Formal Modeling and Analysis of Hierarchical Path Planning

Cesare Tinelli
The University of Iowa

Acknowledgements

Collaborators: Paul Meng (Iowa)

Alessandro Pinto (UTRC)

Funding: United Technology Research Center

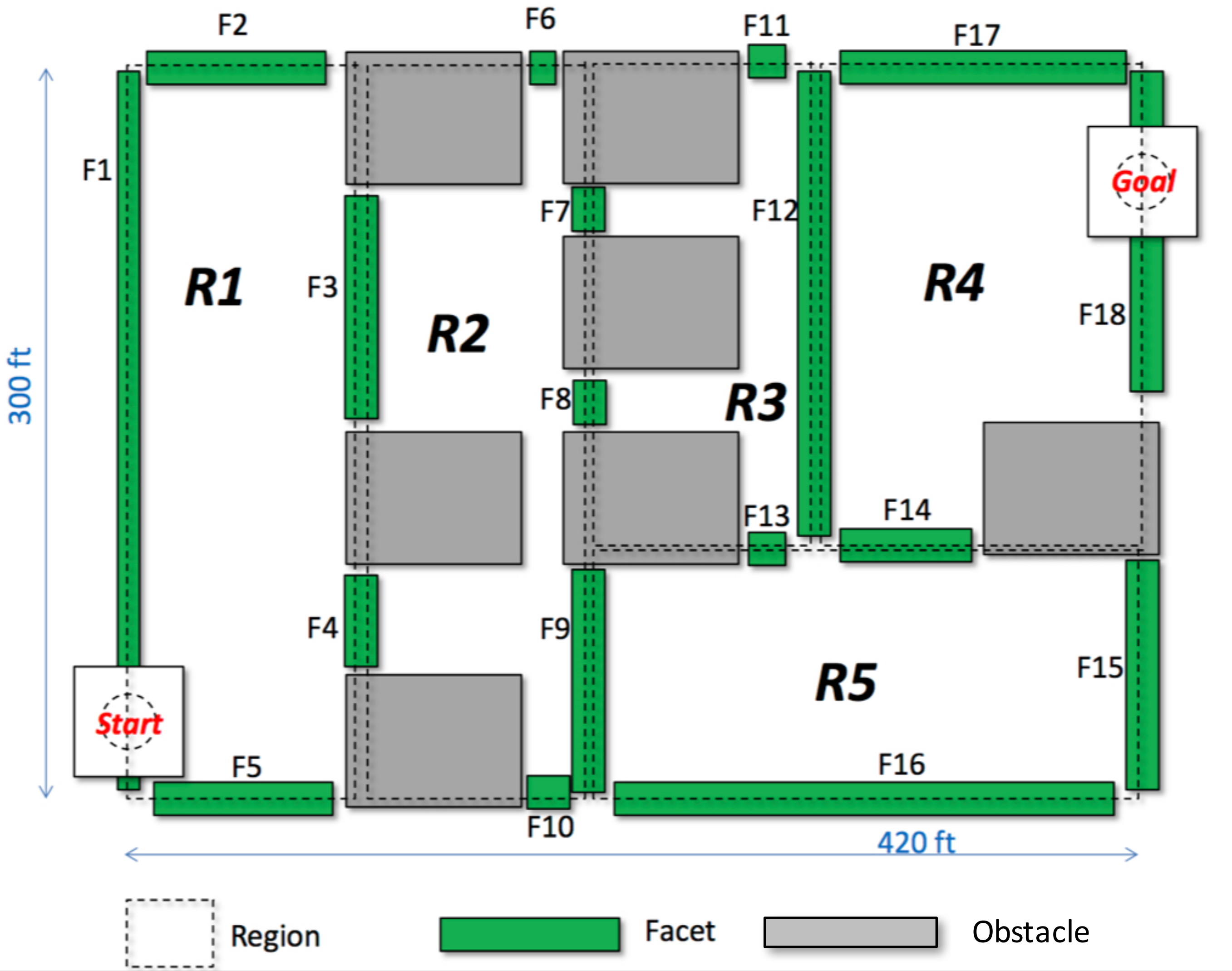
Context

Apply formal methods to model, analyze and verify software components of autonomous ground vehicles

This talk: experiences in the automatic verification of static properties of motion planning systems

2D Motion Planning

Regions
Facets
Obstacles
Locations
Start loc.
End loc.



Hierarchical Planning

Three planners:

- High Level Planner: generates moves between facets
- Path Planner: refines high level moves into sequences of way-points
- Trajectory Planner: refines the path sequences into a finer resolution to account for dynamic constraints

Planning proceeds from higher to lower level

High Level Planner

- **Goal:** generate list of move steps from facet to facet
- **Origin:** A facet f containing the start location
- **Destination:** A facet f' containing the goal location
- **Plan:** a sequence of facets starting with f and ending with f' where every two consecutive facets share a region

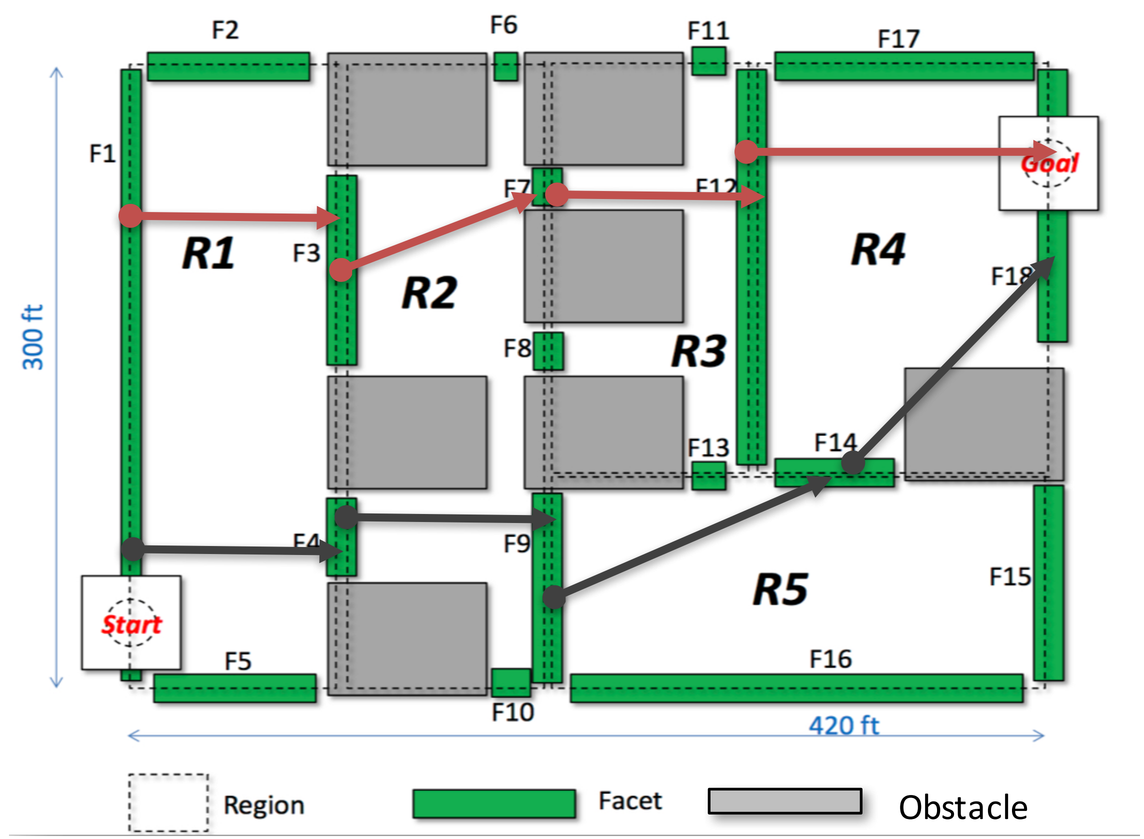
Path Planner

- **Goal:** refine high level commands into sequences of way-points (locations)
- **Origin:** A location p
- **Destination:** A location p'
- **Path Plan:** a sequence of locations starting with p and ending with p' where
  - every two consecutive locations are visible
  - p is on the start facet f of a segment or in a region containing f
  - p' is on the end facet f' of a segment or in a region containing f'

Trajectory Planner

- **Goal:** refines a path plan sequence into a finer resolution of locations
- **Origin:** A location p
- **Destination:** A location p’
- **Trajectory Plan:** a sequence of locations such that every two consecutive locations are “close enough”

High Level Planning



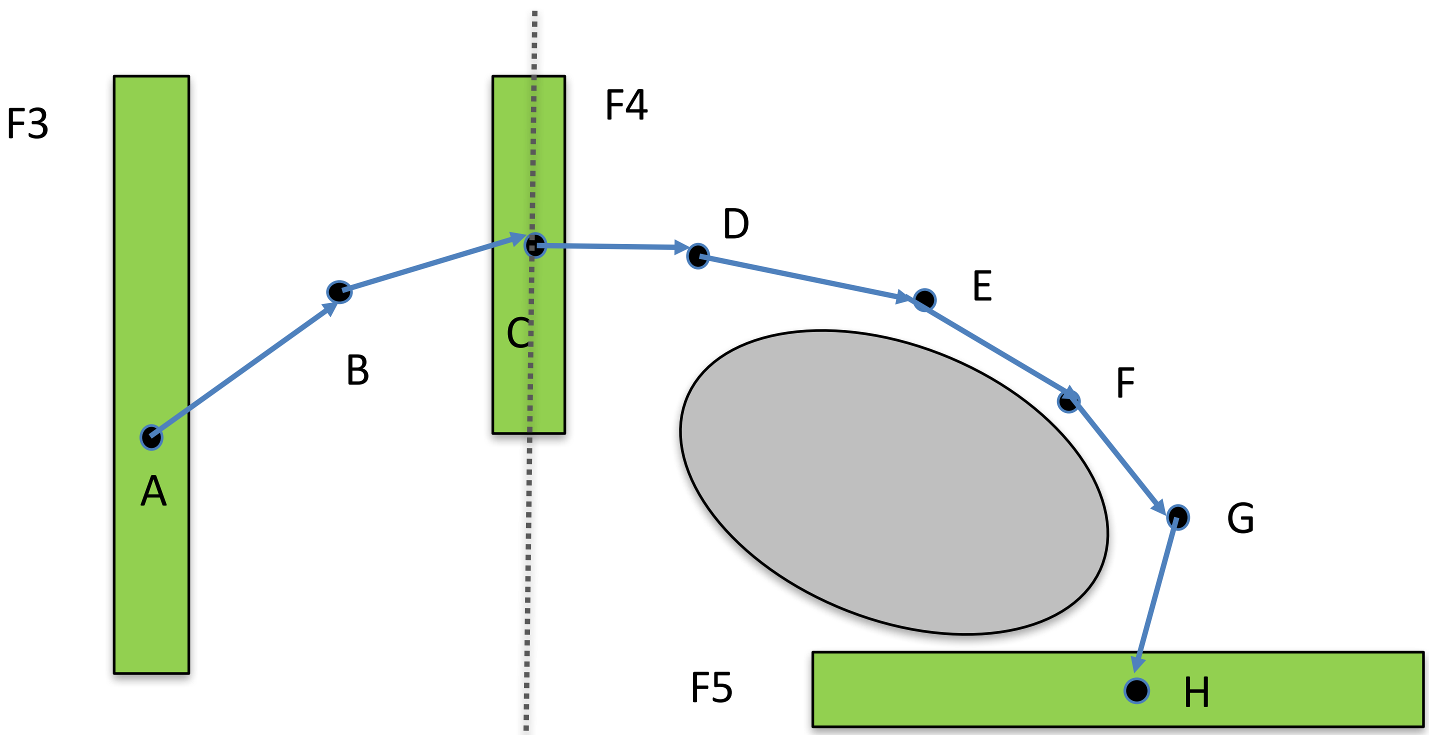
Plans: (F1, F4, F9, F14, F18)

(F1, F3, F7, F12, F18)

...

Path Planning

High-level Plan: (..., F3, F4, F5, ...)

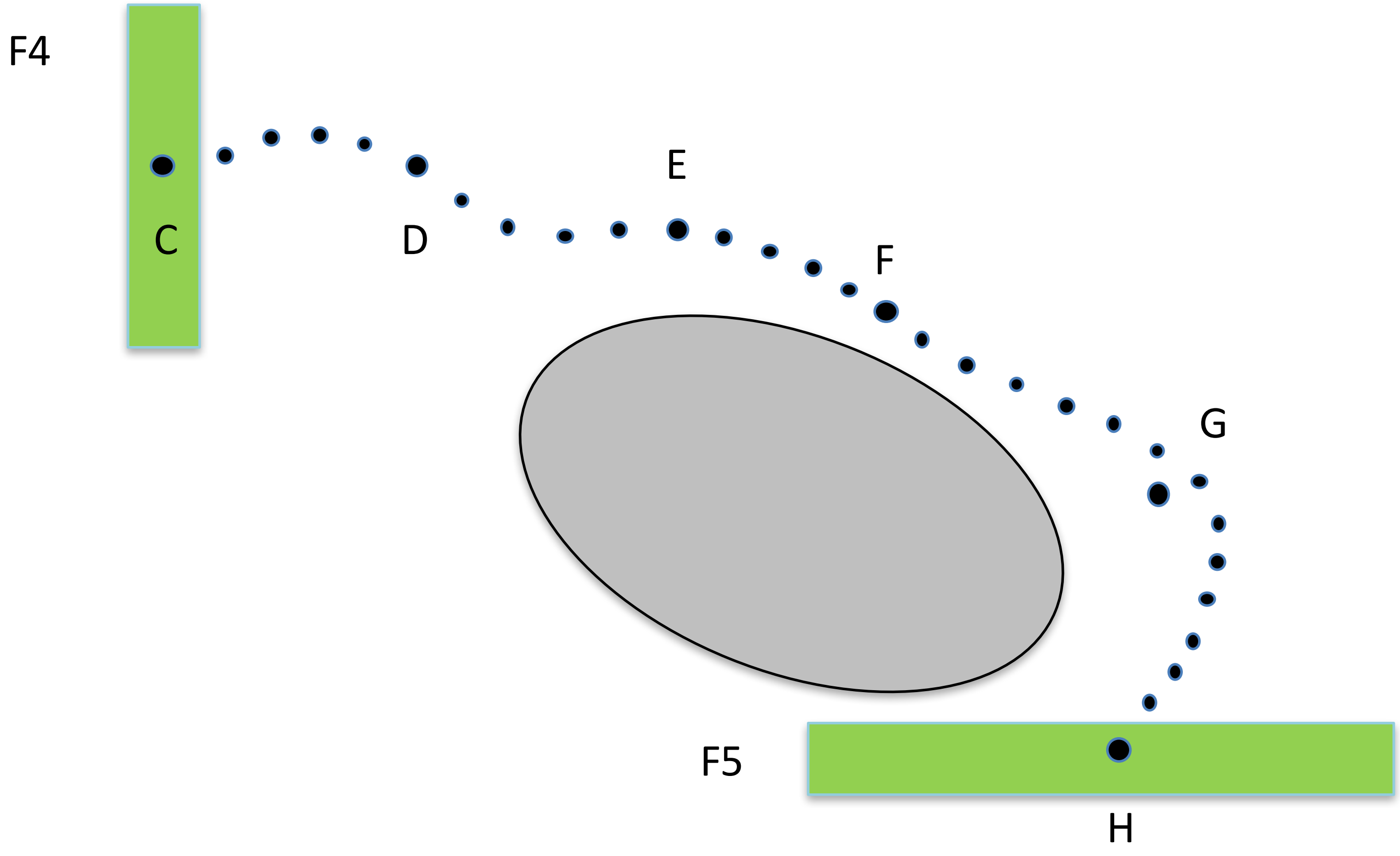


F3-F4 Plan: (A, B, C)

F4-F5 Plan: (C, D, E, F, G, H)

Trajectory Planning

F4-F5 Plan: (C, D, E, F, G, H)



Properties of Interest

Property 1: For every high level plan P_H , there is a path plan P_P for P_H

Property 2: For every high level plan P_H and path plan P_P for P_H, there is a trajectory plan P_T that refines P_P

Modeling Hierarchical Planning

Entities

- regions, facets, location, robot, …

Relationships

- contained: Facet x Region
- occupied: Loc
- visible: Loc x Loc
- close: Loc x Lox
- …

Modeling Hierarchical Planning

Constraints

- Start/end facet contains start/end location
- Individual path plans connected by shared positions in shared facet
- path plan in visible$^+$
- trajectory plan in close$^+$
- trajectory plan follows path plan closely

Modeling Languages

- Alloy
- SMT-LIB 2

Alloy

- Rich modeling language developed at MIT
- Based on first-order relational logic
- Can model any domain of individuals and relations between them
- Fully automated analysis of Alloy models by Alloy Analyzer with respect to a bounded scope for each data domain
- Analyzer has some built-in types (integers)

SMT-LIB 2 Language

- Standard input/output language for SMT solvers
- Based on many-sorted first-order logic
- Refers to a rich set of predefined theories
- Includes a command language for interacting with SMT solvers via a textual interface
- Level of support for language and theories depends on solver
- Major solvers: CVC4, MathSat, Yices, Z3, …

CVC4

- Jointly developed at Iowa and NYU
- Many built-in theories
- Decides several quantifier-free fragments
- Supports quantifier reasoning but with incomplete methods
- Can do finite model finding over uninterpreted sorts or bounded quantifiers

Alloy Analyzer: Property 1

Property 1: For every high level plan P_H , there is a path plan P_P for P_H

✗ Invalid

Counterexample: Scenario with a region fully split by an obstacle (e.g., a wall)

Alloy Analyzer: Property 2

Property 2: For every high level plan P_H and path plan P_P for P_H, there is a trajectory plan P_T that refines P_P

- Unable to prove or disprove because it requires reasoning about arithmetic constraints
- Alloy Analyzer offers limited support for numerical constraints

Limitations of Modeling in Alloy

- Translates constructs to propositional logic and uses a SAT solver
- thus reasoning about properties requires a cardinality bound on each type
- It cannot prove the validity of a property because it only exhaustively searches for models within a bounded scope for each type
- Its ability to reason about arithmetic constraints is very limited

CVC4: Property 1

Property 1: For every high level plan P_H , there is a path plan P_P for P_H

✗ Invalid

Counterexample: As in Alloy

Scalability Issues

- After adding a grid of locations (but no constraints on visibility or neighbors)
- If we allow robot to move freely (all locations are free and reachable), Property 1 trivially holds
- But we can only prove it only for grids up to 7x7, where each location is explicitly specified
- For bigger grids, CVC4 does not terminate within a reasonable timeout

Scalability Issues

- A big problem: transitive closure
  - Encoded by an approximate first-order axiomatization
- General problem: quantified formulas in model
  - default method is incomplete
  - finite-model-finding in CVC4 relies on exhaustive ground quantifier instantiation

CVC4: Property 2

Property 2: For every high level plan P_H and path plan P_P for P_H, there is a trajectory plan P_T that refines P_P

✗ Invalid

Counterexample: Scenario with visible but inaccessible locations (e.g., because separated by a river)

CVC4: Property 2 Redux

Property 2:

Assume that if any location l_2 is visible from a location l_1 then it is reachable from l_1

For every high level plan P_H and path plan P_P for P_H, there is a trajectory plan P_T that refines P_P

✓Valid

Lesson Learned: Alloy

- Very expressive language facilitates modeling
- SAT-engine very effective at finding small models
- Alloy great for model debugging
- Lack of support for built-in types limits ability to model realistic systems
- For invalid properties with large counter-examples scalability is an issue

Lesson Learned: CVC4

- It is possible to prove interesting properties of medium sized (~1000 lines) models of complex systems
- A relational language would facilitate specification
- Better support for transitive closure is crucial
- For invalid properties with large counter-examples scalability is an issue

Resolve

- Add some of the expressiveness of Alloy to CVC4 by building in a theory of finite relations
- Build-in efficient methods to reason about transitive closure
- Continue working on improving support for quantifiers
- Devise new symmetry breaking techniques to improve scalability of finite model finding