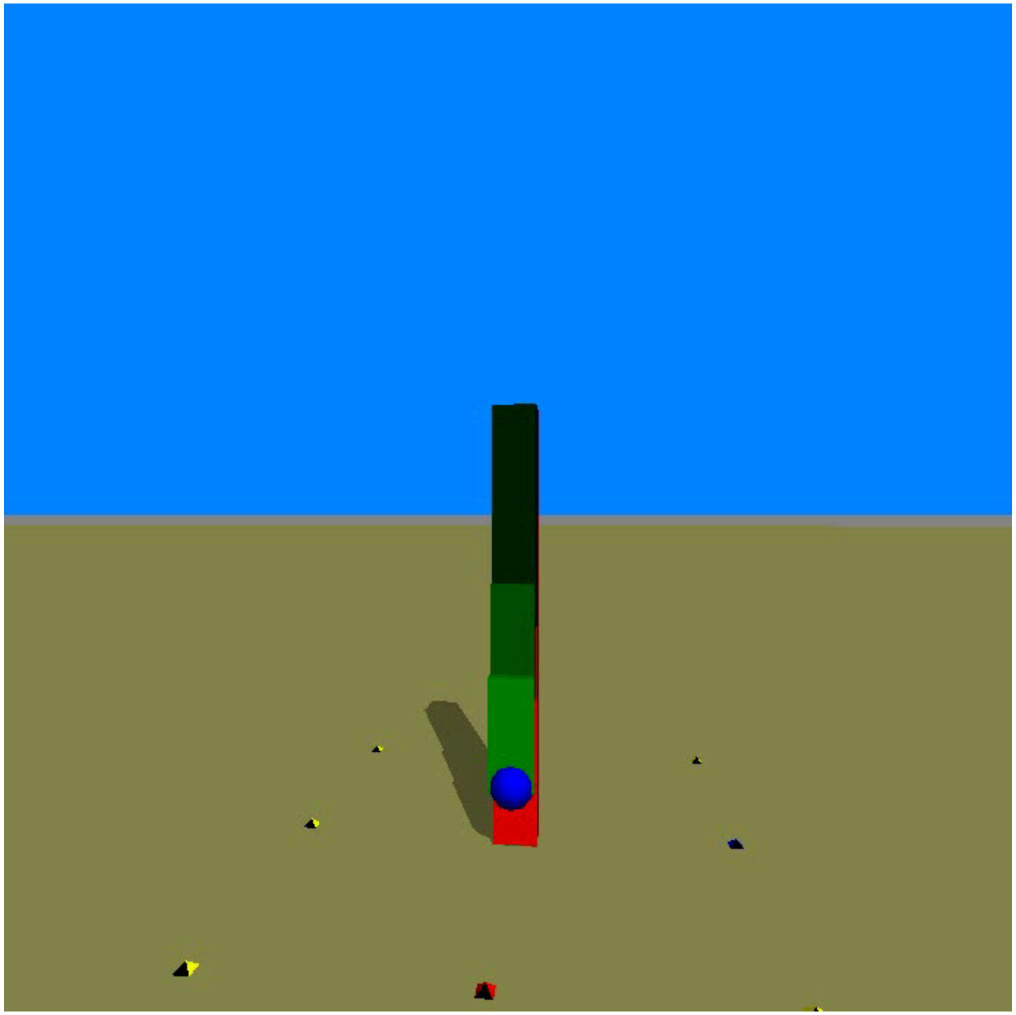
[Sucan, Kavraki - Rice, Yim]

Weight Lifter

- 8 kg weight
- 2 kg arm
- 20 N/m torque at each motor
- Physical Simulator: ODE



Solution Video

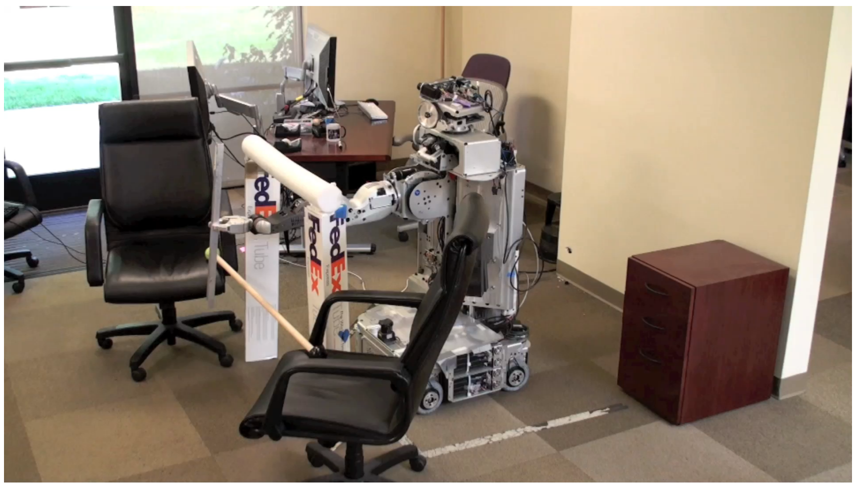


Solve time: 10 seconds

OMPL and OMPL app

OMPL: Open Motion Planning Library:

- Under sourceforge
- Works with ROS



[PR2 – Willow Garage]

OMPL app:

http://www.kavrakilab.org

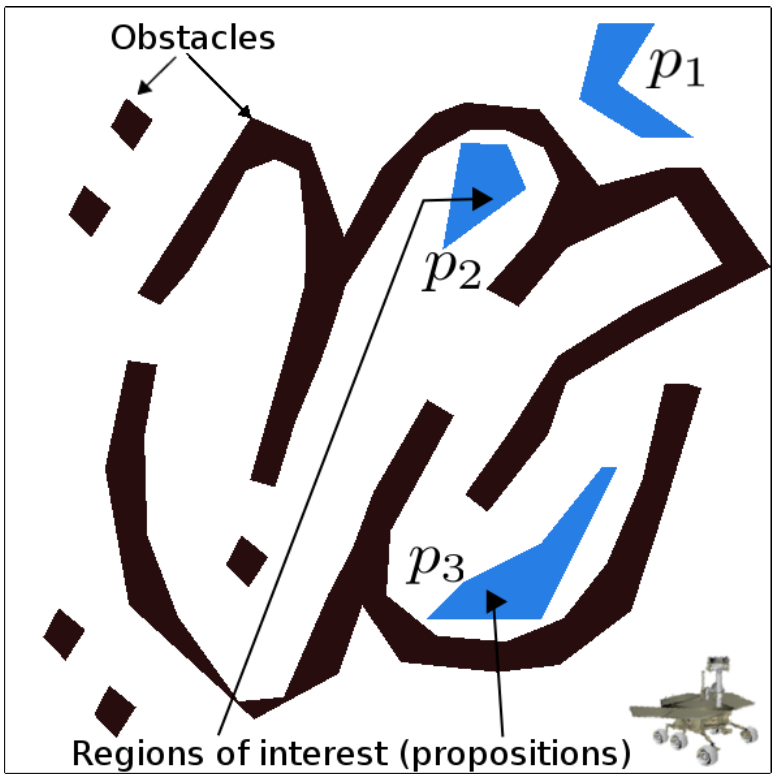
Planning with High-Level Goals

Applications: Motion Planning Falsification of Hybrid Systems

[with Vardi, Plaku, Bhatia]

Motion Planning with Temporal Goals

- **Problem:** Design a motion plan for a given robot model, such that the plan satisfies a prescribed high-level specification.



Example: In future visit p_1, and then visit region p_2 or p_3.

Temporal Logics for Complex Behaviors

- Pnueli introduced Linear Temporal Logic (LTL) in 1977 as a specification mechanism for reactive systems

- Built from:
  - A set of Π of propositional variables: Π **= {p_0,p_1,….,p_N}**
  - Boolean operators: **And** (**&**), **Or** (**|**), **Not** (¬)
  - Temporal operators:

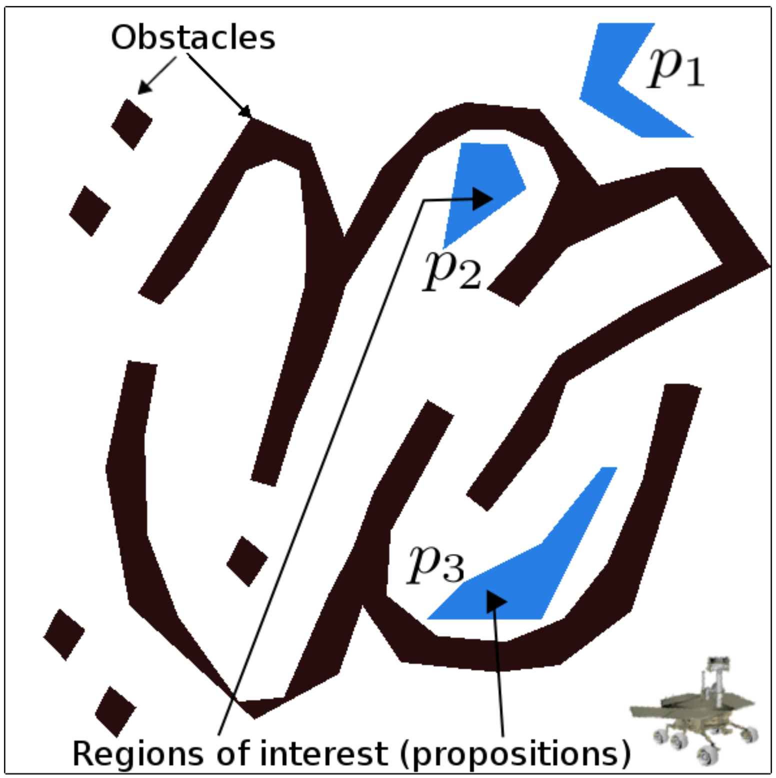
    Next (X), Eventually (F), Always (G), Until (U), Release (R).

- Used to model check properties of computer programs

  [Vardi et al, STOC'84, LICS'86, Clarke et al., '99; Behrmann et al., '01]

Describing High-level Goals using LTL

- Focus on finite-time horizon planning problems
- Describe high-level goals using a finite-horizon LTL formula ϕ
- ϕ is defined over a set of propositions Π **= ($p_0, p_1, \ldots, p_N$)**



Example: In future visit p_1, and then visit region p_2 or p_3.
ϕ**= F (p_1 & F (p_2 | p_3))**

Earlier LTL-related Work

- Automated LTL planning for multi-agent systems: Loizou *et al.,* 04; Kloetzer *et al.,* 07, 08; Karaman *et al.,* 08
- Automated approaches to LTL planning: Kloetzer *et al*., 06; Fainekos *et al.,* 09, Belta et al, 09
- Sampling-based approach for μ-calculus: Karaman *et al*., 09
- Receding horizon approach to LTL planning, Wongpiromsarn *et al*, 09
- Multi-layered approach to LTL planning: Plaku, Kavraki, Vardi, 09
- and many more…….

Distinguishing Features of Our Work

- Focus on complex models

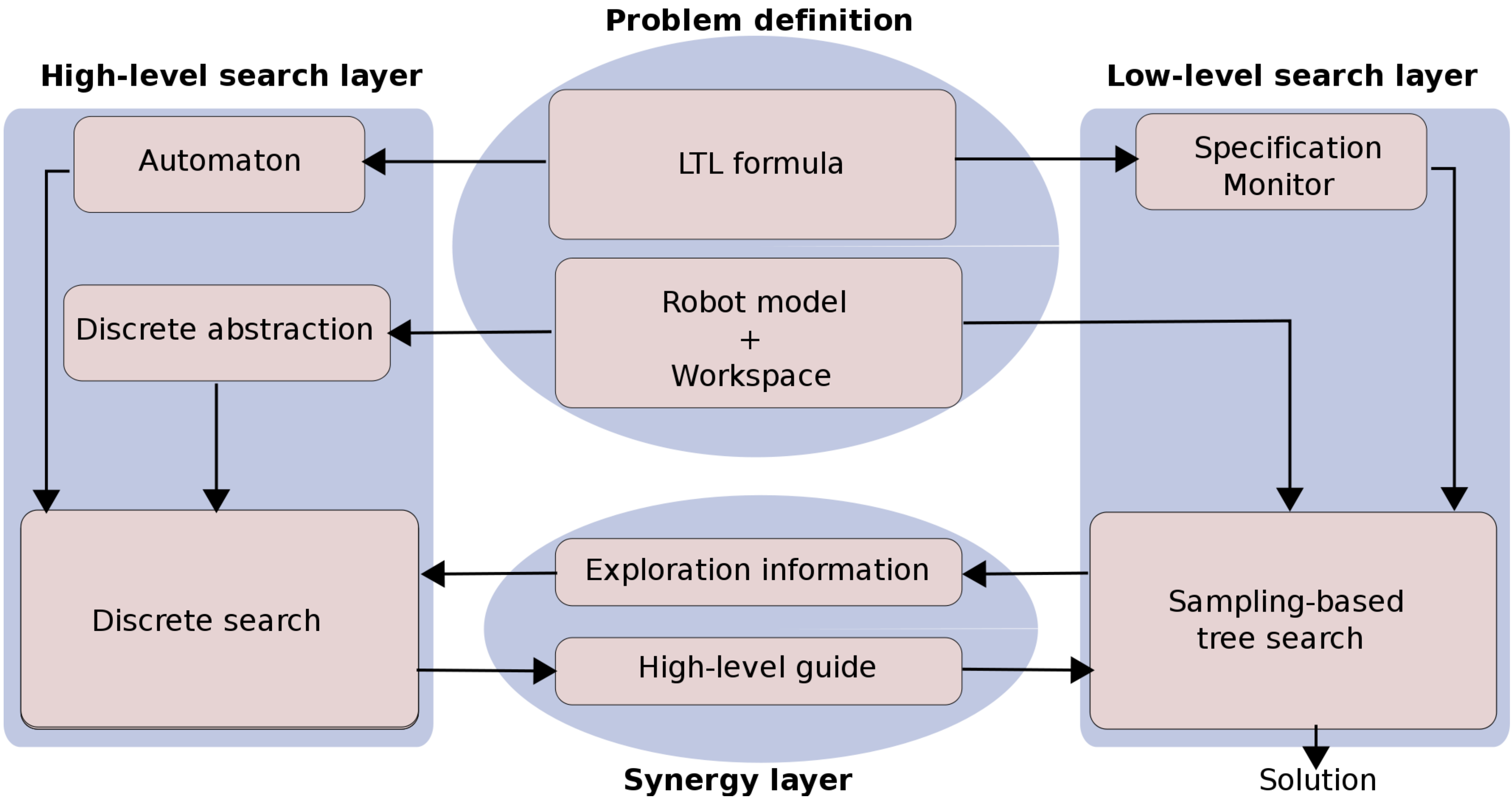
- **Complexity at the physical level:**
  - Nonlinearity of robot model
  - Geometric constraints
  - Possibly hybrid dynamics

- **Complexity at the task level:** operators of the LTL specification

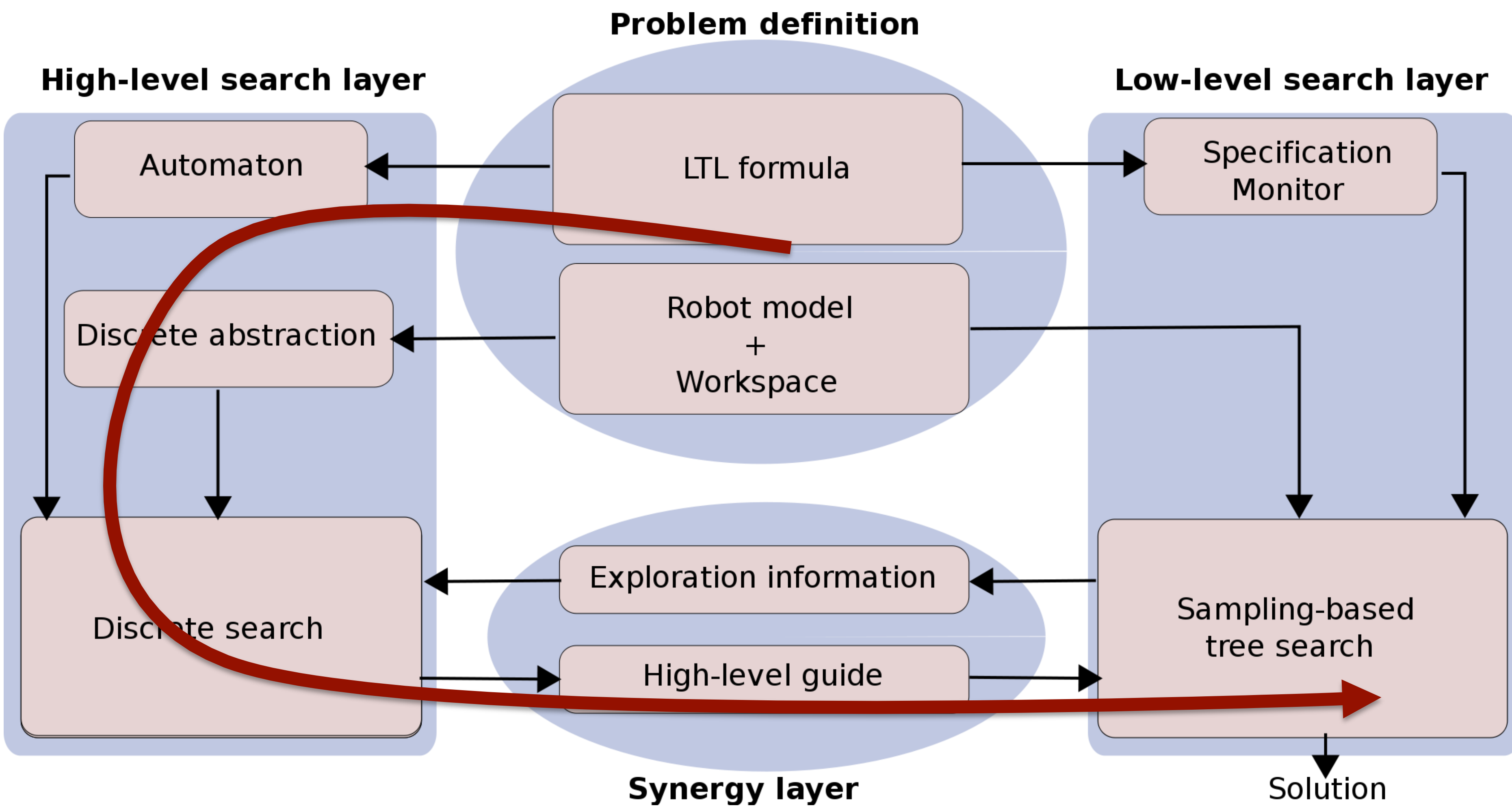
Our approach:

- Relax strong completeness guarantees
- Address both the discrete and continuous nature of problem
- Use a multi-layered synergistic approach

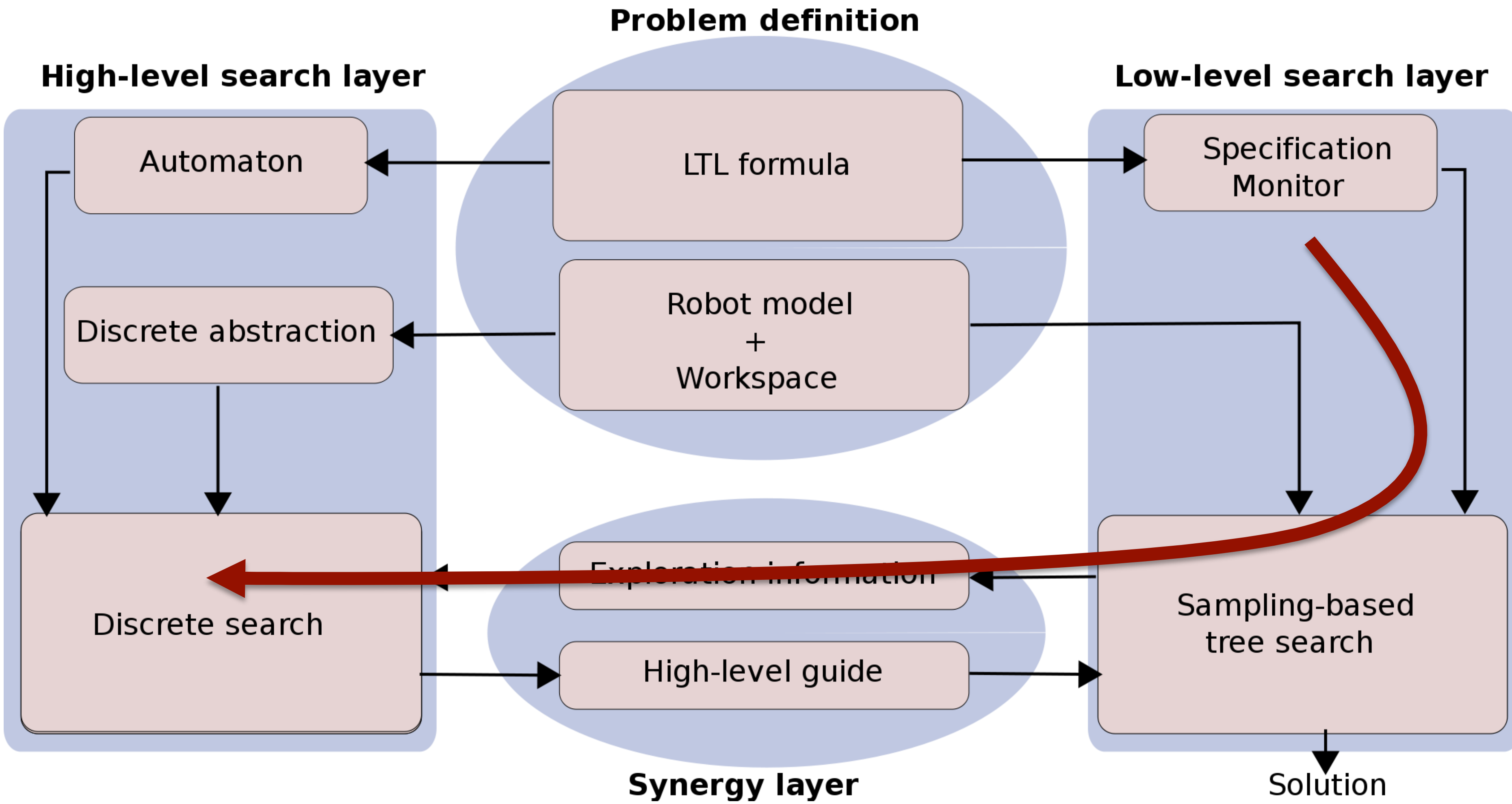
Our Solution



Our Solution



Our Solution

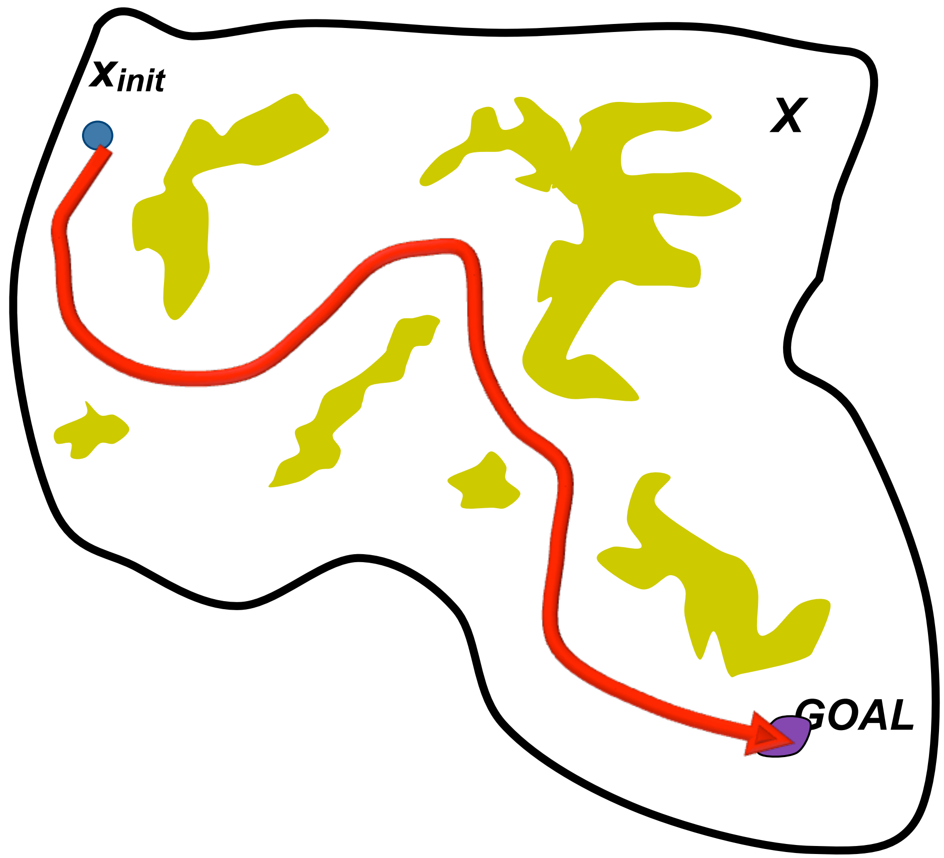


Motion Planning for Continuous Systems

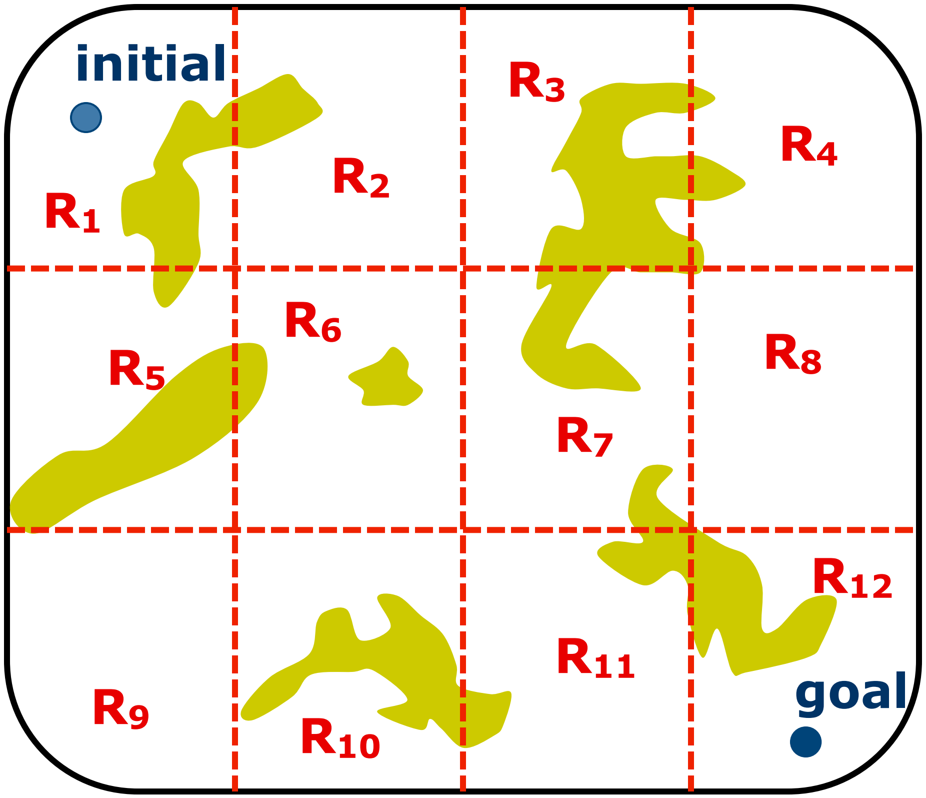
Compute a trajectory from an initial state to a goal region

Trajectory should satisfy all state constraints (e.g., no collisions) and dynamics constraints

We use SyCLoP

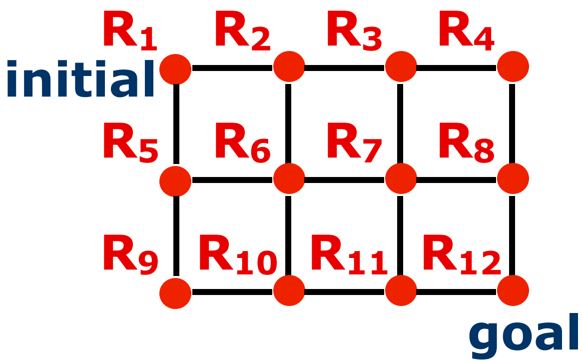


SyCLoP: Discrete Abstraction



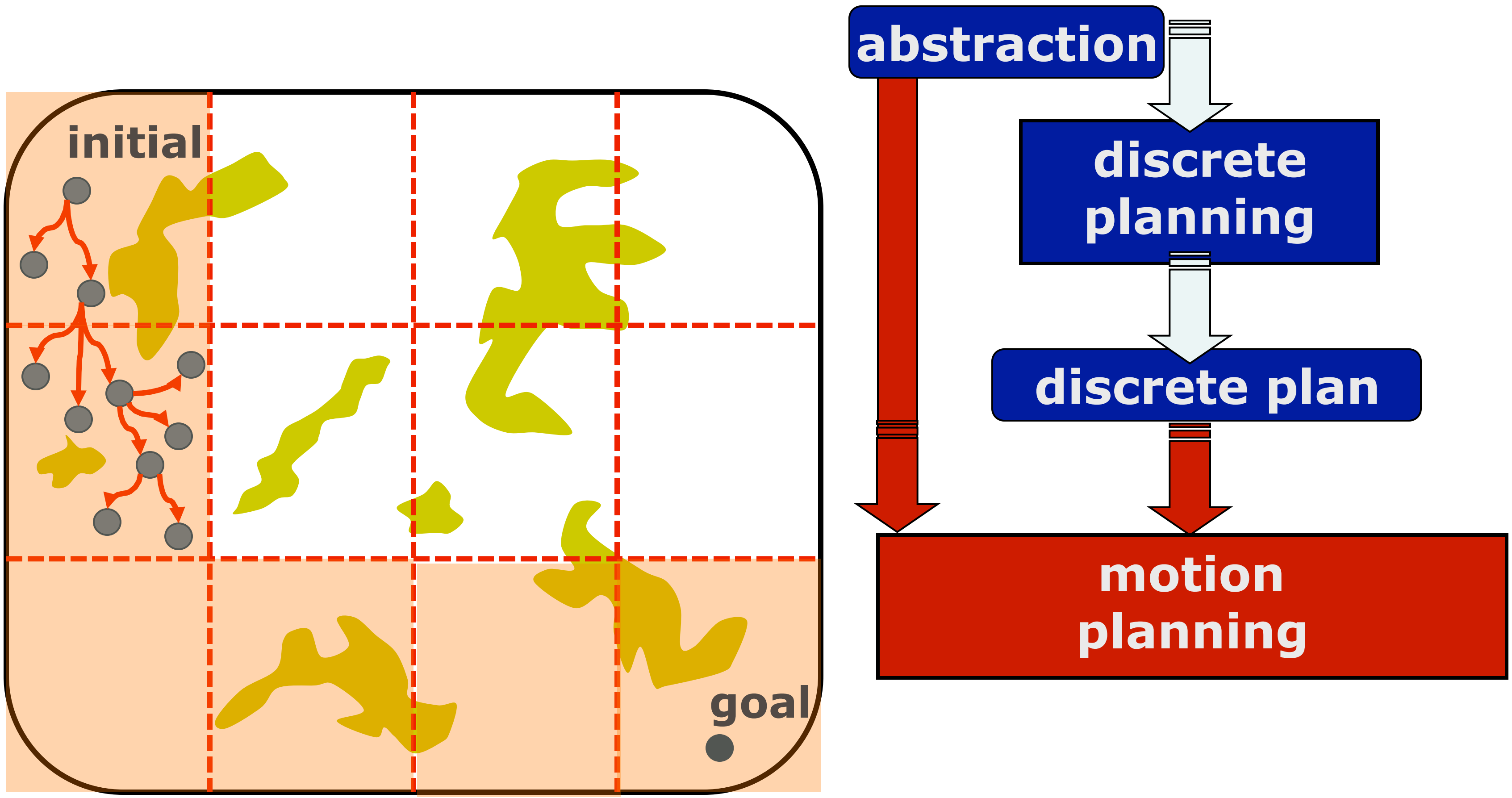
- Decomposition of state space into regions

- Abstraction graph encodes adjacency of regions



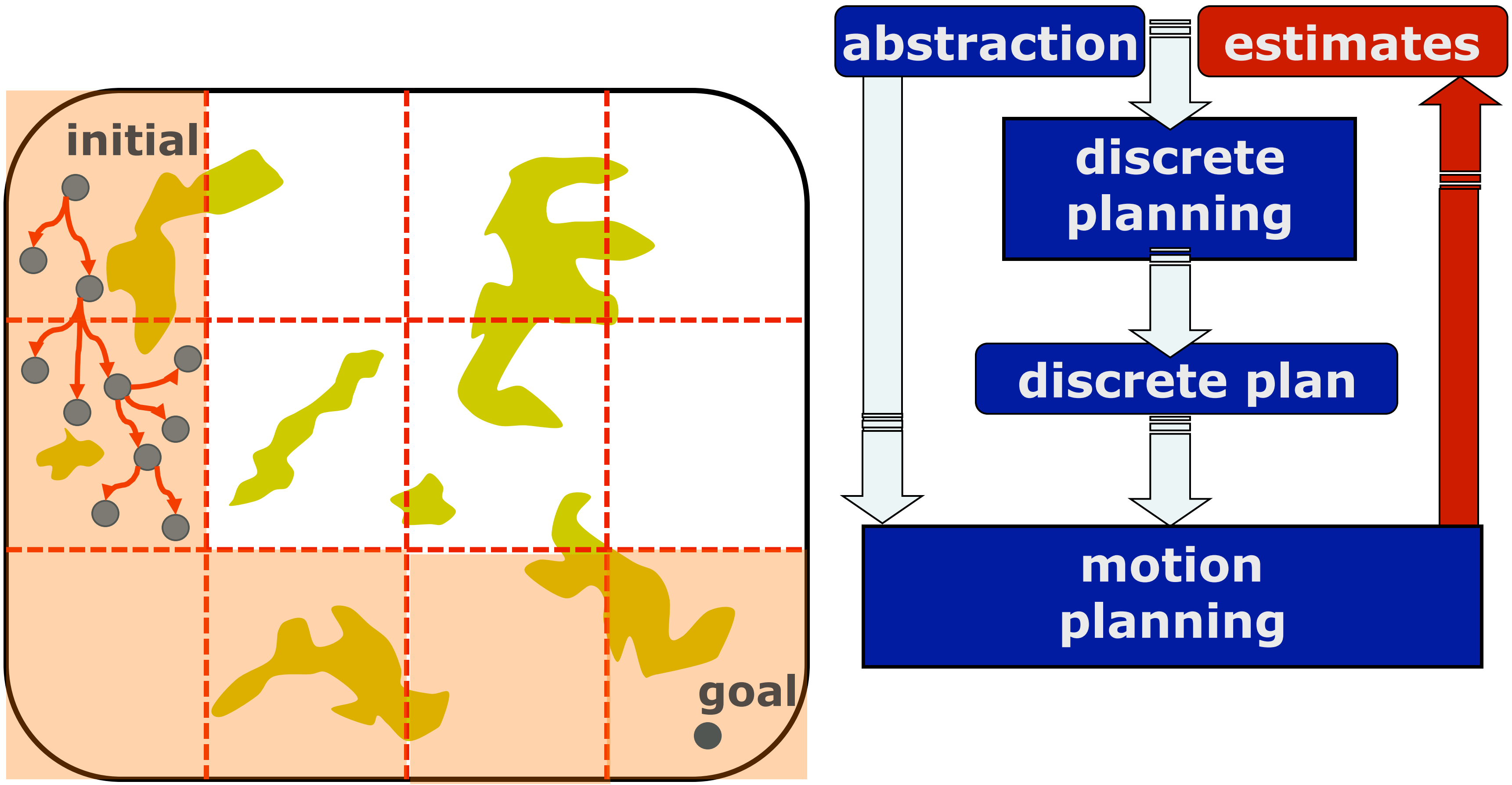
Abstraction provides *discrete plans*: sequences of regions connecting initial to goal

SyCLoP: Synergistic Combination of Discrete and Continuous Search



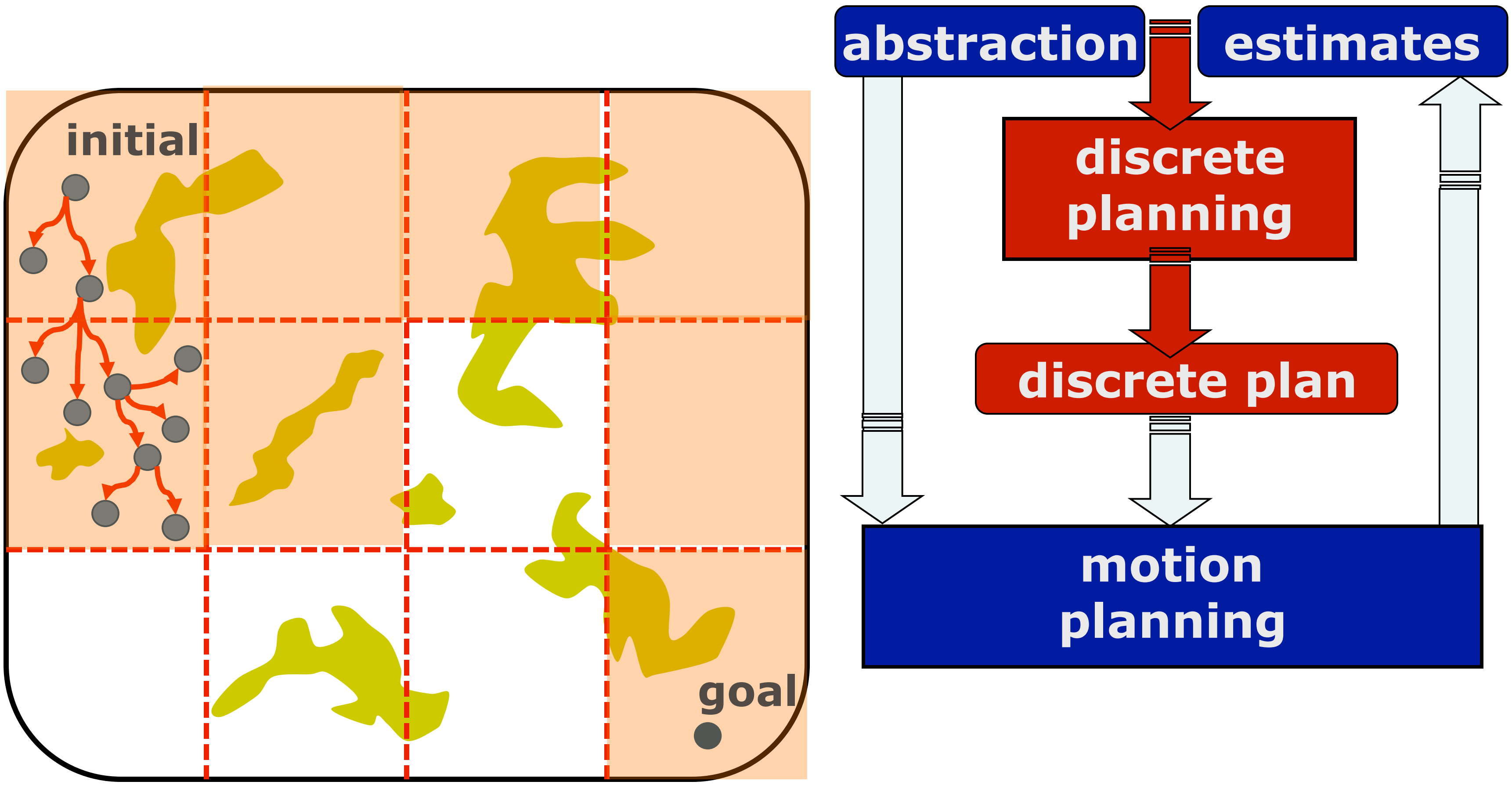
Extend tree branches along regions specified by current discrete plan

SyCLoP: Synergistic Combination of Discrete and Continuous Search



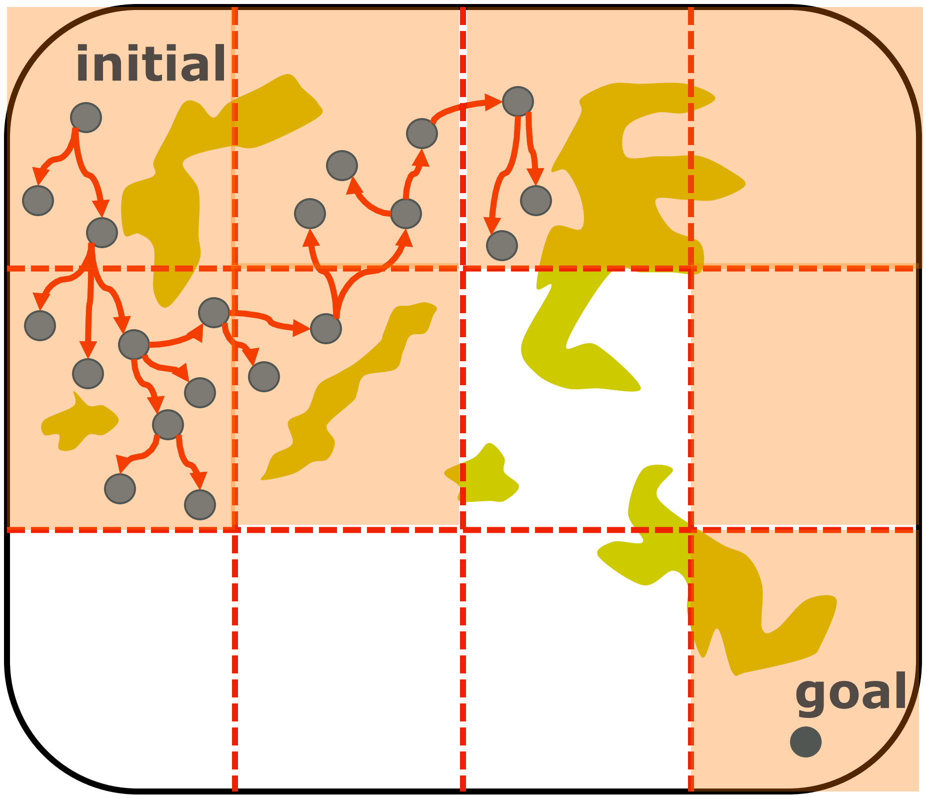
Update feasibility estimates based on information gathered during motion planning

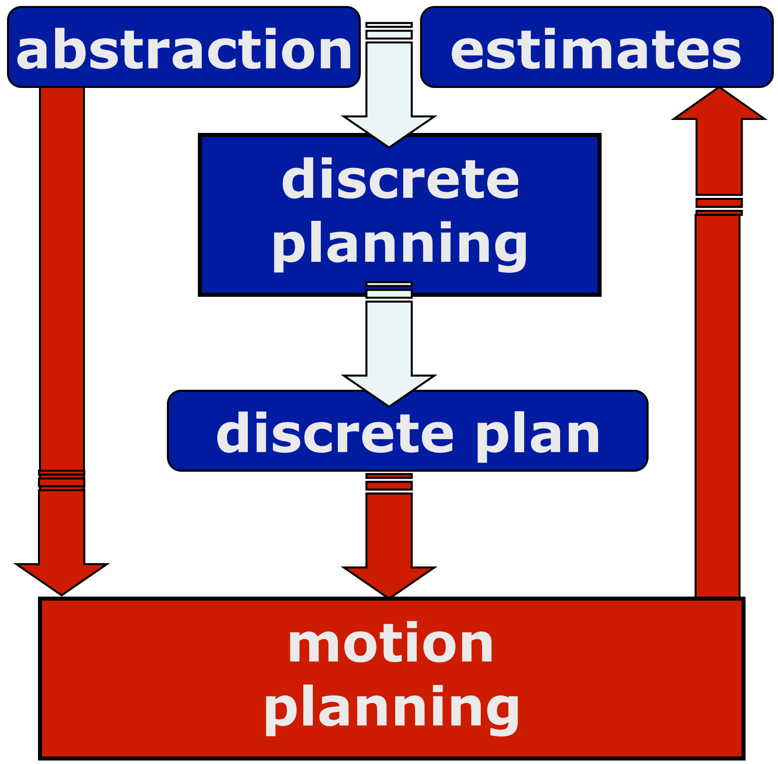
SyCLoP: Synergistic Combination of Discrete and Continuous Search



Compute new discrete plan based on updated feasibility estimates

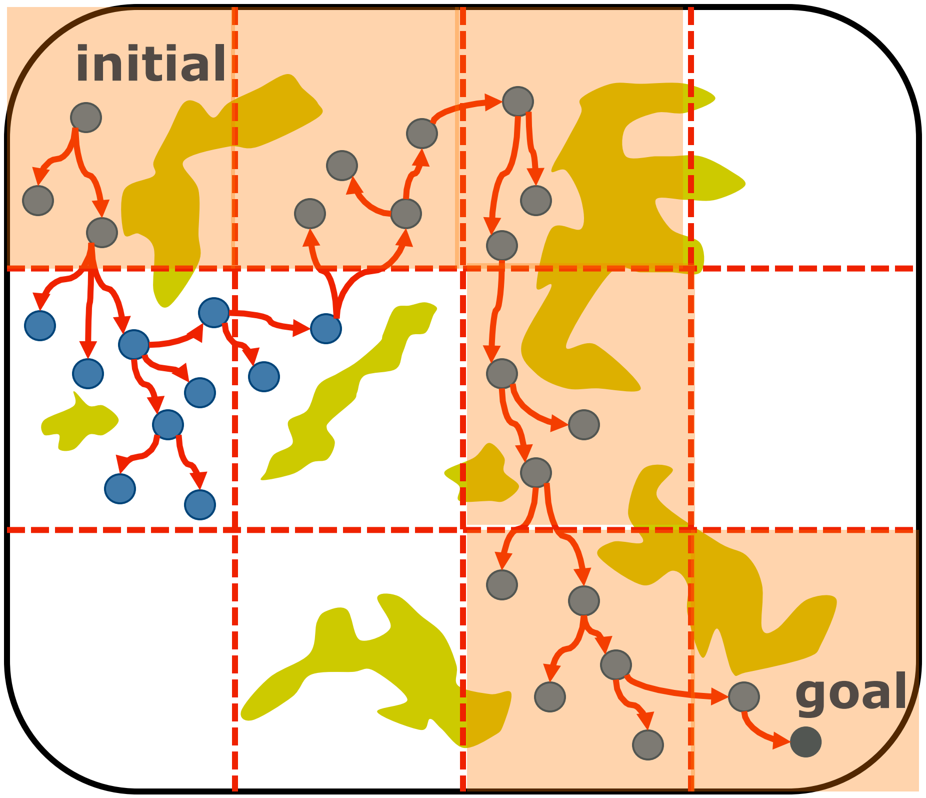
SyCLoP: Synergistic Combination of Discrete and Continuous Search

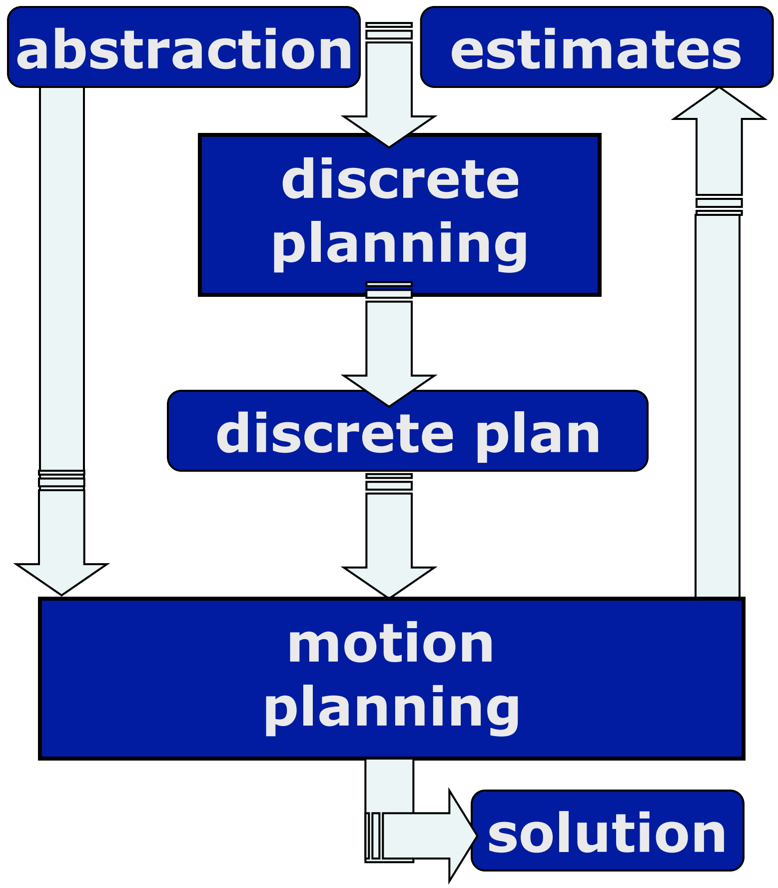




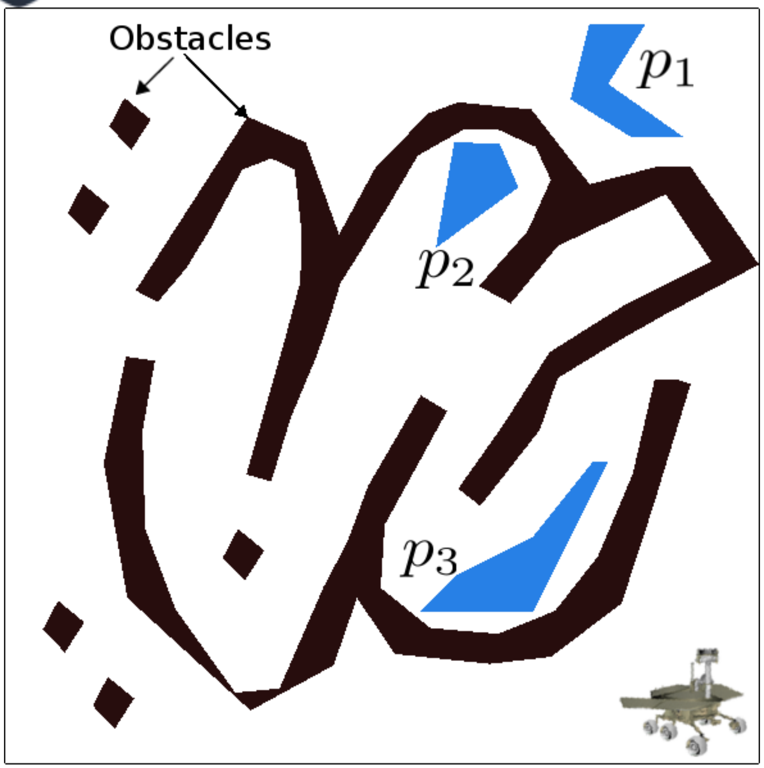
Extend tree branches along regions specified by new discrete plan &
Update feasibility estimates

SyCLoP: Synergistic Combination of Discrete and Continuous Search

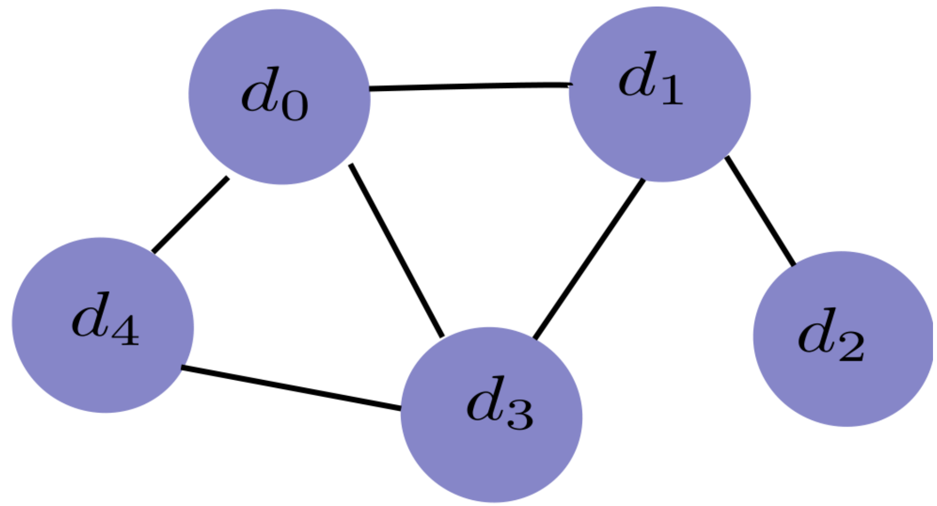




Discrete Abstraction



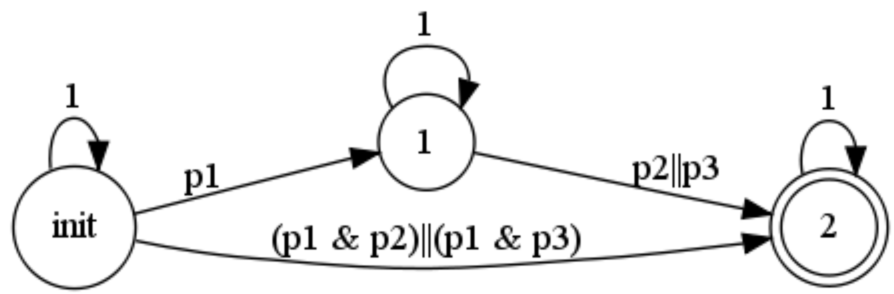
Robot model + Workspace



Discrete abstraction (M)

- **Abstraction: M = (D, d_0, $\rightarrow_D$, h_D)**
- **States: D, d_0 $\in$ D** is the initial state
- **Transitions:** $\rightarrow_D \subset D \times D$, is the transition relation
- **Observation map: h_D : D $\rightarrow$ 2^{Π},** maps states to propositions

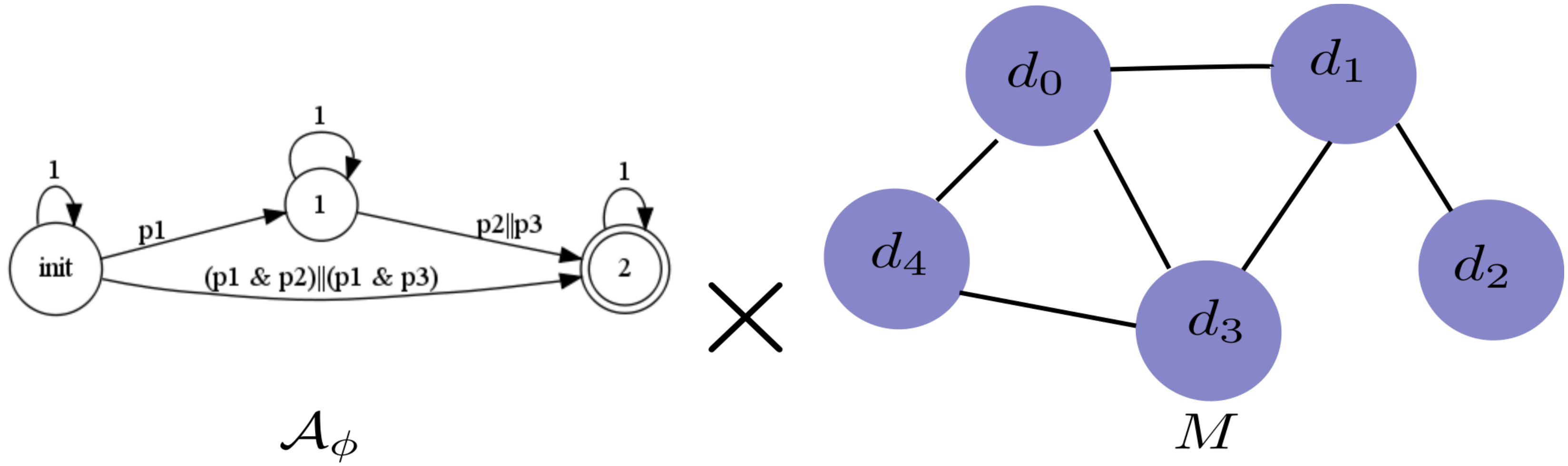
Co-safe LTL Specifications



A_ϕ for ϕ = F (p1 & F (p2 | p3))

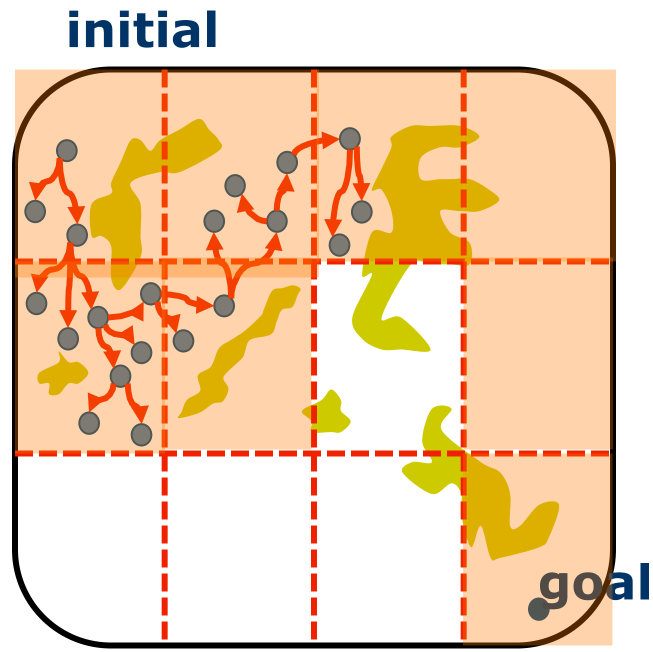
- **Co-safe LTL :** Good trace satisfying ϕ has a finite good prefix
- **Syntactically co-safe LTL formulas:**
  - Write ϕ in Positive Normal Form (PNF)
  - Check the temporal operators in the formula
  - **Next, Eventually, Until** operators only $\Rightarrow$ ϕ is syntactically co-safe (and hence co-safe)
- **Automaton representation:** An NFA $\mathbf{A}_\phi$ describes all the good prefixes for ϕ (automaton on finite words; Vardi, Kupferman, FMSD 01)
- Model checking tools produce a minimized DFA used in this work

High-level Layer



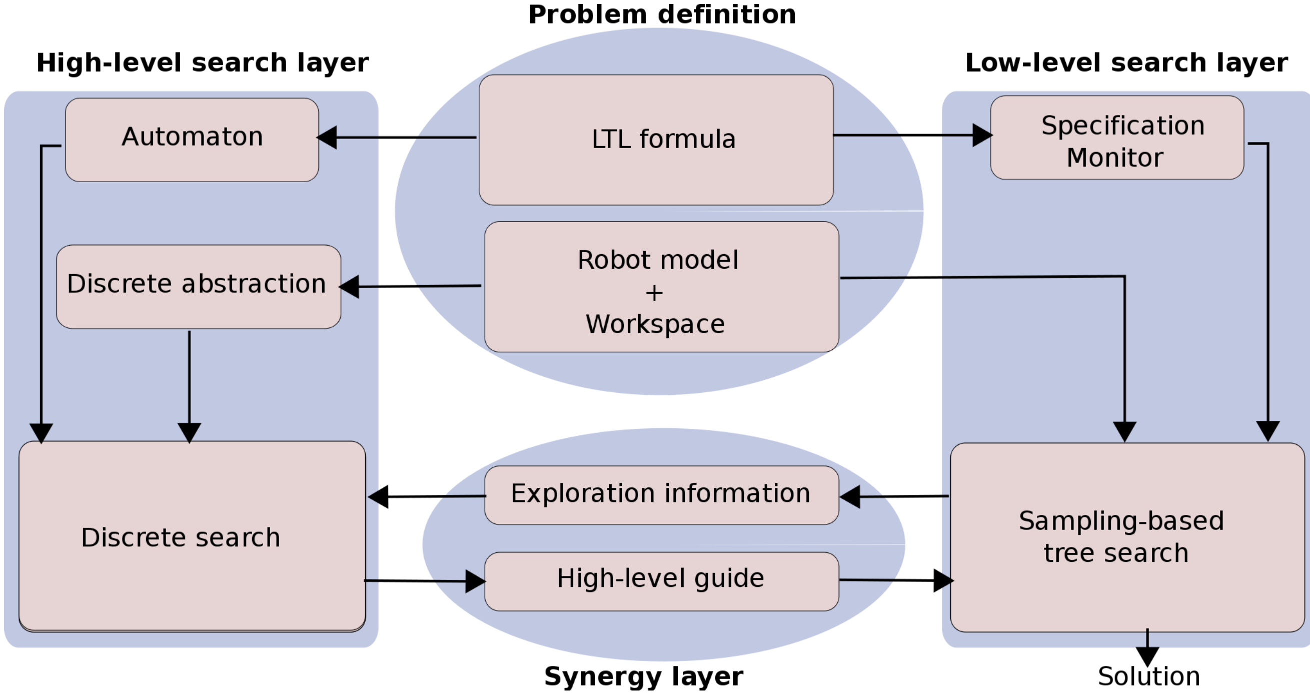
- **Model:** High-level planning for abstraction M
- **Planning space:** Automaton states (**A_ϕ.Z**) $\times$ Abstract states (**D**)
- **High-level state: (z, d)** $\in$ **A_ϕ.Z** $\times$ **D**
- **High-level plan:** Sequence of high-level states, ζ **= $(z_i, d_i)^k_{i=1}$**
  - **$d_i \rightarrow_D d_{i+1}$** $\forall$ **i** $\in$ **[1,k-1]** (feasible transition for abstraction)
  - **$z_i \in \lambda (z_{i-1}, h_D(d_i))$** (feasible transition for automaton)
  - **$z_k \in A_\phi.Z_{acc}$** (last automaton state is an accepting state)

Low-level Layer



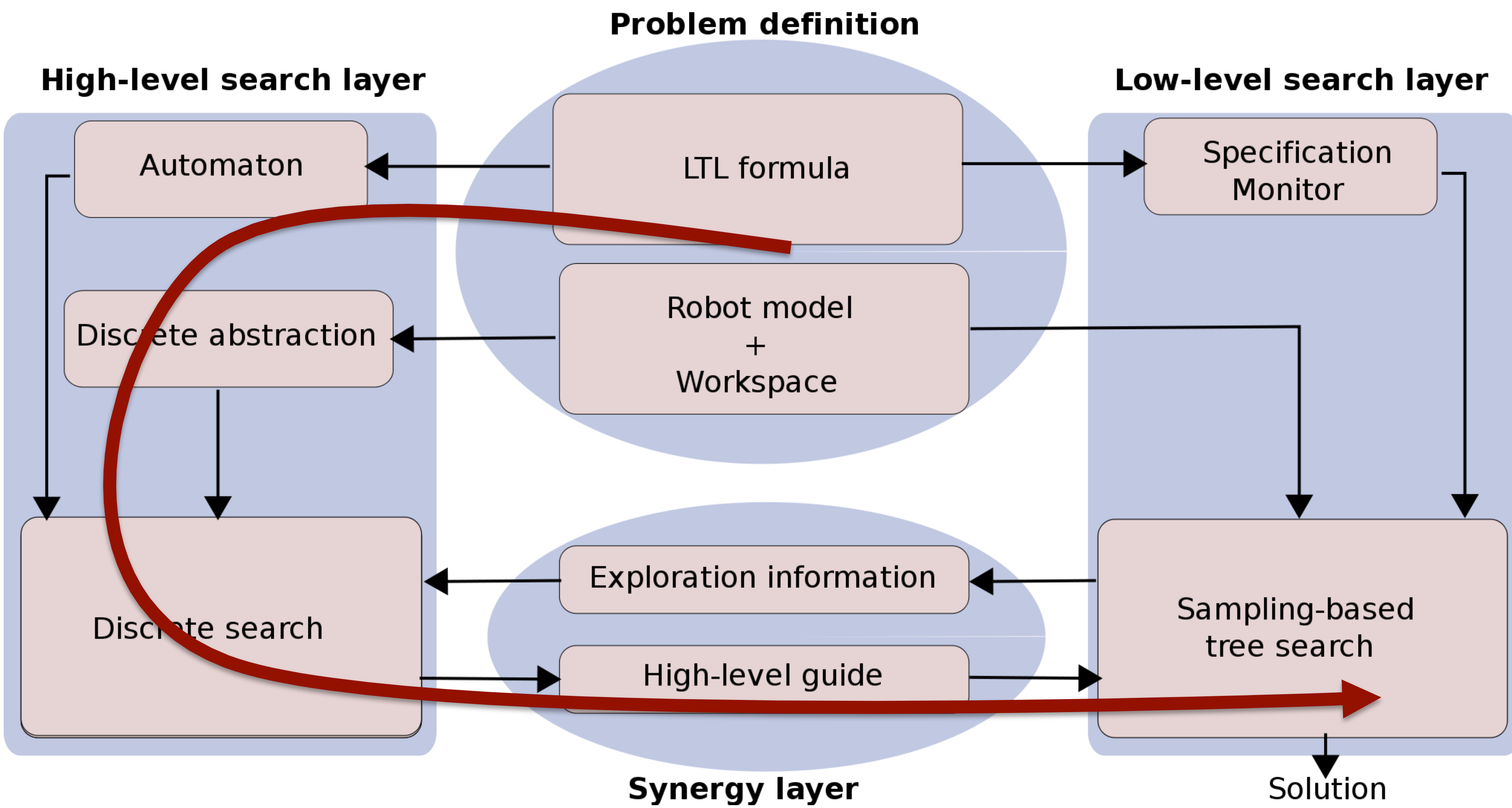
- **Exploration space:** Robot's state space
- **Guide:** High-level plan ζ
- **Data structures:** Tree vertices store edges, automaton state, state of system and other bookkeeping information
- **Search procedure:**
  1. Pick a feasible high-level state (z,d)
  2. Select a tree vertex from (z, d).vertices
  3. Simulate system dynamics, using heuristics of choice
  4. Update feasibility estimates
  5. Raise a flag if accepting state of automaton reached
  6. Repeat from Step 1

Synergy Layer

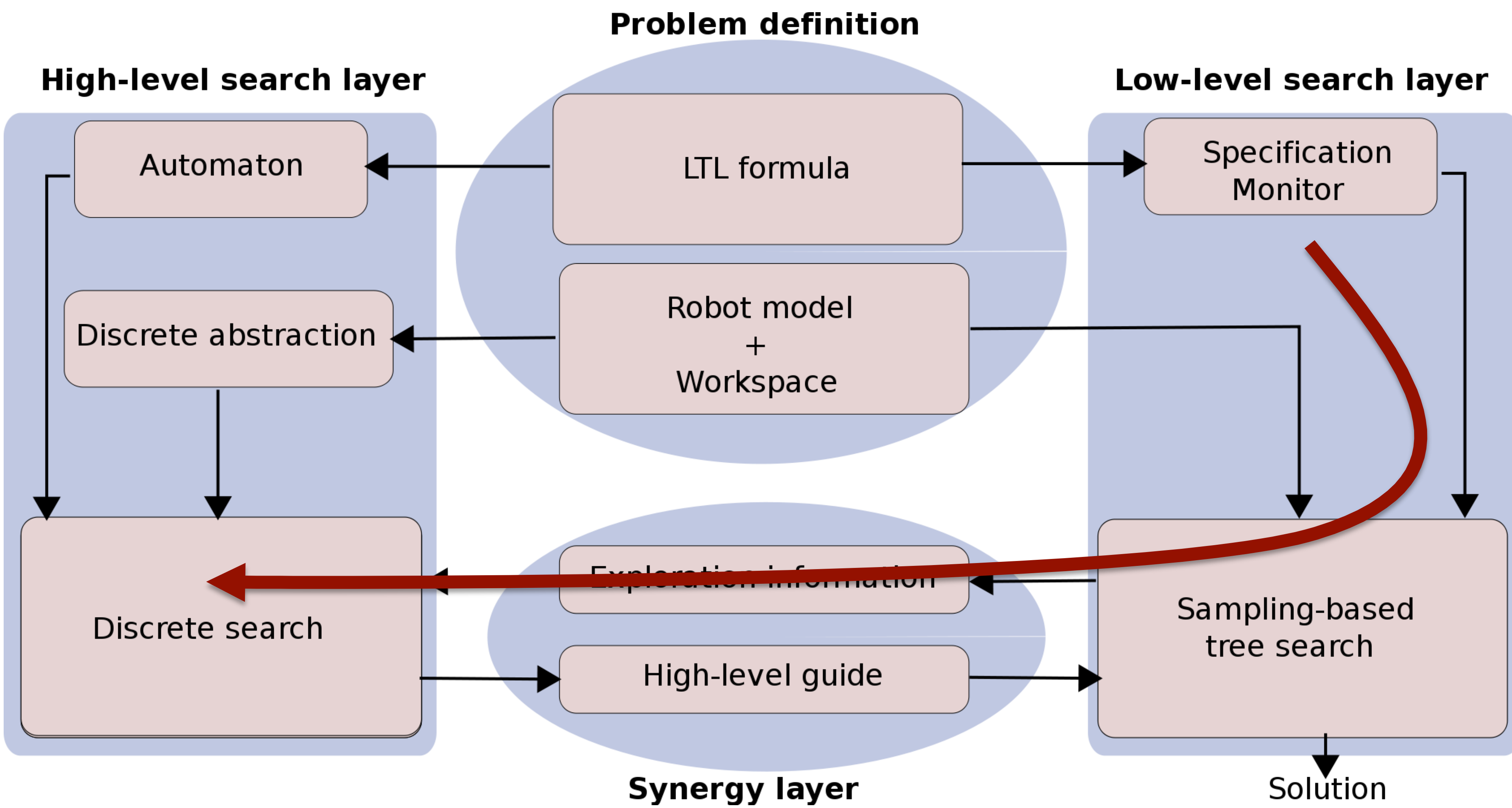


- **Synergy: A_ϕ.Z $\times$ D** represented as weighted graph
- Edge weights represent feasibility of transitions
- Feasibility captured through the notion of feasibility estimate ρ
- $$\rho(z,d) = \frac{\text{Region Coverage }(d) \times \text{Region volume }(d)}{\text{Automaton state }(z) \times \text{Past history }(z,d)}$$

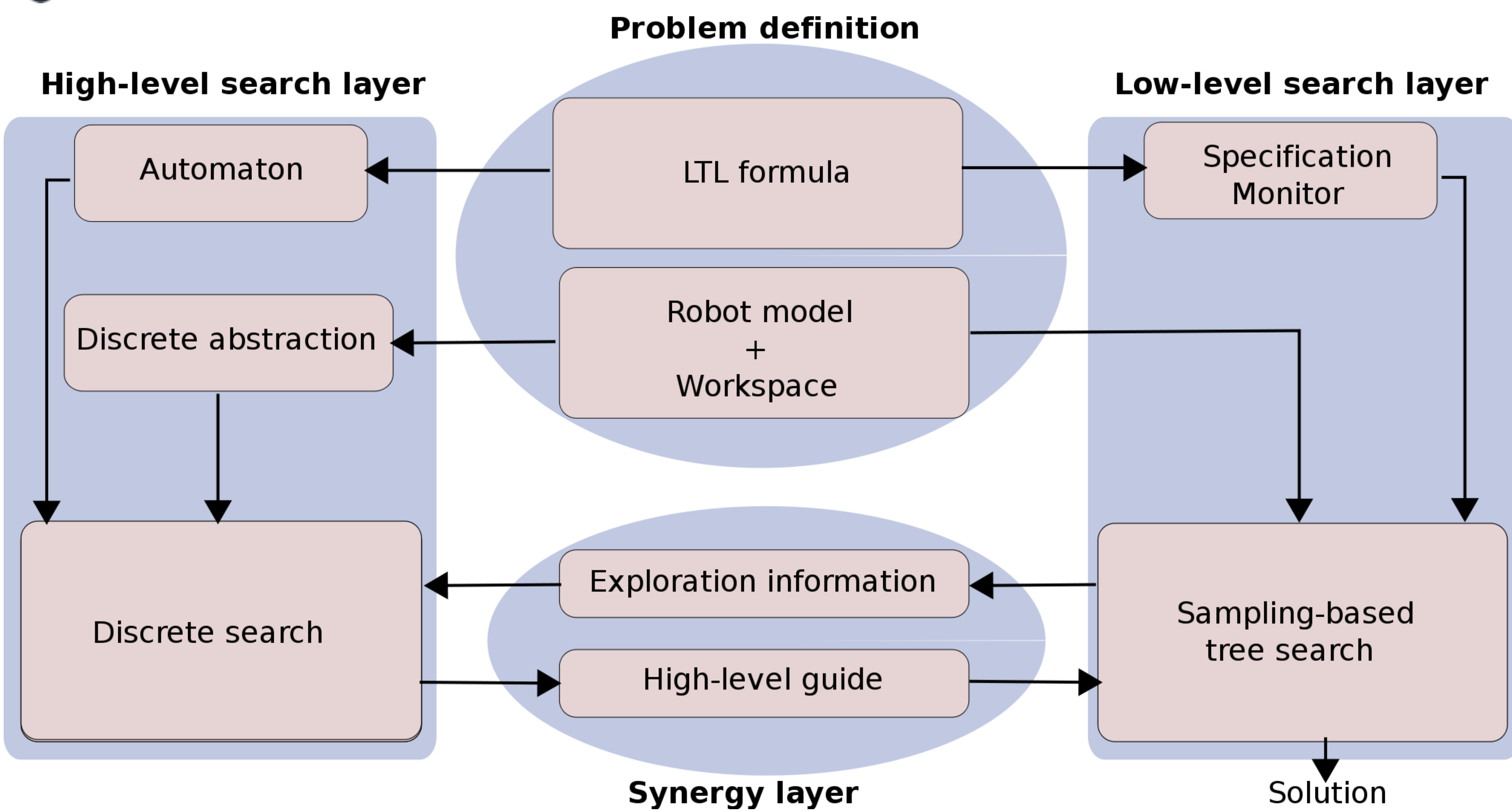
Our Solution



Our Solution



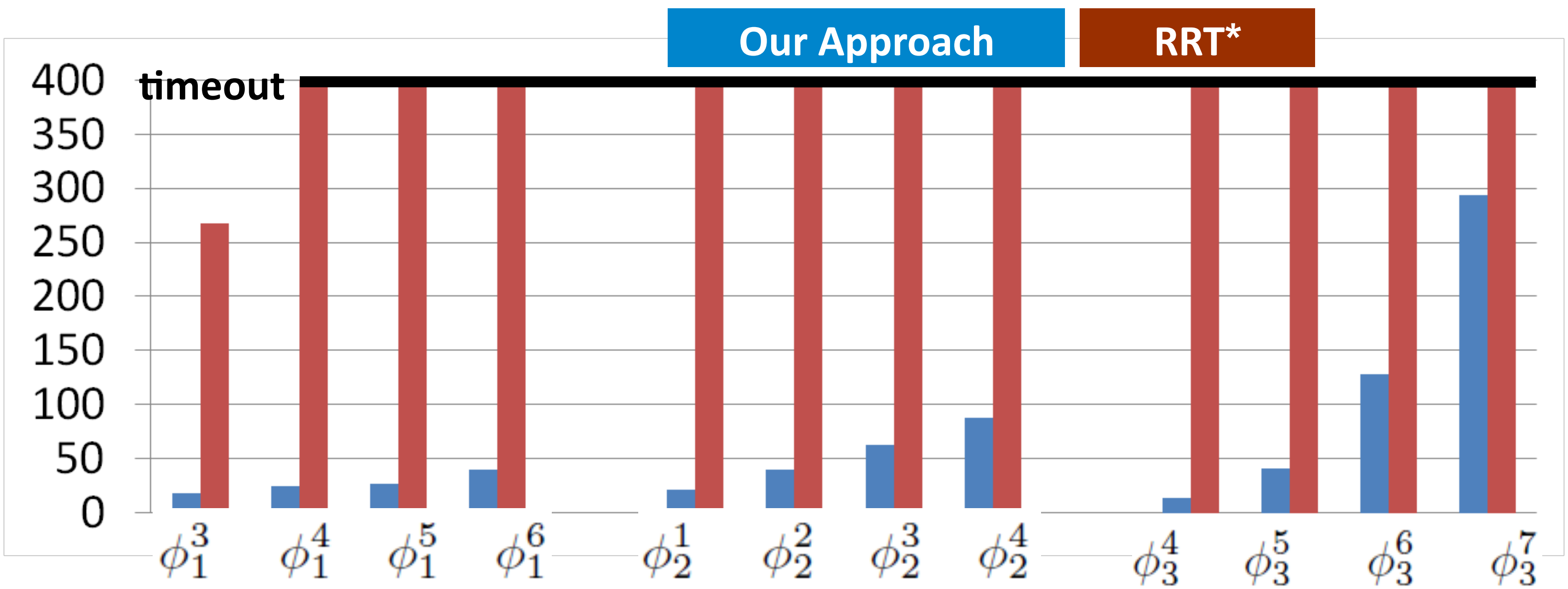
Initial Validation



TACAS2009 (abstraction is provided by the user)

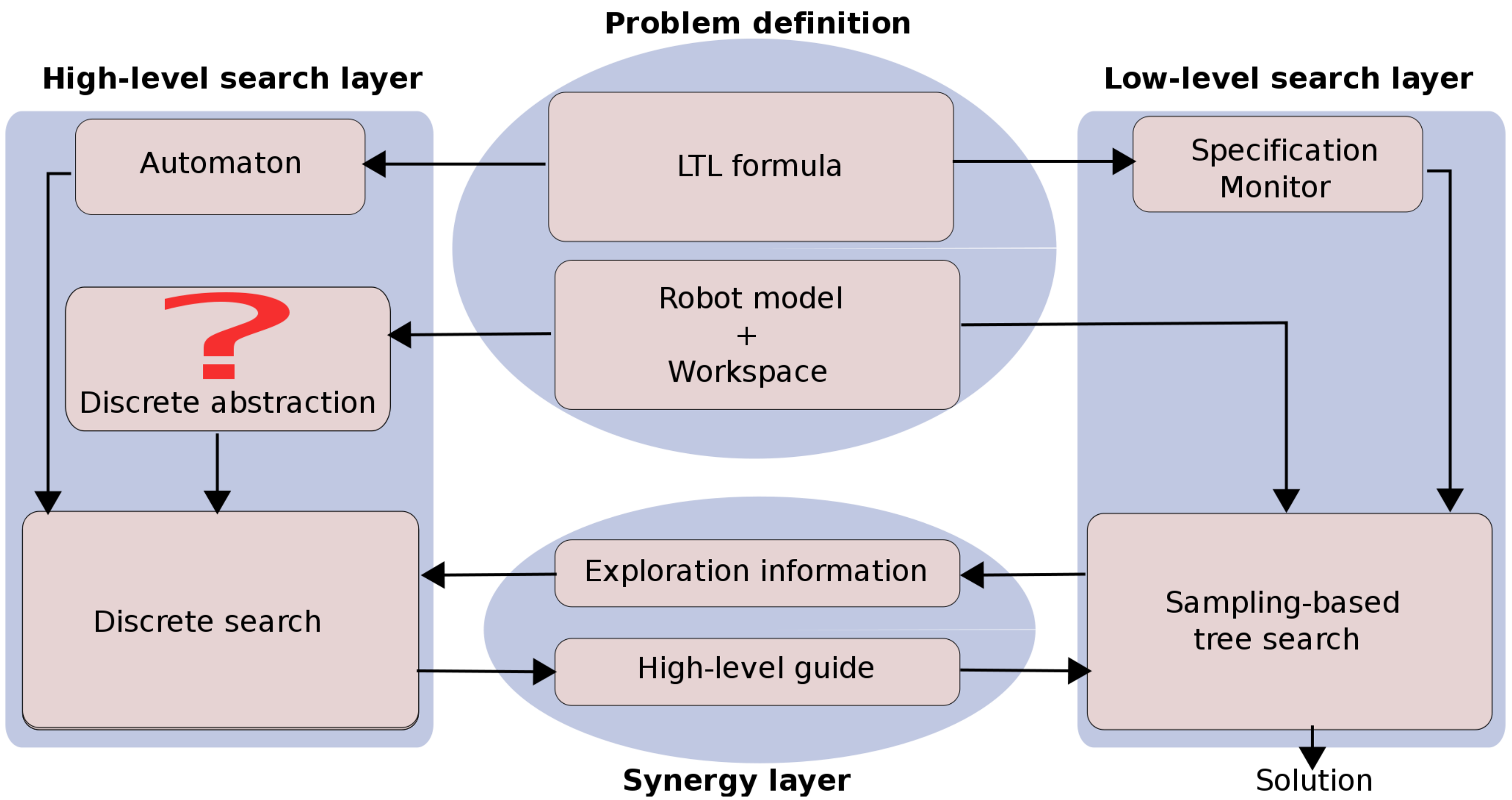
Augmented Tree-based Approach

- The approach lacks guidance
- Difficult to discover new promising search directions
- Experiments show the approach is impractical



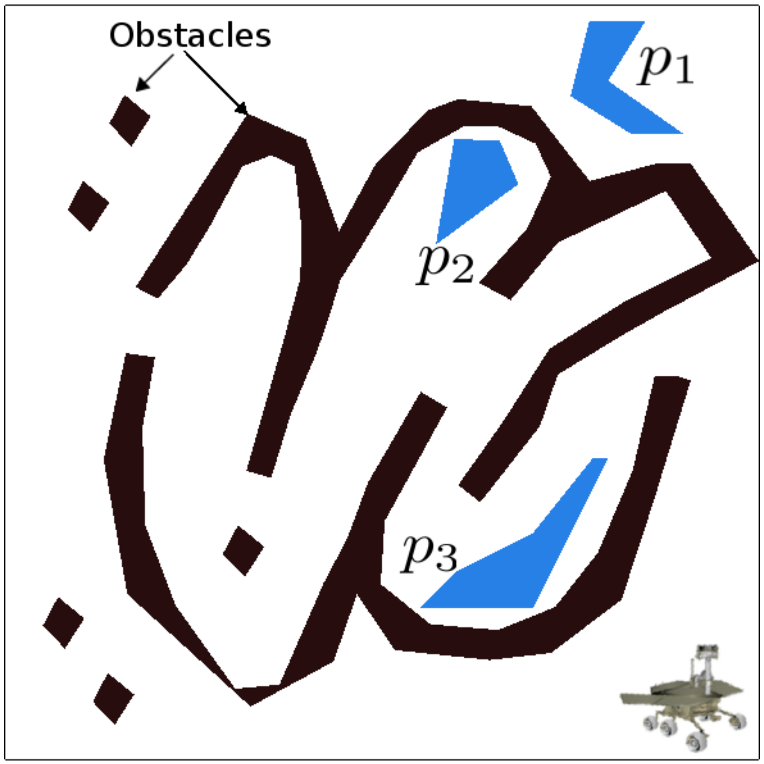
Reported is the average time [seconds] to solve 100 problem instances for each of the LTL safety formulas. Timeout set to 400s. LTL formulas translated to minimized DFA [TACAS 2009]

Ongoing Work

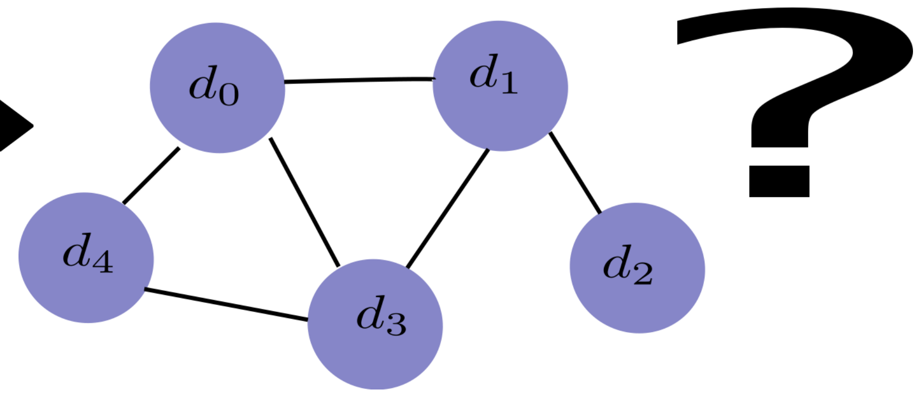


- TACAS'09 did not address the issue of abstraction construction
- **Ongoing work:** Automated construction of discrete abstraction

Construction of Discrete Abstraction



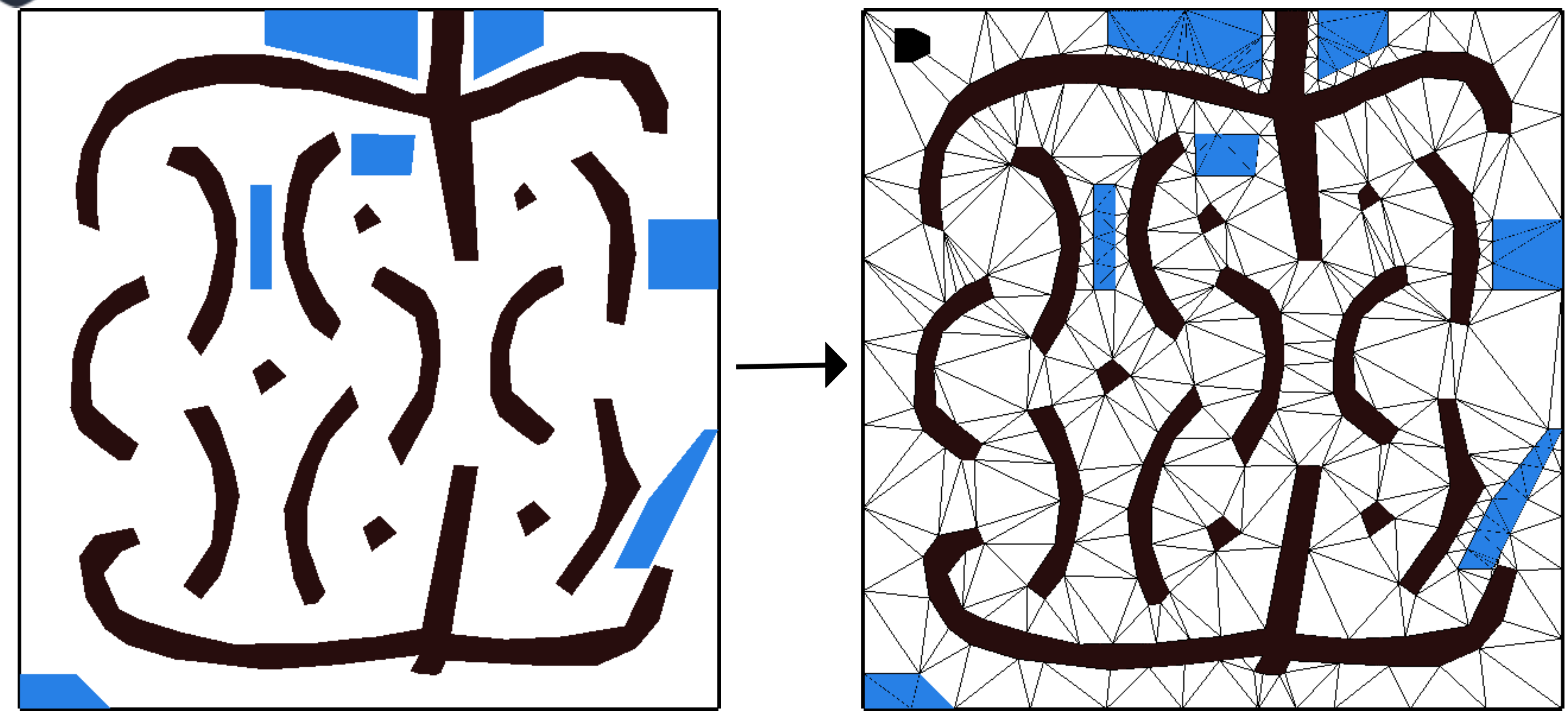
Robot model + Workspace



Discrete abstraction (M)

- **Design goal:** Address scalability issues
- **Challenges:**
  - Temporal logic constraints
  - Workspace constraints (obstacles, propositional sets)
- **Proposal:** Use geometry of specifications and workspace

Discrete Abstraction: Geometric Approach

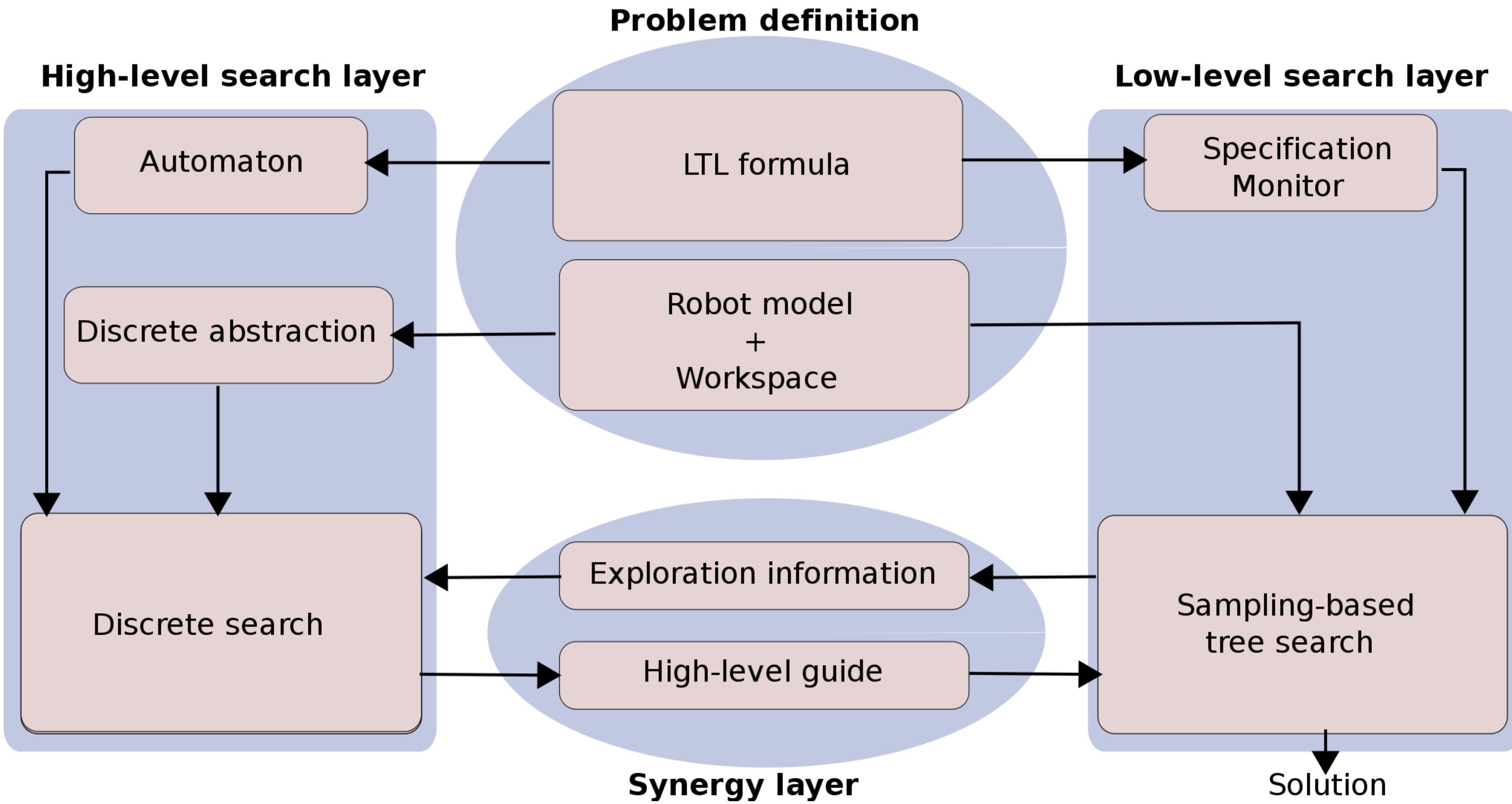


Workspace

Geometry-using (**618 states**)

[Bhatia, Kavraki, Vardi, ICRA 2010]

Our Solution



Acknowledgments

<u>Kavraki's Group:</u>

Amit Bhatia
Drew Bryant
Devin Grady
Allison Heath
Mark Moll
Ioan Sucan

Undergraduate students:
N. Feltman, N. Briddle, C. Pen

<u>Credits to past members of the group:</u>
PDST: A. Ladd, K. Bekris, K. Tsianos
Hybrid Systems, SyCLoP: E. Plaku
Str. Bio: A. Shehu, B. Chen, N. Haspel

<u>Collaborators:</u>

Dr. George Bennett (Rice)
Dr. Wah Chiu (Baylor)
Dr. Cecilia Clementi (Chemistry, Rice)
Dr. Marek Kimmel (Statistics, Rice)
Dr. John Lambris (UPenn)
Dr. Dan Sorensen (CAAM, Rice)
Dr. Mark Yim (UPenn)
Dr. Joe Warren (Computer Science, Rice)
Dr. Moshe Vardi (Computer Science, Rice)

<u>Funding:</u>

NSF, NIH, ARL (MURI),
Sloan and Hamil Foundations

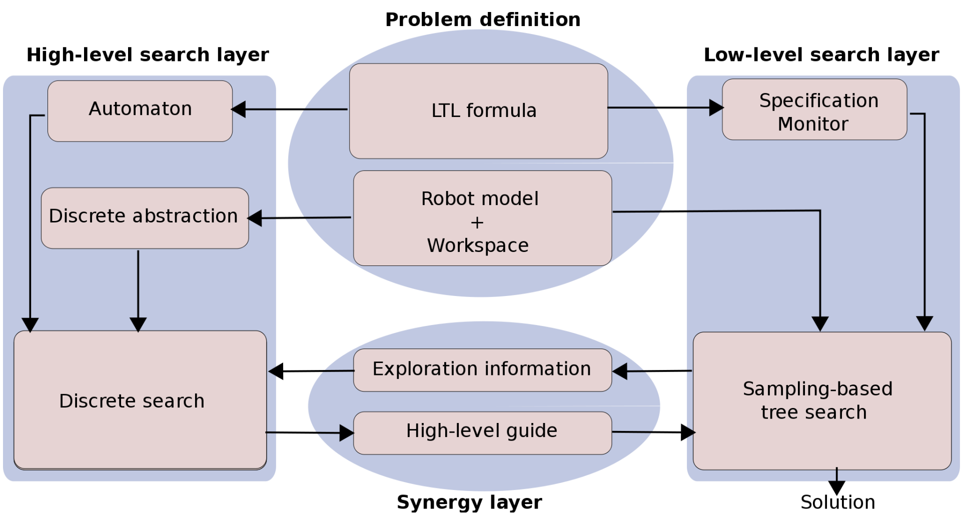
IEEE RAS Distinguished Lecturer Program

For More Information:

http://www.cs.rice.edu/~kavraki

http://www.kavrakilab.org

THANK YOU



Problem definition
High-level search layer
Low-level search layer
Automaton
LTL formula
Specification Monitor
Discrete abstraction
Robot model
+
Workspace
Discrete search
Exploration information
High-level guide
Sampling-based tree search
Solution
Synergy layer