2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Automatic Synthesis of Robot Controllers for Tasks with Locative Prepositions *

Hadas Kress-Gazit[1] and George J. Pappas[2]

[1]Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA
[2]GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104, USA
hadaskg@cornell.edu, pappasg@grasp.upenn.edu

***Abstract*—This paper describes the synthesis of correct robot control from high-level tasks that include non-projective locative prepositions. Here, locative prepositions such as 'near' and 'between' are used to refer to regions in the robot's workspace and are part of a high-level task description such as "Always stay near room 1" or "Visit the area between room 2 and room 3". These prepositions induce a discrete abstraction of the workspace which, together with the rest of the task, is used to synthesize a correct-by-construction robot controller such that the robot is guaranteed to behave as expected, if the task is feasible. This work presents an important step towards allowing linguistic control of robots that is both intuitive and provably correct.**

***Index Terms*—Mission Planning, Language, Motion Planning, Temporal Logic, Controller Synthesis, Hybrid Control.**

I. Introduction

For robots to be truly ubiquitous, they must be easy and intuitive to control while at the same time they must be safe and dependable. To create such systems it is crucial to address topics ranging from Human-Robot interaction and high-level planning to low-level control and dynamics while providing methods and tools for either verification or correct-by-design synthesis of the system.

An exciting approach to intuitive Human-Robot interaction is the use of natural language to control and interact with a robot or team of robots. Work in this direction ranges from robots understanding the implicit (or indirect) meaning of a request [1] to tightly integrating incremental parsing of language with goal and action processing [2]; from methods for enriching the semantics of a spoken utterance, represented using visual schemas, with cognitive and pragmatic information [3] to extracting semantic representations and executable robot procedures from spoken dialog [4]. In [5] the authors discuss different aspects of controlling robots (non autonomous) using natural language.

Recent work has explored high-level representations of tasks, typically given in a Temporal Logic [6] representation, that are automatically converted into provably-correct robot controllers [7]–[10]. These controllers relay on the notions of abstraction and bisimulation [11], [12] thus providing guarantees for correct and safe robot behavior.

Connections between the formalism of Temporal Logic and language have been explored in [13] where natural language utterances were translated into temporal logic representations that were then used to reason about current and future actions and procedures. In [14] Structured English was translated into a subset of Linear Temporal Logic that was then automatically converted into a hybrid controller using synthesis techniques.

Building on the work in [14], [15], this paper extends the Structured English interface by adding non-projective locative prepositions to the grammar while preserving the guarantees of the automatically generated controller. The locative prepositions 'within', 'near', 'between', 'inside' and 'outside' are automatically resolved thus allowing a user to specify a complex and reactive task at a high-level (Structured language), press a button and observe the robot performing the task, if the task is feasible. While locative prepositions have been explored in the context of robotics (e.g. [5], [16] for both projective and non-projective prepositions), here the process of generating the low-level robot velocity commands from a task with locative prepositions is fully automated and is guaranteed to be correct. Furthermore, the prepositions are used in the context of specifying a continuous and complex mission rather than interpreting a scene or grounding a dialog with a human.

This paper is structured as follows: Section II discusses the formal semantics of locative prepositions, Section III describes the Structured English grammar that is used to capture high-level robotic tasks. Section IV gives an overview of the automatic process in which the high-level task is transformed into low-level robot control and then describes the algorithms to resolve the locative prepositions. The paper concludes with examples (Section V), conclusions and future directions (Section VI).

II. Semantics of locative prepositions

Locative prepositions can be roughly divided into non-projective and projective prepositions [17], [18]. The semantics, or meaning, of non-projective locative prepositions such as 'between' and 'near' does not depend on a point of view; even if the speaker and the person spoken to are facing each other or are in different parts of the environment, both will agree on the region of space that is being discussed. In contrast, projective prepositions such as 'to the left of' and 'in front of' highly depend on the point of view of the speaker or the person spoken to.

*This work is supported by ARO MURI (SUBTLE) W911NF-07-1-0216.

This paper provides a framework to automatically generate a correct-by-construction robot controller that will ensure a robot satisfies a task that contains non-projective locative prepositions, specifically 'between', 'near', 'within', 'inside' and 'outside'. Here, the locative prepositions refer to regions of interest in the robot's workspace that are defined with respect to other regions. Other prepositions such as 'above' or 'below' have a non-projective meaning when dealing with motion in three dimensions; in two dimensions however, their linguistic meaning is less clear and therefore this paper does not discuss them.

Inspired by [17], the formal semantics of a locative prepositions is defined as a function mapping from one set of points in the workspace to another. In the context of this work, namely robot mission and motion planning, these sets of points correspond to regions in the two-dimensional workspace of a mobile robot.

A. *'within','near'*

The expression 'within d of A', where $d \in \mathbb{R}^+$ is a distance and $A \subseteq \mathbb{R}^2$ is a region, describes a region $Q \subseteq \mathbb{R}^2$ such that

$$Q = \{q \in \mathbb{R}^2 | q \notin A \text{ and } \exists a \in A, \|q - a\| \leq d\}$$

Intuitively, for a convex region A, Q represents an annulus of radius d around A, as shown in Figure 1(a).

The semantics of 'near' are the same as 'within' without an explicitly given distance. Linguistically speaking, there is evidence that the distance implied when using the 'near' proposition depends on different factors such as the size of the reference object or the presence of other objects in the scene (e.g. [5], [19], [20]). For simplicity here as in other work (e.g. [3], [17]) this distance is assumed to be a fixed number.

B. *'inside','outside'*

Given a region $A \subseteq \mathbb{R}^2$ the semantics of the expression 'inside A' is the set of points that belong to the region

$$Q = \{q \in \mathbb{R}^2 | q \in A\}$$

and the semantics of 'outside A' is the set of points that do not belong to A

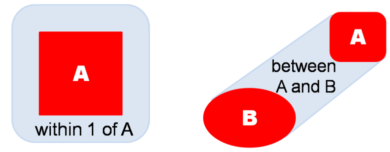
$$Q = \{q \in \mathbb{R}^2 | q \notin A\}$$

C. *'between'*

Unlike the previous prepositions, the locative preposition 'between' is used in conjunction with at least two objects (regions). While in general it can be used with more than two objects, the plurality in such sentences creates interesting linguistic phenomena as discussed in [17] and references therein, therefore here two regions are considered.

The semantics of the expression 'between A and B' is the set of points that belong to the convex hull of the regions but not to the regions themselves as illustrated in Figure 1(b).

$$Q = \{q \in \mathbb{R}^2 | q \in ConvHull(A, B),\ q \notin A,\ q \notin B\}$$



(a) Semantics of 'within'

(b) Semantics of 'between'

Fig. 1: Semantics of locative prepositions

III. Task specification

This section describes the Structured English grammar used to specify tasks that contain non-projective locative prepositions; first the set of terminals of the grammar and then the rules of the grammar. The formal description of the class of Linear Temporal Logic (LTL) formulas that underlay the Structured English specification is omitted for lack of space and can be found in [7], [15], [21].

A. *Atomic propositions*

The atomic propositions, which act as the terminal symbols of the grammar, together with a set of reserved words make up the grammar used to describe high-level tasks. In the following, their syntax and semantics are defined.

1) syntax: The set of atomic propositions contains three types of propositions: (i) Binary sensor inputs $x \in \mathcal{X}$ (ii) Binary robot actions $a \in \mathcal{A}$ and (iii) Regions in the workspace $r \in \mathcal{R}$

The set $\mathcal{Y} = \mathcal{A} \cap \mathcal{R}$ corresponds to all robot controlled propositions (motion and action).

2) semantics: The set $\mathcal{X}$ corresponds to an abstraction of the low-level sensing capabilities of the robot. For example, a vision system abstraction can be a sensor proposition $seePerson$ or a light detector that is abstracted into a proposition $dark$. The truth value of a sensor proposition $x_i \in \mathcal{X}$ reflects the state of the environment as perceived by the robot's sensors:

$$x_i = \begin{cases} \textit{True} & \text{if the environmental state detected by} \\ & \text{sensor } i \text{ is occurring} \\ \textit{False} & \text{otherwise} \end{cases}$$

The set $\mathcal{A}$ corresponds to the robot's available actions such as turning on a camera of flashing a light. These actions are assumed to be binary (ON/OFF) or have values over a finite domain in which case it can be captured by a binary vector (Low/Medium/High). Furthermore, it is assumed that these actions do not have any timing restrictions The truth value of an action proposition $a_i \in \mathcal{A}$ reflects the state of the action:

$$a_i = \begin{cases} \textit{True} & \text{if action } i \text{ is being executed} \\ \textit{False} & \text{if action } i \text{ is not being executed} \end{cases}$$

The position of the robot is denoted as $p \in \mathbb{R}^2$. Propositions belonging to the set $\mathcal{R}$ correspond to whether the robot is in a region of interest in the workspace, where the regions are defined by the user based on the desired task. Unlike previous work [7], [15], [21], these regions do not have to form a partition of the workspace.

Every region of interest P_i in the workspace P is a convex polygon such that

$$P_i = \{p \in \mathbb{R}^2 \mid H_i p \leq K_i\}$$

with $H_i \in \mathbb{R}^{Edges \times 2}, K_i \in \mathbb{R}^{Edges}$. The truth value of a region proposition $r_i \in \mathcal{R}$ corresponds to whether the robot is in region P_i:

$$r_i = \begin{cases} True & \text{if } p \in P_i \\ False & \text{if } p \notin P_i \end{cases} .$$

B. Grammar

The Structured English grammar considered in this paper is given in Table I. This grammar adds the ability to express robotic tasks that include non-projective locative prepositions to the grammar introduced in [14].

The grammar in Table I contains different types of logical formulas and sentences. There, $x \in \mathcal{X}, y \in \mathcal{Y}, a \in \mathcal{A}$ and $r \in \mathcal{R}$ are the atomic propositions of the task as described in Section III-A and $distance \in \mathbb{R}^+$ is a positive number. Intuitively, ϕ captures a logical connection between propositions belonging to both the environment and the robot while ϕ_{env} restricts to propositions relating to the environment, ϕ_{robot} restricts to propositions relating to the robot, ϕ_{reg} to to the robot's region propositions and the locative prepositions and ϕ_{action} to the robot's action propositions.

In each of the sentences, exactly one of the terms written inside of parentheses is required while terms written inside square brackets are optional. Past and present tenses in *Condition* are treated differently only when combined with *EnvSafety* or *RobotSafety* or *Stay*[1].

The user specification may include any combination of sentences and there is no minimal set of instructions that must be written. If some sentences, such as initial conditions, are omitted, their corresponding formulas are replaced by default maximally-permissive values.

Note that this grammar allows for nested locative prepositions such as "within 5 of between A and D"; It does not allow logical connectives over the regions in the locative preposition such as "within 5 of (A or D)". Such requirements can be addressed in ϕ_{reg} (e.g. "within 5 of A or within 5 of D").

IV. Synthesis of controller from Structured English specifications

Once the atomic propositions are defined and the Structured English specifications are written, the rest of the procedure for generating robot control that achieves the task is fully automated, as seen in Algorithm 1. The inputs to the procedure are the sets of atomic propositions $AP = \mathcal{X} \cup \mathcal{A} \cup \mathcal{R}$, the written specifications $Spec$, the regions of interest $\{P_i\}$ corresponding to the set $\mathcal{R}$ and the boundaries of the robot's workspace WS.

This section describes the function $ParseLocPrep$; the rest of the algorithm which include translating the Structure English into a Linear Temporal Logic (LTL) formula, synthesizing an automaton satisfying the formula, and creating the hybrid controller that drives the robot is discussed in detail in [7], [15], [21].

[1]For information regarding the tenses the reader is referred to [14].

Algorithm 1 Automatic synthesis of robot controller

procedure CRTCNTRL($\mathcal{X}, \mathcal{A}, \mathcal{R}, Spec, \{P_i\}, WS$)
  $Spec \leftarrow ParseLocPrep(\mathcal{R}, Regions, Spec)$
  $LTLFormula \leftarrow SEtoLTL(\mathcal{X}, \mathcal{A}, \mathcal{R}, Spec)$
  $Automata \leftarrow Synthesize(LTLFormula)$
  $Controller \leftarrow HybridCntrl(Automata, Regions)$
  return $Controller$
end procedure

A. Automatic resolution of locative prepositions

Initially, to create the controller, the workspace is partitioned into a set of convex polytopes, some of which correspond to the user-defined regions of interest and the rest to a convex partition of the remaining area in the workspace. To resolve the locative prepositions, first the partition of the workspace is recalculated such that each region corresponding to a locative preposition is a union of convex polytopes. Then, the specification is automatically rewritten to replace every region proposition r_i with an expression that is a disjunction of the propositions relating to the regions that the region P_i has been partitioned into. For example, in Figure 3(b) the creation of 'within 5 of B' required the partition of region A into two regions A_1 and A_2. Therefore, every occurrence of A in the Structured English specification must be replaced with 'A_1 or A_2'.

1) 'within' and 'near': The region Q corresponding to the locative prepositions 'within $distance$ of r_i' or 'near r_i' is

$$Q = \{q \in \mathbb{R}^2 | \exists p \in P_i, \|p - q\| \leq distance\}$$

where $distance$ is either given in the specification ('within $distance$ of') or defined before hand (the meaning of 'near', as discussed in Section II).

While the locative prepositions can be used to modify sets of regions $\{P_i\}_{i \in I}$, the discussion here is focused on a single convex polytope P_i. When modifying a set of regions, each region is treated separately and the result is the union of the respective regions.

Recall that a region P_i is defined by a set of half spaces:

$$P_i = \{p | H_i p \leq K_i\}$$

To create the modified region, the edges of P_i are first expanded by the desired amount:

$$Q = \{p | H_i p \leq K_i + distance\}$$

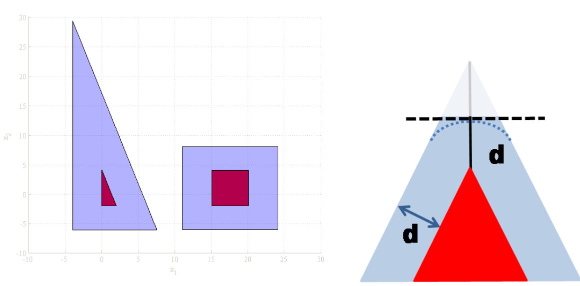
This results in a region that along the edges of the original polytope contains points that are at most $distance$ away; however, near the vertices of the original polytope, the points in the new region can be significantly further away, as illustrated in Figure 2(a). In the figure, the dark (red) polytopes are the original polytopes that are expanded by 4 resulting in the polytope in gray (light blue). While for

<table>
<tr><td>ϕ</td><td>::=</td><td>x | y | not ϕ | ϕ or ϕ | ϕ and ϕ | ϕ implies ϕ | ϕ iff ϕ</td></tr>
<tr><td>ϕ_{env}</td><td>::=</td><td>x | not ϕ_{env} | ϕ_{env} or ϕ_{env} | ϕ_{env} and ϕ_{env} | ϕ_{env} implies ϕ_{env} | ϕ_{env} iff ϕ_{env}</td></tr>
<tr><td>ϕ_{robot}</td><td>::=</td><td>y | not ϕ_{robot} | ϕ_{robot} or ϕ_{robot} | ϕ_{robot} and ϕ_{robot} | ϕ_{robot} implies ϕ_{robot} | ϕ_{robot} iff ϕ_{robot}</td></tr>
<tr><td>ϕ_{action}</td><td>::=</td><td>a | not ϕ_{action} | ϕ_{action} or ϕ_{action} | ϕ_{action} and ϕ_{action} | ϕ_{action} implies ϕ_{action} | ϕ_{action} iff ϕ_{action}</td></tr>
<tr><td>ϕ_{reg}</td><td>::=</td><td>r | LocativePrep | not ϕ_{reg} | ϕ_{reg} or ϕ_{reg} | ϕ_{reg} and ϕ_{reg} | ϕ_{reg} implies ϕ_{reg} | ϕ_{reg} iff ϕ_{reg}</td></tr>
<tr><td>LocativePrep</td><td>::=</td><td>“within distance of (r | LocativePrep)” | “near (r | LocativePrep)” | “inside (r | LocativePrep)
| “outside (r | LocativePrep) | “between (r | LocativePrep) and (r | LocativePrep)”</td></tr>
<tr><td>EnvInit</td><td>::=</td><td>“Environment starts with (ϕ_{env} | false | true) ”</td></tr>
<tr><td>EnvInit</td><td>::=</td><td>“Environment starts with (ϕ_{env} | false | true) ”</td></tr>
<tr><td>RobotInit</td><td>::=</td><td>“Robot starts [in ϕ_{reg}] [with ϕ_{action} | with false | with true] ”</td></tr>
<tr><td>EnvSafety</td><td>::=</td><td>“Always ϕ_{env}”</td></tr>
<tr><td>RobotSafety</td><td>::=</td><td>“(Always | Always do | Do) ϕ_{robot}”</td></tr>
<tr><td>EnvLiveness</td><td>::=</td><td>“Infinitely often ϕ_{env}”</td></tr>
<tr><td>RobotLiveness</td><td>::=</td><td>“(Go to | Visit | Infinitely often do) ϕ_{robot}”</td></tr>
<tr><td>RobotGoStay</td><td>::=</td><td>“Go to ϕ_{reg} and stay [there]”</td></tr>
<tr><td>Conditional</td><td>::=</td><td>“If Condition then Requirement ” | “Requirement unless Condition ” |
| “Requirement if and only if Condition ”</td></tr>
<tr><td>Condition</td><td>::=</td><td>“Condition and Condition” | “Condition or Condition” |
| “you (were | are) [not] [in] ϕ_{reg} ” |
| “you (sensed | did not sense | are [not] sensing) ϕ_{env} ” |
| “you (activated | did not activate | are [not] activating) ϕ_{action}</td></tr>
<tr><td>Requirement</td><td>::=</td><td>EnvSafety | RobotSafety | EnvLiveness | RobotLiveness | “stay [there]”</td></tr>
</table>

TABLE I: The basic grammar rules for the Structured English to Robot control framework

the square the maximal distance between the original and expanded polytopes is $\sqrt{2}distance$, the maximal distance for the triangle is much larger.

The hybrid controller requires all regions to be convex polytopes; therefore, the modified region is approximated by creating an additional hyperplane for each vertex that “chops off” the corners. In this work, the modified region is over approximated as seen in Figure 2(b). There, the arc represents the points that are d away from the corner of the original polytope and the black broken line represents the additional hyperplane added to create the approximation.



(a) Regions expanded by 4

(b) Over approximation of the region ‘within’

Fig. 2: Geometric resolution of ‘within’.

Each additional hyperplane added to Q is perpendicular to the vector connecting vertex j in the original polytope P_i to the corresponding vertex in Q and it intersects that vector at a point that is $distance$ away from vertex j. Future work will examine the use of over and under approximations based on the high-level specifications.

Finally, the original polytope is removed from Q to obtain the modified region:

$$Q = Q \backslash P_i$$

Figure 3(a) depict the regions (in light blue) that are within 5 of a square region (shown in yellow).

2) ‘inside’ and ‘outside’: The region Q corresponding to the locative prepositions ‘inside r_i’ is defined as region P_i itself. $Q = \{q \in P_i\}$ and the region Q corresponding to the locative prepositions ‘outside r_i’ represents all the regions except P_i and is defined as $Q = \{q \in \bigcup_{j \neq i} P_j\}$. The blue region in Figure 3(b) represents the region ‘outside B’ while the yellow region represents ‘inside B’.

These functions can be used to modify a region that is a union of convex regions but that can be non-convex and can contain holes (such as in Figure 3(d)). The semantics of locative prepositions over regions that are not topologically simple is a topic of research in psycholinguistics (e.g. [18]) since in certain situations, human psychology leads to different linguistics effects. For example, a point in the hole inside of a ring is seldom considered ‘outside’ the ring even though topologically it is. In this paper the notion of inside and outside relate to the region and its complement and such linguistic effects will be addressed in future work.

Note that writing ‘inside r_i’ in the specification is equivalent to writing ‘r_i’ only and writing ‘outside r_i’ is equivalent to writing ‘$\neg r_i$’; however, using the locative preposition results in a more natural way to refer to those regions, as shown in the examples in Section V.

3) ‘between’: As described in Section II the region Q corresponding to the locative preposition ‘between r_i and r_j’ is defined as

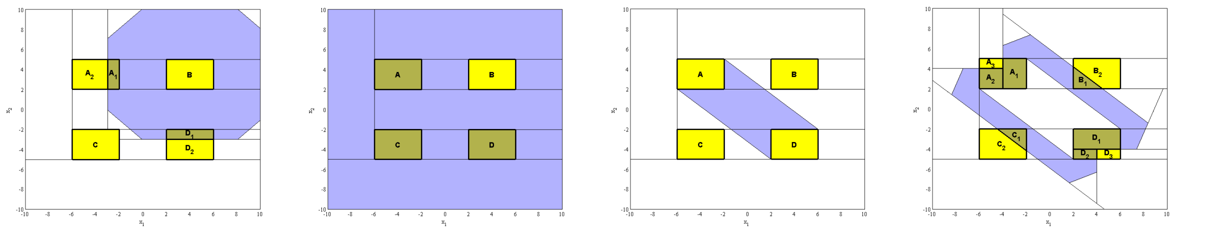
$$Q = \{q \in ConvexHull(P_i, P_j) \backslash P_i \backslash P_j\}$$

Figure 3(c) depicts the regions corresponding to ‘between A and D’.

As indicated by the grammar, the locative prepositions can be nested to create more complex regions, for example the regions depicted in Figure 3(d).

V. Examples

The following examples illustrates the process of automatically transforming a task given in Structured English



(a) 'Within 5 of B' (b) 'Outside B' (c) 'Between A and D' (d) 'Within 2 of between A and D'

Fig. 3: Examples for regions corresponding to locative prepositions. The regions in yellow (light gray) are the regions of interest and the regions in blue (gray) correspond to the semantics of the locative preposition.

and containing locative prepositions into robot low-level control. The algorithms to resolve the locative preposition were implemented in MATLAB using the Multi-Parametric Toolbox [22]. The controller synthesis from Structured English was done using Python and TLV [23] and the robot was simulated using MATLAB.

Consider a robot patrolling the environment depicted in Figure 3 which contains fours regions of interest: A, B, C and D. The robot's patrol routine depends on whether an alarm is sounding in the environment.

The atomic propositions for the following scenarios are the sensor proposition $alarm$ and the regions propositions A, B, C, D [2]. Given the dimensions of the workspace, the regions of interest and the propositions, if a new robot behavior is needed, it simply requires the user to change a few words in the task description as demonstrated next.

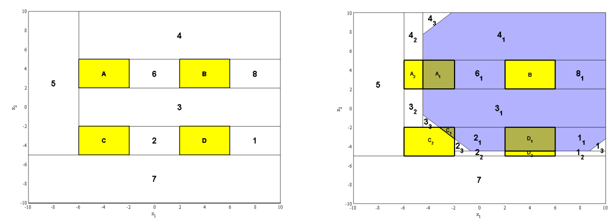
A. 'Stay near B unless the alarm is sounding'

For the first example, the desired behavior is captured by the following Structured English sentences where the 'near' distance is defined to be 6.5:

1) Environment starts with not alarm
2) Visit A
3) If you are not sensing alarm then visit B
4) If you are sensing alarm then go to outside near B
5) Visit C
6) Visit D
7) If you are sensing alarm and you were not inside near B or B then always outside near B and outside B
8) If you are not sensing alarm and you were near B then always near B or inside B

The initial step of creating the controller requires resolving the locative preposition 'near B'. Figure 4(a) depicts the original partition of the workspace, generated automatically from the region definitions and the workspace dimensions. The partition after the automatic resolution of the locative preposition is shown in Figure 4(b). In the specification, the expression 'near B' is replaced with 'A_1 or C_1 or D_1 or 1_1 or 2_1 or 3_1 or 4_1 or 6_1 or 8_1', A is replaced with 'A_1 or A_2', C is replaced with 'C_1 or C_2' and D is replaced with 'D_1 or D_2'.

[2]With a slight abuse of notation, the convex regions and the propositions relating to these regions are given the same name.



(a) Initial partition (b) Partition after resolving 'near B'

Fig. 4: Partition of the workspace for Example 1. The shaded area in (b) corresponds to the region that is 'near B'.

Figure 5 shows a simulation of the robot with the generated control. The blue circles indicate the trajectory of the robot when it is not sensing the alarm (the proposition $alarm$ is false) and the red stars indicate that the robot is sensing the alarm (the proposition $alarm$ is true). As seen, the robot is satisfying the task no matter what the environment is doing.

B. Additional constraints

Figure 6 depicts the behavior of the robot when the sentence "Always not between A and C" is added to the requirements in Section V-A. In this figure, the borders of the regions that are 'between A and C' are denoted by a broken red line.

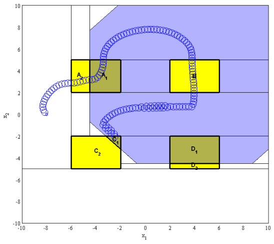
C. Task feasibility

In addition to logical consistency, a task may be deemed infeasible based on the definition of the locative prepositions. For example, if the 'near' distance is changed to 5 instead of 6.5 (as shown in Figure 3(b)) the synthesis procedure returns that the specification is unrealizable (cannot be executed by the robot). The reason for the infeasibility of the task is that when the 'near' distance is set to 5, region C is no longer 'near B' so when $alarm$ is off the robot cannot visit C while staying 'near B'.

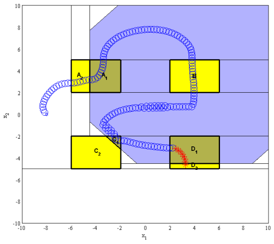
VI. CONCLUSION AND FUTURE DIRECTIONS

This paper presented an automatic method for creating correct-by-construction robot controllers for high-level tasks that include non-projective locative prepositions relating to regions within a robot's workspace.

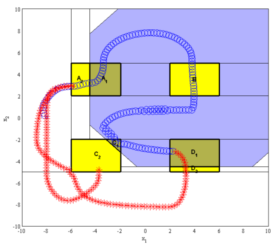
Future work will include creating over and under approximations of the regions corresponding to 'within' and



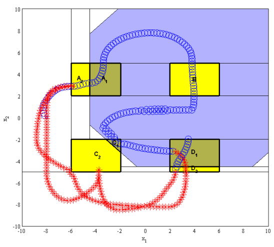
(a) At first $alarm$ is false. The robot visited A, B and C



(b) While in D $alarm$ becomes true and the robot is moving away from 'near B'

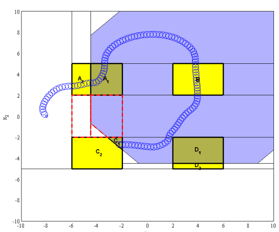


(c) The robot visited A and C while avoiding 'near B'

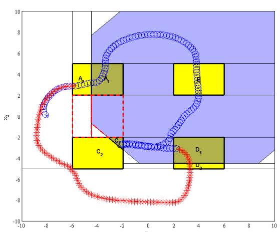


(d) When in D $alarm$ becomes false and the robot returns to the area 'near B'

Fig. 5: Robot patrolling the environment. The blue circles denote the trajectory of the robot when $alarm$ is off and red stars when $alarm$ is on.



(a) $alarm$ is off



(b) $alarm$ is on

Fig. 6: Robot should not enter the area between A and C.

'near' depending on the task; over-approximation when the regions should be avoided and under-approximation when they must be visited. This will require feedback between the task parsing and the locative preposition resolution.

Another challenging direction that will be pursued is incorporating projective prepositions. Resolving prepositions such as 'to your left' and 'behind' will require tight integration with the synthesis algorithm as well as the Structured English parsing; depending on the generated control, these prepositions will have different semantics.

ACKNOWLEDGMENTS

We thank Prof. Anna Papafragou for introducing us to [17]. Interactions between engineers and cognitive scientists are often very fruitful.

REFERENCES

[1] Sabrina Wilske and Geert-Jan Kruijff. Service robots dealing with indirect speech acts. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4698–4703, Beijing, China, October 2006.

[2] Timothy Brick, Paul Schermerhorn, and Matthias Scheutz. Speech and action: Integration of action and language for mobile robots. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1423–1428, San Diego, CA, October/November 2007.

[3] D. Paul Benjamin, Deryle Lonsdale, Damian Lyons, and Siddtharth Patel. Using cognitive semantics to integrate perception and motion in a behavior-based robot. In *LAB-RS '08: Proceedings of the 2008 ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems*, pages 77–82, Washington, DC, USA, 2008.

[4] Stanislao Lauria, Theocharis Kyriacou, Guido Bugmann, JohanBos, and Ewan Klein. Converting natural language route instructions into robot-executable procedures. In *Proceedings of the 2002 IEEE International Workshop on Robot andHuman Interactive Communication*, pages 223–228, Berlin, 2002.

[5] Colleen Crangle and Patrick C. Suppes. *Language and Learning for Robots*. CSLI Publications, Stanford, CA, USA, 1994.

[6] E. Allen Emerson. Temporal and modal logic. In *Handbook of theoretical computer science (vol. B): formal models and semantics*, pages 995–1072. MIT Press, Cambridge, MA, USA, 1990.

[7] Hadas Kress-Gazit, Gerogios E. Fainekos, and George J. Pappas. Temporal logic based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381, 2009.

[8] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *Automatic Control, IEEE Transactions on*, 53(1):287–297, February 2008.

[9] Morteza Hlahijanian, Marius Kloetzer, Sara Itani, Calin Belta, and Sean Andersson. Automatic deployment of autonomous cars in a robotic urban-like environment (rule). In *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009.

[10] S. Loizou and K. Kyriakopoulos. Automatic synthesis of multi-agent motion tasks based on ltl specifications. In *IEEE Conference on Decision and Control*, Bahamas, December 2004.

[11] Paulo Tabuada and George J. Pappas. Linear time logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 51(12):1862–1877, December 2006.

[12] R. Alur, T.A. Henzinger, G. Lafferriere, and G.J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88:971–984, 2000.

[13] juraj dzifcak, matthias scheutz, chitta baral, and paul 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. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009.

[14] Hadas Kress-Gazit, Gerogios E. Fainekos, and George J. Pappas. Translating structured english to robot controllers. *Advanced Robotics Special Issue on Papers from IROS 2007*, 22(12):13431359, 2008.

[15] Hadas Kress-Gazit. *Transforming high-level tasks to low-level controllers*. PhD thesis, University of Pennsylvania, Philadephia, PA, 2008.

[16] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock. Spatial language for human-robot dialogs. *IEEE Transactions on SMC, Part C, Special Issue on Human-Robot Interaction*, 34(2):154–167, May 2004.

[17] Joost Zwarts and Yoad Winter. Vector space semantics: A model-theoretic analysis of locative prepositions. *J. of Logic, Lang. and Inf.*, 9(2):169–211, 2000.

[18] Annette Herskovits. Semantics and pragmatics of locative expressions. *Cognitive Science*, 9(3):341 – 378, 1985.

[19] Fintan J. Costello and John D. Kelleher. Spatial prepositions in context: The semantics of 'near' in the presence of distractor objects. In *SIGSEM Workshop On Prepositions*, 2006.

[20] John D. Kelleher, Geert-Jan M. Kruijff, and Fintan J. Costello. Proximity in context: an empirically grounded computational model of proximity for processing topological spatial expressions. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 745–752, 2006.

[21] Hadas Kress-Gazit, Georgios E. Fainekos, and George J. Pappas. Where's waldo? sensor-based temporal logic motion planning. In *IEEE International Conference on Robotics and Automation*, pages 3116–3121, Rome, Italy, 2007.

[22] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004.

[23] A. Pnueli and E. Shahar. The TLV system and its applications, 1996.