Multi-tier Automated Planning for Adaptive Behavior

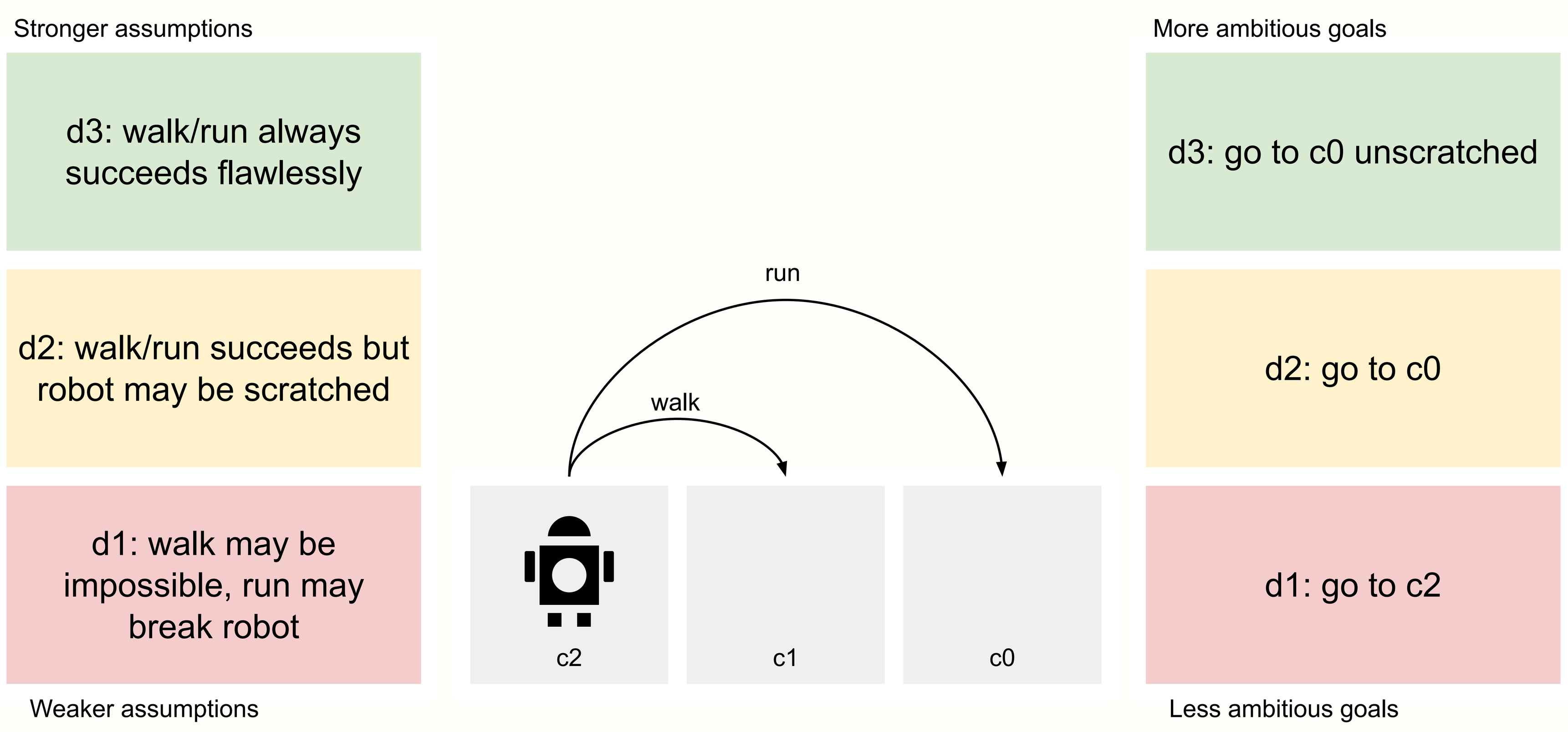
Daniel Ciolek, Nicolás D'Ippolito, Alberto Pozanco, Sebastian Sardiña
Universidad de Buenos Aires, Universidad Carlos III de Madrid, RMIT University

In 30^{th} *International Conference on Automated Planning and Scheduling*, 2020.

Multi-tier Planning

A novel approach to synthesize a policy capable of coping with the progressive violation of environment assumptions, for which the agent "gracefully degrades" to less refined models with (in general) less ambitious goals.

Example: A robot starting on the left of a 1×3 grid should reach the right-end unscratched when possible.



A plan must strive for the best possible solution, but always remain open to a potential violation of the assumptions. In particular:

1. The robot should start assuming the best scenario and try to reach the right-end unscratched;
2. If an obstacle along the way scratches the robot, the goal needs to be degraded to just reach the right-end; and
3. If the corridor is blocked the goal should be degraded to go back to the starting point (i.e., never breaking the robot).

Multi-tier Problem Definition

A *multi-tier planning domain* (MTD) is a tuple $(\Omega, \leq)$ such that:

1. Ω is a set of FOND planning domains over the same variables V and operator signatures, and every operator has the same preconditions across all domains in Ω;
2. $\leq$ is a partial-order relation over Ω such that $\mathcal{D}_1 \leq \mathcal{D}_2$ implies $Ex(\mathcal{D}_2, s) \subseteq Ex(\mathcal{D}_1, s)$ for all states $s \in S$, where $Ex(\mathcal{D}, s)$ denotes the set of possible executions in $\mathcal{D}$ from state s; and
3. $\leq$ has a greatest element in Ω, denoted $\hat{D}$, as well as a minimum element.

A *multi-tier planning problem* (MTP) is a tuple $\mathcal{M} = ((\Omega, \leq), s_I, \mathcal{G})$ where $(\Omega, \leq)$ is an MTD, s_I is the initial state, and $\mathcal{G}$ is a function mapping each domain $\mathcal{D}$ in Ω to a goal $\mathcal{G}_\mathcal{D}$ (i.e., one goal at each tier).

A *multi-tier controller* (MTC) for an MTD $(\Omega, \leq)$ is a function $\mathcal{C} : \Omega \mapsto (S \mapsto 2^O)$ mapping each domain $\mathcal{D} \in \Omega$ to a specific policy $\mathcal{C}_\mathcal{D}$.

An MTC $\mathcal{C}$ is a *solution controller* for an MTP $\mathcal{M} = ((\Omega, \leq), s_I, \mathcal{G})$ *iff* for every domain $\mathcal{D} \in \Omega$, the projected policy $\mathcal{C}_\mathcal{D}$ is a solution plan for planning problem $(\mathcal{D}, s, \mathcal{G}_\mathcal{D})$, for every state $s \in Init(\mathcal{D}, \mathcal{C})$.

Unlike standard planning, this problem requires each policy to work from more than one initial state ($Init(\mathcal{D}, \mathcal{C})$), to account for the degradations that can occur in different states.

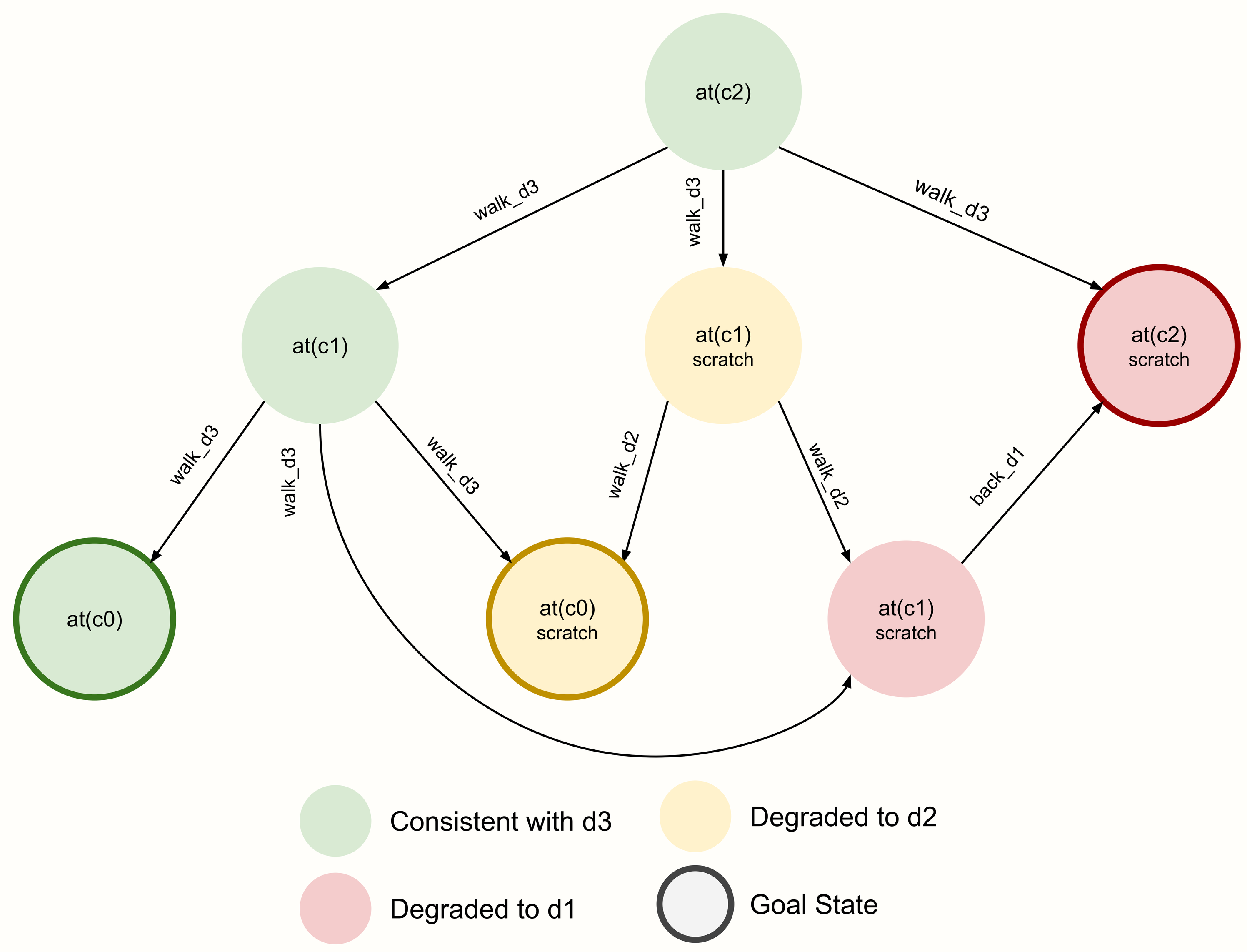
Dual FOND Compilation

A Fully Observable Non-Deterministic (FOND) planning problem (Rintanen 2008, Gervini, et.al. 2006) allow us to consider planning domains in which actions may have observable but non-deterministic effects.

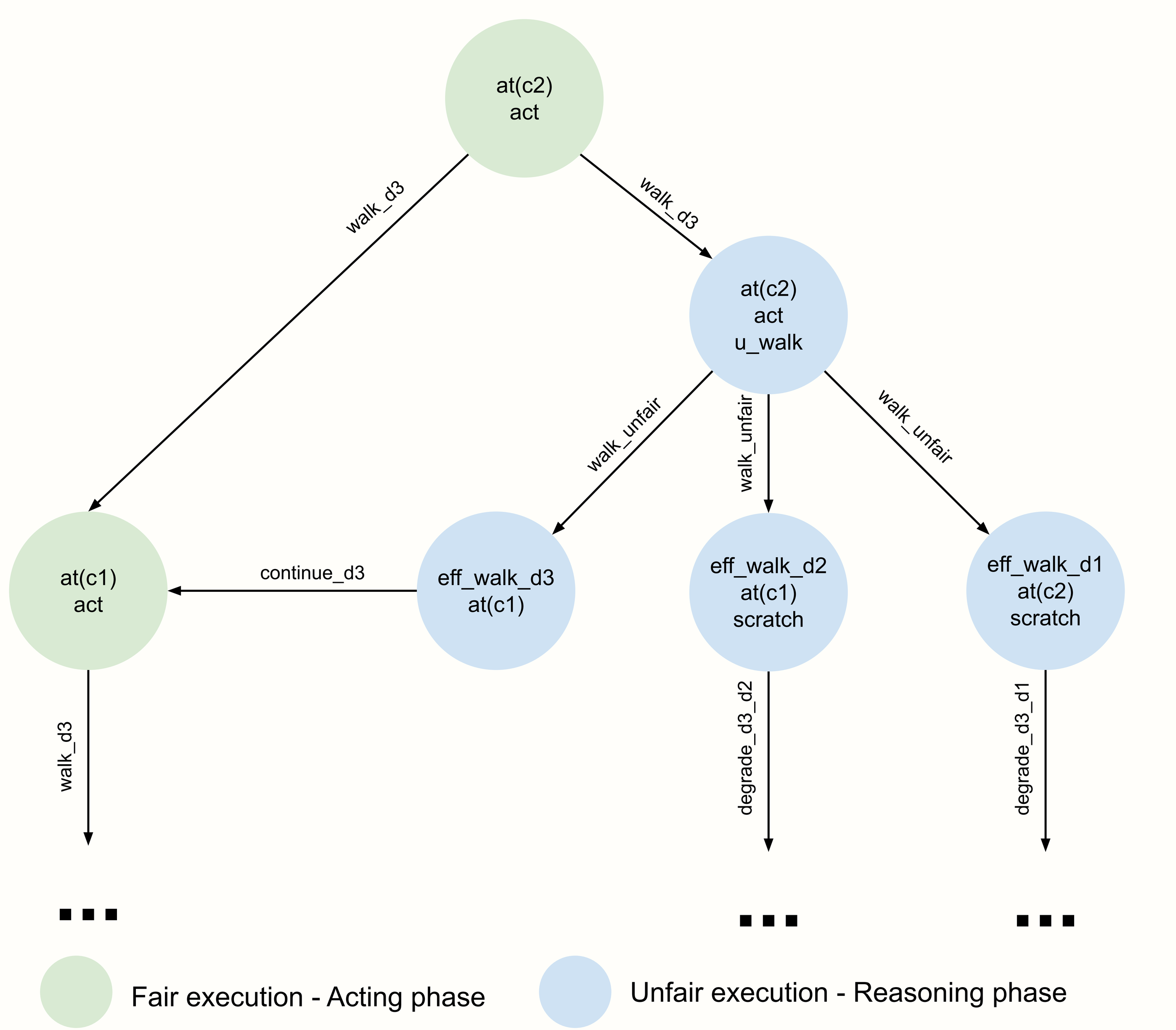
There has been several solution concepts for FOND planning depending on the "fairness" of non-deterministic actions (Geffner and Bonet 2013, Sardina and D'Ippolito 2015). A fair action is one in which all effects occur infinitely often when the action is executed infinitely many times.

The Dual FOND hybrid variation has recently been introduced (Camacho and McIlraith 2016; Geffner and Geffner 2018) to deal with domains that have both fair and unfair actions.

We show how to compile an MTP as a Dual FOND planning problem. Using our compilation we synthesized a MTC for the robot example using the FOND-SAT system. The resulting controller effectively degrades when necessary and is able to achieve the appropriate goal of each tier.



This (arguably simple) example takes 600 seconds to produce a controller in an i7-4510 CPU with 8GB of RAM. The compilation to Dual FOND requires the policy to include 22 additional intermediate decision states.



While it is possible to solve MTPs using existing planning technology, our results indicate the need for better Dual FOND implementations or the development of specialized optimizations for MTPs.