# Multi-Agent Intention Recognition and Progression

Brian Logan

University of Aberdeen

# Acknowledgements

joint work with Natasha Alechina, Michael Dann,
Felipe Meneguzzi, John Thangarajah and Yuan Yao

# Outline

- action selection
- BDI agents
- intention progression in BDI agents
- multi-agent intention progression
- multi-agent intention recognition and progression

# Action selection

# What to do next

- a key problem for rational agents is '*what to do next*'

- an agent typically has multiple goals to achieve and multiple ways of achieving each goal

- at any given point the agent must decide which of its goal to work towards, and how to achieve that goal

- this can be viewed as an '*action selection problem*', e.g.:

$$Event^{+} \times State \rightarrow Action \times State$$

# Action selection is a hard problem

- the agent's goals and its environment change over time
- the presence of other agents introduces a strategic dimension
- practical agents are resource-bounded
- the amount of computation (time) an agent can spend deciding what to do is limited

# Approaches to action selection

- **(reinforcement) learning:** fast, robust, context sensitive behaviour; typically limited to one goal at a time, requires a model of the environment, sensitive to choice of reward function

- **classical AI planning:** flexible, can handle complex goals; slow, brittle, requires an action theory

- **reactive planning:** fast, (reasonably) flexible, robust and context sensitive behaviour; typically limited to one goal at a time

# Practical reasoning

- Bratman (1987) argued that *practical reasoning* can be viewed as the weighing of "*multiple, conflicting considerations for and against conflicting choices, in light of what the agent believes, desires, values and cares about*"

- practical reasoning consists of two steps:

  - **deliberation:** deciding what state(s) of affairs to bring about from the possibly conflicting desires of the agent

  - **means-ends reasoning:** deciding how to bring about that state of affairs

- these two steps interact – what goals the agent should commit to depends on the means it has available to achieve them

# Example

- agent with two goals, *r* and *u*
  - *r* can be achieved by the plan $\pi_1$ (cost 10), or the plan $\pi_2$ (cost 5)
  - *u* can be achieved by the plan $\pi_3$ (cost 10), or the plan $\pi_4$ (cost 5)
- if the agent has < 20 units of resource, its choice of plans are constrained (if it wants to achieve both goals)
- if the agent has < 10 units of resource, it can achieve at most one goal

# BDI agents

# Belief-Desire-Intention agents

- in BDI-based agent programming languages, the behaviour of an agent is specified in terms of beliefs, goals, and plans
  - **beliefs** represent the agent's information about the environment, itself and other agents
  - **goals** represent desired states of the environment the agent is trying to bring about
  - **plans** are composed of *steps* which are either *primitive actions* or *subgoals* which are in turn achieved by other plans

$$\pi_i = g : \chi \leftarrow s_1; \ldots; s_m$$

# Goal-plan trees

- we use **goal-plan trees** to represent the *relations between goals, plans and actions*

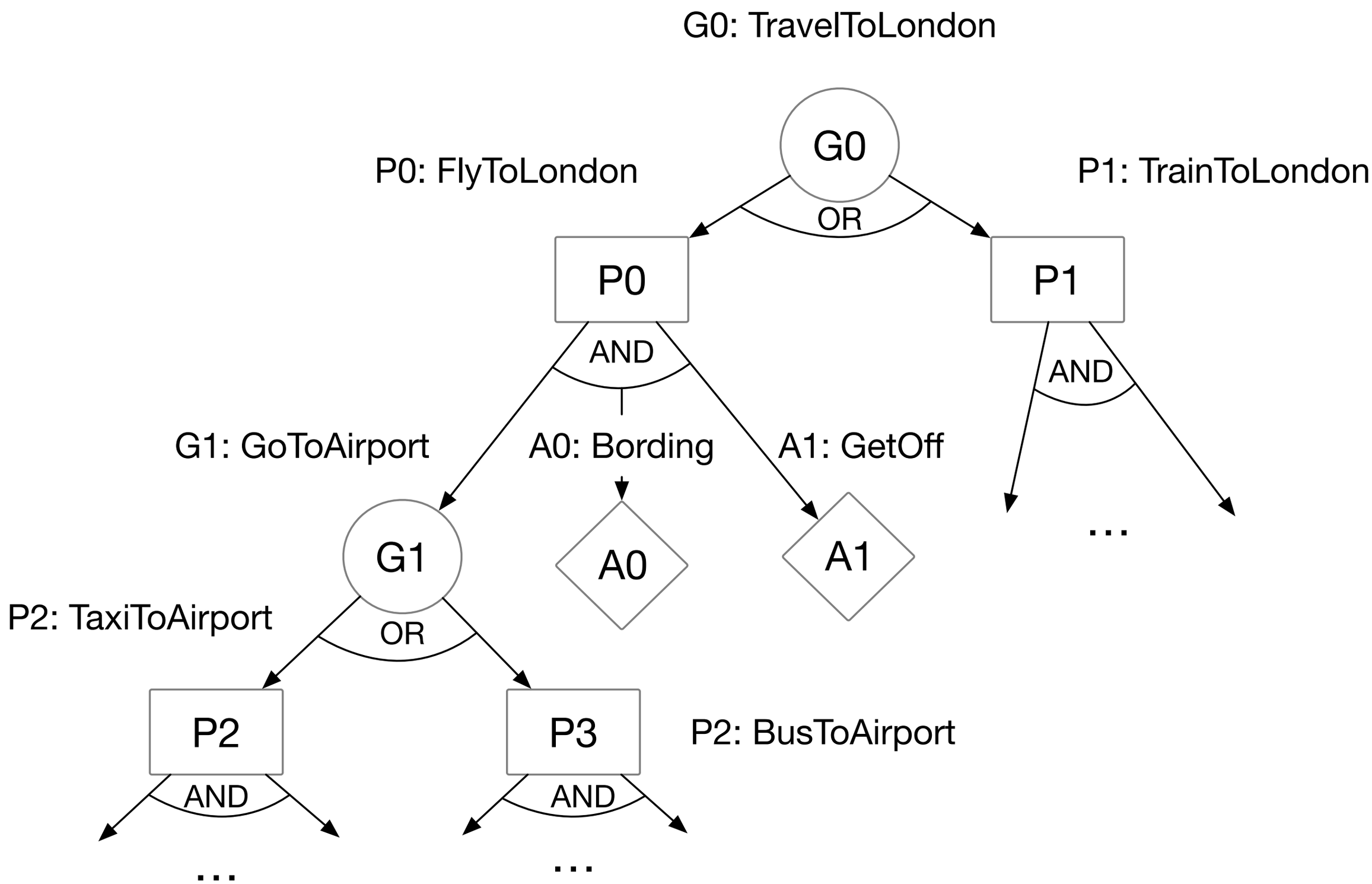
- a goal plan tree is an alternating and-or tree

  - the root is a **goal-node** (or-node) representing the top-level goal

  - children are **plan-nodes** (and-nodes) representing the plans that can be used to achieve the goal

  - children of plans are **(sub)goal nodes** and **action nodes** (or-nodes)

- a goal-plan tree represents *all possible ways an agent can achieve a goal*


G0: TravelToLondon
G0
OR
P0: FlyToLondon
P1: TrainToLondon
P0
P1
AND
AND
G1: GoToAirport
A0: Bording
A1: GetOff
G1
A0
A1
...
P2: TaxiToAirport
OR
P2
P3
P2: BusToAirport
AND
AND
...
...

# Intentions

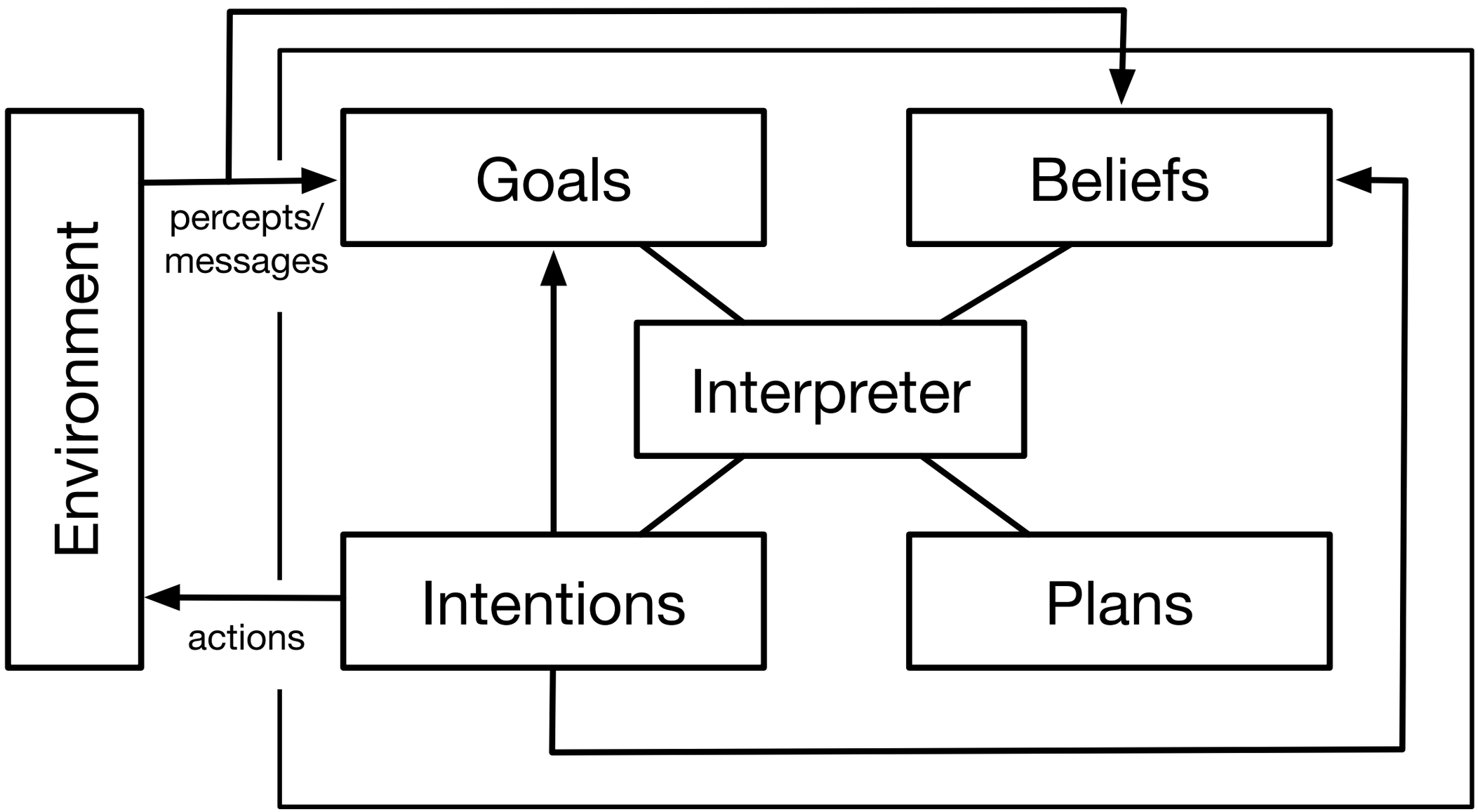
- for each top-level goal, the agent selects a plan which forms the root of an **intention**, and commences executing the steps in the plan

- if the next step in an intention is an **action**, the action is simply executed (assuming its preconditions hold in the current environment)

- if the next step in an intention is a **subgoal**, a (sub)plan is selected to achieve the subgoal and added to the intention, and the steps in the (sub)plan are then executed and so on

- an intention can be thought of as a *stack of partially executed plans*

# BDI architecture



# Intention progression problem

- in the BDI approach, *'what to do next'* reduces to the problem of **intention progression**:
  - which intention to execute (progress) next; and
  - if the next step in the selected intention is a (sub)goal, which of the plans for this goal should be used to achieve it
- to produce an interleaving of the agent's intentions that maximises the agent's utility, e.g., expected utility, number of goals achieved, etc …
- combines *deliberation* (which goals get achieved) and *means-ends reasoning* (how the goals are achieved)

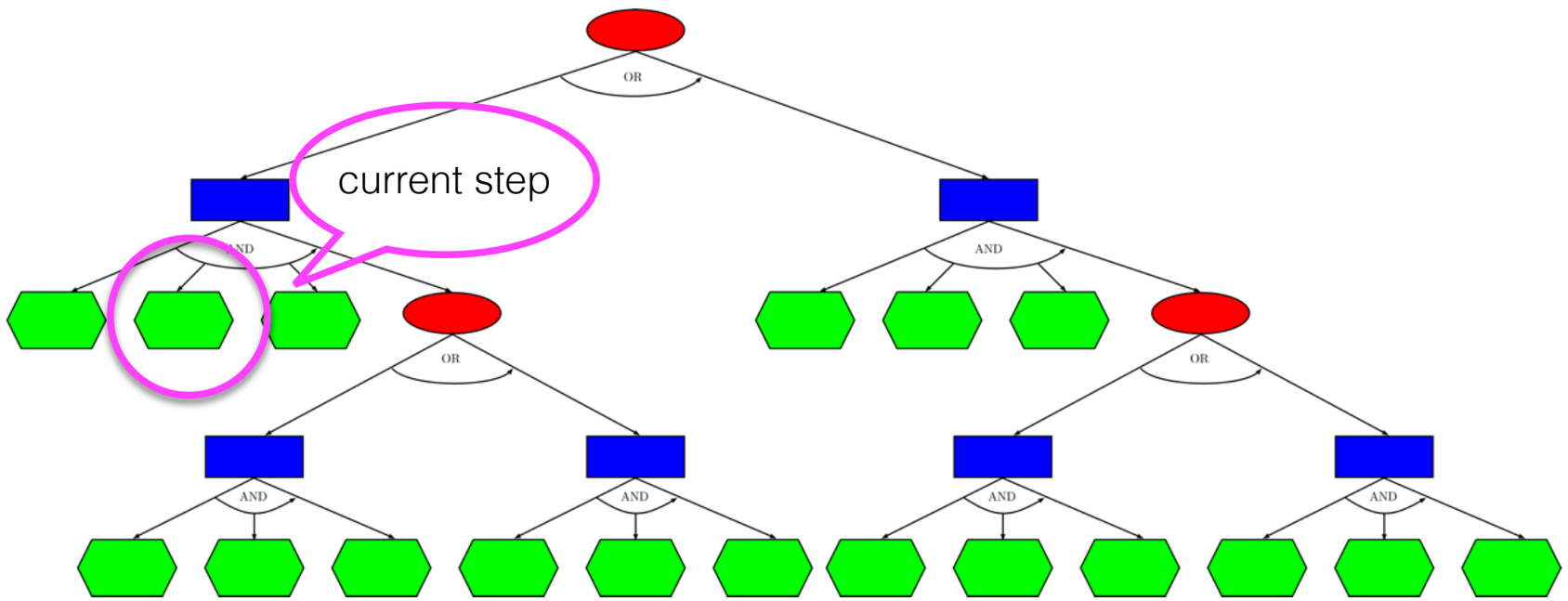
# Representing intentions

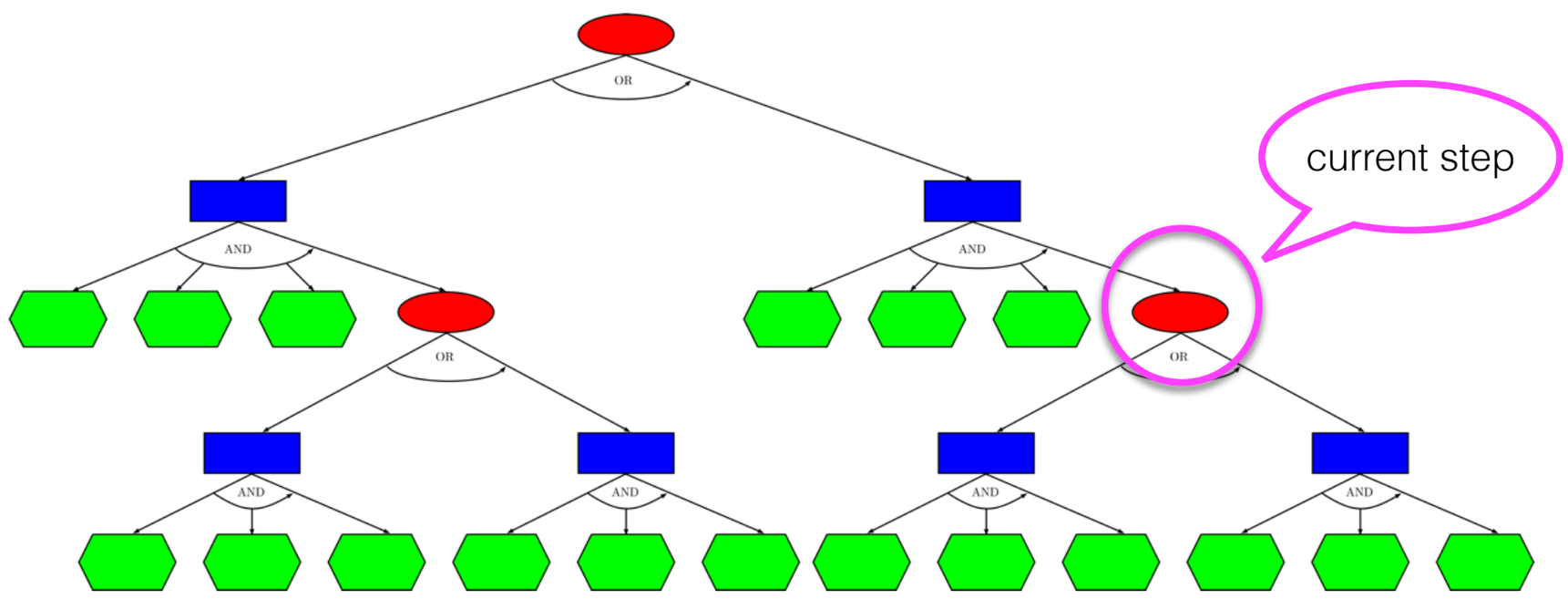
- we represent the agent's intentions as a set of goal-plan trees $T = \{t_1, \ldots, t_n\}$ corresponding to the agent's top-level goals $\{g_1, \ldots, g_n\}$

- the progression of an intention to achieve a top-level goal $g_i$ amounts to **choosing a path** through the goal-plan tree $t_i$ for goal $g_i$

- path specifies a sequence of plans, actions, subgoals and sub-plans that will be executed to achieve $g_i$

- execution of an agent program corresponds to an **interleaving** of paths through (possibly a subset of) $t_1, \ldots, t_n$




which intention should we progress next?

# Conflicts between intentions

- in many BDI agent architectures, the plans comprising the agent's intentions are **executed in parallel**,
- e.g., by default, many BDI architectures execute one step of an intention at each cycle in round robin (RR) fashion
- interactions between interleaved steps in different intentions may result in **conflicts**
- interactions can also result in **synergies,** e.g., where two intentions share a common subgoal – with appropriate interleaving the goal need only be achieved once [not in this talk]

# Example: (single agent) intention conflicts

- agent with two goals, *r* and *u*

- *r* can be achieved by the plan $\pi_1 = a_1; a_2$

  action $a_1$ has precondition *p* and action $a_2$ has postcondition $\neg q$

- *u* can be achieved by the plan $\pi_2 = b_1; b_2$

  action $b_1$ has precondition *q* and action $b_2$ has postcondition $\neg p$

- achieving *r* and *u* requires **appropriate scheduling** of plan steps, e.g.,

$$a_1, b_1, a_2, b_2$$

# Implementing autonomous decision-making

# Solving the intention progression problem

- approach based on Monte-Carlo Tree Search (MCTS)

- MCTS is a best-first search in which pseudorandom simulations are used to guide expansion of the search tree

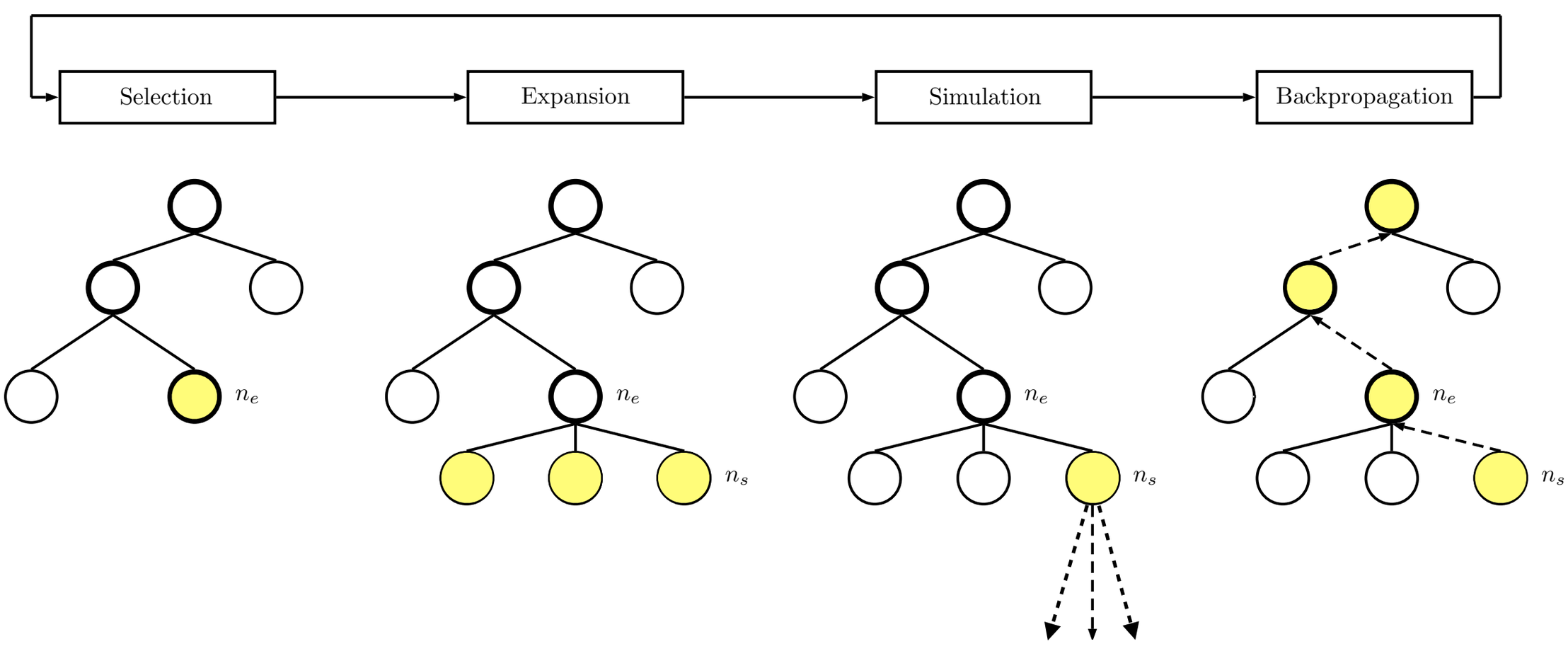
- **nodes** represent a partial **interleaving** of steps from (a subset of) the goal-plan trees in *T*

  - each node records the state of the agent's environment and the agent's utility (e.g., goals achieved) resulting from the interleaving

- **edges** represent the selection of a plan for a subgoal or the execution of primitive action in a plan

# MCTS algorithm

- four phases: **selection**, **expansion**, **simulation**, and **backpropagation**



# MCTS phases

- **Selection:** starting from the root node, we recursively follow child nodes with the highest UCT value until a leaf node $n_e$ is reached (tree policy)

- **Expansion:** $n_e$ is expanded by adding child nodes representing all moves possible from the state represented by $n_e$

- **Simulation:** one of the newly created child nodes, $n_s$, is selected at random, and the *value* of $n_s$ is estimated by performing $\beta$ pseudorandom simulations (random interleavings from $n_s$ – default or simulation policy)

- **Back-propagation:** the estimated value of $n_s$ is then back-propagated all nodes on the path to the root node

- after $\alpha$ iterations, the action leading to the *best* child of the root node is returned

# Exploration vs exploitation

- choice of node to expand in the selection phase is usually based on some form of UCT (Upper Confidence Bound applied to Trees)

$$UCT = \overline{X}_j + 2C_p \sqrt{\frac{2ln\ n}{n_j}}$$

- where $n$ is the number of times the current node has been visited, $n_j$ is the number of times child $j$ has been visited, $X_j$ is the average reward from child $j$, and $C_p > 0$ is a constant

- the reward term $X_j$ encourages **exploitation** of high reward choices, while the right hand term encourages **exploration** of less visited choices

# Anytime behaviour

- MCTS is an **anytime** algorithm, i.e., it is guaranteed to return solutions of monotonically increasing quality with time

- the number of node expansions $\alpha$ and simulations to perform $\beta$ when choosing the next action to execute are parameters

- together $\alpha$ and $\beta$ define a *computational budget* that can be varied at runtime, e.g., in response to the dynamism of the environment

- agent effectively samples the space of possible interleaving until the computational budget is reached, and returns the next step in the best interleaving found

# Integrating MCTS with BDI agent programs

- MCTS merges *intention selection* and *plan selection* into a single process
- determines which plans should be selected for a (sub)goal and how the plans should be interleaved so as to maximise the agent's utility
- agent developer no longer has to:
  - define plan selection and/or intention selection functions (e.g., $S_O$, $S_I$)
  - anticipate and control possible conflicts between intentions using *atomic constructs*

# MCTS variants

- basic idea can be applied in a range of different settings
- static environment (Single-Player Monte Carlo Tree Search)
- non-deterministic/dynamic environments (games against the environment)
- multi-agent environments (strategic games) [this talk]

# Intention progression in multi-agent settings

# Intention-aware intention progression

- in a multi-agent setting, how an agent progresses its intentions has implications for both the achievement of its **own goals** and the **goals of other agents**:

- e.g., when the execution of a step in a plan of the agent makes the execution of a step in a plan of another agent impossible

- in a multi-agent setting, to progress its intentions effectively, an agent must be able to **predict the actions of other agents**

- how to predict the actions of other agents?

# Example: multi-agent intention conflicts

- two agents with goals, *r* and *u*

- *r* can be achieved by the plan $\pi_1 = a_1; a_2$

  action $a_1$ has precondition *p* and action $a_2$ has postcondition $\neg q$

- *u* can be achieved by the plan $\pi_2 = b_1; b_2$

  action $b_1$ has precondition *q* and action $b_2$ has postcondition $\neg p$

- achieving *r* and *u* requires **appropriate scheduling** of plan steps by **both agents**, e.g.,

$$a_1, b_1, a_2, b_2$$

# $I_A$: Intention-aware intention progression

- $I_A$ is an "**intention-aware**" a multiplayer MCTS-based scheduler (Dann et al 2020)

- in $I_A$, the objectives of each agent are represented by a payoff matrix, where each element, $p_{ij}$ , is the assumed payoff that agent *i* receives on completion of a goal of agent *j*

- (some) of the top-level goals and goal-plan trees of all agents are assumed to be common knowledge, e.g., the agents are co-designed, or have the same agent program

- $I_A$ is able to effectively schedule intentions when paired with agents using a range of intention scheduling approaches (single-agent MCTS, RR, FIFO)

# $I_B$: Egocentric intention progression

- $I_A$ assumes that the programs (goal-plan trees) of the other agents are known
- $I_B$ is an "**egocentric**" MCTS-based scheduler which uses an abstraction of its own program to predict the actions of other agents (Dann et al 2021)
- the $I_B$ agent's program is abstracted into a **partially ordered goal-plan tree** which encodes *all possible executions of plan steps* that do not violate action preconditions
- the goal-plan trees of the other agents are not known, but are assumed to be similar to those of the $I_B$ agent
- $I_B$ is able to effectively schedule intentions when paired with agents which are not co-designed

# $I_{RM}$: Intention progression with reward machines

- $I_A$ and $I_B$ assume that the other agents in the environment are BDI agents
- $I_{RM}$ agents predict the actions of other agents based on **high-level declarative specifications** of the tasks performed by an agent, e.g., Linear Time Temporal Logic (LTL) formulas (Dann et al 2022)
- LTL task specification is converted to a **reward machine** [Toro Icarte et al ICML 2018]
- reward machines are *Mealy machines* (automata with outputs) that encode the *logical structure* of a task
- $I_{RM}$ agents can effectively inter-operate with *learning-based*, *model-based* and *BDI agents*

# Multi-agent intention progression

- in summary, previous work on multi-agent intention progression has assumed either:
  - **co-designed agents:** agents have access to the plans used by other agents to achieve their goals ($I_A$) (Dann et al AAMAS 2020)
  - **egocentric agents:** agents predict the actions of other agents based on an *abstraction* of their own BDI program ($I_B$) (Dann et al IJCAI 2021)
  - **reward machine-based agents:** agents predict the actions of other agents based on a high-level specification of the tasks performed by an agent in the form of a reward machine ($I_{RM}$) rather than on its (assumed) program (Dann et al IJCAI 2022)
- however, all these approaches assume that the current intentions of the other agents are known …

# Multi-agent intention recognition and progression

# Multi-goal recognition problem

A **multi-goal recognition problem** $P_{MG}$ is a tuple $\langle \Xi, s_0, G, \Omega \rangle$:

- $\Xi = \langle F, A \rangle$ is the planning domain
- *F* is a set of fluents
- Act is a set of actions
- $s_0$ is the initial state
- *G* is the set of possible goals
- $\Omega = [s_0, a_0, s_1, a_1, \ldots, s_n, a_n]$ is a sequence of observations, where $s_i \in S$ and $a_i \in Act$

A *solution* is a set of goals $G' \subseteq G$. A solution is *correct* if $G' = G^*$ where $G^*$ is the set of current goals of the agent that generated the observations

# Multi-agent intention recognition and progression

A *multi-agent intention recognition and progression problem* is a tuple $P_{RP} = \langle \{P_{MG_i} \mid i \in Agt\}, B, T, C \rangle$ where:

- $B$ are the agents current beliefs
- $T, C$ are the agent's current intentions

A solution to a multi-agent intention recognition and progression problem $P_{RP}$ is a policy $\Pi(P_{RP})$, that, at each deliberation cycle, selects a current step $c_i \in C$ to progress so as to maximise some overall utility function $U^{\Pi}(P_{RP})$:

$$\Pi(P_{RP}) = c_i \text{ and } \nexists\ \Pi' \text{ s.t. } U^{\Pi'}(P_{RP}) > U^{\Pi}(P_{RP})\,.$$

# $I_{GR}$: intention recognition and progression

- $I_{GR}$ uses multi-goal recognition to infer the possible goals of other agents and MCTS to predict their behaviour given the inferred goals

- $I_{GR}$ builds on Goal Recognition as Reinforcement Learning (GRAQL) [Amado et al 2022] which does single goal recognition

- we take the set of inferred current goals, $G'$, to be those goals *whose KL divergence is within some threshold*, $\delta$, of the goal with the minimum KL divergence:

$$G' = \{g \in G \mid KL(\Omega, Q_g) \leq \min_{\hat{g} \in G} KL(\Omega, Q_{\hat{g}}) + \delta\}$$

- we use an *exponential moving average of the KL divergence* which is more sensitive to recent observations

# Evaluation: scenario

- two agent version of the Craft World environment
- agents must gather or craft goal items
- for example, to make an axe, the agent must gather iron and a stick and craft them into an axe in a toolshed
- agents have six possible actions: *movement* in four directions, *collect* and *craft* (actions are assumed to be fully observable)
- goal-plan trees were generated algorithmically for each goal item

# Evaluation: baselines

- we evaluated the performance of $I_{GR}$ when paired with three different agent types:
  - **Q-learn:** a DQN agent trained in a single-agent version of the Craft World environment
  - ***$S_P$*:** single agent scheduler based on single-player Monte Carlo Tree Search (Yao et al 2016)
  - ***$I_{RM}$*:** multi-agent scheduler which assumes *a priori* knowledge of the other agent's goals (Dann et al 2022)
- we would expect $I_{GR}$*'s* performance to be somewhere between that of $S_P$ and $I_{RM}$

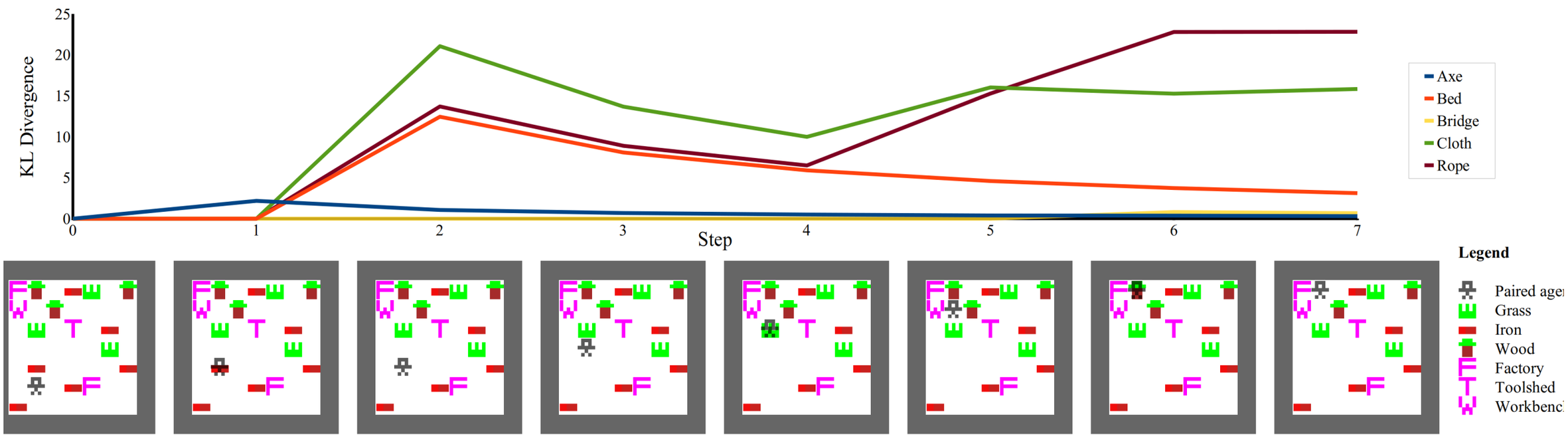
# Evaluation: agents & goals

- each scenario consisted of two agents:
  - **blue agent:** agent predicting the actions of the other agent ($I_{RM}$, $I_{GR}$)
  - **red agent:** agents whose actions are being predicted (Q-learn, $S_P$, $I_{RM}$, $I_{GR}$)
- the goals of each agent depended on the scenario
- the possible goals of the red agent were assumed to be a subset of its actual goals

# Evaluation: agent relationships & goals

- we considered three types of relationships between agents:
  - **allied:** blue agent seeks to maximise the achievement of both agents goals
  - **selfish:** blue agent seeks to maximise the achievement of its own goals
  - **adversarial:** blue agent seeks to maximise achievement of its own goals while preventing the red agent from achieving its goals

# Example: goal recognition




Change in KL divergence over time

# Results

- $I_{GR}$ out-performs Q-learn and $S_P$ in all settings (allied, selfish and adversarial ).
- $I_{GR}$ performs nearly as well as $I_{RM}$ which has complete a priori knowledge of the other agent's current goals.
- $I_{GR}$ still performs well (and can out-perform $I_{RM}$) if it doesn't know all the other agent's possible goals.
- time spent on goal recognition is negligible (less than a millisecond) compared to the time spent on MCTS rollouts.

# References


Yuan Yao, Brian Logan, and John Thangarajah (2014). SP-MCTS-based Intention Scheduling for BDI Agents. (ECAI-2014).

Yuan Yao, Brian Logan, and John Thangarajah (2016). Robust Execution of BDI Agent Programs by Exploiting Synergies Between Intentions. (AAAI-16).

Yuan Yao and Brian Logan (2016). Action-Level Intention Selection for BDI Agents. (AAMAS 2016)

Yuan Yao, Brian Logan, and John Thangarajah (2016). Intention Selection with Deadlines. (ECAI-2016).

Sam Leask, Natasha Alechina, and Brian Logan (2018). A Computationally Grounded Model for Goal Processing in BDI Agents. (GR 2018).

Yuan Yao, Natasha Alechina, Brian Logan, and John Thangarajah (2020). Intention Progression under Uncertainty. (IJCAI 2020).

Michael Dann, John Thangarajah, Yuan Yao, and Brian Logan (2020). Intention-Aware Multiagent Scheduling. (AAMAS 2020).

Michael Dann, Yuan Yao, Brian Logan, and John Thangarajah (2022). Multi-Agent Intention Progression with Reward Machines. (IJCAI-ECAI 2022)

Michael Dann, Yuan Yao, Natasha Alechina, Brian Logan, Felipe Meneguzzi and John Thangarajah (2023). Multi-Agent Intention Recognition and Progression. (IJCAI 2023)

Yuan Yao, Natasha Alechina, Brian Logan (2024) *Intention Progression with Temporally Extended Goals*. (IJCAI 2024)

# Intention Progression Competition

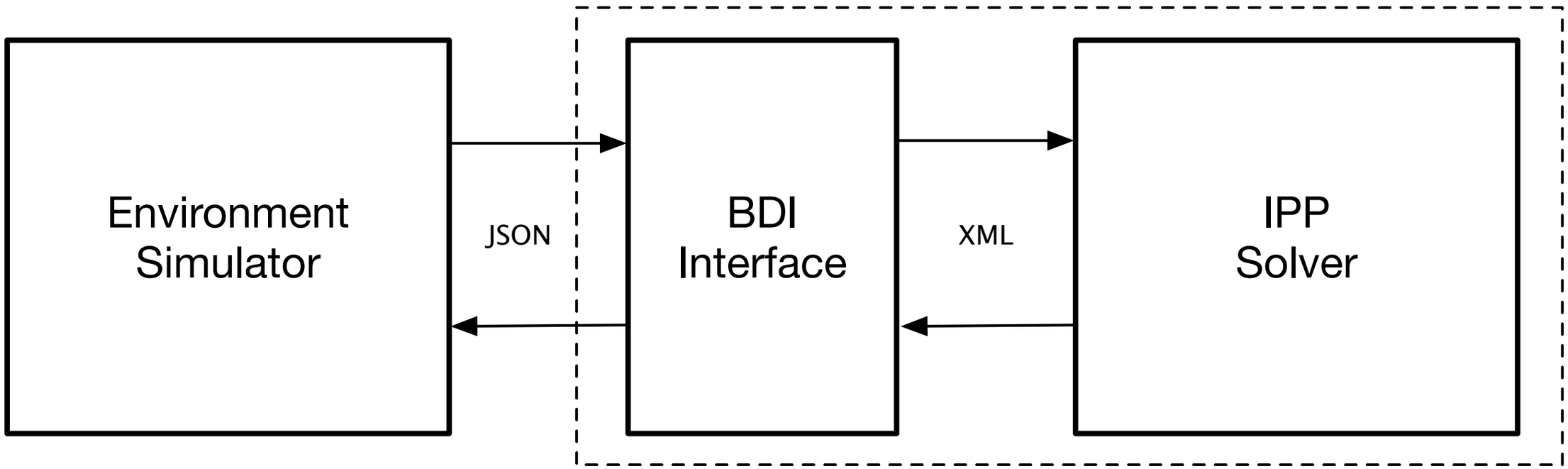
- in a blue skies paper at AAMAS'17, we proposed an **intention progression competition** to incentivise research around the IPP

- inspired by competitions such as the international planning competition:

  - submissions take the form of executable code

  - evaluated on a set of (unseen) benchmark intention progression problems

# IPC software architecture



# IPC 2020

- the first edition of the Intention Progression Competition was announced at EMAS/AAMAS 2019
- registrations opened in July/August 2019
- competition platform available September 2019
- calibrated test problems released in January 2020
- final entries were due in March 2020 with winners to be announced at AAMAS 2020

# IIPC 2023

- the first edition of the Intention Progression Competition was announced at EMAS/AAMAS 2019
- in 2023 we relaunched the IIPC
- competition software has been re-packaged in a Docker container
- registration and software download available at:

intentionprogression.org

# Participating in the IPC

- if you have questions email:
  - Rafael Cordoso (rafael.cordoso@abdn.ac.uk) and/or
  - me (brian.logan@abdn.ac.uk)
- register for the competition at

intentionprogression.org