EFP 2.0: A Multi-Agent Epistemic Solver with Multiple e-State Representations

Francesco Fabiano[1], Alessandro Burigana[1], Agostino Dovier[1], Enrico Pontelli[2]

[1]University of Udine - Udine, Italy [2]New Mexico State University - Las Cruces, NM, U.S.A.

Multi-Agent Epistemic Reasoning

Reasoning about actions and information has been one of the prominent interests since the beginning of the AI [6].

In particular, our work studies a family of problems recently considered in the automated reasoing scenario [7]. That is, the *Multi-agent Epistemic Planning problem* (MEP) that, differently from most of the other approaches, is not only interested in the state of the world but also in the *knowledge* or *beliefs* of the agents.

Epistemic reasoning, initially formalized by logicians in the early sixties, rapidly evolved into *Dynamic Epistemic Logic*, a formalism used to reason not only on the state of the world but also on *information change* in dynamic domains.

As discussed in [7]: "***information*** **is something that is relative to a subject who has a certain perspective on the world, called an *agent*, and that is meaningful as a whole, not just loose bits and pieces. This makes us call it *knowledge* and, to a lesser extent, *belief*.**"

A New Epistemic-State Representation

Reasoning about beliefs is not as direct as reasoning on the "physical" state of the world. One of the main issues is that expressing belief relations between agents often implies to consider nested and group beliefs that are not easily extracted from the state description by a human reader. This inherent complexity is reflected in computational overhead that brings, most of the time, infeasibility to the solving process. That is why, in this work, we present an Epistemic Forward Planner integrated with a new *epistemic state* representation called ***Possibilities***.

Possibilities (firstly introduced in [4]) are *non-well-founded* data structures that corresponds with a whole class of *bisimilar* Kripke structures.

Possibilities

Let $\mathcal{AG}$ be a set of agents and $\mathcal{F}$ a set of propositional variables:

- A *possibility* u is a function that assigns to each propositional variable $f \in \mathcal{F}$ a truth value $u(f) \in \{0, 1\}$ and to each agent $ag \in \mathcal{AG}$ an information state $u(ag) = \sigma$.
- An *information state* σ is a (non-well-founded) set of possibilities.

Each possibility u contains both an *interpretation of the world* and of each *agent's beliefs*. That is, the component $u(f)$ assigns a truth value to the fluent f in u, while $u(ag)$ represents the (non-well-founded) set of possibilities that could be true w.r.t. the agent ag.

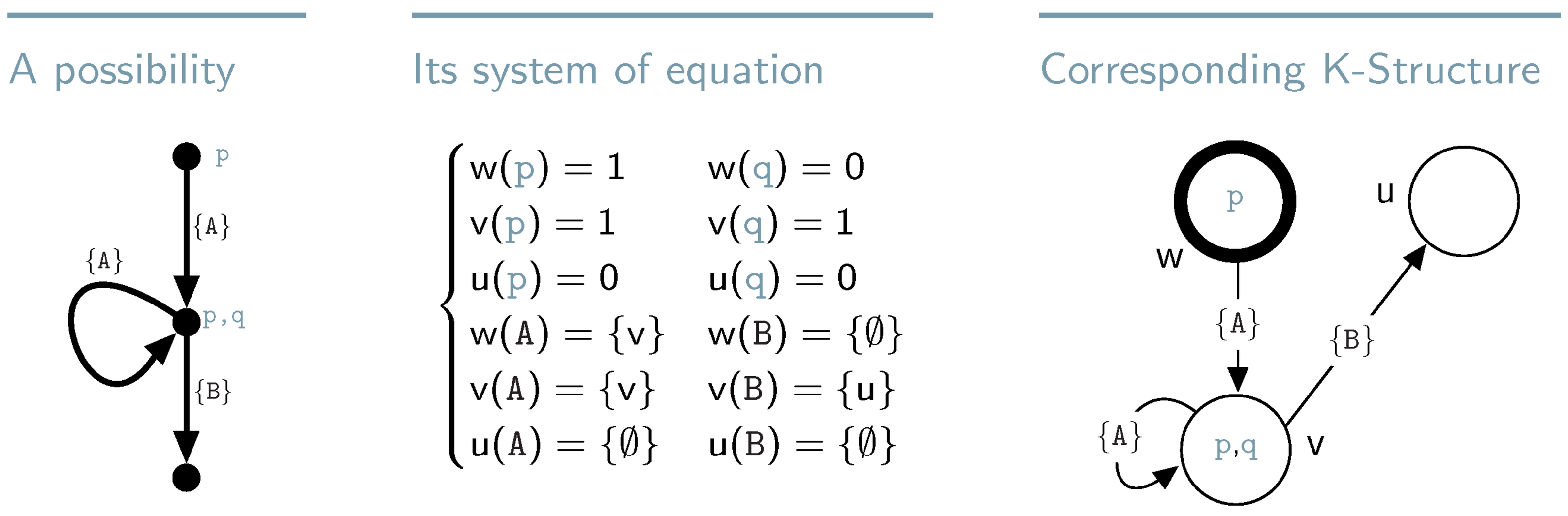

Figure 1: Transition from a possibility to a Kripke structure

An updated Transition Function

As first main contribution we presented the *formalization of a new transition function* for the action language $m\mathcal{A}^\rho$, an epistemic action language initially introduced in [3]. The updated epistemic action language borrows its syntax from $m\mathcal{A}^*$ [1] but changes the underlying e-state representation from Kripke structures to possibilities.

In particular, as $m\mathcal{A}^*$, $m\mathcal{A}^\rho$ distinguishes between three types of actions:

- **Ontic** actions: used to modify certain properties (*i.e.*, fluents) of the world
- **Sensing** actions: used by an agent to refine her beliefs about the world
- **Announcement** actions: used by an agent to affect the beliefs of other agents

Moreover, the languages also identify three possible levels of observability for an agent w.r.t. to an action a:

- **Fully observant** if ag knows about the execution of a and about its effects on the world
- **Partially observant** if ag knows about the execution of a but she does not know how a affected the world
- **Oblivious** if ag does not know about the execution of a

The transition function formalized in our work (the details are available in the paper), is more compact and, therefore, more understandable than the original one introduced in [3].

The "simplicity" of the e-states update formalization is reflected in a much cleaner and faster implementation and allowed us to formally demonstrate that $m\mathcal{A}^\rho$ can be used for multi-agent epistemic reasoning. In particular, we ensured that any planner based on $m\mathcal{A}^\rho$ satisfies the following propositions, that fully capture the concept of beliefs update:

- If an agent is fully aware of the execution of an action instance then her beliefs will be updated with the effects of such action execution
- An agent who is only partially aware of the action occurrence will believe that the agents who are fully aware of the action occurrence are certain about the actions effects
- An agent who is oblivious of the action occurrence will also be ignorant about its effects.

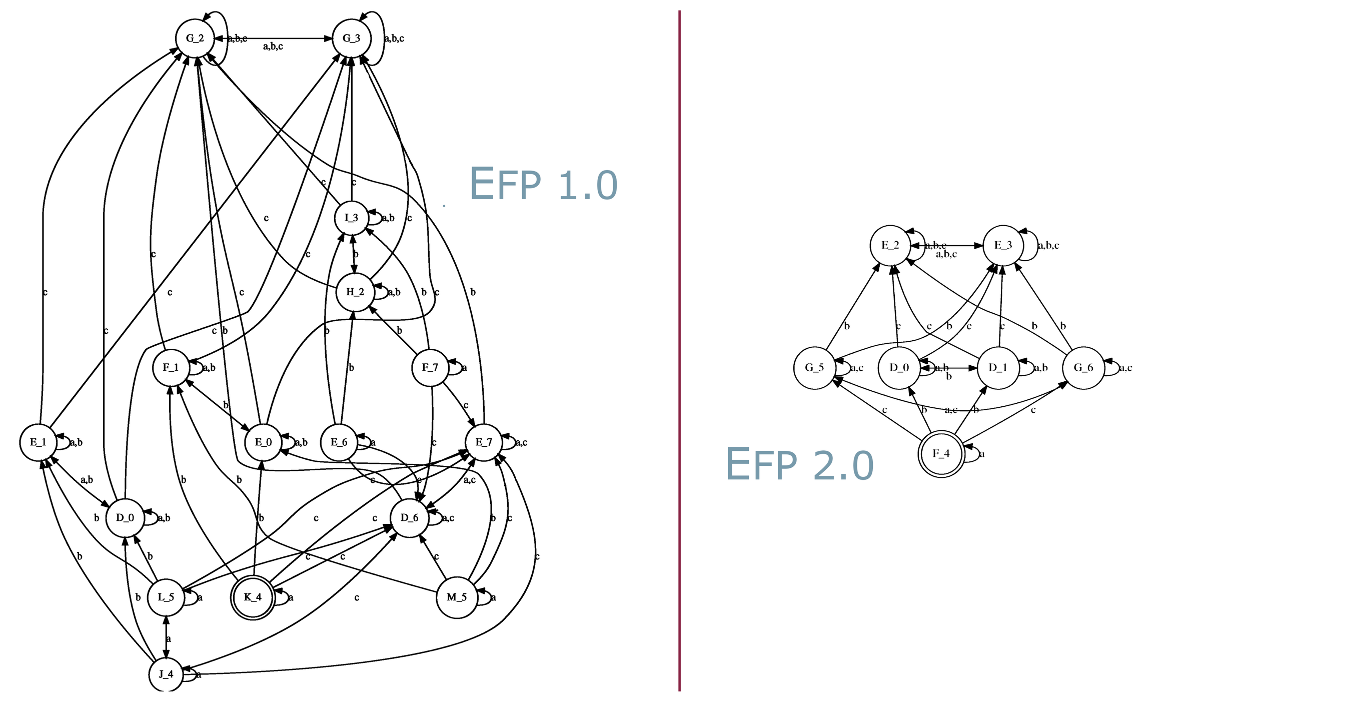

Figure 2: e-states' size generated following the semantics of $m\mathcal{A}^*$ (left) and of $m\mathcal{A}^\rho$ (right) after a plan of length four.

A Comprhensive Epistemic Planner: EFP 2.0

EFP 2.0 is a comprehensive **E**pistemic **F**orward **P**lanner derived from a complete refactoring of EFP 1.0 [5]. Let us briefly list the main characteristics of EFP 2.0:

- The planning process executed by EFP 2.0 is a *breadth-first search*
- Allowing for a multiple e-state representation, is able to reason on both $m\mathcal{A}^*$ (based on Kripke structures) and $m\mathcal{A}^\rho$(based on possibilities)
- Integrates a *Kripke structures size reduction* following the algorithm introduced in [2]. That is, starting from a generic Kripke structure, EFP 2.0 computes the *bisimilar* state with minimal size.
- Finally, EFP 2.0 introduces the concept of "*already visited e-state*"

Results

EFP 1.0 = planner of [5]
K-MAL = EFP 2.0 + K. structures
K-OPT = K-MAL + e-state reduction
P-MAR = EFP 2.0 + possibilities
TO = Time Out (25 m)
WP = Wrong Plan
-NV = w/o visited check

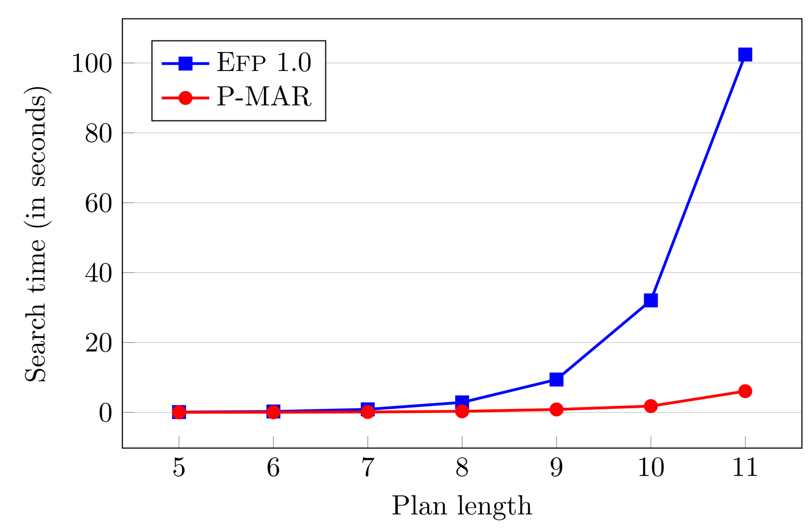

Figure 3: Comparison between EFP 1.0 and EFP 2.0's best configuration on **SC**

Grapevine									
$\|\mathcal{AG}\|$	$\|\mathcal{F}\|$	$\|\mathcal{A}\|$	L	K-MAL−NV	K-MAL	K-OPT−NV	K-OPT	P-MAR−NV	P-MAR
3	9	24	2	.09	**.09**	**.14**	.15	.03	**.02**
			4	9.19	**8.13**	10.20	**9.95**	1.34	**1.25**
			5	94.49	**75.32**	84.07	**75.87**	8.67	**7.71**
			6	372.64	**278.93**	291.62	**230.69**	27.63	**20.26**
4	12	40	2	1.85	**1.786**	**2.33**	2.34	**.17**	.18
			4	403.11	**274.53**	205.00	**152.07**	13.49	**7.31**
			5	TO	TO	TO	**1315.38**	123.54	**36.54**
			6	TO	TO	TO	TO	427.97	**108.64**

Table 1: Runtimes for the Grapevine domain comparing the configs. with and w/o (−NV) the visited e-states check.

References

[1] Chitta Baral, Gregory Gelfond, Enrico Pontelli, and Tran Cao Son. An action language for multi-agent domains: Foundations. *CoRR*, abs/1511.01960, 2015.

[2] Agostino Dovier, Carla Piazza, and Alberto Policriti. An efficient algorithm for computing bisimulation equivalence. *Theoretical Computer Science*, 311(1-3):221–256, 2004.

[3] Francesco Fabiano, Idriss Riouak, Agostino Dovier, and Enrico Pontelli. Non-well-founded set based multi-agent epistemic action language. In *Proceedings of the 34th CILC*, volume 2396 of *CEUR Workshop Proceedings*, pages 242–259, Trieste, Italy, June 19-21 2019.

[4] Jelle Gerbrandy. *Bisimulations on planet Kripke*. Inst. for Logic, Language and Computation, Univ. van Amsterdam, 1999.

[5] Tiep Le, Francesco Fabiano, Tran Cao Son, and Enrico Pontelli. EFP and PG-EFP: Epistemic forward search planners in multi-agent domains. In *Proceedings of the 28th ICAPS*, pages 161–170, Delft, The Netherlands, June 24–29 2018. AAAI Press.

[6] John McCarthy. Programs with common sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, pages 75–91, London, United Kingdom, 1959.

[7] Hans Van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.

Acknowledgements

This research is partially supported by: the NSF HRD 1914635 and NSF HRD 1345232 projects, the University of Udine PRID ENCASE project, and the GNCS-INdAM 2017–2020 projects.