Dynamic Epistemic Logic

B. Nebel, R. Mattmüller, T. Engesser
Winter Semester 2016/2017

University of Freiburg
Department of Computer Science

Exercise Sheet P2

Due: January 9th, 2016, 20:00

We now want to extend our model checker with multi-pointed event models and the action modality. For this, we represent action models (analogously to epistemic states) as JSON dictionaries with a field `domain` (containing a list of all events) and a field `indist` specifying the indistinguishability relations for all of the agents (using the same nested list representation we use for the indistinguishability relations of epistemic states). Event names follow the same conventions as world and proposition names. The events' preconditions are specified in an additional field `pre`, containing a dictionary that maps from event names to precondition formulas. Effects are specified analogously in a field `eff`. The set of designated events is given as list `designated`. An *action library* then is a dictionary that maps from action names (which also follow our conventions for world/event/proposition names) into such actions. E.g., consider the following action library:

```
{"a1tella2whetherp": {"domain": ["e1", "e2"],
                      "indist": [[1, "e1"], [1, "e2"],
                                 [2, "e1"], [2, "e2"]],
                      "pre": {"e1": "K1p",
                              "e2": "K1~p"},
                      "eff": {"e1": "T",
                              "e2": "T"},
                      "designated": ["e1", "e2"]},
 "a2setq": {"domain": ["e1"],
            "indist": [[1, "e1"], [2, "e1"]],
            "pre": {"e1": "K2p"},
            "eff": {"e1": "q"},
            "designated": ["e1"]}}
```

For simplicity, we extend our formula language with the literals `T` and `F` representing $\top$ and $\bot$. Furthermore, we introduce a new construct `[a]`ϕ, where a is an action name and ϕ is a formula. As this construct represents the action modality, it may not be used in event preconditions. For the epistemic state, we now allow the specification of multiple designated worlds (by using a list of world names instead of just one world name for the field `designated`). In the following example, the ellipsis has to be replaced with the action library from above:

```
{"model": {"domain": ["w1", "w2", "w3"],
           "indist": [[1, "w1", "w2"], [2, "w2", "w3"]],
           "val": {"p": ["w1", "w2"], "q": []}},
 "actionlib": ...,
 "formulas": ["K1p", "K2p", "~K1K2p",
              "K1[a2setq]q", "K2[a2setq]q",
              "K1K2[a2setq]q", "K1K2~[a2setq]~q",
              "K1[a1tella2whetherp]K2[a2setq]q",
              "K1~[a1tella2whetherp]~K2~[a2setq]~q"],
 "designated": ["w1", "w2"]}
```

Again, the field `designated` is optional. If it is omitted, the formulas are to be evaluated in all worlds. The output format is analogous to the one specified in the last practical exercise sheet.

Exercise P2.1 (Implementation, 10 points)

Extend your model checker with the functionality as described above. In particular, your model checker should be able to model check formulas containing the action modality.

Exercise P2.2 (Examples, 4 points)

(a) Extend the `birthday.json` example from the previous practical sheet (which contains a specification of the initial state from *Cheryl's Birthday*) with an action library containing appropriate specifications of both Albert's and Bernard's announcements. Use your modelchecker to verify that after the announcements, it is common knowledge that Cheryl's birthday is on June 16th. Include your modified version of `birthday.json` in your submission.

(b) Devise another interesting example. Your action library should contain at least four actions, including

- a non-deterministic ontic action
- a partially observable ontic action
- a public announcement action
- a sensing action

Exercise P2.3 (Documentation, 4 points)

Make a short presentation (maximally three or four slides for a five minute talk) describing both your implementation (discussing its advantages/disadvantages) and your example. How does your model checker check announcement formulas? How do you deal with multi-pointed states / event models? Be prepared to present it in the exercise sessions.