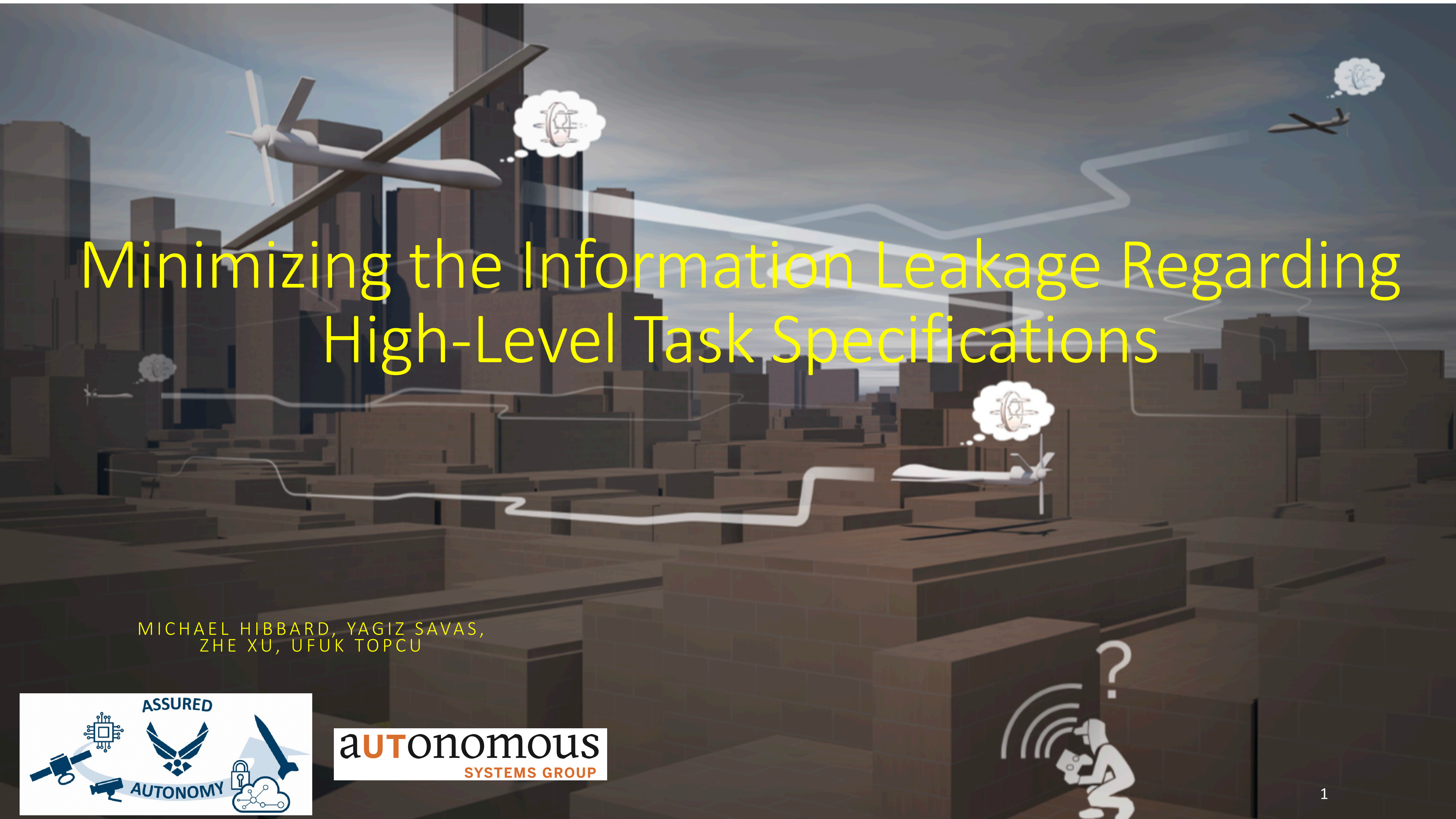


Minimizing the Information Leakage Regarding High-Level Task Specifications

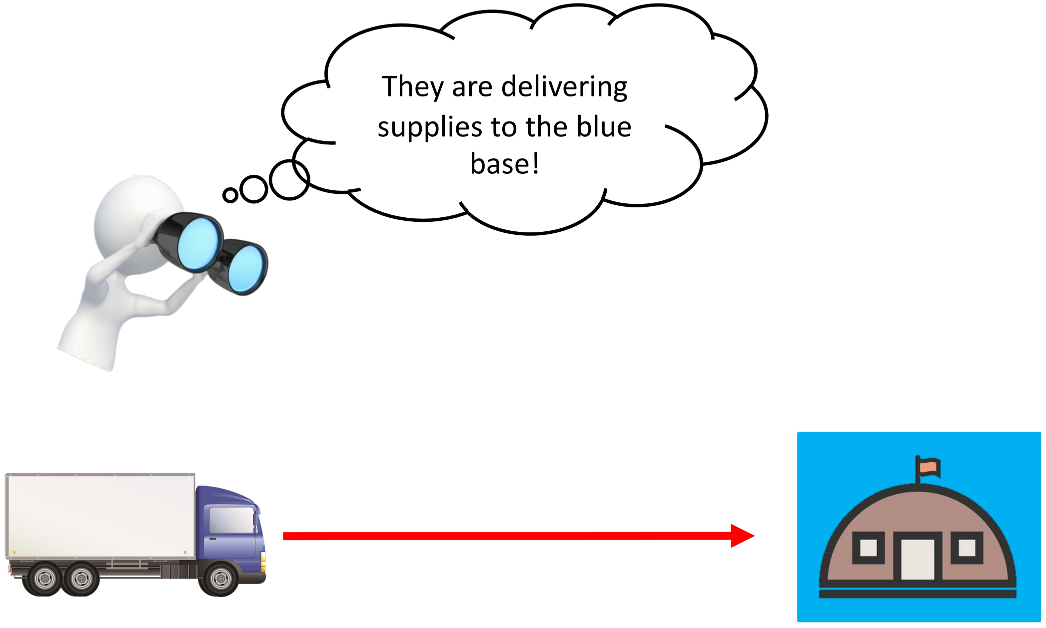
MICHAEL HIBBARD, YAGIZ SAVAS,
ZHE XU, UFUK TOPCU

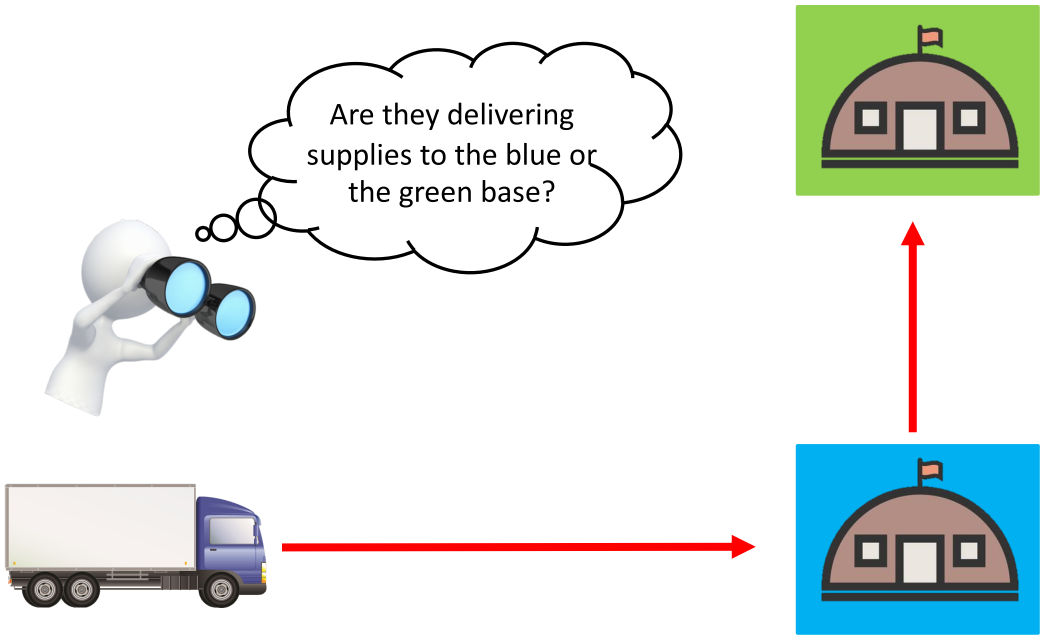
ASSURED AUTONOMY

autonomous SYSTEMS GROUP

Motivation

- By only completing a single task, the objective of an agent is clear to an observer.
- By completing multiple tasks, an observer must attempt to infer the true objective of the agent.
    - Uncertainty in which task the agent *cares* about completing, observer cannot optimally allocate resources

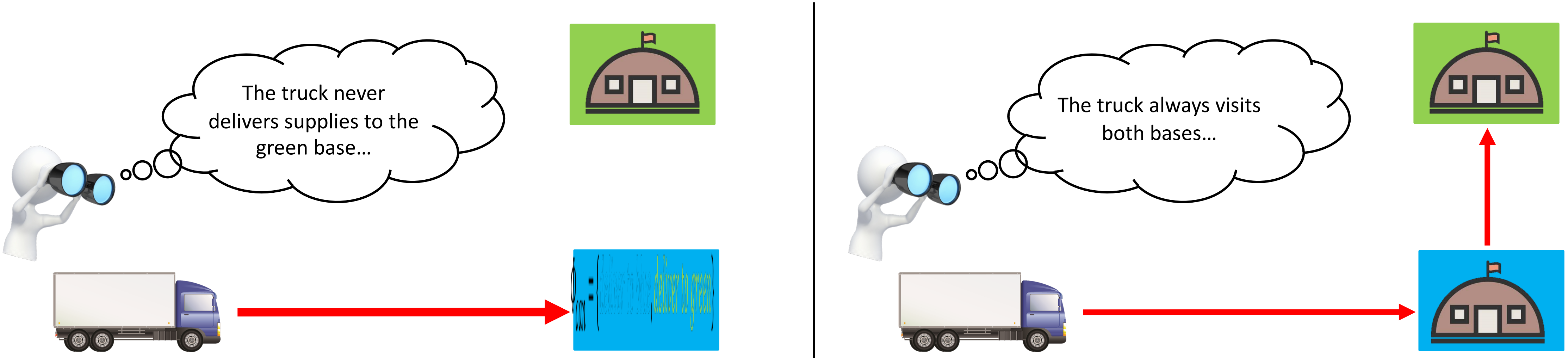




High-Level Problem Formulation

- The agent and the observer each know a set of ***specifications***, $\Phi = \{\phi_1, \dots, \phi_N\}$.
    - Only the agent knows the ***ground-truth specification*** , $\phi^* \in \Phi$.
    - The observer seeks to infer ϕ^* from the set of ***candidate specifications*** $\Phi_{can} \subseteq \Phi$, the specifications completed with a probability above some common-knowledge threshold.
- ***Goal:*** Synthesize the agent's policy such that it completes ϕ^* with a desired probability while also leading an observer to believe that each $\phi_i \in \Phi_{can}$ is equally likely to be ϕ^*.

$$\Phi = \{deliver\ to\ blue, deliver\ to\ green\}$$

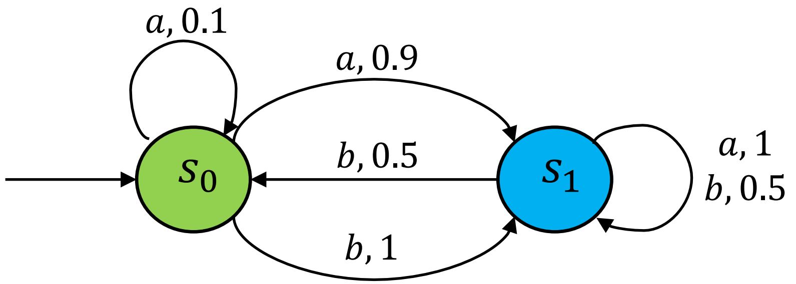


$\Phi_{can} = \{deliver\ to\ blue\}$ | $\Phi_{can} = \{deliver\ to\ blue, deliver\ to\ green\}$

Agent Model

- The agent operates in a stochastic environment modelled as a ***Markov decision process (MDP)***, given by $\mathcal{M} = \{S, s_0, \mathcal{A}, \mathcal{P}, \mathcal{AP}, \mathcal{L}\}$
  - S is a finite set of states
  - s_0 is a unique initial state
  - $\mathcal{A}$ is a finite set of actions
  - $\mathcal{P}$ is a transition function, $\mathcal{P}: S \times \mathcal{A} \rightarrow \Delta(S')$
  - $\mathcal{AP}$ is a set of atomic propositions
  - $\mathcal{L}$ is a labelling function, $\mathcal{L}: S \rightarrow 2^{\mathcal{AP}}$
- A ***policy*** for an agent is a sequence $\pi = (d_1, d_2, \dots)$ where each $d_t: S \rightarrow \Delta(\mathcal{A})$. Denote the set of all policies by $\Pi(\mathcal{M})$.

$\Delta(\cdot)$ is a probability mass function



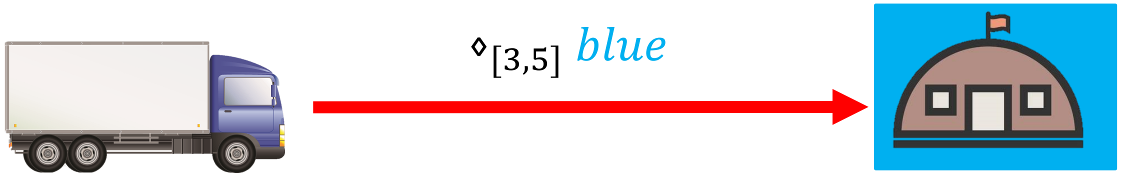
$\mathcal{AP} = \{Green, blue\}$
$\mathcal{A} = \{a, b\}$
$S = \{s_0, s_1\}$

Agent Specifications

- Use ***temporal logic*** to express agent specifications.
  - Relate occurrence of an event, causality between events, and ordering of successive events
- Focus on syntactically co-safe ***parametric linear temporal logic***, includes parameterized temporal operators:
  - ***Parameterized Always***: $\square_{[a,b]} p$
  - ***Parameterized Eventually***: $\diamond_{[a,b]}\, p$
  - Can nest together: $\square_{[a,b]}\, \diamond_{[c,d]}\, p$

    $\diamond_{[a,b]}\, \square_{[c,d]} p$

Will focus on these four structures



Eventually, between the third and fifth time steps, visit the blue base

Observer Inference Model

- Ignores specifications satisfied with low probability when inferring which is ϕ^*
- Uses a simple ***averaging rule*** to assign inference probabilities:

$$\Pr(\phi = \phi^* | \phi \in \Phi_{can}) \coloneqq \frac{\overbrace{Pr_{\mathcal{M}}^{\pi}(s_0 \vDash \phi)\mathbb{I}\{\phi \in \Phi_{can}\}}}{\sum_{\phi \in \Phi} \underbrace{Pr_{\mathcal{M}}^{\pi}(s_0 \vDash \phi_i)}\underbrace{\mathbb{I}\{\phi \in \Phi_{can}\}}}$$

Probability of satisfying ϕ, ignore if $\phi \notin \Phi_{can}$

Probability of satisfying ϕ in MDP $\mathcal{M}$ under policy π

Indicator function taking value 1 if $\phi \in \Phi_{can}$ and 0 otherwise.

- Uncertainty of the observer measured using the ***entropy*** of its inference probabilities.

$$H^{\pi}(\Phi_{can}) \coloneqq -\sum_{\phi \in \Phi_{can}} \Pr(\phi = \phi^* | \phi \in \Phi_{can}) \log(\Pr(\phi = \phi^* | \phi \in \Phi_{can}))$$

Problem: Synthesis of Minimum Information Leakage Policy

- Synthesize a policy $\pi \in \Pi(\mathcal{M})$ for the agent solving:

$$\max_{\pi \in \Pi(\mathcal{M})} H^{\pi}(\Phi_{can})$$

$\mathcal{M}$: given MDP

$$\text{subject to: } Pr^{\pi}_{\mathcal{M}}(s_0 \vDash \phi^*) \geq \Gamma$$

Agent must satisfy ϕ^* with probability above this threshold

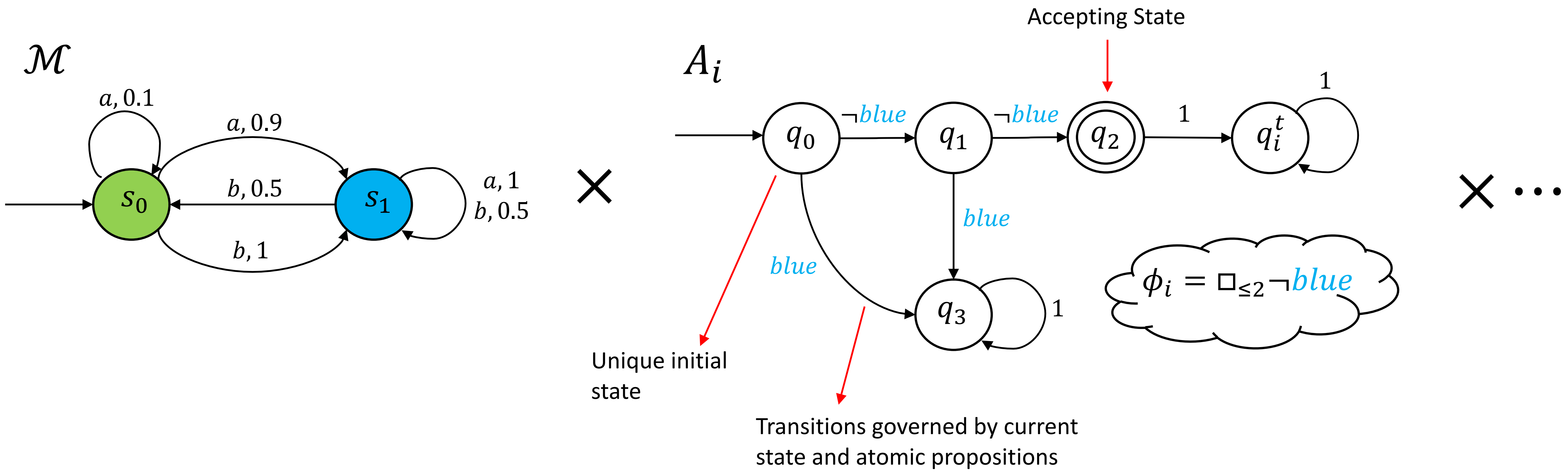
ϕ^*: given ground-truth specification

$$\phi \in \Phi_{can} \Leftrightarrow Pr^{\pi}_{\mathcal{M}}(s_0 \vDash \phi) \geq \beta$$

Agent must satisfy ϕ_i with probability above this threshold for observer to consider it a candidate specification

Solution – Product MDP

- For each specification ϕ_i, construct its corresponding ***deterministic finite automaton (DFA)***, A_i.
- To determine the satisfaction probabilities, form the ***product MDP*** $\mathcal{M} \otimes A_i$ for each $i = 1 \dots N$.



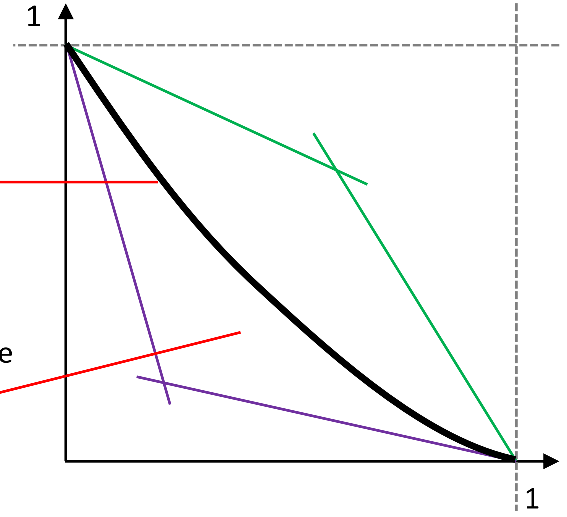
Solution – "Exact Approximation" of the Objective

$$H^{\pi}(\Phi_{can}) \coloneqq - \sum_{\phi \in \Phi_{can}} \underbrace{\Pr(\phi = \phi^* | \phi \in \Phi_{can})}_{= \frac{\nu(i)}{\sum \nu(i)}} \log(\Pr(\phi = \phi^* | \phi \in \Phi_{can}))$$

Bilinear constraint!

$= \frac{\nu(i)}{\sum \nu(i)}$ where $\nu(i) = Pr^{\pi}_{\mathcal{M}}(s_0 \vDash \phi)\mathbb{I}\{\phi \in \Phi_{can}\}$

- Replace this bilinear constraint by a ***McCormick Envelope*** (affine in $Pr^{\pi}_{\mathcal{M}}(s_0 \vDash \phi_i)$, $\mathbb{I}\{\phi_i \in \Phi_{can}\}$):



Affine over-approximators:
- $\nu(i) \leq \mathbb{I}\{\phi_i \in \Phi_{can}\}$
- $\nu(i) \leq Pr^{\pi}_{\mathcal{M}}(s_0 \vDash \phi_i)$

Affine under-approximators:
- $\nu(i) \geq 0$
- $\nu(i) \geq \mathbb{I}\{\phi_i \in \Phi_{can}\} + Pr^{\pi}_{\mathcal{M}}(s_0 \vDash \phi_i) - 1$

- The relaxation is exact, $\nu(i) = Pr^{\pi}_{\mathcal{M}}(s_0 \vDash \phi_i)$ if $\mathbb{I}\{\phi_i \in \Phi_{can}\} = 1$ and 0 otherwise.

Solution – Optimization Problem

- Mixed-integer program (MIP) with quasi-concave objective function
- Solve using bisection method together with MIP solvers
- State space is the product of the MDP with all specification DFAs

$$\max_{\lambda(s,a),x(i)} -\sum_{i\in[N]} \frac{\nu(i)}{\sum_{j\in[N]}\nu(j)} \log\left(\frac{\nu(i)}{\sum_{j\in[N]}\nu(j)}\right)$$

← Entropy of observer's likelihood probabilities

subject to:

$$\forall s \in S_p \backslash \mathrm{B}, \quad \sum_{a\in\mathcal{A}} \lambda(s,a) - \sum_{s'\in S_p}\sum_{a\in\mathcal{A}} \mathbb{P}_{s',a,s}\lambda(s',a) = \alpha(s)$$

← Flow constraint: If you visit a state, must also leave that state

$$\forall i \in [N], \quad \mu_i = \sum_{s\in S_p : s[i+2]\in\mathcal{F}_i}\sum_{a\in\mathcal{A}} \lambda(s,a)$$

← Probability of satisfying the i^{th} specification, ϕ_i

$$\mu(1) \geq \Gamma$$

← Must satisfy ϕ^* with probability Γ (Assume $\phi^* = \phi_1$

$$\forall i \in [N], \quad \mu(i) \geq \beta x(i)$$
$$\forall i \in [N], \quad \nu(i) = \mu(i)x(i)$$
$$\forall i \in [N], \quad x(i) \in \{0,1\}$$

← Only consider satisfaction probability if $\phi_i \in \phi_{can}$. Use McCormick Envelope for middle constraint.

$$\forall s \in S_p, \forall a \in \mathcal{A}, \quad \lambda(s,a) \geq 0$$

← Expected residence is non-negative

$$\pi(s,a) = \frac{\lambda(s,a)}{\sum_{a\in\mathcal{A}}\lambda(s,a)}$$

← Afterwards, obtain policy from flow variables

Approximate Solution Method: Probabilities at Each Time Step

- Predict whether a specification holds from whether its atomic propositions hold at each time step.
- Use the ***Fréchet inequalities*** to relate the probabilities of satisfying each ϕ_i at an individual time step to the probability of satisfying ϕ_i over the entire time interval
- $\phi_i = \square_{[k_1,k_2]} p_i$: p_i should hold at every time step over $[k_1, k_2]$.

Fréchet inequality for conjunction:

$$\underbrace{\Pr\left(\bigwedge_{t=k_1}^{k_2} p_i(t)\right)}_{Pr_{\mathcal{M}}^{\pi}(s_0 \vDash \phi = \square_{[k_1,k_2]} p_i)} \geq \max\{0, \sum_{t=k_1}^{k_2} \eta_i(t) - (k_2 - k_1)\}$$

- $\phi_i = \diamond_{[k_1,k_2]} p_i$: p_i should hold at least once over $[k_1, k_2]$.

Fréchet inequality for disjunction:

$$\underbrace{\Pr\left(\bigvee_{t=k_1}^{k_2} p_i(t)\right)}_{Pr_{\mathcal{M}}^{\pi}(s_0 \vDash \phi = \diamond_{[k_1,k_2]} p_i)} \geq \max\{\eta(k_1), \ldots, \eta(k_2)\}$$

Approximate Solution – Optimization Problem

- Remains a MIP with a quasi-concave objective function.
- The program uses the state space of an expanded MDP (but not the product of many automata).

$$\max_{\lambda(s,a),x(i)} -\sum_{i\in[N]} \frac{\nu(i)}{\sum_{j\in[N]}\nu(j)} \log\left(\frac{\nu(i)}{\sum_{j\in[N]}\nu(j)}\right)$$

subject to:

$\forall s \in S_p \backslash \mathrm{B},\quad \sum_{a\in\mathcal{A}} \lambda(s,a) - \sum_{s'\in S_p} \sum_{a\in\mathcal{A}} \mathbb{P}_{s',a,s}\lambda(s',a) = \alpha(s)$

$\forall i \in [N],\quad \sum_{\substack{s\in S^{[\tau+1]} \\ s[2]=t, p_i\in\mathcal{L}(s)}} \sum_{a\in\mathcal{A}} \lambda(s,a) = \eta_i(t)$

For each specification, determine the probability that it holds at each time step over interval of interest

$\forall i \in [N],\quad \mu_i = \sum_{s\in S_p : s[i+2]\in\mathcal{F}_i} \sum_{a\in\mathcal{A}} \lambda(s,a)$

For each specification, replace right-hand side with lower bounds derived from Fréchet inequalities

$\mu(1) \geq \Gamma$

$\forall i \in [N],\quad \mu(i) \geq \beta x(i)$

$\forall i \in [N],\quad \nu(i) = \mu(i)x(i)$

$\forall i \in [N],\quad x(i) \in \{0,1\}$

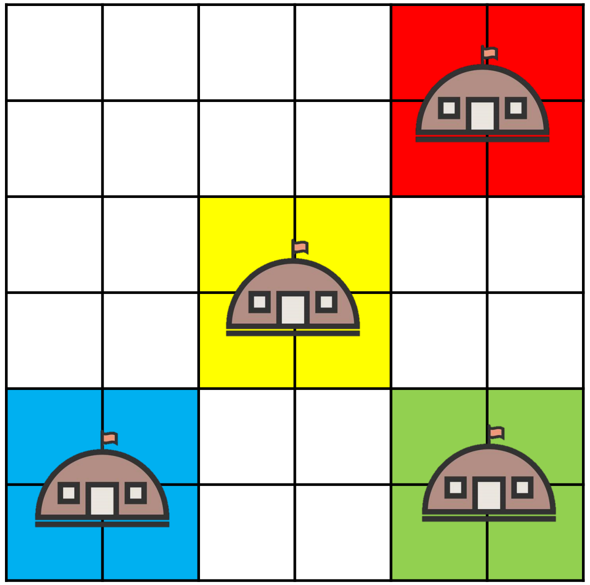
$\forall s \in S_p, \forall a \in \mathcal{A},\quad \lambda(s,a) \geq 0$

$$\pi(s,a) = \frac{\lambda(s,a)}{\sum_{a\in\mathcal{A}}\lambda(s,a)}$$

Example – Delivering Supplies to Bases

- Agent must resupply a specific base at a specific time over the timespan of interest
- Consider two different sets of specifications – require the agent to visit different numbers of bases

Example	Specifications ϕ	Num. Vars. (Continuous, binary)	Sol. Time, Exact	Sol. Time, approx.	$H^{\pi}(\phi_{can})$, exact	$H^{\pi}(\phi_{can})$, approx
Resupply-1	$\phi^* = \square_{[9,10]} blue$ $\phi_2 = \square_{[29,30]} red$	Exact: 4870 con. 1 bin. Approx: 3079 con. 1 bin.	9.25 s	6.75 s	1.000 bits	0.999 bits
Resupply-2	$\phi^* = \square_{[9,10]} blue$ $\phi_2 = \square_{[16,18]} yellow$ $\phi_3 = \square_{[23,25]} green$ $\phi_4 = \square_{[29,30]} red$	Exact: 6980 con. 3 bin. Approx: 3125 con. 3 bin.	53.71 s	25.11 s	1.999 bits	1.999 bits



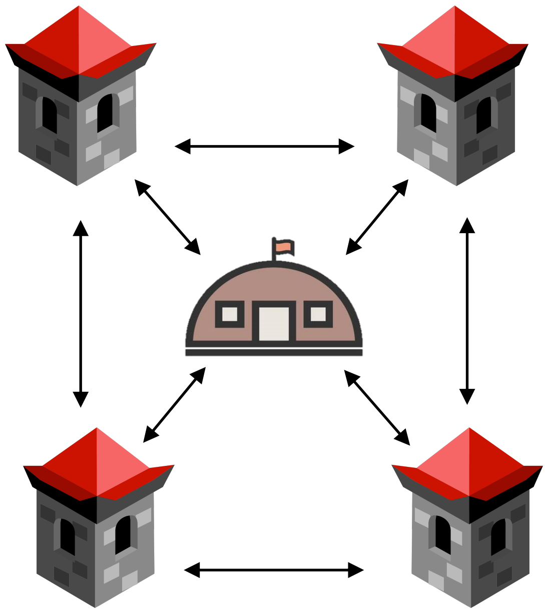
$\Gamma = 0.95, \beta = 0.80.$

For both sets of specifications, each method maximizes the uncertainty of the adversary.

Example – Patrolling a Perimeter

- One outpost is assumed to store critical information. Based on patrolling of agent, observer seeks to infer the outpost using agent's probability of visiting

- Agent must repeatedly visit each outpost over time span, attempt to do so with equal probability for each

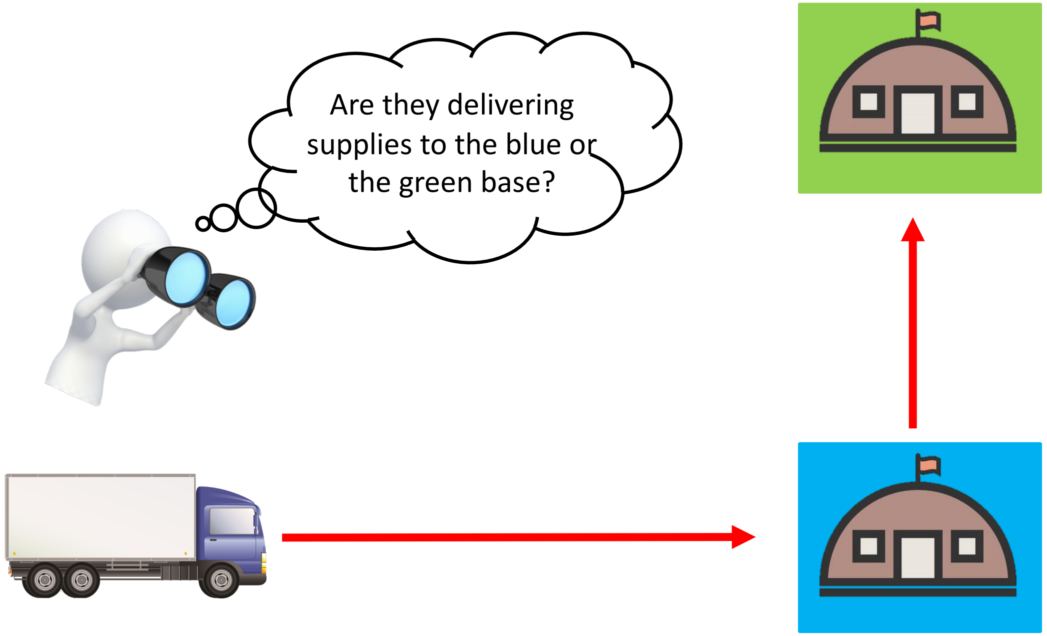


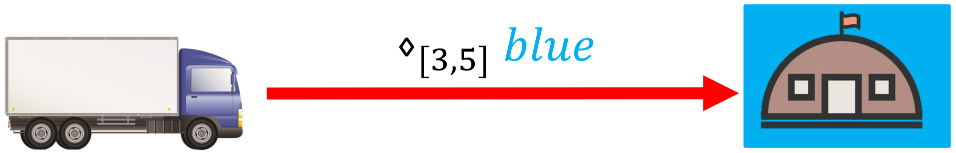
Example	Specifications ϕ	Num. Vars. (Continuous, binary)	Sol. Time, Exact	Sol. Time, approx.	$H^{\pi}(\phi_{can})$, exact	$H^{\pi}(\phi_{can})$, approx
Surveillance	$\phi^* = \square_{[1,10]} \diamond_{[0,5]} upper\ left$ $\phi_2 = \square_{[1,10]} \diamond_{[0,5]} upper\ right$ $\phi_3 = \square_{[1,10]} \diamond_{[0,5]} lower\ left$ $\phi_4 = \square_{[1,10]} \diamond_{[0,5]} lower\ right$	Exact: 22209 con. 1 bin. Approx: 617 con. 239 bin.	148.37 s	26.59 s	1.999 bits	1.999 bits

Even with a large number of binary variables (needed to utilize off-the-shelf solvers), the approximate solution method is quicker and performs as well as the exact solution method.

Summary

- Considered the problem of minimizing the ability of an observer to predict the specification that an agent *actually* cares about completing
- Developed two algorithms, an exact and an approximate solution method to synthesize a policy for the agent
  - Exact solution method gives exact satisfaction probabilities but can be cumbersome
  - Approximate solution is quicker but may underreport the set of candidate specifications





Eventually, between the third and fifth time steps, visit the blue base