Behavior Synthesis for an ATLAS Humanoid Robot from High-level User Specifications

Spyros Maniatopoulos, Philipp Schillinger, Vitchyr H. Pong, David C. Conner, and Hadas Kress-Gazit

List of TODOs, Fixes, Open Issues

***Abstract*— In this paper we . . .**

Body of my TODO example

I. Introduction

Contributions (brain dump):

- Partial to full specification
  - Most intuitive from the users point-of-view
  - Limited message size over bad comms (send partial specification → compile and synthesize onboard)
- Multi-paradigm specification (objectives and initial conditions from user, topology/modes, preconditions, task)
- Generalization of activation–completion paradigm [1]
- Integration with FlexBE and ROS
- Experimental validation on ATLAS

II. Preliminaries

A. ATLAS Humanoid Robot

. . .

B. Team ViGIR's Approach to High-level Control

. . .

C. Linear Temporal Logic and Reactive LTL Synthesis

. . .

III. Problem Statement

. . .

IV. LTL Specification Compilation

A. Multi-Paradigm Specification

. . .

B. Discrete Abstraction & Proposition Grounding

. . .

C. LTL Specification for ATLAS

1) Generic Formulas: The system safety requirements (1) dictate that an activation proposition should turn `False` once an outcome has been returned.

$$\bigwedge_{o\in Out(a)} \square\left(\pi_a \wedge \bigcirc \pi_a^o \Rightarrow \bigcirc \neg\pi_a\right) \tag{1}$$

The environment safety assumptions (2) dictate that the outcomes of an action are mutually exclusive. For example, an action cannot both succeed and fail.

$$\bigwedge_{o\in Out(a)} \square\left(\bigcirc \pi_a^o \Rightarrow \bigwedge_{o'\neq o} \bigcirc \neg\pi_a^{o'}\right) \tag{2}$$

This work was supported by . . . .

Philipp Schillinger is with BOSCH . . . `philipp.schillinger@de.bosch.com`. David C. Conner is with . . . `conner@. . .`. The remaining authors are with the Verifiable Robotics Research Group, Autonomous Systems Lab, Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA, {`sm2296`, `vhp22`, `hadaskg`}`@cornell.edu`.

2) Action-specific Formulas: The environment safety assumptions (3) govern the value of outcomes in the next time step. Specifically, formula (3a) says that if an outcome has been returned, and the corresponding action is re-activated, then any outcome can become `True`. Formula (3b) dictates that, if an outcome is `False` and the corresponding action is not activated, then that outcome should remain `False`. This pair of formulas is a generalization of the "fast-slow" formulas (3) and (4) in [1].

Formula (3a) is outdated. It doesn't account for the activation–outcomes paradigm!

$$\square\Big(\bigvee \pi_a^o \wedge \pi_a \Rightarrow \bigvee \bigcirc \pi_a^o\Big) \tag{3a}$$

$$\bigwedge_{o \in Out(a)} \square\Big(\neg\pi_a^o \wedge \neg\pi_a \Rightarrow \bigcirc \neg\pi_a^o\Big) \tag{3b}$$

The environment safety assumptions (4) dictate that the value of an outcome should not change if the corresponding action has not been activated again. In other words, the outcome persists.

$$\bigwedge_{o \in Out(a)} \square\left(\pi_a^o \wedge \neg\pi_a \Rightarrow \bigcirc \pi_a^o\right) \tag{4}$$

The environment liveness assumption (5c) is a fairness condition. It states that, (always) eventually, either the activation of an action will return an outcome, (5a), or that the robot will "change its mind", (5b). Formula (5a) is a generalization of $\varphi_a^{completion}$ in [1], whereas formula (5b) is exactly the same as φ_a^{change} in [1], since it consists of activation propositions only.

$$\varphi_a^{return} = \Big(\pi_a \wedge \bigvee \bigcirc \pi_a^o\Big) \vee \Big(\neg\pi_a \wedge \bigwedge \bigcirc \neg\pi_a^o\Big) \tag{5a}$$

$$\varphi_a^{change} = \left(\pi_a \wedge \bigcirc \neg\pi_a\right) \vee \left(\neg\pi_a \wedge \bigcirc \pi_a\right) \tag{5b}$$

$$\square\Diamond\left(\varphi_a^{return} \vee \varphi_a^{change}\right) \tag{5c}$$

The system safety requirement (6) demonstrates how a formula encoding the preconditions of an action, $Pre(a)$, looks like in the activation–outcomes paradigm.

$$\square\big(\bigvee_{p \in Pre(a)} \neg\pi_p^c \Rightarrow \neg\pi_a\big) \tag{6}$$

where the superscript $c \in Out(p)$ stands for "completion".

3) Control Mode Formulas: The system safety requirements (7) encode a topological transition relation, e.g., the valid transitions (motion) in the robot's workspace. Similar to [1], but with the addition of $\varphi_{\mathcal{R}}^{none}$.

$$\bigwedge_{r \in \mathcal{R}} \square\Big(\bigcirc \pi_r^c \Rightarrow \bigvee_{r' \in Adj(r)} \bigcirc \varphi_{r'} \vee \bigcirc \varphi_{\mathcal{R}}^{none}\Big) \tag{7}$$

where $\varphi_{\mathcal{R}}^{none} = \bigwedge_{r \in \mathcal{R}} \neg\pi_r$ being `True` stands for not activating any topological transitions.

The environment safety assumptions (8) enforce mutual exclusion between the topology (e.g. region) propositions, just like formula (1) in [1].

Can't remember why this is written in the future tense. Try getting rid of the $\bigcirc$ operators.

$$\bigwedge_{r \in \mathcal{R}} \square\Big(\bigcirc \pi_r^c \Leftrightarrow \bigwedge_{r' \neq r} \bigcirc \neg\pi_{r'}^c\Big) \tag{8}$$

The environment safety assumptions (9) govern how the robot's "state" in the transition system (e.g. location in the workspace) can change in a single time step in response to the activation of a transition. If the only outcome of topological transitions is "completion", i.e., $Out(r) = \{c\}$, $\forall r \in R$, then formula (9) collapses to formula (2) in [1].

$$\bigwedge_{r \in \mathcal{R}} \bigwedge_{r' \in Adj(r)} \square\Big(\pi_r^c \wedge \varphi_{r'} \Rightarrow \big(\bigcirc \pi_r^c \vee \bigvee_{o \in Out(r')} \bigcirc \pi_{r'}^o\big)\Big) \tag{9}$$

The environment safety assumptions (10) constrain the outcomes of topological transitions. Their addition is necessitated by $\varphi_{\mathcal{R}}^{none}$ being an option in the transition relation (7).

$$\bigwedge_{r \in \mathcal{R}} \bigwedge_{o \in Out(r)} \square\left(\neg\pi_r^o \wedge \neg\pi_r \Rightarrow \bigcirc \neg\pi_r^o\right) \tag{10}$$

The environment safety assumptions (11) dictate that the value of the outcomes of topological transitions must not change if no transition is being activated, i.e., they must persist.

$$\bigwedge_{r \in \mathcal{R}} \bigwedge_{o \in Out(r)} \square\left(\pi_r^o \wedge \varphi_{\mathcal{R}}^{none} \Rightarrow \bigcirc \pi_r^o\right) \tag{11}$$

The environment liveness assumption (12c) is the equivalent of the fairness condition (5c) for topology propositions. Formulas (12a) and (12b) are equivalent to $\varphi_{loc}^{completion}$ and φ_{loc}^{change} in [1].

$$\varphi_{\mathcal{R}}^{return} = \bigvee_{r \in \mathcal{R}} \Big(\varphi_r \wedge \bigvee_{o \in Out(r)} \bigcirc \pi_r^o\Big) \tag{12a}$$

$$\varphi_{\mathcal{R}}^{change} = \bigvee_{r \in \mathcal{R}} \left(\varphi_r \wedge \bigcirc \neg\varphi_r\right) \tag{12b}$$

$$\square\Diamond\left(\varphi_{\mathcal{R}}^{return} \vee \varphi_{\mathcal{R}}^{change} \vee \varphi_{\mathcal{R}}^{none}\right) \tag{12c}$$

4) Initial Conditions: Last but not least, this is how initial conditions look like in the "activation–outcomes" paradigm. **Note**: The choice of semantics has been influenced by ATLAS and the integration with FlexBE. There are definitely other choices that also make sense.

For each action (or topology proposition) in the initial conditions, the completion proposition should be `True` in the environment initial conditions (13a). All other outcomes of those actions (13b), as well as all outcomes of any other actions (13c), should be `False`.

$$\bigwedge_{i\in\mathcal{I}} \pi_i^c \tag{13a}$$

$$\bigwedge_{i\in\mathcal{I}} \bigwedge_{o\in Out(i), o\neq c} \neg\pi_i^o \tag{13b}$$

$$\bigwedge_{i\notin\mathcal{I}} \bigwedge_{o\in Out(i)} \neg\pi_i^o \tag{13c}$$

The semantics for activation are *chosen* such that the activation propositions are `False` regardless of whether that action (or topology proposition) is in the initial conditions or not (14). The reason being that, intuitively, if we want something to be an initial condition, then we shouldn't have the resulting controller re-activate it at the beginning of execution.

$$\bigwedge_{i\in\mathcal{I}} \neg\pi_i \tag{14a}$$

$$\bigwedge_{i\notin\mathcal{I}} \neg\pi_i \tag{14b}$$

5) Success and Failure: The system initial condition (15), safety requirements (16), and liveness requirement (17) are used to reason about the satisfaction of the system's goals in a finite run (as opposed to infinite execution, which is what LTL is defined over). Informally, the system propositions in $Out(SM)$ being `True` is equivalent to that state being an "accepting" / "final" state of a traditional automaton.[1] The proposition $\pi_{success} \in Out(SM)$ in formula (16d) is one of those "special" propositions.

Technically, all π_o, $\forall o \in Out(SM)$ should also be persistent, like memory props (16b).

$$\bigwedge_{g\in\mathcal{G}} \neg m_g \tag{15}$$

$$\bigwedge_{g\in\mathcal{G}} \bigcirc \pi_g^c \Rightarrow \bigcirc m_g \tag{16a}$$

$$\bigwedge_{g\in\mathcal{G}} m_g \Rightarrow \bigcirc m_g \tag{16b}$$

$$\bigwedge_{g\in\mathcal{G}} \neg m_g \wedge \bigcirc \neg\pi_g^c \Rightarrow \bigcirc \neg m_g \tag{16c}$$

$$\pi_{success} \Leftrightarrow \bigwedge_{g\in\mathcal{G}} m_g \tag{16d}$$

$$\square\diamond \bigvee_{o\in Out(SM)} \pi_o \tag{17}$$

Come up with alternative formulation of (16) that enforces strict ordering on the liveness requirements. It will be optional from the user's point-of-view.

D. *Other subsection*

...

V. ROS Implementation

VI. Experimental Validation

VII. Conclusions and Future Work

Acknowledgments

The authors would like to thank all other members of Team ViGIR and particularly Alberto Romay and Stefan Kohlbrecher.

References

[1] V. Raman, N. Piterman, and H. Kress-Gazit, "Provably Correct Continuous Control for High-Level Robot Behaviors with Actions of Arbitrary Execution Durations," in *IEEE Int'l. Conf. on Robotics and Automation*, 2013.

[1] In FlexBE terms, this is where execution would exit the current state machine.