Programming with Goals (2)

M. Birna van Riemsdijk, TU Delft, The Netherlands

GOAL slides adapted from MAS course slides by Hindriks

4/24/11

Recap

- BDI philosophy & logics
  - goals as consistent set of chosen desires
  - intention as goals that agent commits to
- Explicit representation of goals in agent systems
  - enables reasoning about goals
- Representation of goals: XML, (conjunctions of) atoms
- GOAL mental states: beliefs, goals, knowledge
- Goal types

Outline

- Dynamics of goals
- Dynamics of goals in GOAL: Action Specification & Selection

1.

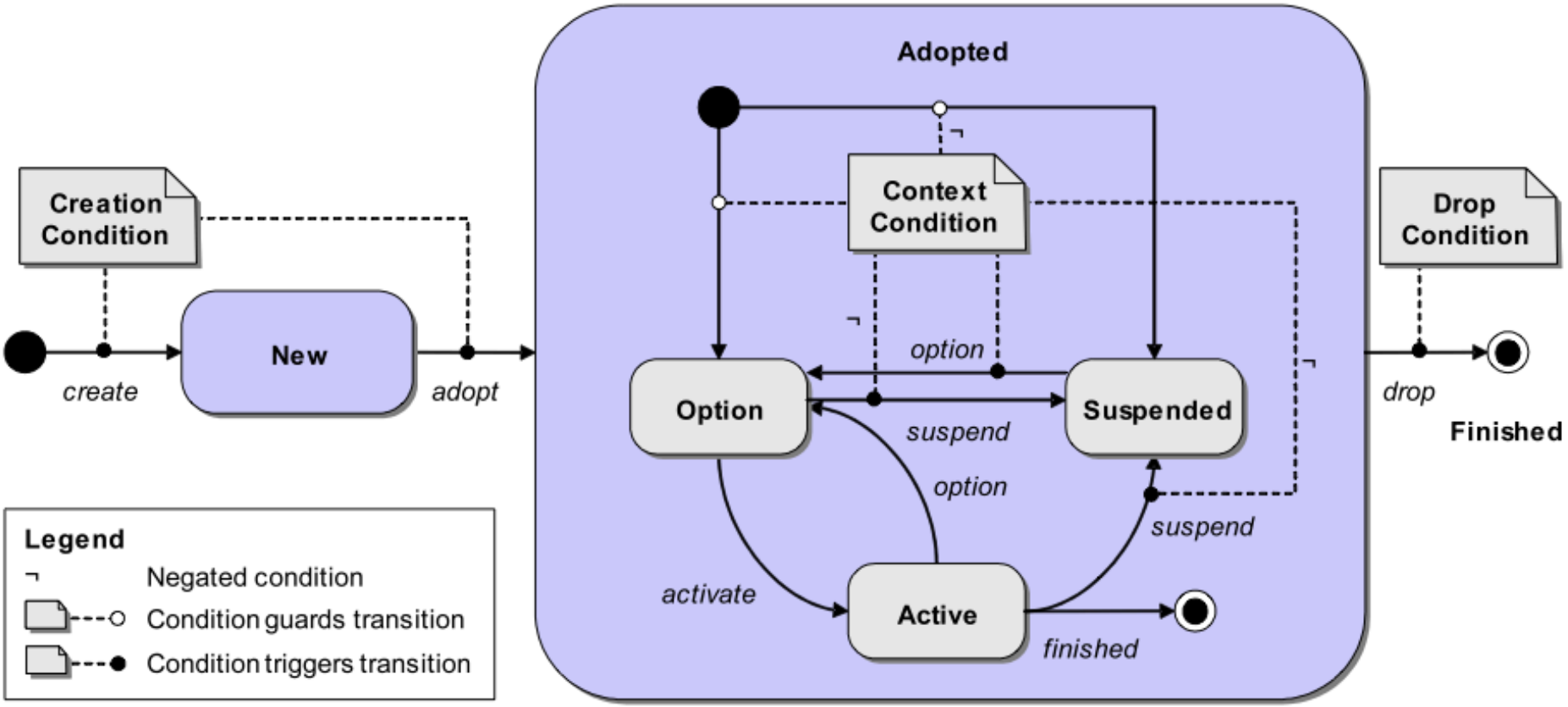
Dynamics of Goals

Dynamics of Goals

- When and how to adopt goals?
  - Internal motivation: beliefs, other goals, desires
  - External motivation: requests, obligations
- When and how to drop goals?
  - Commitment strategies
- When and how to suspend goals?
  - Priorities, goal conflict

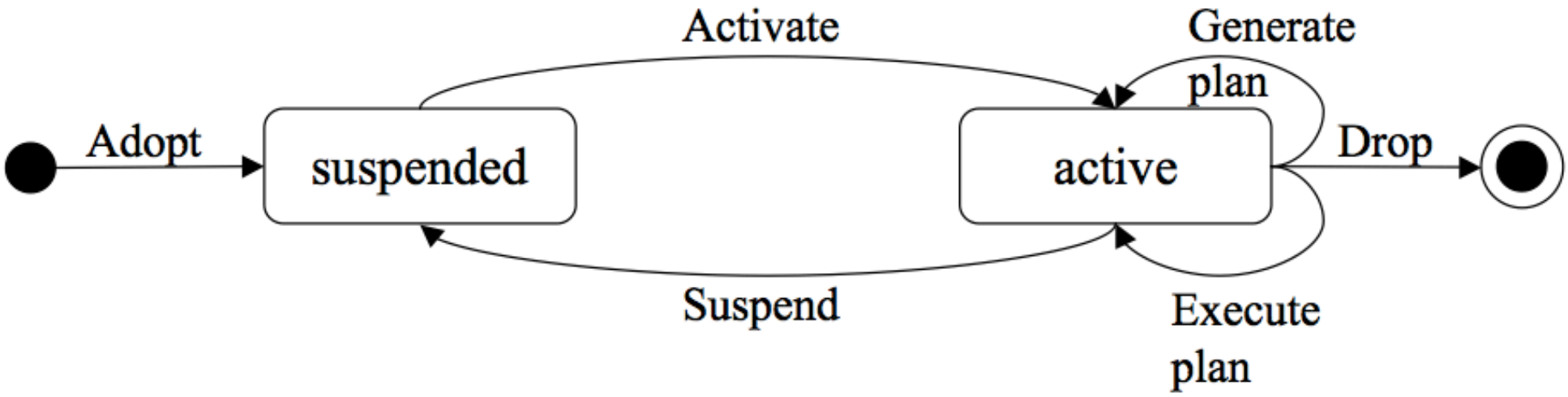
Goal Life Cycle (1)

L. Braubach, A. Pokahr, D. Moldt, and W. Lamersdorf. Goal representation for BDI agent systems. In ProMAS'04, volume 3346 of LNAI, pages 44–65. Springer, Berlin, 2005



Goal Life Cycle (2)

M. B. van Riemsdijk, M. Dastani, and M. Winikoff. Goals in agent systems: A unifying framework. In Padgham, Parkes, Müller, and Parsons, editors, *Autonomous Agents and Multi-Agent Systems (AAMAS), pages 713–720. IFAAMAS, 2008.*



Dropping goals: Recall…

- Intention is choice with commitment
- But… what does commitment mean exactly?

Commitment Strategies

Anand S. Rao and Michael P. Georgeff, 'Modeling rational agents within a BDI-architecture', in Proc. of KR'91, (1991).

- Blind commitment: keep intentions until believed they are achieved
- Single-minded commitment: keep intentions until believed they are achieved, or believed they are impossible to achieve
- Open-minded commitment: keep intentions until believed they are achieved, or they are no longer goals of the agent

Commitment strategies concern when (not) to drop intentions

Commitment Strategies for Goals (1)

M. Winikoff, L. Padgham, J. Harland, and J. Thangarajah, 'Declarative and procedural goals in intelligent agent systems', in Proc. of KR'02, (2002).

Somehow...

- commitment strategies for intentions as proposed in BDI logics translated to similar commitment strategies for goals in agent programming languages

Commitment Strategies for Goals (2)

M. Winikoff, L. Padgham, J. Harland, and J. Thangarajah, 'Declarative and procedural goals in intelligent agent systems', in Proc. of KR'02, (2002).

Winikoff et al. (2002):

- Persistent: A goal should only be deleted when it succeeds or where there is a good reason for dropping it.
- Unachieved: A goal to achieve *s* should be dropped when s is true.
- Possible: Failure condition *f* defines when a goal should be dropped with failure; failure condition expresses impossibility to achieve.

≈ single-minded commitment

Perspectives on Dropping Goals

M. B. van Riemsdijk, M. Dastani, F. Dignum, and J.-J. Ch. Meyer. Dynamics of declarative goals in agent programming. In Proc of DALT'04, volume 3476 of LNAI, pages 1–18, 2005.

- Dropping over transition
  - transition from one configuration into another: $c \rightarrow c'$
  - dropping of goal ϕ over $c \rightarrow c'$
    agent has goal ϕ in *c* but not in *c'*
- Two perspectives
  - deletion perspective: deletion of goal from goal base
  - satisfaction perspective: formula $\mathbf{G}\phi$ becomes unsatisfied

Commitment Strategy of GOAL

- Goals are dropped from goal base when believed to be achieved (deletion perspective)
  - ≈ blind commitment
- Mental state condition `a-goal(`ϕ`)` holds if ϕ is not believed (satisfaction perspective)
- Goals can also be dropped using the built-in `drop` action
  - can be used to implement single-minded commitment

Commitment Strategy of Jason

J. F. Hubner, R. H. Bordini, and M. Wooldridge. Programming declarative goals using plan patterns. In M. Baldoni and U. Endriss, editors, Declarative Agent Languages and Technologies IV, pages 123–140. Springer, 2006.

- Drop achievement goal from event base as soon as plan for it has been selected
- But, commitment strategies can be programmed using plan patterns

```
+!g: g ← true.
+!g: c ← p; ?g.
-!g: true ← !g
```

Suspending Tasks (= Goals or Plans)

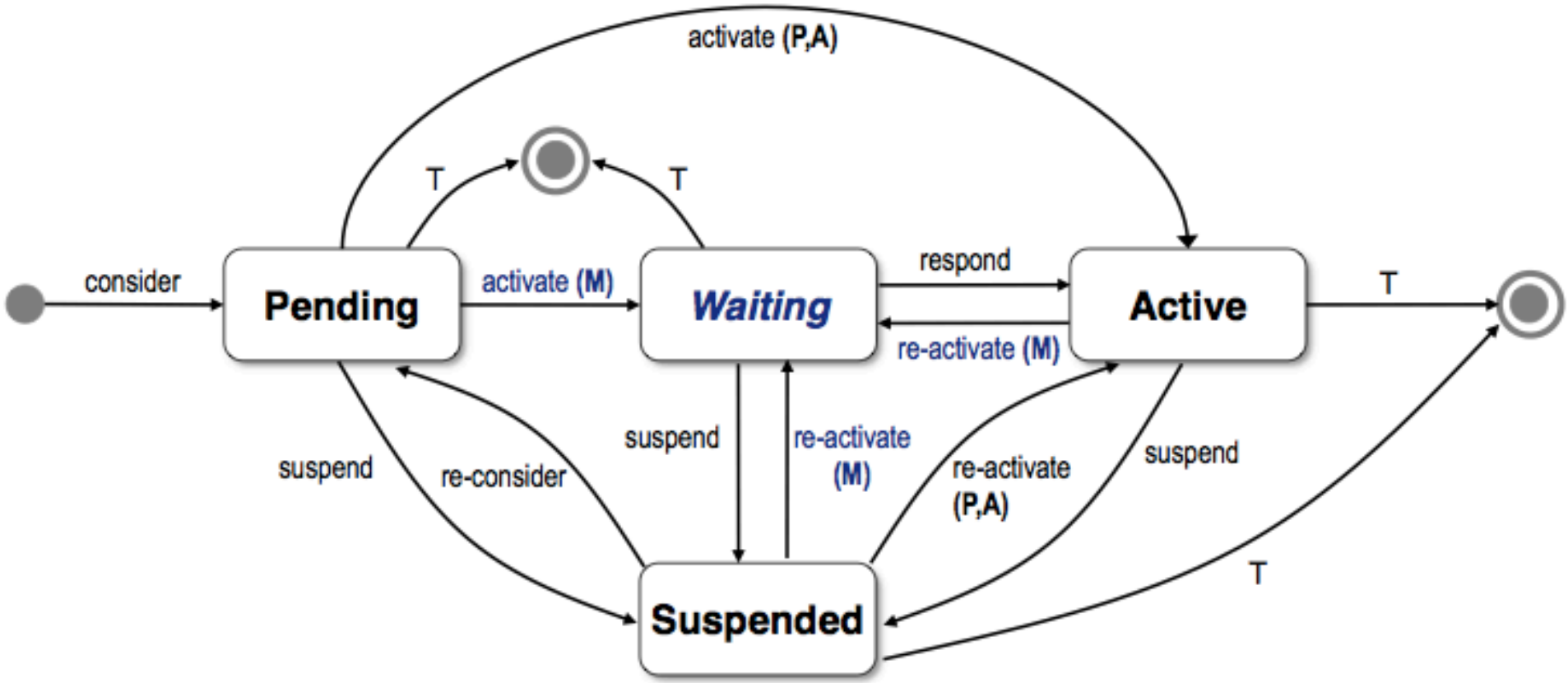
J. Thangarajah, J. Harland, D. Morley, and N. Yorke-Smith. Suspending and resuming tasks in BDI agents. In Proc. of AAMAS'08, 2008.

Reasons for suspension:

- Conflicts between tasks
- Positive interaction
- Invalid context
- No applicable plan
- Changing priorities
- Requests from other agents

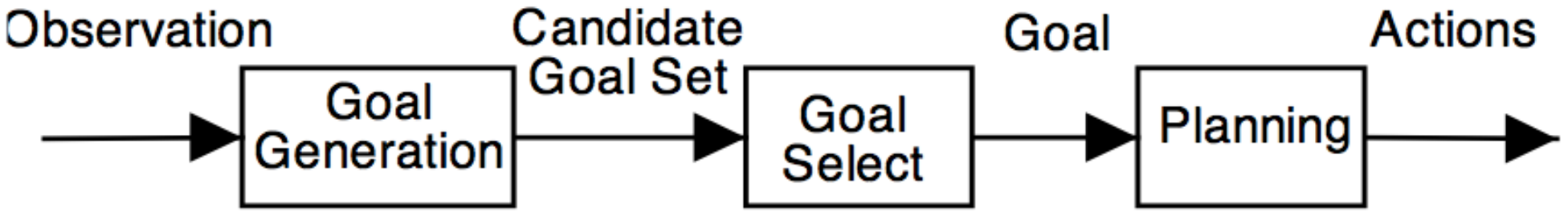
Suspending & Aborting Goals

J. Thangarajah, J. Harland, D. Morley, and N. Yorke-Smith. Operational Behaviour for Executing, Suspending, and Aborting Goals in BDI Agent Systems. In Proc. of DALT'10, 2010.



Goal Generation in BOID

Goal Generation in the BOID Architecture, Jan Broersen, Mehdi Dastani, Joris Hulstijn and Leendert van der Torre, Cognitive Science Quarterly, Volume 2, Issue 3-4, 2002.



- Conditional rules for generating goals, based on mental attitudes
  - Beliefs, Obligation, Intention, Desire
  - Technically, goal set is extension in default logic
- Priorities over mental attitudes create agent types

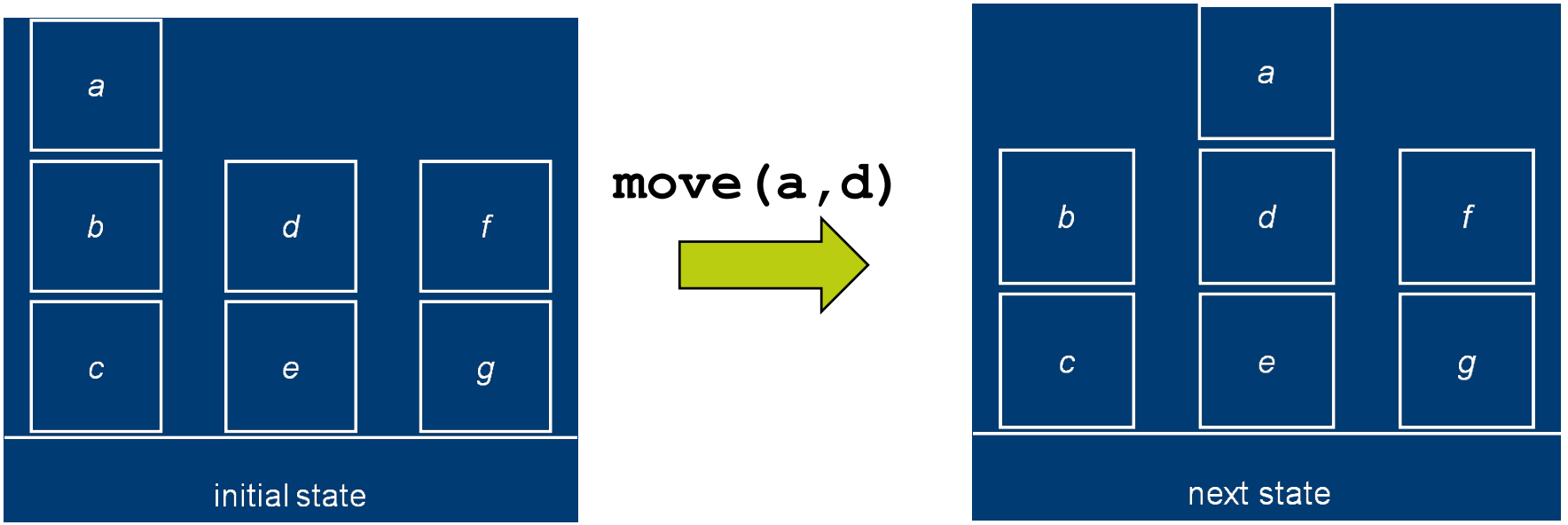
Goal Adoption in GOAL

- Goals can be inserted into goal base in initial mental state
- Goals can be adopted at run-time using the built-in `adopt` action, conditional on mental state

2.

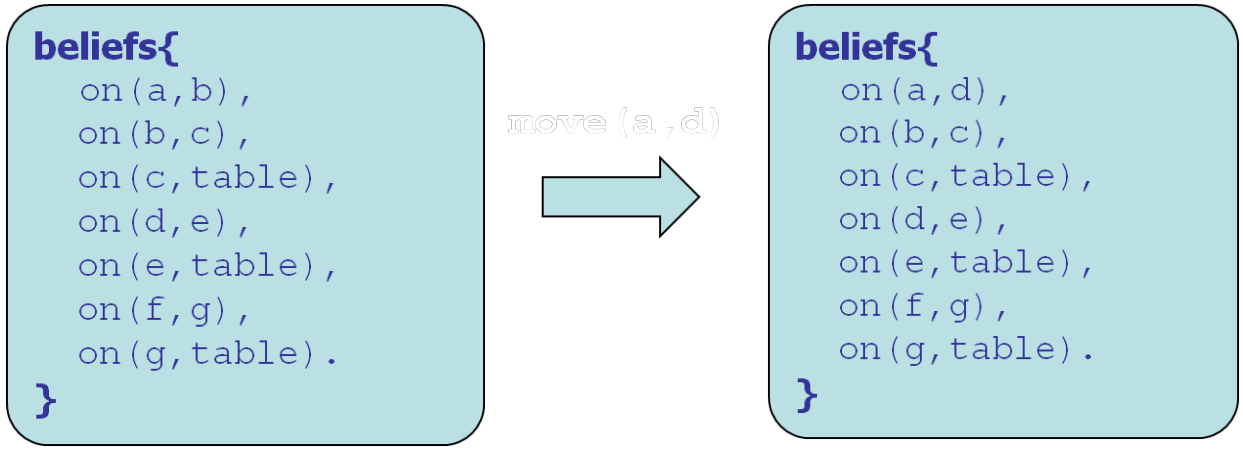
Dynamics of Goals in GOAL: Action Specification & Action Selection

Actions Change Environment...



…and Require Updating Mental States: Beliefs

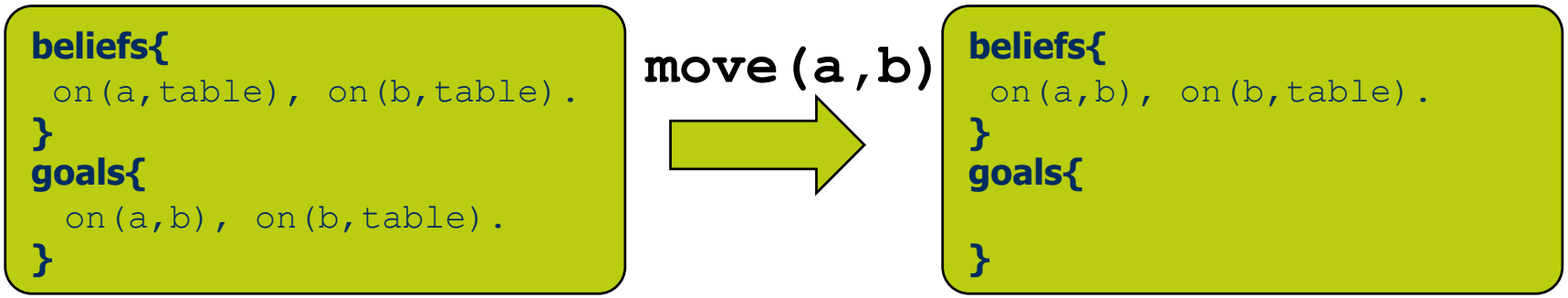
- To ensure adequate beliefs after performing an action the belief base needs to be updated (and possibly the goal base).



- *Add effects* to belief base: insert **`on(a,d)`** after **`move(a,d)`**.
- *Delete* old beliefs: delete **`on(a,b)`** after **`move(a,d)`**.

...and Require Updating Mental States: Goals

- If a goal has been (believed to be) completely achieved, the goal is removed from the goal base.



- Default update implements a blind commitment strategy.
- Goal base updates as "side effect" of belief base updates

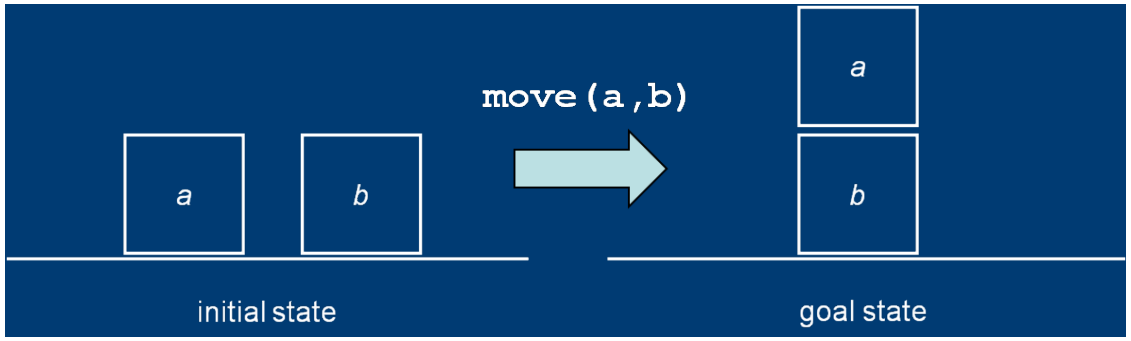
Action Specifications

- Actions in GOAL have preconditions and postconditions (STRIPS-style)
- Executing an action in GOAL means:
  - Preconditions are conditions that need to be true:
    - Check preconditions on the belief base.
  - Postconditions (effects) are add/delete lists:
    - Add positive literals in the postcondition
    - Delete negative literals in the postcondition

```
move(X,Y){
    pre { clear(X), clear(Y), on(X,Z), not( on(X,Y) ) }
    post { not(on(X,Z)), on(X,Y) }
}
```

Actions Specifications

```
move(X,Y){
  pre { clear(X), clear(Y), on(X,Z), not( on(X,Y) )}
  post { not(on(X,Z)), on(X,Y) }
}
```



Example: `move(a,b)`

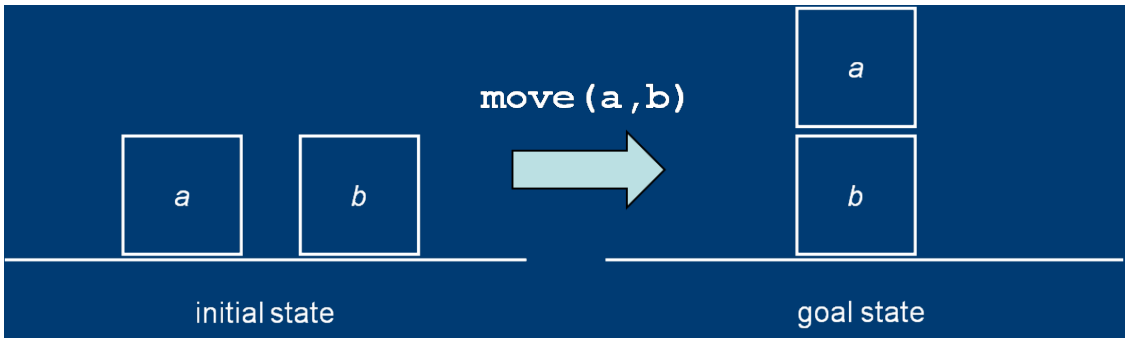
- Check: `clear(a), clear(b), on(a,Z), not(on(a,b))`
- Remove: `on(a,Z)`
- Add: `on(a,b)`

`Z = table`

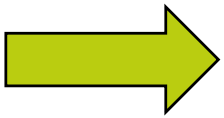
Actions Specifications

```
move(X,Y){
  pre { clear(X), clear(Y), on(X,Z) }
  post { not(on(X,Z)), on(X,Y) }
}
```

Example: **move(a,b)**




```
beliefs{
  on(a,table),
  on(b,table).
}
```



```
beliefs{
  on(b,table).
  on(a,b).
}
```

Built-in Actions

Adopting and dropping goals:

- adopt(<conjunction of positive literals>)
  meaning: add a **new** goal to goal base (if not already **implied** by a goal)
- drop(<conjunction>)
  meaning: remove **all** goals that imply <conjunction> from the goal base

Inserting and deleting beliefs:

- insert(<conjunction>)
- delete(<conjunction>)

Drop Action

`drop(on(b,a), not(on(c,table)))`

```
knowledge{
 block(X) :- on(X, Y).
 clear(X) :- block(X), not(on(Y,X)).
 clear(table).
 tower([X]) :- on(X, table).
 tower([X,Y|T]):- on(X,Y),tower([Y|T]).
}
goals{
 on(a,table), on(b,a), on(c,b),
 on(d,table), on(e,table), on(f,e),
 on(g,f), on(h,g), on(i,h).
}
```

- Is goal in goal base dropped?
- Check: does goal imply `on(b,a), not(on(c,table))` ?
- A: Yes, so goal is removed by drop action.

Action Selection in Agent-Oriented Programming

- How do humans choose and/or explain actions?
- Examples:
  - I believe it rains; so, I will take an umbrella with me.
  - I go to the video store because I want to rent I-robot.
  - I don't believe busses run today so I take the train.
- BDI not only for explaining & predicting, but also for programming!
- Use intuitive common sense concepts:

beliefs + goals => action

Selecting Actions: Action Rules

- Action rules are used to define a strategy for action selection.
- Defining a strategy for blocks world:
  - If constructive move can be made, make it.
  - If block is misplaced, move it to table.

```
program{
    if bel(tower([Y|T])), a-goal(tower([X,Y|T])) then move(X,Y).
    if a-goal(tower([X|T])) then move(X,table).
}
```

- What happens:
  - Check condition, e.g. can **`a-goal(tower([X|T]))`** be derived given current mental state of agent?
  - Yes, then (potentially) select **`move(X,table)`**.

Order of Action Rules

- Action rules are executed by default in linear order.
- The first rule that fires is executed.

```
program{
    if bel(tower([Y|T])), a-goal(tower([X,Y|T])) then move(X,Y).
    if a-goal(tower([X|T])) then move(X,table).
}
```

- Default order can be changed to random.
- Arbitrary rule that is able to fire may be selected.

```
program[order=random]{
    if bel(tower([Y|T])), a-goal(tower([X,Y|T])) then move(X,Y).
    if a-goal(tower([X|T])) then move(X,table).
}
```

Summary

- Dynamics of goals
  - goal life cycle
  - adopting, dropping & suspending goals
- Action specification
  - pre- and postcondition
  - update beliefs
  - built-in actions for adopting/dropping goals
- Action selection
  - action rules with mental state condition