LTL$_f$ Synthesis on First-Order Agent Programs in Nondeterministic Environments

AAAI 2025

Till Hofmann, Jens Claßen

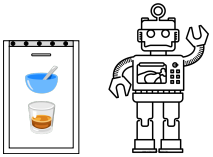
June 2, 2025

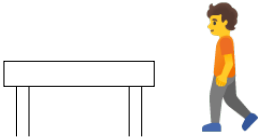
Roskilde University

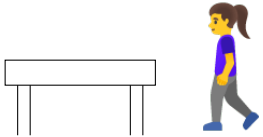
Chair of Computer Science 6 (Machine Learning and Reasoning)

RWTH AACHEN UNIVERSITY

Motivating Example: The Dishwasher Robot







High-Level Programming with Golog

GOLOG: High-level agent programming language

- Based on **situation calculus** $\rightarrow$ first-order logic

$\Rightarrow$ Open-world reasoning with incomplete information

- Allows nondeterministic constructs

loop:
  while $\exists x.\ OnRobot(x)$ **do**
    $\pi x.\ unload(x)$;
  $\pi y.\ goto(y)$;
  while $\exists x.\ DirtyDish(x, y)$ **do**
    $\pi x.\ load(x, y)$;
  $goto(kitchen)$

What is missing?

This GOLOG program does not (fully) solve our task:

- Assumes angelic nondeterminism, i.e., agent has complete control
  - → Actions may have **unintended effects**, e.g., dropping a cup while moving
  - → Users may **interfere** and change the world, e.g., placing a new dirty dish on the table
- Initial state satisfies the goal!
  - → The agent may need to *react* to requests → **temporal goals**

On the other hand:

- Acting in presence of an uncontrollable environment is well-understood in reactive synthesis
- Agent and environment alternate:
  - Environment chooses propositional symbols from $\mathcal{X}$
  - Agent reacts by choosing propositional symbols from $\mathcal{Y}$
- **LTL$_f$ synthesis** (De Giacomo and Vardi 2015): LTL$_f$ formula Φ over $\mathcal{X} \cup \mathcal{Y}$

→ Determine strategy such that Φ is satisfied for all possible environment choices

Program Realization as Synthesis

- Idea: Also model uncontrollable behavior as part of the agent program, e.g., during execution, a new dirty dish may appear in any room:

  loop: $\pi x, y.\, addDish(x, y)$
- Partition all actions into **agent** and **environment** actions: $\mathcal{A} = \mathcal{A}_C \,\dot\cup\, \mathcal{A}_E$
- Formulate goal as LTL$_f$ formula Φ, e.g.,

$$\mathcal{F}\,\mathcal{G}\,\neg\exists x, y.\, DirtyDish(x, y)$$

- Task: Given a GOLOG program $\mathcal{P}$, goal Φ, partitioning $\mathcal{A}_C \,\dot\cup\, \mathcal{A}_E$, find *policy* π:
  - π must follow the program $\mathcal{P}$
  - π must allow all possible environment actions
  - Every π trace must satisfy Φ

Solving the Synthesis Problem: Main Challenges

1. First-order logic and thus first-order synthesis is undecidable

→ Restrict to a decidable fragment

2. Check the satisfaction of the temporal goal Φ

→ Split Φ:

   1. Sub-formulas that are satisfied in the current state
   2. Sub-formulas that must be satisfied in some future state

3. Determine a strategy that executes the program δ and satisfies Φ

→ Game-theoretic approach: construct a finite game arena and label recursively

Step 1: Decidable Fragment

Zarrieß and Claßen 2016 describe a decidable fragment for verification:

1. The base logic is restricted to the **two-variable fragment of FOL with counting** (C^2)
2. **Successor state axioms must be acyclic**
3. The pick operator π may only **pick from a finite set** of ground terms

With these restrictions, we can define a finite abstraction of a GOLOG program $\mathcal{P} = (\mathcal{D}, \delta)$:

- A program may only accumulate finitely many effects $\mathfrak{E}^{\mathcal{D},\mathcal{A}}$
- A program may only mention **finitely many formulas** $\rightarrow$ finite context $\mathcal{C}(\mathcal{P})$
- Identify types of models w agreeing on context formulas after action effects:

$$\mathrm{type}(w) \doteq \{(\psi, E) \mid w \models \mathcal{R}[E, \psi]\}$$

"ψ holds in w after applying E"

context condition from $\mathcal{C}(\mathcal{P})$

effect from $\mathfrak{E}^{\mathcal{D},\mathcal{A}}$

Step 2: Tracking the Satisfaction of LTLf Goal Φ

- Here: LTLf syntax, but replacing propositions with fluent sentences

$$\Phi ::= \phi \mid \Phi \wedge \Phi \mid \mathcal{X}\,\Phi \mid \Phi\,\mathcal{U}\,\Phi$$

- We adopt two notions from (Li et al. 2020):

  ***Tail Normal Form* (TNF):** mark the last state of a trace with $Tail$, e.g.,

$$\mathcal{F}\,a \doteq \top\,\mathcal{U}\,a \rightsquigarrow (\neg Tail)\,\mathcal{U}\,a \wedge \mathcal{F}\,Tail$$

  ***neXt Normal Form* (XNF):** only allow $\mathcal{X}$ as outermost temporal operator, e.g.,

$$a\,\mathcal{U}\,b \rightsquigarrow b \vee (a \wedge \mathcal{X}\,(a\,\mathcal{U}\,b))$$

$$(\neg Tail)\,\mathcal{U}\,a \wedge \mathcal{F}\,Tail \rightsquigarrow \big(a \vee \neg Tail \wedge \mathcal{X}((\neg Tail)\,\mathcal{U}\,a)\big) \wedge \big(Tail \vee \mathcal{X}\,\mathcal{F}\,Tail\big)$$

→ Treat Φ as **propositional formula** with propositional assignment P

⇒ Split P intro three parts:

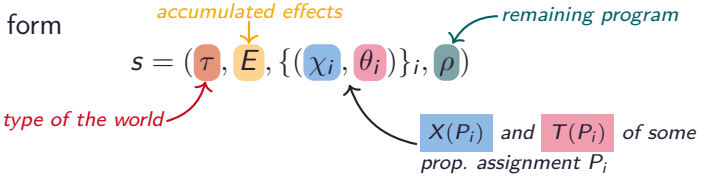
$$L(P) = \{l \mid l \in P \text{ is a literal other than } (\neg)\,Tail\ \} \quad = \{\neg a\}$$

$$X(P) = \{\theta \mid \mathcal{X}\,\theta \in P\} \quad = \{(\neg Tail)\,\mathcal{U}\,a,\ \mathcal{F}\,Tail\}$$

$$T(P) = \top \text{ if } Tail \in P \text{ and } T(P) = \bot \text{ otherwise} \quad = \bot$$

Putting Things Together: The Game Arena

- Construct a *game arena* $\mathbb{A}_{\mathcal{G}}^{\Phi}$ that captures the execution of $\mathcal{P}$ while tracking the satisfaction of Φ
- Each state is of the form



- A state is *final* if ρ is in a terminating configuration
- A state is *accepting* if $(\emptyset, \top) \in A$, i.e., Φ is satisfied
- **Transitions:**

$$\mathrm{xnf}(\neg Tail\, \mathcal{U}\, a \wedge \mathcal{F}\, Tail) = (a \vee \mathcal{X} \neg Tail\, \mathcal{U}\, a) \wedge (Tail \vee \mathcal{X}\, \mathcal{F}\, Tail)$$

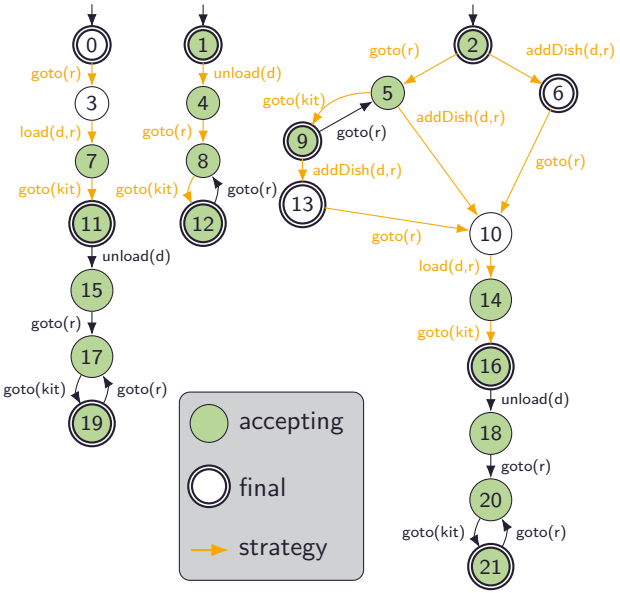
$$L(P) = \{a\} \quad X(P) = \emptyset \quad T(P) = \top$$

Example with 1 Room and 1 Cup

- Program: *agent* *environment*

  loop:
  while $\exists x.\ OnRobot(x)$ **do**
  $\pi x : \{d_1\}.\ unload(x)$;
  $\pi y : \{r_1\}.\ goto(y)$;
  while $\exists x.\ DirtyDish(x, y)$ **do**
  $\pi x : \{d_1\}.\ load(x, y)$;
  $goto(kitchen)$
  $\|$
  loop: $\pi x : \{d_1\}, y : \{r_1\}.\ addDish(x, y)$

- Goal: $\mathcal{F}\,\mathcal{G}\,\neg\exists x, y.\ DirtyDish(x, y)$



Conclusion

- GOLOG is an expressive agent programming language based on first-order logic
- Assumption so far: The agent is under complete control
- More realistic view: Agent acts in a partially controllable environment

→ Program realization is now a synthesis task with an LTL_f goal Φ

- Approach:
  - Finite abstraction of the infinite program configuration space
  - Track goal Φ by splitting into *local* and *next* parts
  - Use a game-theoretic approach to determine a policy

⇒ Resulting policy guarantees to satisfy the goal,
independent of the environment's choices

Appendix

Computing a Strategy

- Based on the finite game arena $\mathbb{A}_{\mathcal{G}}^{\Phi}$, determine a *terminating* and *winning* strategy
  terminating: The agent must eventually terminate by not choosing any actions
  winning: In every terminating state, the temporal goal Φ must be satisfied
- In principle, we can just start with the final+accepting states $\mathcal{S}_F \cap \mathcal{S}_A$ and label bottom up
- Problem: Even in a final+accepting state, the environment may continue and eventually lead into bad states

$\rightarrow$ *Guess* a subset $H \subseteq \mathcal{S}_F \cap \mathcal{S}_A$

- Check whether there is a strategy that enforce each play to end in H
- Label nodes bottom up with $\top/\bot$
- Any strategy that remains in the $\top$-labeled sub-graph is a terminating and winning strategy

Bibliography (I)

- De Giacomo, Giuseppe and Moshe Y. Vardi (2015). "Synthesis for LTL and LDL on Finite Traces". In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI Press, pp. 1558–1564.
- Li, Jianwen et al. (Dec. 2020). "SAT-based Explicit LTLf Satisfiability Checking". In: *Artificial Intelligence* 289, p. 103369. DOI: 10.1016/j.artint.2020.103369.
- Zarrieß, Benjamin and Jens Claßen (2016). "Decidable Verification of Golog Programs over Non-Local Effect Actions". In: *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, pp. 1109–1115.

Till Hofmann, Jens Claßen

Philadelphia, PA, USA, June 2, 2025