# Interpretable apprenticeship learning with temporal logic specifications

Daniel Kasenberg Matthias Scheutz

December 15, 2017

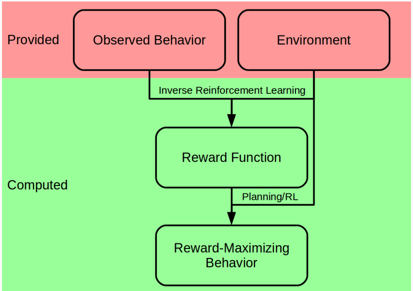
Human-Robot Interaction Laboratory (HRILab)
Tufts University

# Inverse Reinforcement Learning (IRL)

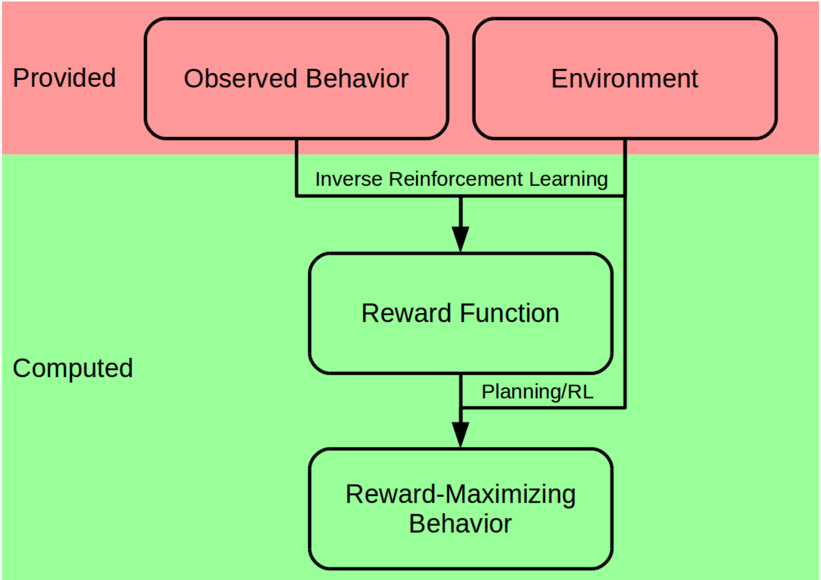
- Given a set of trajectories $\tau^1, \cdots, \tau^m$, where

$$\tau^i = s_0, a_0, s_1, a_1, \cdots, s_{T_i}, a_{T_i}, s_{T_i+1}$$

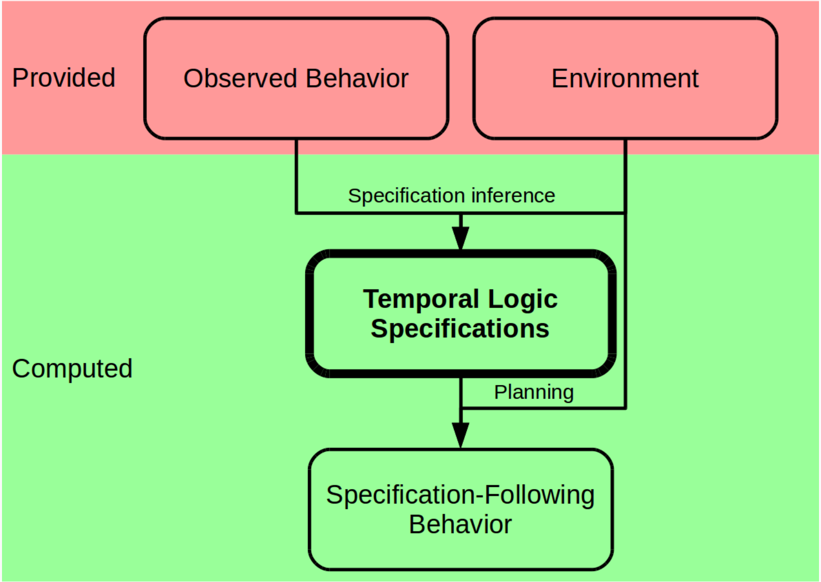
- … figure out which reward function $R$ "best explains" those trajectories.



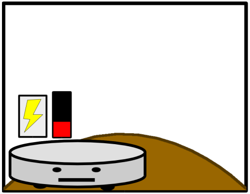
# What if we could swap out the reward function...



# ...with a statement in linear temporal logic?



# Example: CleaningWorld



- Vacuum cleaning robot in messy room
- Limited battery life
- Actions: *vacuum*, *wait*, *dock*, *undock*
- *vacuum* and *wait* actions deplete battery life

# Linear temporal logic (LTL)

A simple propositional logic encoding time

$$\phi ::= p \mid \neg\phi_1 \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid \\ \mathbf{X}\phi_1 \mid \mathbf{G}\phi_1 \mid \mathbf{F}\phi_1 \mid \phi_1 \mathbf{U} \phi_2$$

where $\phi_1, \phi_2$ are LTL statements; $p$ a proposition from some set $\Pi$.

- $\mathbf{X}\phi_1$: "in the next time step, $\phi_1$"
- $\mathbf{G}\phi_1$: "in all present and future time steps, $\phi_1$"
- $\mathbf{F}\phi_1$: "in some present or future time step, $\phi_1$"
- $\phi_1 \mathbf{U} \phi_2$: "$\phi_1$ will be true until $\phi_2$ becomes true"

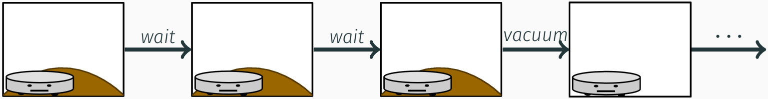
# Advantages of specifications over reward functions

- Handle more temporally complex properties and behaviors than (Markovian) reward functions
- Generalize to new MDPs and unseen states, if propositions in common
- Interpretable! (useful, e.g., for AI ethics and safety)

(Arnold, Kasenberg, and Scheutz 2017)

# Relating MDPs to LTL

- Augment the MDP with a set $\Pi$ of atomic propositions (e.g. *roomClean*, *batteryDead*)
- $\mathcal{L}(s)$: which propositions true in state $s$ ( *valuation* of $s$)
- LTL formulas are evaluated over an infinite sequence of *valuations* $\sigma_1, \sigma_2, \cdots$; that is, $\sigma_1, \sigma_2, \cdots \models \phi$
- We say that $\tau \models \phi$ iff $\mathcal{L}(s_0), \mathcal{L}(s_1), \cdots \models \phi$




$$\tau = \quad s_0, \quad a_0, \quad s_1, \quad a_1, \quad s_2, \quad a_2, \quad s_3,$$

| $s_0$ | $s_1$ | $s_2$ | $s_3$ |
|---|---|---|---|
| $\Downarrow$ | $\Downarrow$ | $\Downarrow$ | $\Downarrow$ |
| $\neg$*roomClean* | $\neg$*roomClean* | $\neg$*roomClean* | *roomClean* |
| $\neg$*batteryDead* | $\neg$*batteryDead* | $\neg$*batteryDead* | $\neg$*batteryDead* |

# Deterministic Rabin Automata (DRAs)

- Each LTL statement $\phi$ has a corresponding *Deterministic Rabin Automaton* $\mathcal{D}(\phi)$
  - A finite state machine over infinite sequences of *valuations*
  - $\mathcal{D}(\phi)$ accepts on input $\sigma_1, \sigma_2, \cdots$ iff $\sigma_1, \sigma_2, \cdots \vDash \phi$
- Can construct a **product MDP** – the Cartesian product of the original MDP and the DRA $\mathcal{D}(\phi)$

# Table of contents

# Related work

**Inverse Reinforcement Learning** (A. Ng and Russell 2000; Abbeel and A. Y. Ng 2004)

**MDP Planning with LTL specifications** (Ding et al. 2011; Wolff, Topcu, and Murray 2012; Fu and Topcu 2014; Svoreňová et al. 2015; Sharan and Burdick 2014; Leahy et al. 2015; Guo and Dimarogonas 2014; Reyes Castro et al. 2013; Tumova et al. 2013; Lahijanian et al. 2015)

**Specification Mining** (Gabel and Su 2008a; Gabel and Su 2008b; Gabel and Su 2010; Lemieux, Park, and Beschastnikh 2015; Kong et al. 2014; Chivilikhin, Ivanov, and Shalyto 2015)

# Table of contents

# Specification inference vs IRL

## Specification inference

- Given a set of trajectories $\tau^1, \cdots, \tau^m$, where

$$\tau^i = s_0, a_0, s_1, a_1, \cdots, s_{T_i}, a_{T_i}, s_{T_i+1}$$

- … figure out which **LTL statement** $\phi$ "best explains" those trajectories.

## Inverse reinforcement learning

- Given a set of trajectories $\tau^1, \cdots, \tau^m$, where

$$\tau^i = s_0, a_0, s_1, a_1, \cdots, s_{T_i}, a_{T_i}, s_{T_i+1}$$

- … figure out which **reward function** $R$ "best explains" those trajectories.

# “Best explains”

- We prefer specifications which are **simpler**
  - e.g. $\mathbf{G}\ \mathit{roomClean}$ vs $\mathbf{G}((p \vee \neg p) \wedge \neg \mathbf{F} \neg(\neg \mathit{roomClean} \to \bot))$

# “Best explains”

- We prefer specifications which are **simpler**
  - e.g. $\mathbf{G}\ \mathit{roomClean}$ vs $\mathbf{G}((p \vee \neg p) \wedge \neg\mathbf{F}\neg(\neg \mathit{roomClean} \rightarrow \bot))$
- We prefer specifications which “specifically describe” the observed behaviors
  - No trivial ($\mathbf{G}\top$) or contradictory ($\mathbf{G}\bot$) specifications
  - If $\tau^1, \cdots, \tau^m$ completely satisfy $\phi$ and $\phi$ is very hard to satisfy without trying, then $\phi$ describes $\tau^1, \cdots, \tau^m$ well

# As an optimization problem...

- A multi-objective optimization problem over the set of LTL statements:

$$\min_{\phi \in LTL} (Obj^S(\phi), Obj^X(\phi))$$

  where smaller values of $Obj^S$ correspond to simplicity, and smaller values of $Obj^X$ correspond to statements which specifically describe the trajectories

# $Obj^S$: simplicity

- We (simply) say that a candidate statement is simpler if it consists of fewer symbols than another statement:

$$Obj^S(\phi) = \ell(\phi)$$

where $\ell(\phi)$ is the length of $\phi$ in symbols

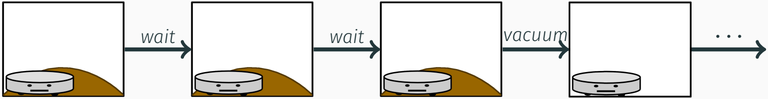
- (each connective, operator, and proposition counts as one symbol)
- $\mathsf{G}((\mathsf{X}\mathit{vacuum})\ \mathsf{U}\ \mathit{roomClean}))$ consists of 5 symbols

# Specifically describing trajectories

- A candidate specification specifically describes an agent's behavior if the observed behavior (in expectation) deviates less from the specification than random behavior does
- How to measure deviation from specification?
- **Idea:** allow agent to temporarily "suspend" the specification, but pay a cost for doing so

# Violation cost

For an infinite trajectory $\tau = s_0, a_0, s_1, a_1, \cdots$ and $N$ a set of nonnegative integers, let $\tau \backslash N$ be the sub-sequence of $\tau$ omitting the time steps indexed by elements of $N$




$$\tau = \quad s_0, \quad a_0, \quad s_1, \quad a_1, \quad s_2, \quad a_2, \quad s_3, \quad \cdots$$

# Violation cost

For an infinite trajectory $\tau = s_0, a_0, s_1, a_1, \cdots$ and $N$ a set of nonnegative integers, let $\tau \backslash N$ be the sub-sequence of $\tau$ omitting the time steps indexed by elements of $N$


$$\tau \backslash \{0, 2\} = \quad s_1, \quad a_1, \quad \quad s_3, \quad \cdots$$

# Violation cost

For an infinite trajectory $\tau = s_0, a_0, s_1, a_1, \cdots$ and $N$ a set of nonnegative integers, let $\tau \backslash N$ be the sub-sequence of $\tau$ omitting the time steps indexed by elements of $N$


$$\tau \backslash \{0, 2\} = \quad s_1, \quad a_1, \quad s_3, \quad \cdots$$

We define the *violation cost* of an infinite trajectory $\tau$ with respect to an LTL statement $\phi$ to be the (discounted) number of time steps that need to be omitted from $\tau$ to make $\tau$ satisfy $\phi$:

$$Viol_{\phi}(\tau) = \min_{\substack{N \subseteq \mathbb{N}_0 \\ \tau \backslash N \vDash \phi}} \sum_{t=0}^{\infty} \gamma^t \mathbf{1}_{t \in N}$$

# Computing expected violation cost for a policy

The total violation cost from a product state $(s, q)$ under product-space policy $\pi^{\otimes}$ satisfies the following **Bellman-like** equation:

$$
\begin{aligned}
Viol_{\phi}^{\pi^{\otimes}}((s,q)) = \sum_{a \in A} \pi^{\otimes}((s,q),a) \sum_{s' \in S} T(s,a,s') \min\{1 \\
+ \gamma Viol_{\phi}^{\pi^{\otimes}}((s',q)), \gamma Viol_{\phi}^{\pi^{\otimes}}((s,\delta(q,\mathcal{L}(s'))))\}
\end{aligned}
$$

where $q$ is the state of the DRA $\mathcal{D}(\phi)$, and $\delta$ is the transition function of $\mathcal{D}(\phi)$.

We can thus use value iteration to compute this for all product states $(s, q)$[1].

[1] There are a few caveats regarding initialization, etc. - see the paper for details.

# $Obj^X$: specifically describing trajectories

- We define our objective as

$$Obj^X(\phi) = Viol_\phi^{\pi^\otimes}(s_0^\otimes) - Viol_\phi^{\pi^{rand}}(s_0^\otimes)$$

  where $\pi^{rand}$ is the random policy and $s_0^\otimes$ is the initial product state, and $\pi^\otimes$ is the *observed product-space policy*

# The observed product-space policy $\pi^{\otimes}$

- For each finite trajectory $\tau = s_0, a_0, s_1, a_1, \cdots, s_T, a_T, s_{T+1}$, we can compute a corresponding product space trajectory $\tau^{\otimes} = (s_0, q_0), a_0, (s_1, q_1), a_1, \cdots, (s_T, q_T), a_T, s_{T+1}, q_{T+1}$

# The observed product-space policy $\pi^{\otimes}$

- For each finite trajectory $\tau = s_0, a_0, s_1, a_1, \cdots, s_T, a_T, s_{T+1}$, we can compute a corresponding product space trajectory $\tau^{\otimes} = (s_0, q_0), a_0, (s_1, q_1), a_1, \cdots, (s_T, q_T), a_T, s_{T+1}, q_{T+1}$
- We can then compute a "product space action restriction" $A^{\star}((s,q)) \subseteq A(s)$ for every product state $(s,q)$ by the following rules:
    - If some observed trajectory $\tau^{\otimes}$ contains $(s,q)$, then $A^{*}((s,q))$ is the set of all actions observed at $(s,q)$ in any trajectory
    - If $(s,q)$ is never observed in any trajectory, $A^{*}((s,q)) = A(s)$

# The observed product-space policy $\pi^{\otimes}$

- For each finite trajectory $\tau = s_0, a_0, s_1, a_1, \cdots, s_T, a_T, s_{T+1}$, we can compute a corresponding product space trajectory $\tau^{\otimes} = (s_0, q_0), a_0, (s_1, q_1), a_1, \cdots, (s_T, q_T), a_T, s_{T+1}, q_{T+1}$
- We can then compute a "product space action restriction" $A^{\star}((s, q)) \subseteq A(s)$ for every product state $(s, q)$ by the following rules:
  - If some observed trajectory $\tau^{\otimes}$ contains $(s, q)$, then $A^{*}((s, q))$ is the set of all actions observed at $(s, q)$ in any trajectory
  - If $(s, q)$ is never observed in any trajectory, $A^{*}((s, q)) = A(s)$
- We define the observed product-space policy $\pi^{\otimes}$ as the random policy over $A^{*}$

# Algorithm for Inferring LTL specifications

Given some set of trajectories $\tau^1, \cdots, \tau^m$:

For each candidate LTL specification $\phi$:

# Algorithm for Inferring LTL specifications

Given some set of trajectories $\tau^1, \cdots, \tau^m$:

For each candidate LTL specification $\phi$:

- Compute the DRA $\mathcal{D}(\phi)$ and product MDP $\mathcal{M}^{\otimes}$

# Algorithm for Inferring LTL specifications

Given some set of trajectories $\tau^1, \cdots, \tau^m$:

For each candidate LTL specification $\phi$:

- Compute the DRA $\mathcal{D}(\phi)$ and product MDP $\mathcal{M}^{\otimes}$
- Compute the observed product-space policy $\pi^{\otimes}$

# Algorithm for Inferring LTL specifications

Given some set of trajectories $\tau^1, \cdots, \tau^m$:

For each candidate LTL specification $\phi$:

- Compute the DRA $\mathcal{D}(\phi)$ and product MDP $\mathcal{M}^{\otimes}$
- Compute the observed product-space policy $\pi^{\otimes}$
- Compute the "violation cost" objective function by

$$Obj^X(\phi) := Viol_{\phi}^{\pi^{\otimes}}(s_0^{\otimes}) - Viol_{\phi}^{\pi_{rand}^{\otimes}}(s_0^{\otimes})$$

- Compute

$$Obj^S(\phi) := \ell(\phi)$$

# Algorithm for Inferring LTL specifications

Given some set of trajectories $\tau^1, \cdots, \tau^m$:

For each candidate LTL specification $\phi$:

- Compute the DRA $\mathcal{D}(\phi)$ and product MDP $\mathcal{M}^{\otimes}$
- Compute the observed product-space policy $\pi^{\otimes}$
- Compute the "violation cost" objective function by

$$Obj^X(\phi) := Viol_{\phi}^{\pi^{\otimes}}(s_0^{\otimes}) - Viol_{\phi}^{\pi_{rand}^{\otimes}}(s_0^{\otimes})$$

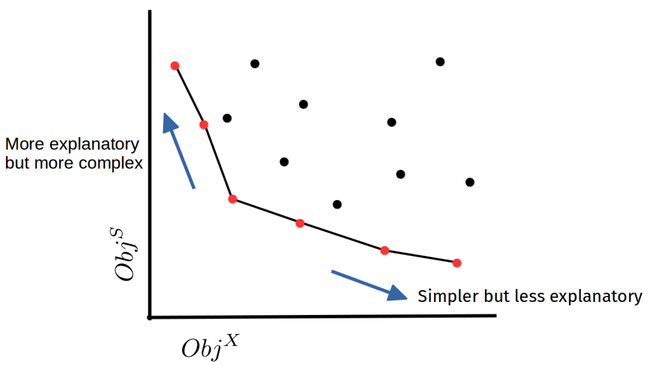
- Compute

$$Obj^S(\phi) := \ell(\phi)$$

Compute $\min_{\phi}(Obj^S(\phi), Obj^X(\phi))$

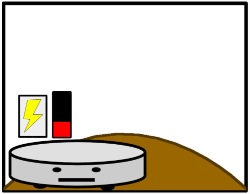
# Multi-objective optimization

- Any multi-objective optimization algorithm will do, if it can optimize over grammars (we used NSGA-II)
- Running any such algorithm will result in a set of Pareto-efficient candidate specifications $\phi_1, \cdots, \phi_k$



# Table of contents

# Evaluation: CleaningWorld



- Mess takes 5 vacuum actions to clean; initial battery life: 3
- Propositions: *roomClean*, *batteryDead*
- Proposition for each action: *vacuum*, *wait*, *dock*, *undock*
- Trajectories: robot continually vacuums, docking and recharging only when necessary
    - Cut off after 10 time steps, before room completely clean
- Ran specification inference 20 times

# Evaluation: CleaningWorld

**Table 1:** Pareto efficient solutions in action-based CleaningWorld

| $\phi$ | $Obj^X(\phi)$ | $Obj^S(\phi)$ | # Runs |
|---|---|---|---|
| $\mathbf{G}(roomClean)$ | -72.74240 | 2 | 20 |
| $\mathbf{G}(\mathbf{F}\ roomClean)$ | -75.15686 | 3 | 20 |
| $\mathbf{G}(vacuum \vee \mathbf{F}\ roomClean)$ | -75.15832 | 5 | 3 |
| $\mathbf{G}(\mathbf{F}(roomClean \vee dock))$ | -75.15782 | 5 | 3 |
| $\mathbf{G}((\mathbf{F}\ roomClean) \vee dock)$ | -75.15832 | 5 | 2 |
| $\mathbf{G}((\mathbf{X} roomClean) \vee vacuum)$ | -75.64639 | 5 | 2 |

# Table of contents

# Contribution

An algorithm for inferring *linear temporal logic (LTL) specifications* from agent behavior in Markov Decision Processes.

# Future work

- Efficiency/scalability
- Unknown transition dynamics, POMDPs, multi-agent domains
- Active learning

# Acknowledgments

- This project was in part supported by ONR grant N00014-16-1-2278.

# References i


Abbeel, Pieter and Andrew Y Ng (2004). "Apprenticeship learning via inverse reinforcement learning". In: *Proc. 21st International Conference on Machine Learning (ICML)*, pp. 1–8. DOI: 10.1145/1015330.1015430. arXiv: 1206.5264. URL: http://www.scopus.com/inward/record.url?eid=2-s2.0-14344251217%7B%5C&%7DpartnerID=40.

Arnold, Thomas, Daniel Kasenberg, and Matthias Scheutz (2017). "Value Alignment or Misalignment–What Will Keep Systems Accountable?". In: *3rd International Workshop on AI, Ethics, and Society*. URL: https://hrilab.tufts.edu/publications/aaai17-alignment.pdf.

Chivilikhin, Daniil, Ilya Ivanov, and Anatoly Shalyto (2015). "Inferring Temporal Properties of Finite-State Machine Models with Genetic Programming". In: *Proc. 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 1185–1188. ISBN: 9781450334884. DOI: 10.1145/2739482.2768475. URL: http://dl.acm.org/citation.cfm?doid=2739482.2768475.

Ding, Xu Chu et al. (2011). "LTL control in uncertain environments with probabilistic satisfaction guarantees". In: *Proceedings - IFAC World Congress*. Vol. 18, pp. 3515–3520. ISBN: 9783902661937. DOI: 10.3182/20110828-6-IT-1002.02287. arXiv: 1104.1159.

Fu, Jie and Ufuk Topcu (2014). "Probably Approximately Correct MDP Learning and Control With Temporal Logic Constraints". In: *Robotics: Science and Systems X*. arXiv: 1404.7073. URL: http://arxiv.org/abs/1404.7073.

Gabel, Mark and Zhendong Su (2008a). "Javert: Fully Automatic Mining of General Temporal Properties from Dynamic Traces". In: *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. SIGSOFT '08/FSE-16. Atlanta, Georgia: ACM, pp. 339–349. ISBN: 978-1-59593-995-1. DOI: 10.1145/1453101.1453150. URL: http://doi.acm.org/10.1145/1453101.1453150.

# References iii


Gabel, Mark and Zhendong Su (2008b). “Symbolic Mining of Temporal Specifications”. In: *Proc. 30th International Conference on Software Engineering*. ICSE ’08. Leipzig, Germany: ACM, pp. 51–60. ISBN: 978-1-60558-079-1. DOI: `10.1145/1368088.1368096`. URL: `http://doi.acm.org/10.1145/1368088.1368096`.

– (2010). “Online Inference and Enforcement of Temporal Properties”. In: *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 1*. ICSE ’10. Cape Town, South Africa: ACM, pp. 15–24. ISBN: 978-1-60558-719-6. DOI: `10.1145/1806799.1806806`. URL: `http://doi.acm.org/10.1145/1806799.1806806`.

Guo, M. and D. V. Dimarogonas (2014). “Multi-agent plan reconfiguration under local LTL specifications”. In: *The International Journal of Robotics Research* 34.2, pp. 218–235. ISSN: 0278-3649. DOI: `10.1177/0278364914546174`. URL: `http://ijr.sagepub.com/content/34/2/218.short`.

# References iv


Kong, Zhaodan et al. (2014). “Temporal Logic Inference for Classification and Prediction from Data”. In: *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*, pp. 273–282. DOI: 10.1145/2562059.2562146.

Lahijanian, Morteza et al. (2015). “This Time the Robot Settles for a Cost: A Quantitative Approach to Temporal Logic Planning with Partial Satisfaction”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29, pp. 3664–3671. ISBN: 9781577357032.

Leahy, Kevin et al. (2015). “Distributed Information Gathering Policies under Temporal Logic Constraints”. In: *IEEE Conference on Decision and Control (CDC)*. Vol. 54, pp. 6803–6808. ISBN: 9781479978854. DOI: 10.1109/CDC.2015.7403291.

Lemieux, Caroline, Dennis Park, and Ivan Beschastnikh (2015). “General LTL specification mining”. In: *Automated Software Engineering (ASE), 30th IEEE/ACM International Conference on*. IEEE, pp. 81–92.

Ng, Andrew and Stuart Russell (2000). "Algorithms for inverse reinforcement learning". In: *Proc. Seventeenth International Conference on Machine Learning*. Vol. 0, pp. 663–670. ISBN: 1-55860-707-2. DOI: 10.2460/ajvr.67.2.323. arXiv: arXiv:1011.1669v3. URL: http://www-cs.stanford.edu/people/ang/papers/icml00-irl.pdf.

Reyes Castro, Luis I. et al. (2013). "Incremental sampling-based algorithm for minimum-violation motion planning". In: *Proc. IEEE Conference on Decision and Control*, pp. 3217–3224. ISBN: 9781467357173. DOI: 10.1109/CDC.2013.6760374. arXiv: arXiv:1305.1102v1.

Sharan, Rangoli and Joel Burdick (2014). "Finite state control of POMDPs with LTL specifications". In: *Proceedings of the American Control Conference*, pp. 501–508. ISBN: 9781479932726. DOI: 10.1109/ACC.2014.6858909.

# References vi


Svoreňová, Mária et al. (2015). "Temporal logic motion planning using POMDPs with parity objectives". In: *Proceedings of the 18th International Conference on Hybrid Systems Computation and Control*, pp. 233–238. ISBN: 9781450334334. DOI: 10.1145/2728606.2728617. URL: http://dl.acm.org/citation.cfm?id=2728606.2728617.

Tumova, Jana et al. (2013). "Least-violating control strategy synthesis with safety rules". In: *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control*, pp. 1–10. ISBN: 9781450315678. DOI: 10.1145/2461328.2461330.

Wolff, Eric M., Ufuk Topcu, and Richard M. Murray (2012). "Robust control of uncertain Markov Decision Processes with temporal logic specifications". In: *IEEE Conference on Decision and Control (CDC)*. Vol. 51, pp. 3372–3379. ISBN: 978-1-4673-2066-5. DOI: 10.1109/CDC.2012.6426174. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6426174.