

# Hierarchies of Reward Machines

Daniel Furelos-Blanco

Mark Law

Anders Jonsson

Krysia Broda

Alessandra Russo

**Imperial College  
London**

**ILASP**  
*Learning Logically*



**Universitat  
Pompeu Fabra  
Barcelona**

# Introduction

Humans describe tasks in some language to instruct other humans:

- ‘bring coffee to the office, but be careful with the plants!’,
- ‘make a cake’,
- ‘patrol locations A, B, C and D in that order’,
- ...

We want to do the same with AI agents. Example: MineRL BASALT competition.



MakeWaterfall



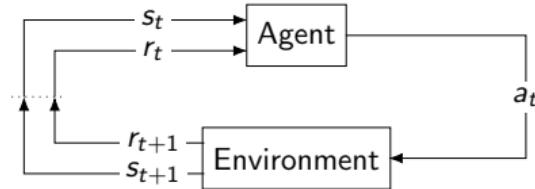
CreateVillageAnimalPen



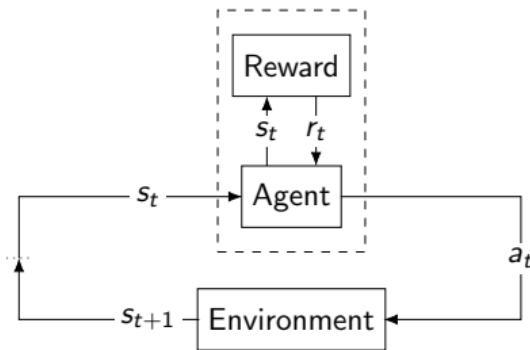
BuildVillageHouse

# Introduction

- The interaction hides the reward structure from the agent...



- Why not providing these *structured task descriptions* (e.g., LTL formulas, finite-state machines, grammars) to the agent?

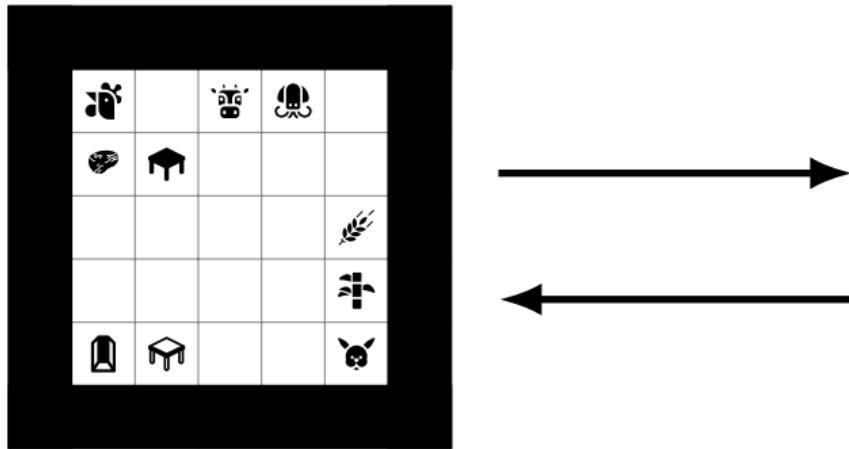


- Advantages & drawbacks:
  - + Interpretability.
  - + Enable task decomposition.
  - /+ Handcrafted – but we can learn them!
- **Our focus:** Express-Exploit-Learn descriptions of tasks that depend on other tasks.

$$\text{'make a cake'} = \begin{cases} \text{'make batter'} = \begin{cases} \text{'collect wheat'}, \\ \text{'collect eggs'}, \\ \text{'mix them'} \end{cases} \\ \text{'make sweetened milk'} = \begin{cases} \text{'make milk'} = \begin{cases} \text{'make bucket'} = \begin{cases} \text{'collect iron'}, \\ \text{'transform it'} \end{cases}, \\ \text{'go to a cow'}, \\ \text{'fill the bucket with milk'} \end{cases} \\ \text{'make sugar'} = \begin{cases} \text{'collect a sugarcane'}, \\ \text{'transform it'} \end{cases}, \\ \text{'mix them'} \end{cases} \end{cases}$$

# Motivation

## Reward Machines



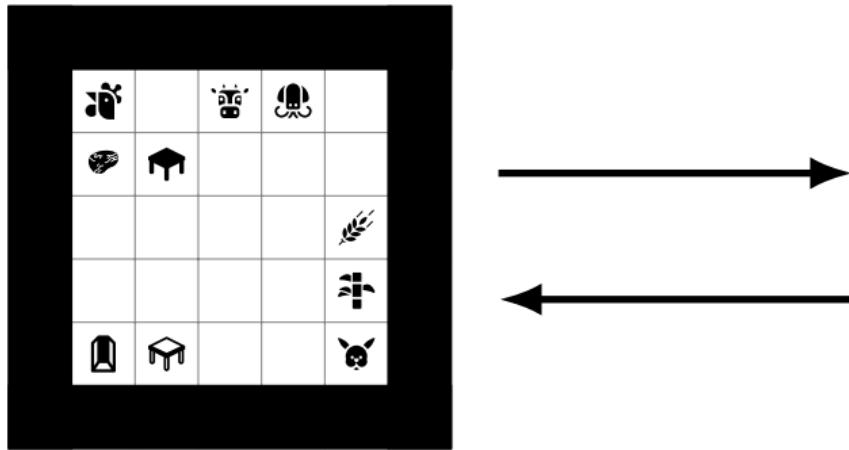
Events

{, , , , , ,  
, }

Toro Icarte et al. "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning". ICML, 2018.

# Motivation

## Reward Machines



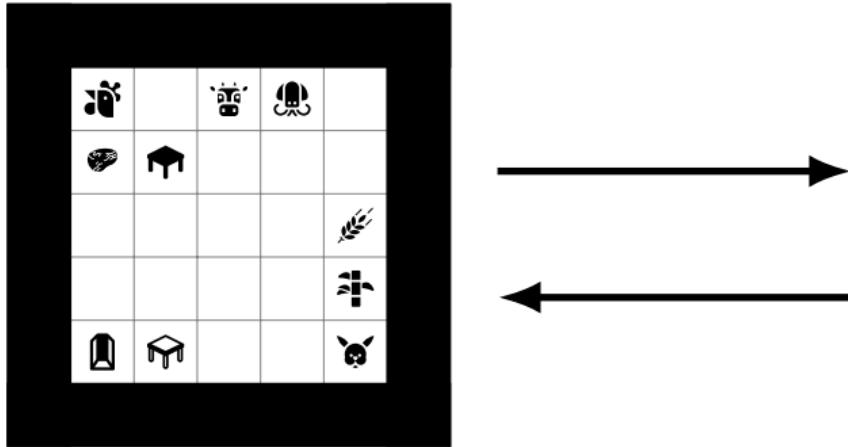
Events

{, , , , , , , }

# Motivation

## Reward Machines

Task Collect and (in any order), then go to .



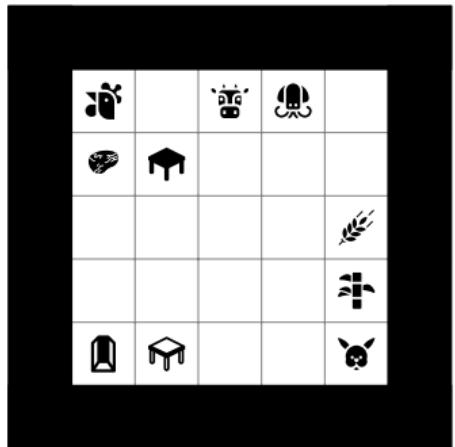
Events

$\{\text{treasure chest}, \text{house}, \text{book}, \text{key}, \text{meat}, \text{shovel}, \text{wheat}, \text{pickaxe}, \text{skull}, \text{small house}\}$

# Motivation

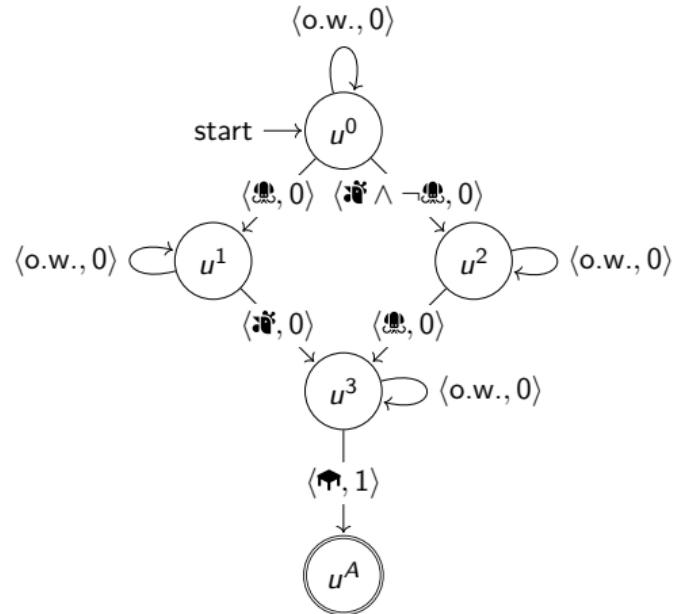
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

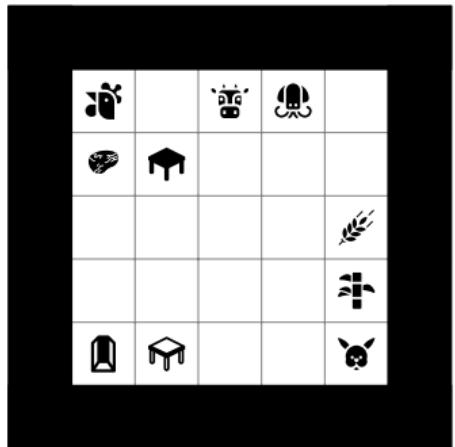
$\{\text{apple}, \text{key}, \text{house}, \text{cross}, \text{book}, \text{chest}, \text{skull}\}$



# Motivation

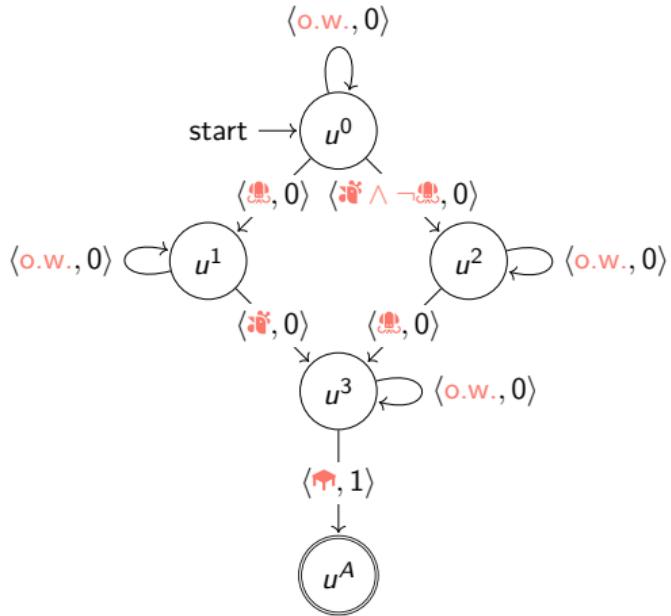
## Reward Machines

**Task** Collect and (in any order), then go to .



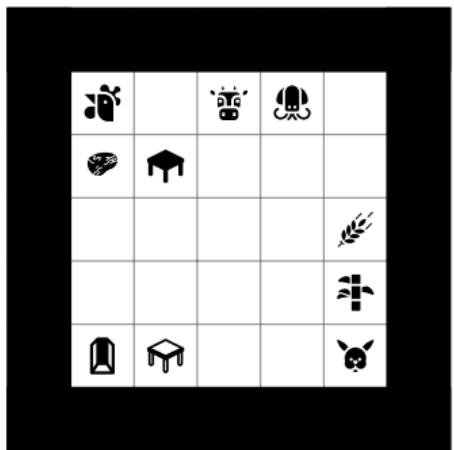
Events

$\{\text{apple}, \text{house}, \text{book}, \text{apple}, \text{house}, \text{apple}, \text{book}, \text{house}\}$



# Motivation

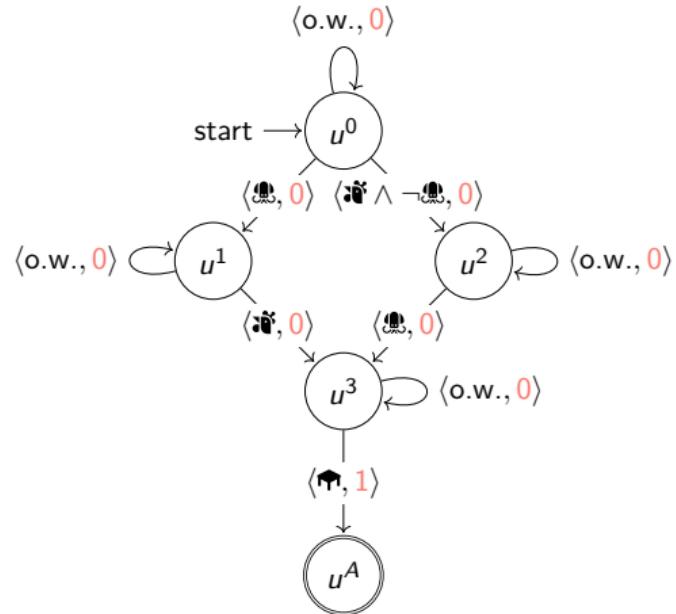
## Reward Machines



Events

$\{\text{chest}, \text{house}, \text{wheat}, \text{key}, \text{apple}, \text{mushroom}, \text{book}, \text{star}, \text{goal}\}$

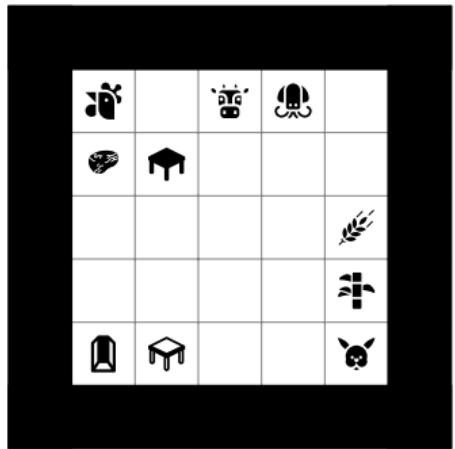
**Task** Collect  $\text{chest}$  and  $\text{apple}$  (in any order), then go to  $\text{goal}$ .



# Motivation

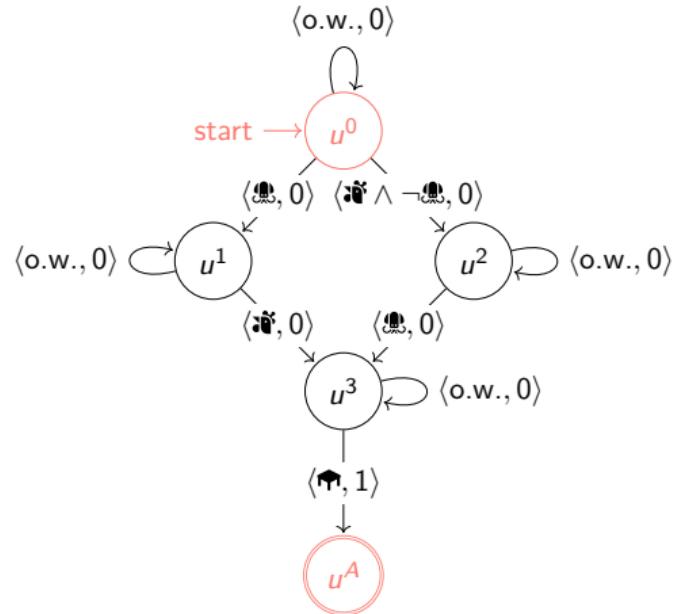
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

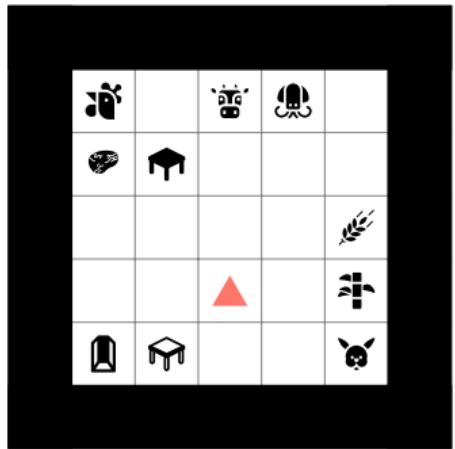
$\{\text{apple}, \text{house}, \text{book}, \text{wheat}, \text{apple}, \text{book}, \text{house}, \text{book}\}$



# Motivation

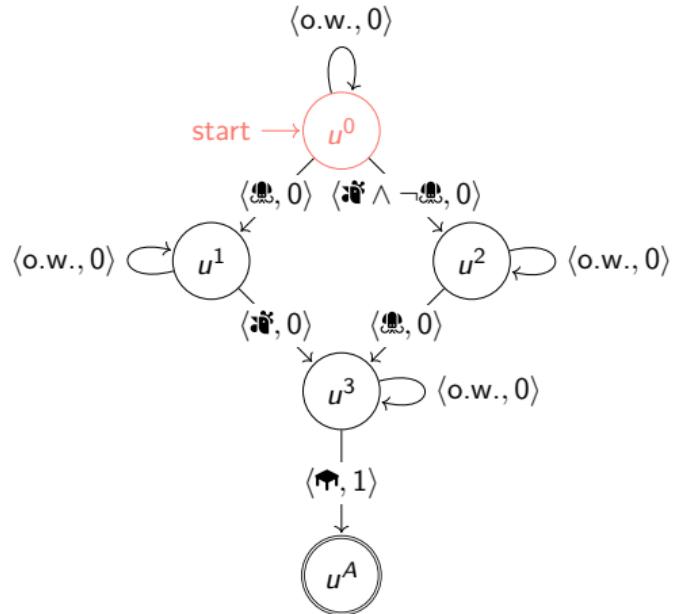
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

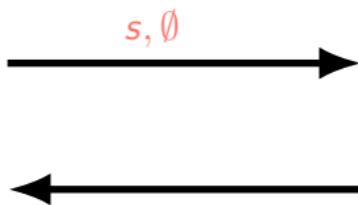
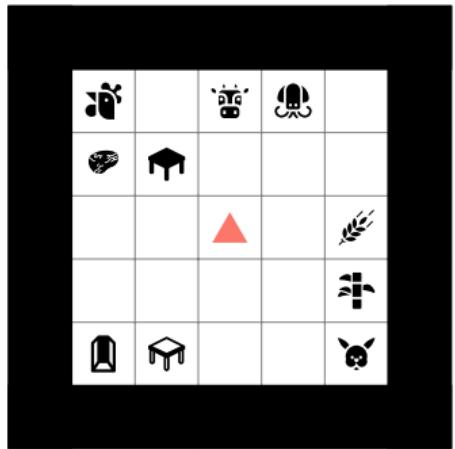
$\{\text{apple}, \text{house}, \text{book}, \text{table}, \text{chair}, \text{mushroom}, \text{key}, \text{keyhole}\}$



# Motivation

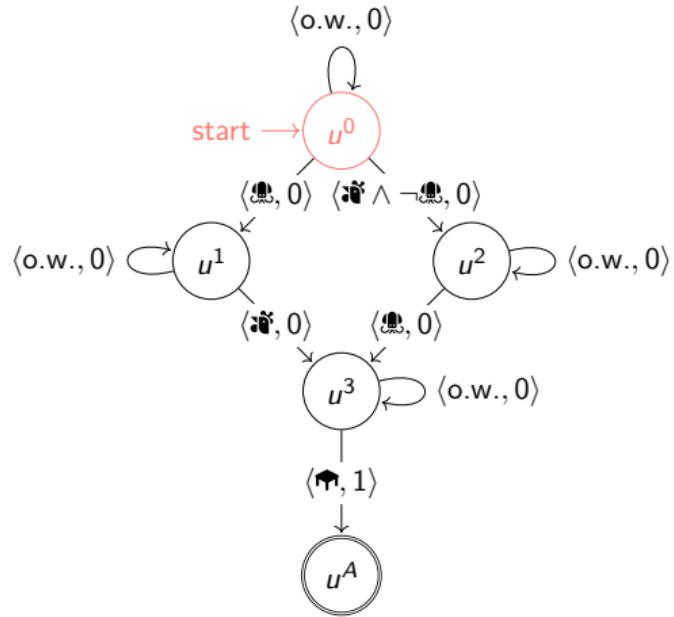
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

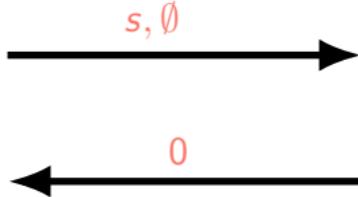
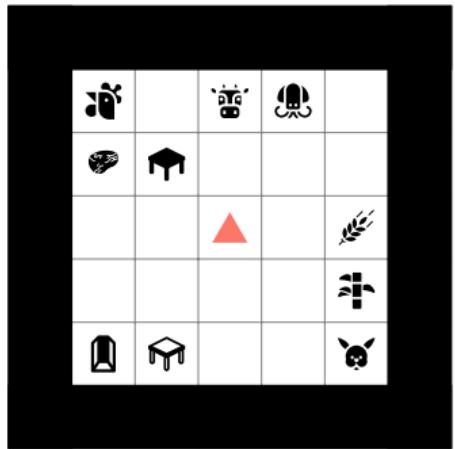
$\{\text{apple}, \text{house}, \text{mushroom}, \text{book}, \text{wheat}, \text{triangle}, \text{empty slot}\}$



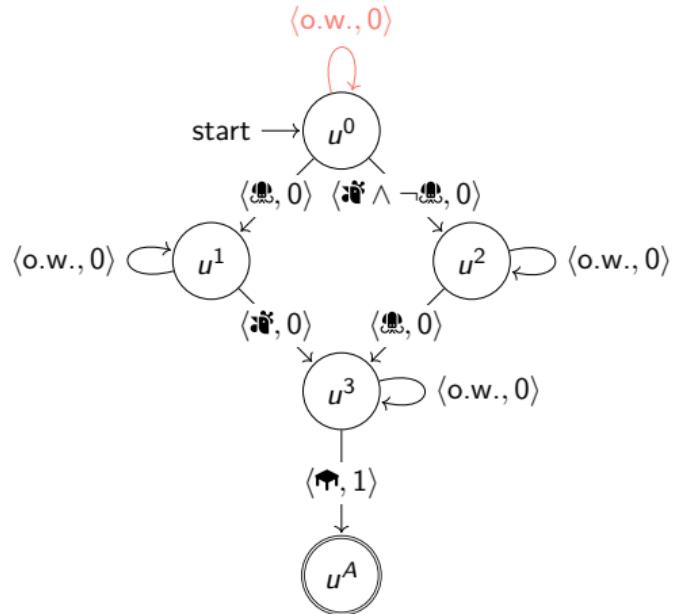
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



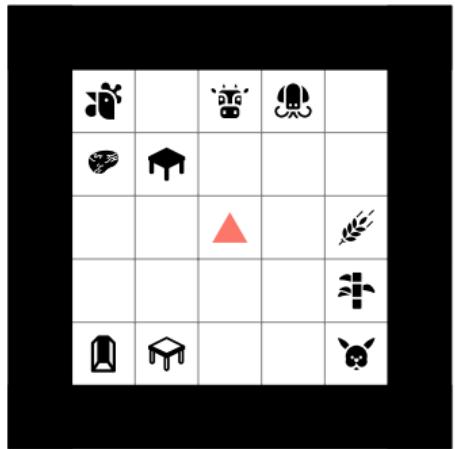
Events  
 $\{\text{apple}, \text{house}, \text{mushroom}, \text{book}, \text{wheat}, \text{apple}, \text{mushroom}, \text{house}\}$



# Motivation

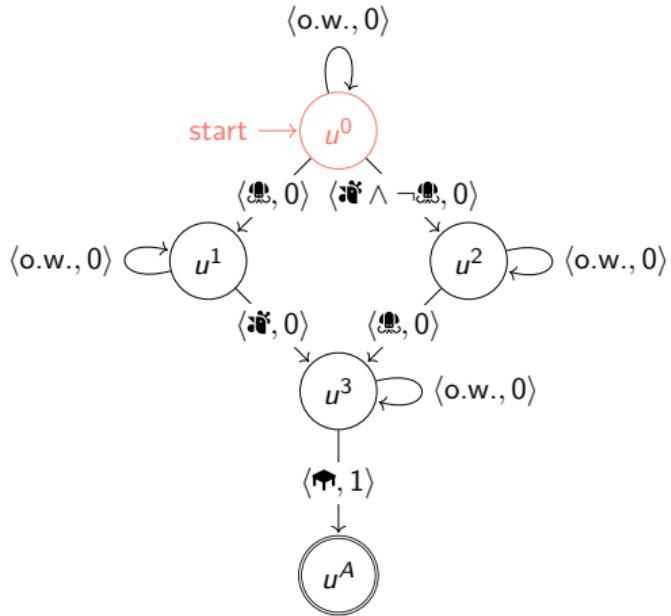
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

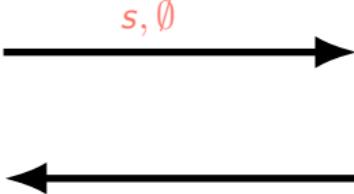
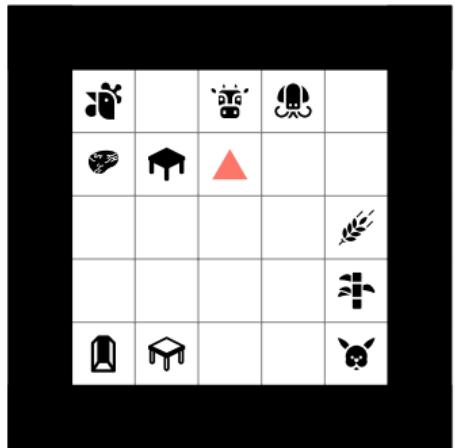
$\{\text{apple}, \text{house}, \text{mushroom}, \text{book}, \text{wheat}, \text{triangle}, \text{empty}\}$



# Motivation

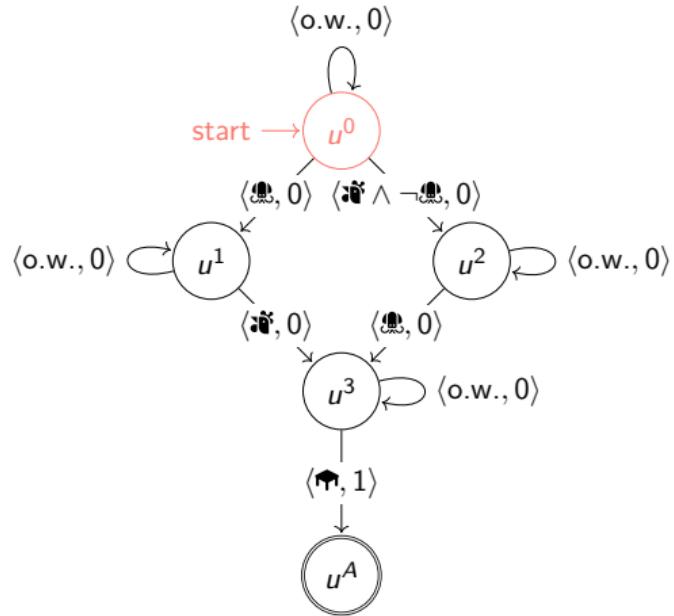
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

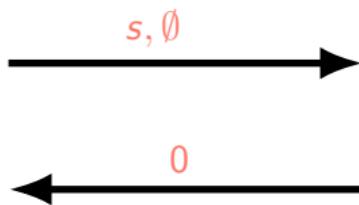
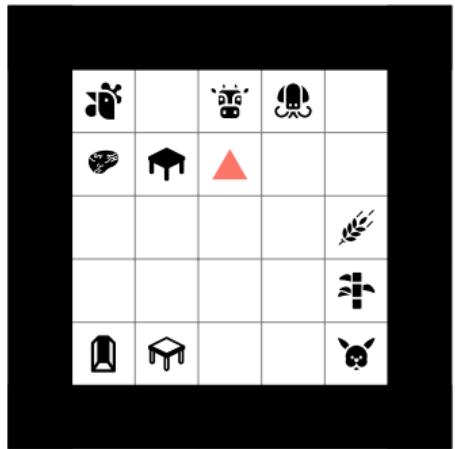
$\{\text{key}, \text{desk}, \text{book}, \text{cross}, \text{wheat}, \text{coin}, \text{house}\}$



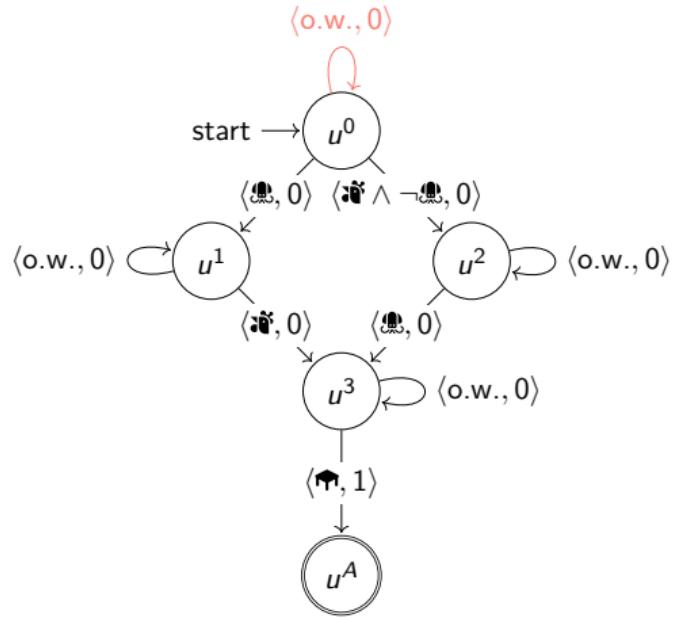
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



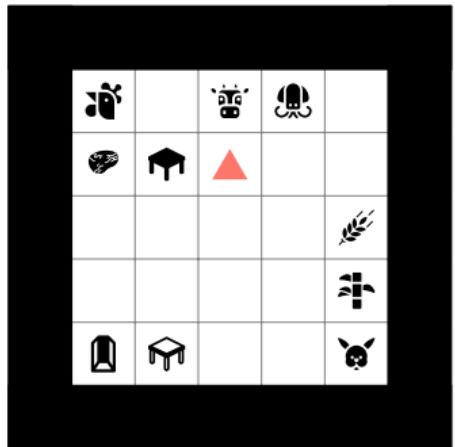
Events  
 $\{\text{apple}, \text{house}, \text{mushroom}, \text{cross}, \text{wheat}, \text{book}, \text{red triangle}\}$



# Motivation

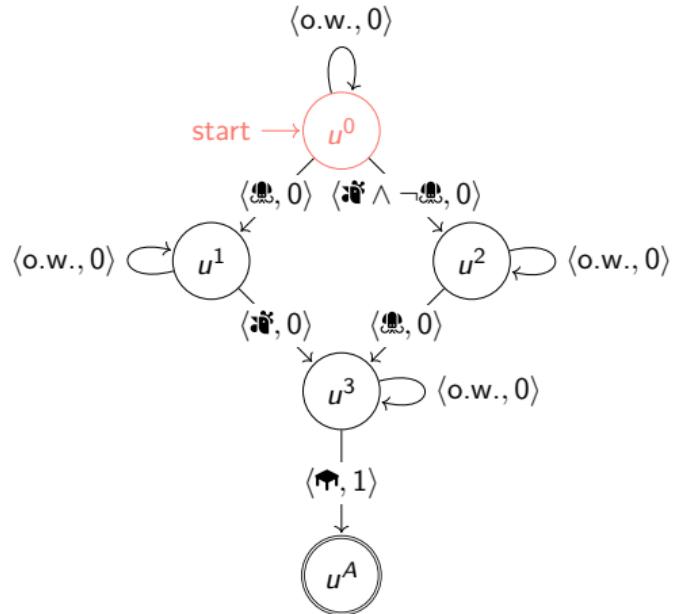
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

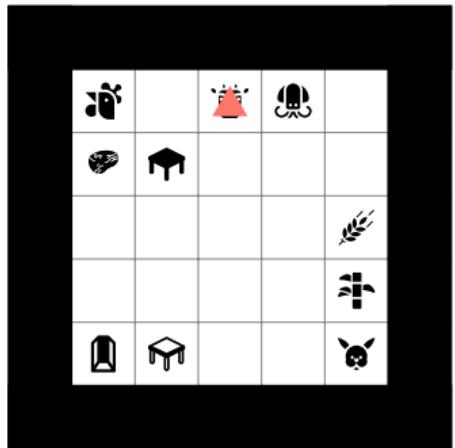
{, , , , , ,  
, , , }



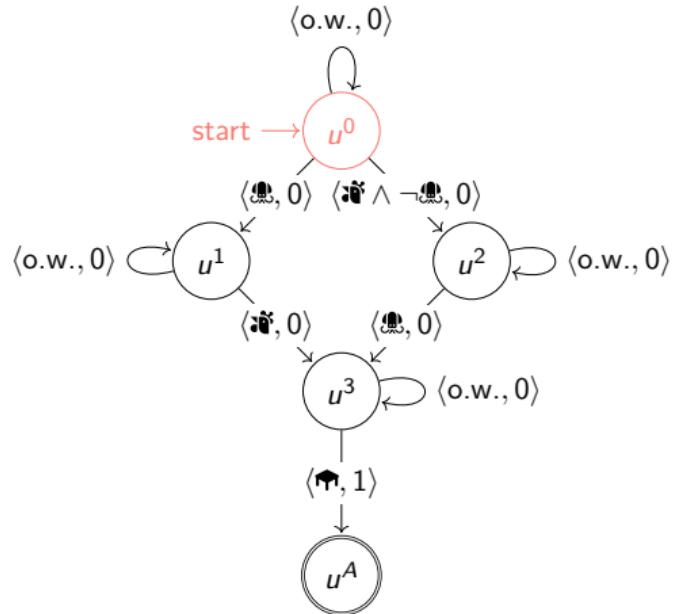
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



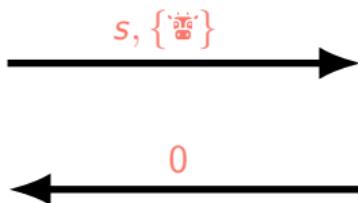
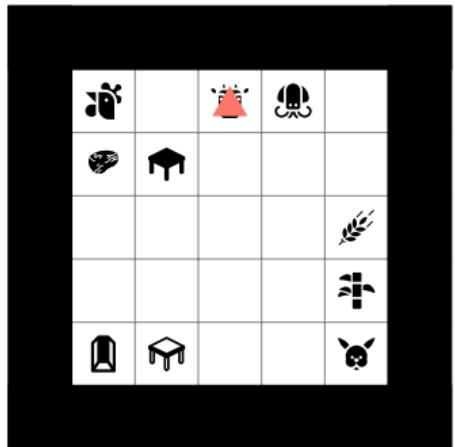
Events  
 $\{\text{key}, \text{desk}, \text{bell}, \text{cross}, \text{wheat}, \text{key}, \text{book}, \text{flower}, \text{skull}, \text{house}\}$



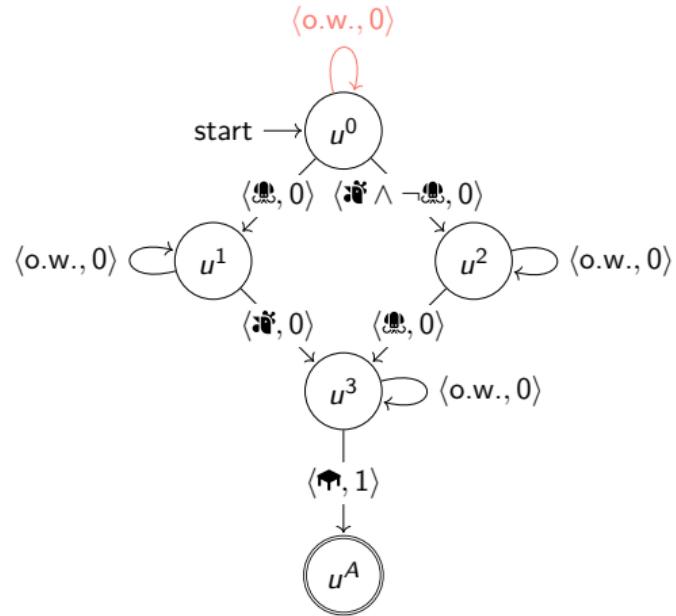
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



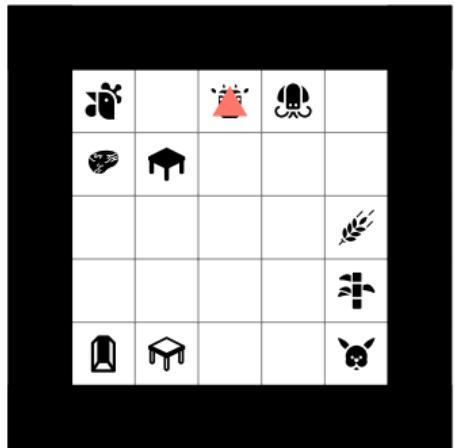
Events  
 $\{\text{apple}, \text{house}, \text{bell}, \text{cross}, \text{wheat}, \text{apple}, \text{bell}, \text{house}, \text{apple}\}$



# Motivation

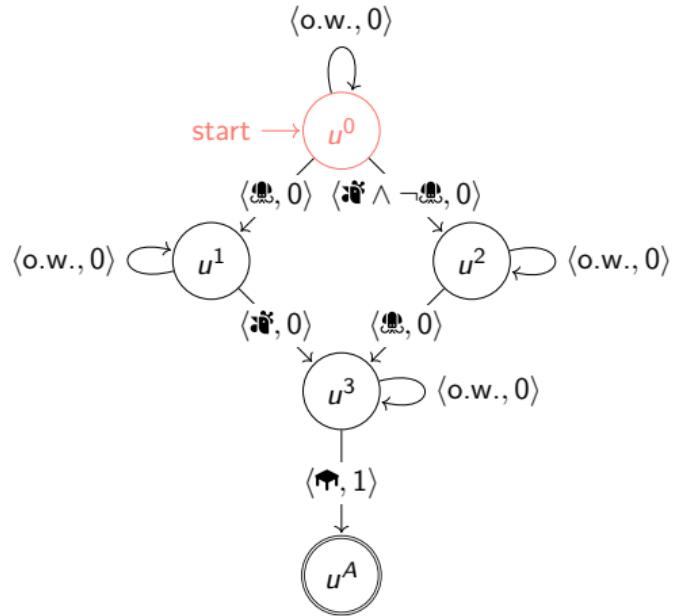
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

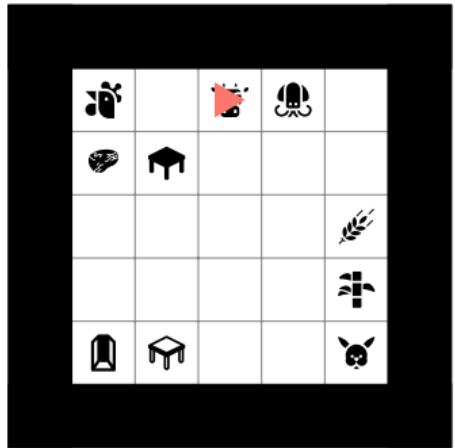
$\{\text{apple}, \text{house}, \text{bell}, \text{flower}, \text{wheat}, \text{apple}, \text{bell}, \text{house}, \text{wheat}\}$



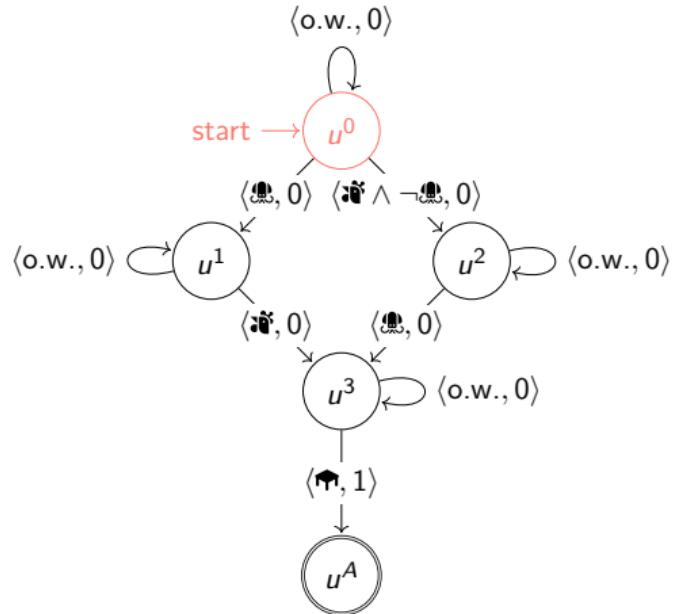
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



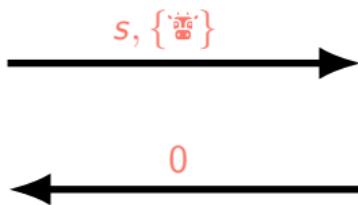
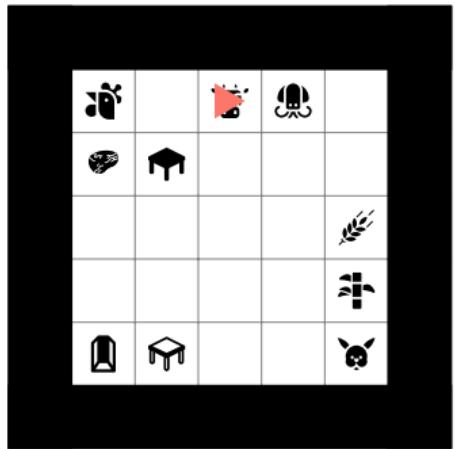
Events  
 $\{\text{apple}, \text{house}, \text{book}, \text{wheat}, \text{key}, \text{book}, \text{key}, \text{house}\}$



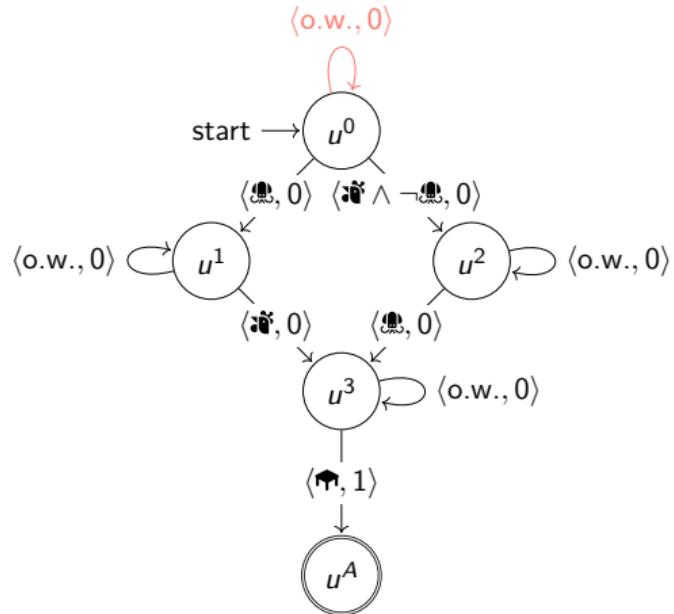
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



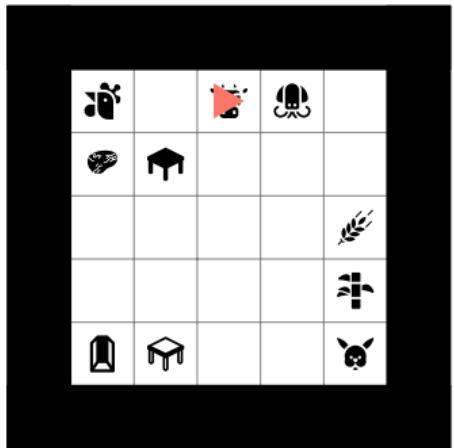
Events  
 $\{\text{book}, \text{house}, \text{apple}, \text{cross}, \text{wheat}, \text{mushroom}, \text{empty slot}\}$



# Motivation

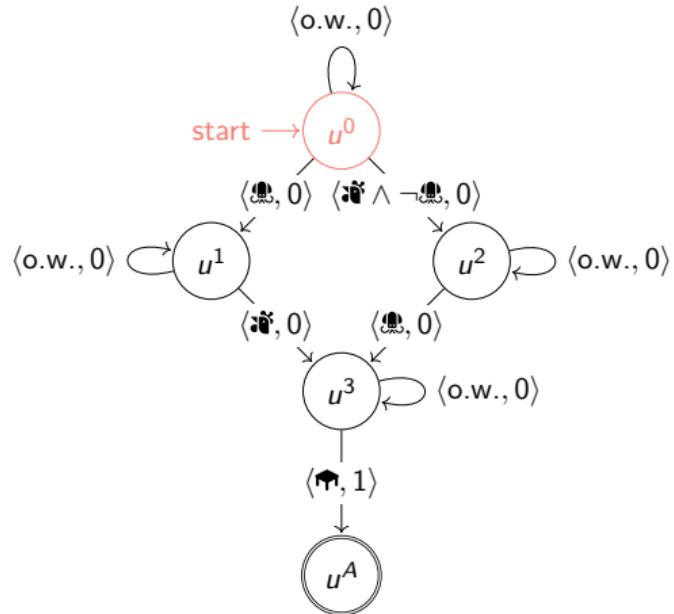
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

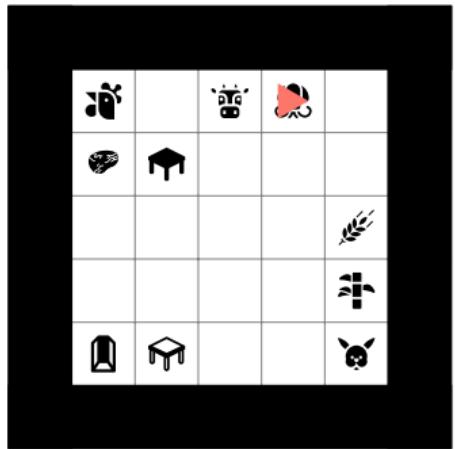
$\{\text{apple}, \text{house}, \text{mushroom}, \text{book}, \text{wheat}, \text{key}, \text{star}, \text{cross}\}$



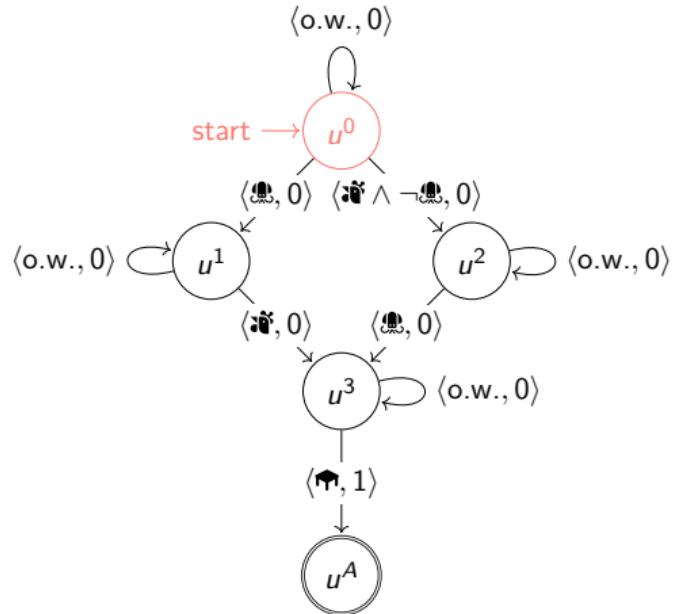
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



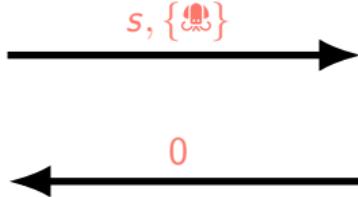
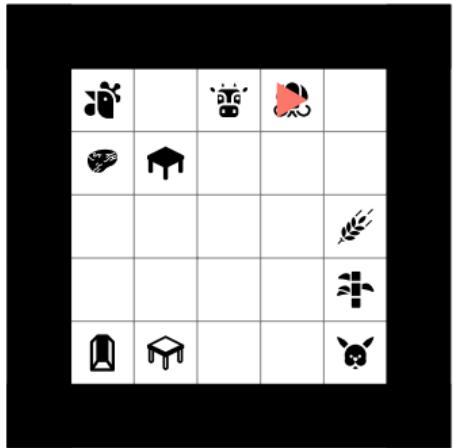
Events  
 $\{\text{apple}, \text{house}, \text{book}, \text{cross}, \text{leaf}, \text{skull}, \text{door}, \text{table}, \text{chair}, \text{sword}, \text{shield}\}$



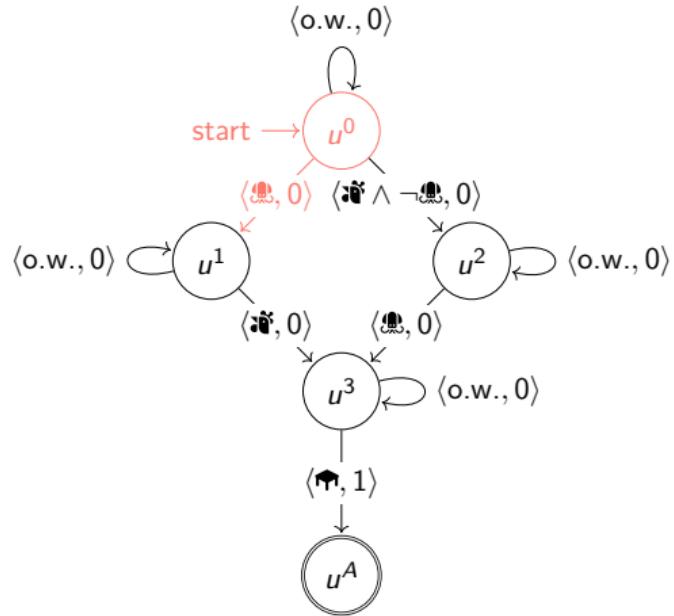
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



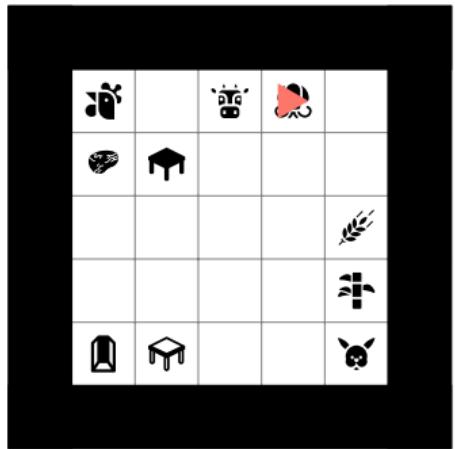
Events  
 $\{\text{apple}, \text{book}, \text{key}, \text{chest}, \text{wheat}, \text{bomb}, \text{sword}, \text{shield}\}$



# Motivation

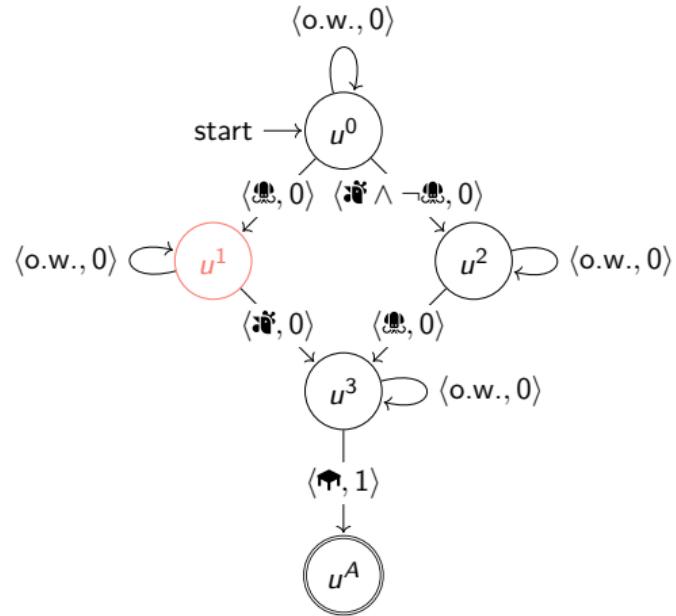
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

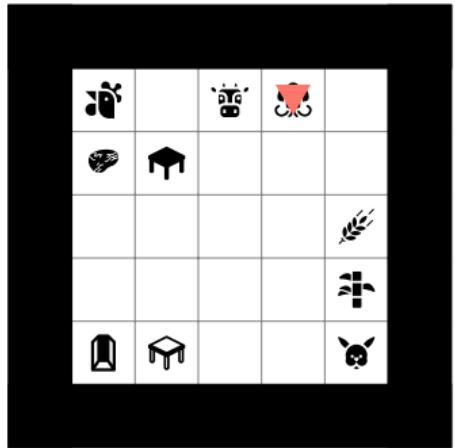
$\{\text{book}, \text{house}, \text{apple}, \text{mushroom}, \text{wheat}, \text{key}, \text{keyhole}, \text{door}\}$



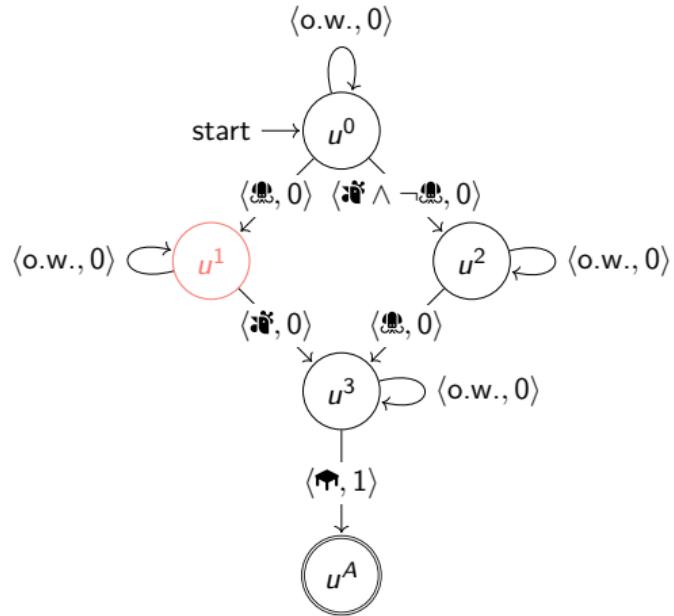
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



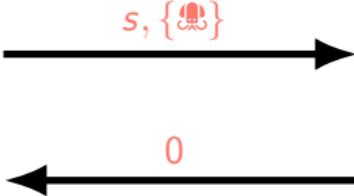
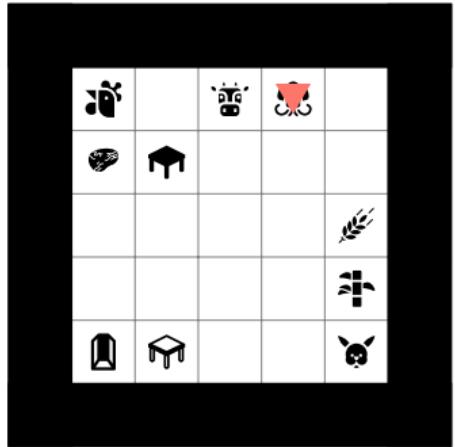
Events  
 $\{\text{apple}, \text{house}, \text{key}, \text{book}, \text{wheat}, \text{apple}, \text{key}, \text{book}\}$



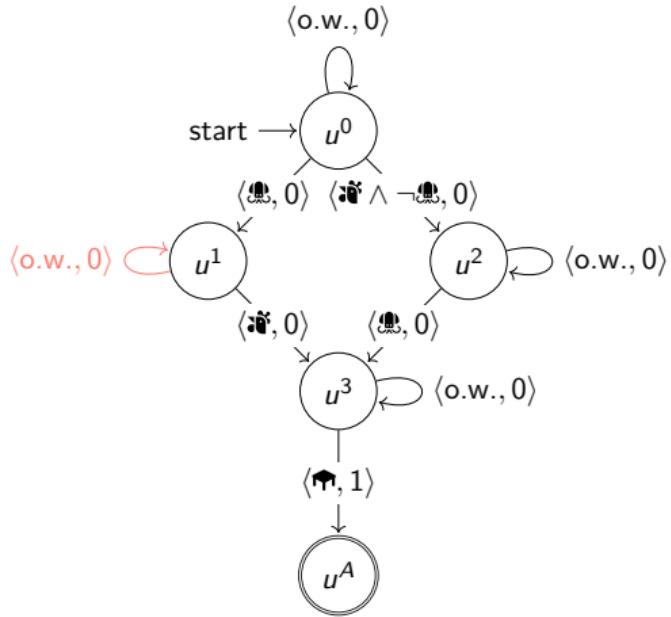
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



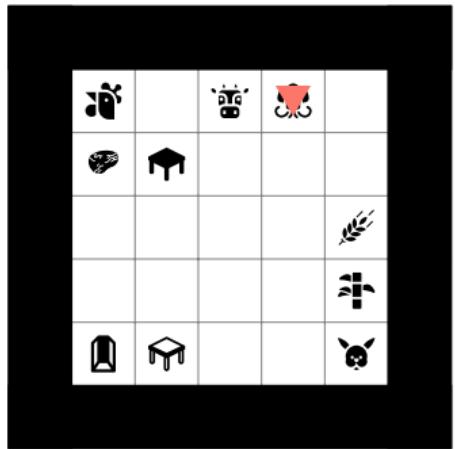
Events  
 $\{\text{apple}, \text{book}, \text{house}, \text{empty}, \text{wheat}, \text{key}, \text{book}, \text{apple}, \text{house}\}$



# Motivation

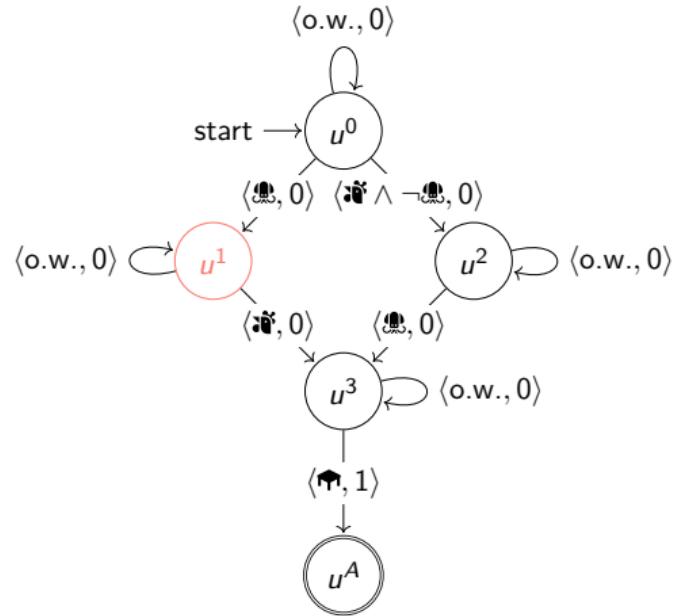
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

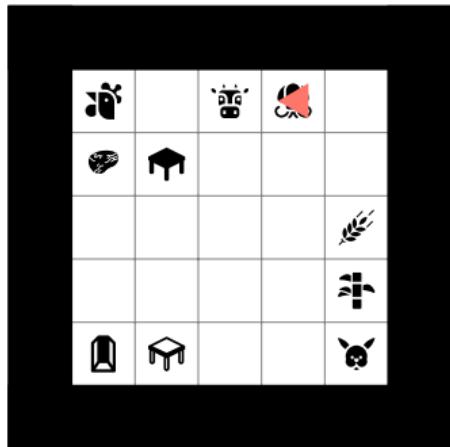
$\{\text{book}, \text{house}, \text{apple}, \text{mushroom}, \text{wheat}, \text{key}, \text{keyhole}, \text{door}\}$



## Motivation

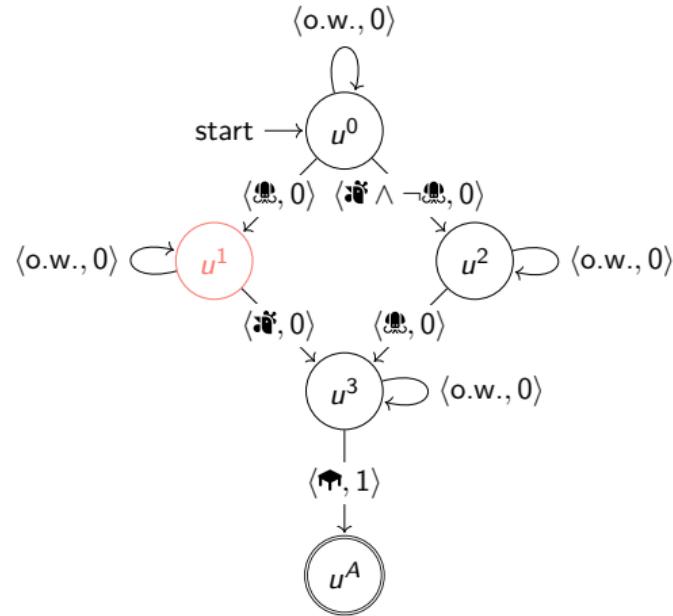
## Reward Machines

**Task** Collect  and  (in any order), then go to .



## Events

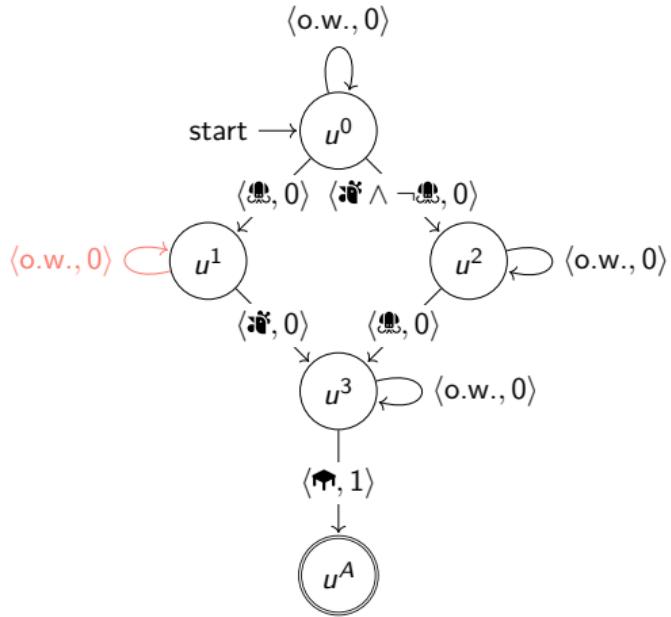
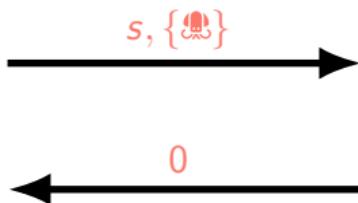
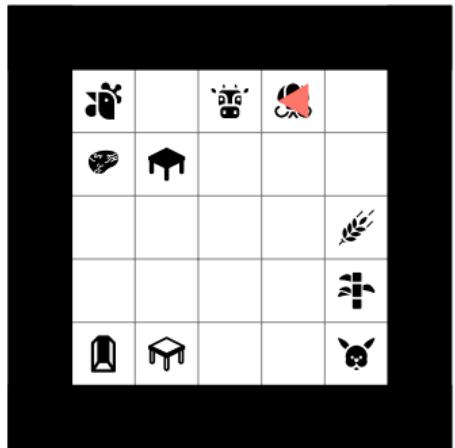
{匚, 壴, 酉, 才, 穀, 食, 肴, 鱼, 饣}



# Motivation

## Reward Machines

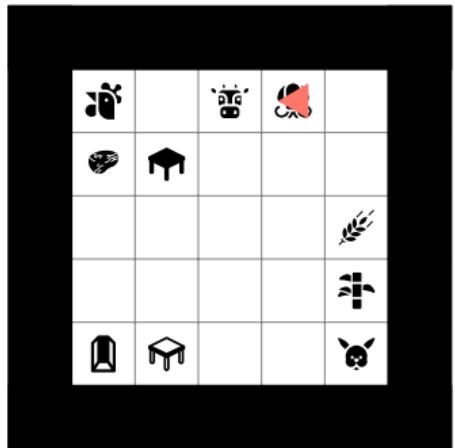
**Task** Collect and (in any order), then go to .



# Motivation

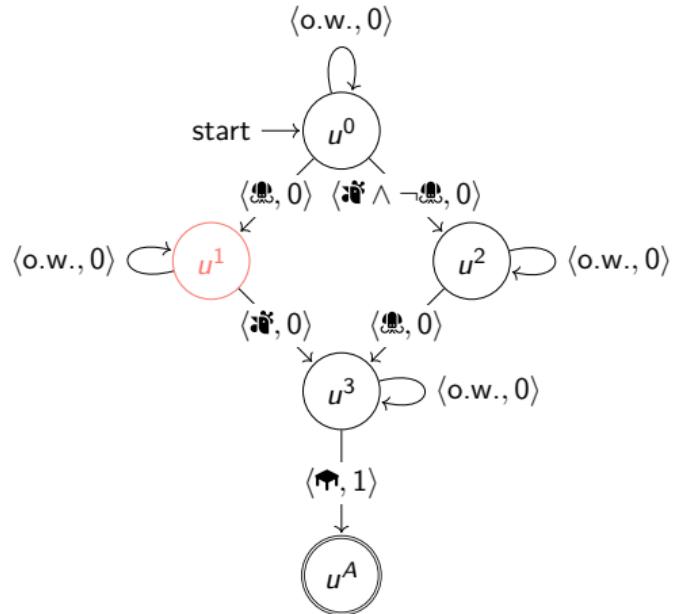
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

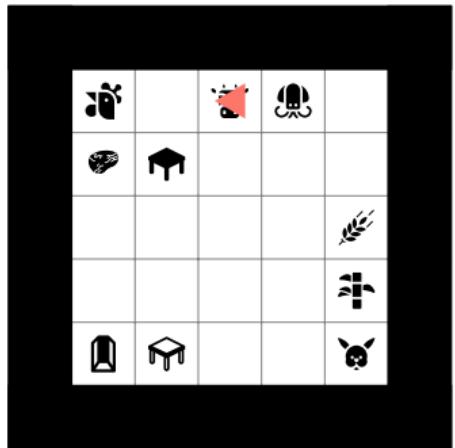
$\{\text{book}, \text{house}, \text{mushroom}, \text{apple}, \text{wheat}, \text{cross}, \text{sword}, \text{shield}\}$



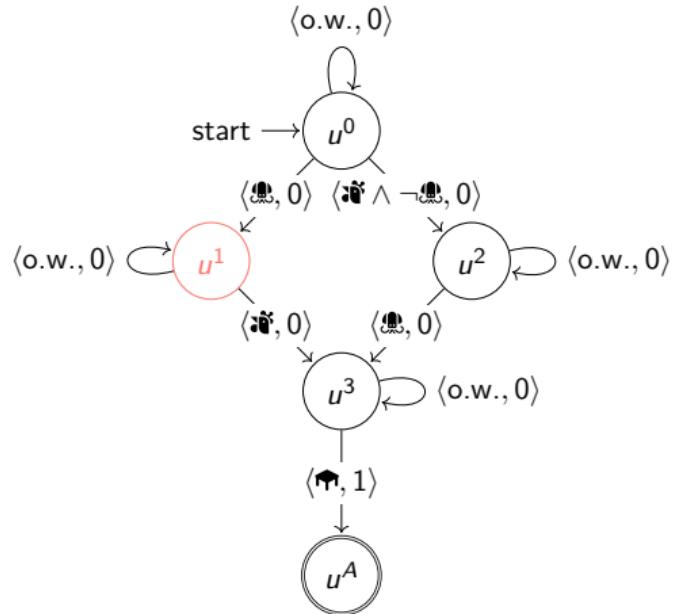
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



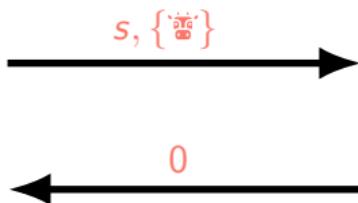
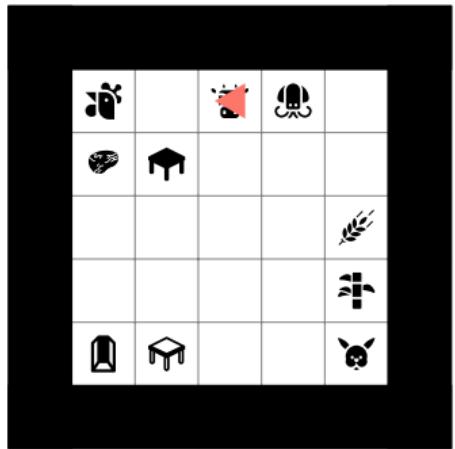
Events  
 $\{\text{apple}, \text{house}, \text{tree}, \text{flower}, \text{skull}, \text{book}, \text{star}, \text{tree}\}$



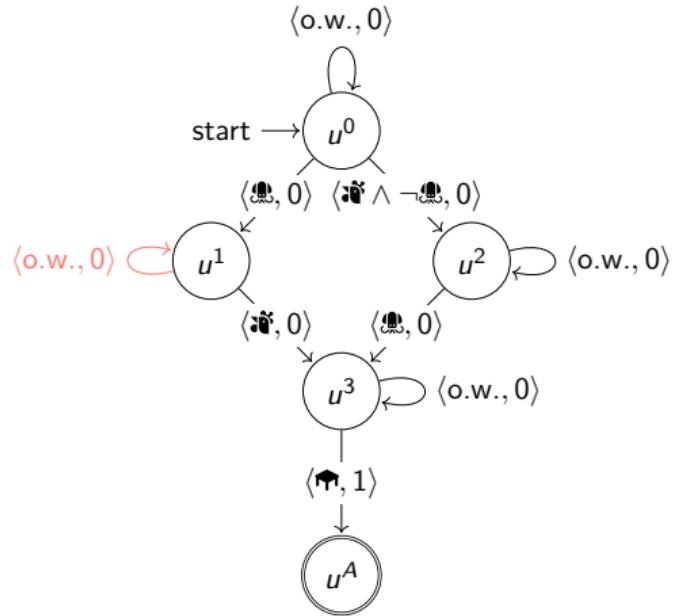
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



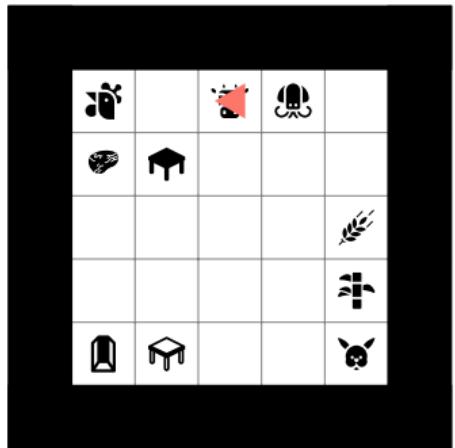
Events  
 $\{\text{apple}, \text{house}, \text{empty slot}, \text{cross}, \text{wheat}, \text{mushroom}, \text{book}, \text{key}, \text{magnet}\}$



# Motivation

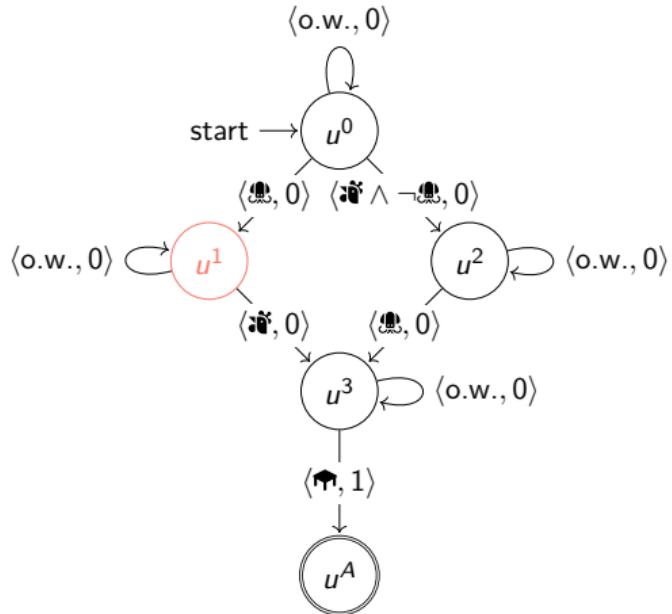
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

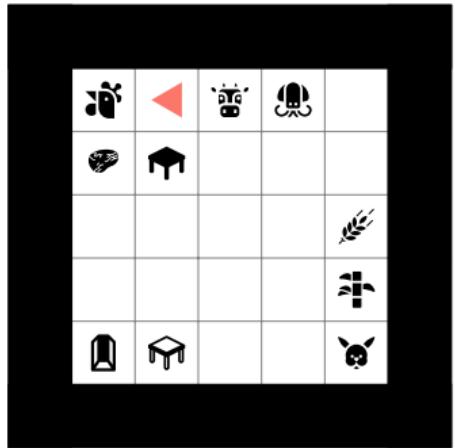
$\{\text{apple}, \text{key}, \text{house}, \text{book}, \text{wheat}, \text{skull}, \text{house}, \text{key}\}$



# Motivation

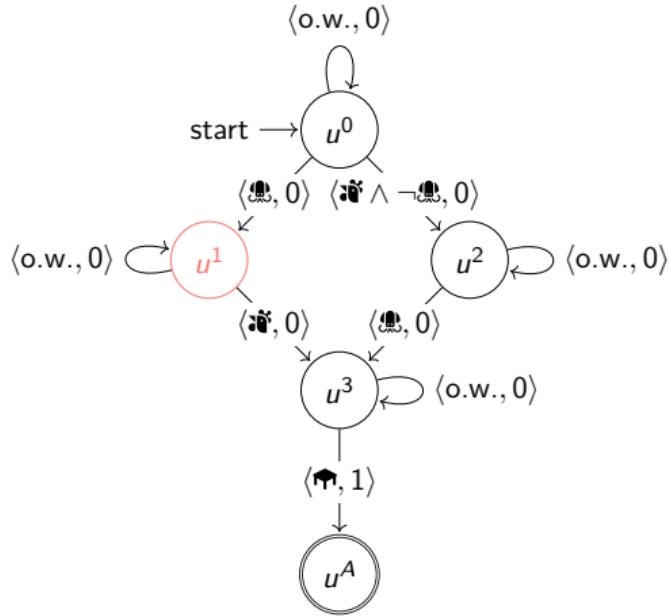
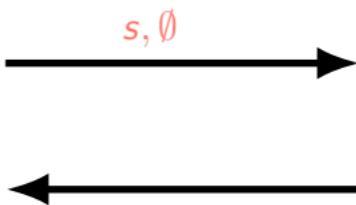
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

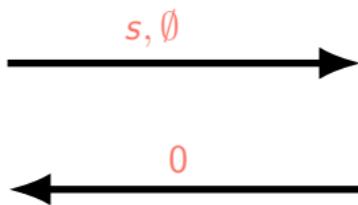
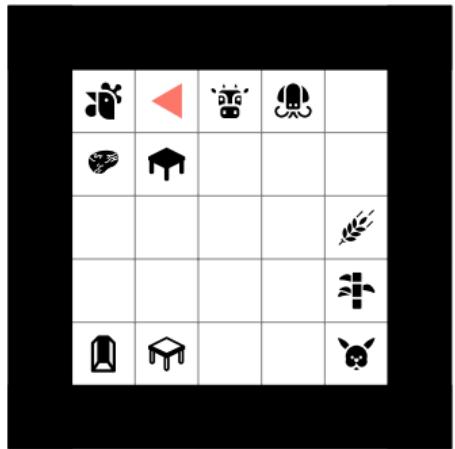
$\{\text{book}, \text{house}, \text{key}, \text{apple}, \text{wheat}, \text{key}, \text{apple}, \text{wheat}\}$



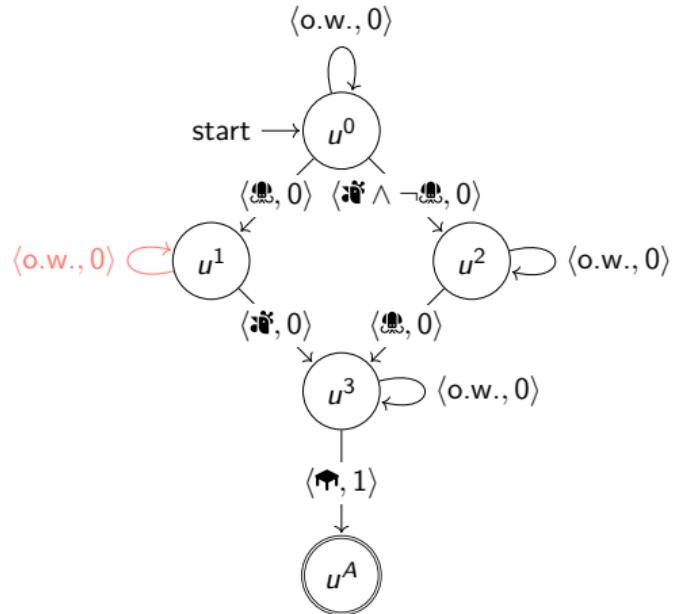
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



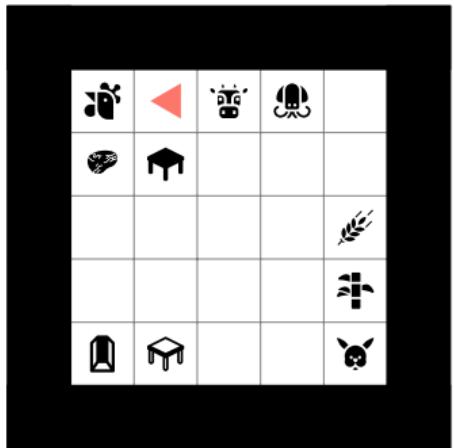
Events  
 $\{\text{apple}, \text{house}, \text{key}, \text{book}, \text{wheat}, \text{key}, \text{apple}, \text{house}, \text{key}\}$



# Motivation

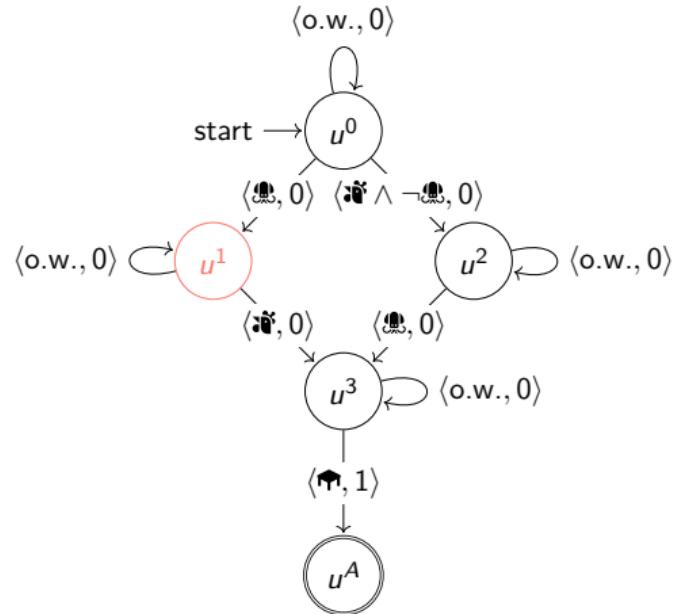
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

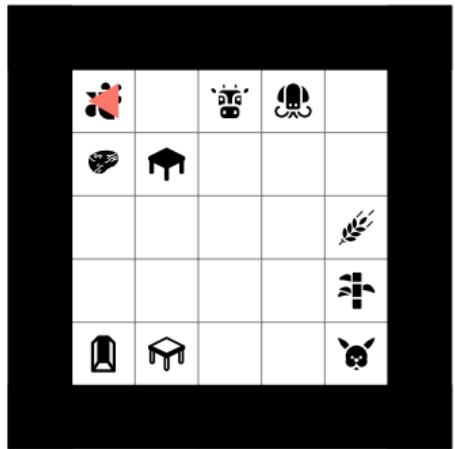
$\{\text{book}, \text{house}, \text{key}, \text{apple}, \text{wheat}, \text{skull}, \text{red arrow}\}$



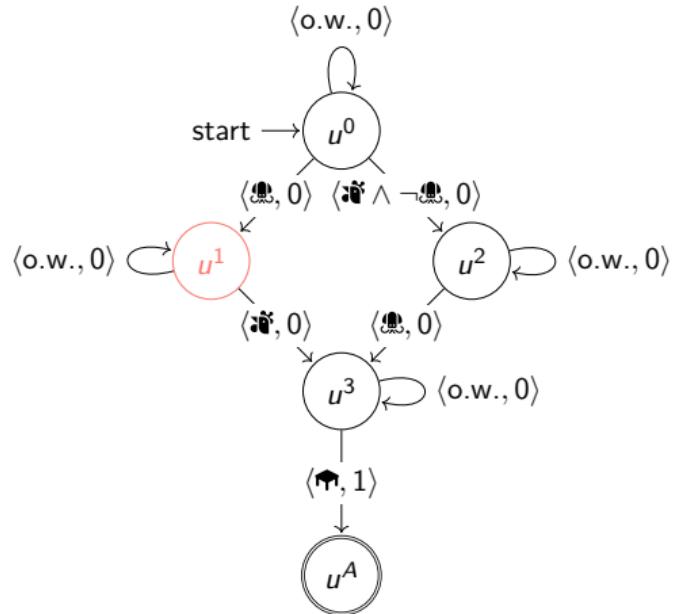
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



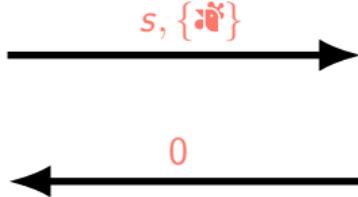
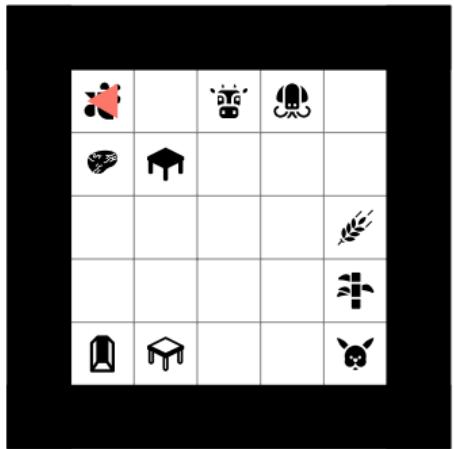
Events  
 $\{\text{apple}, \text{house}, \text{book}, \text{table}, \text{leaf}, \text{mushroom}, \text{chest}\}$



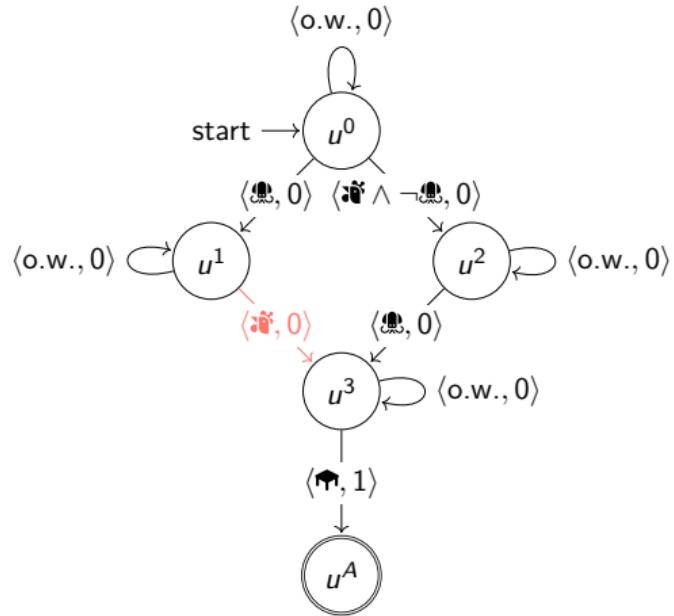
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



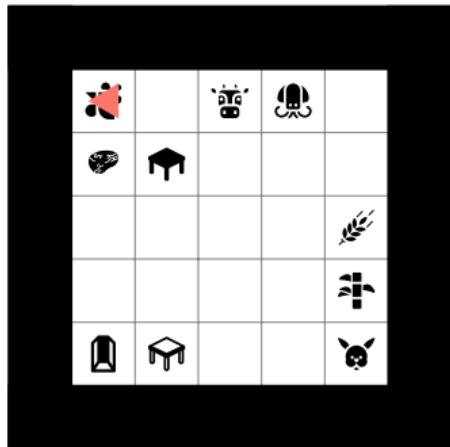
Events  
 $\{\text{apple}, \text{house}, \text{book}, \text{flower}, \text{mushroom}, \text{sword}\}$



# Motivation

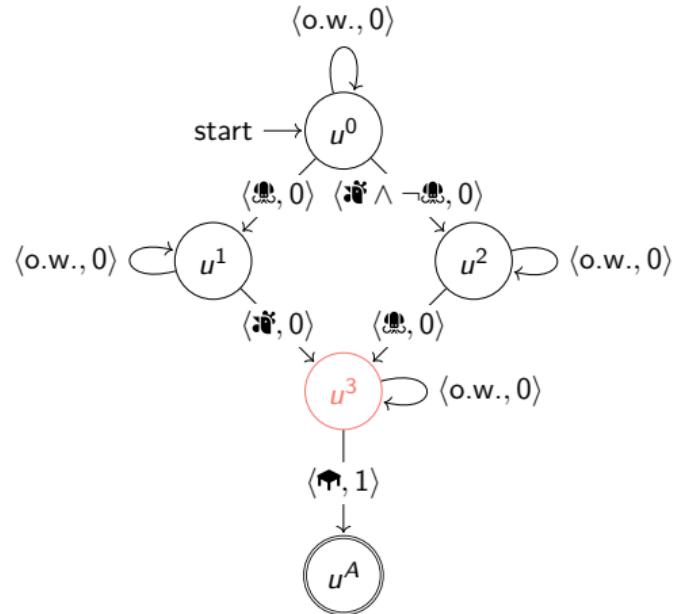
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

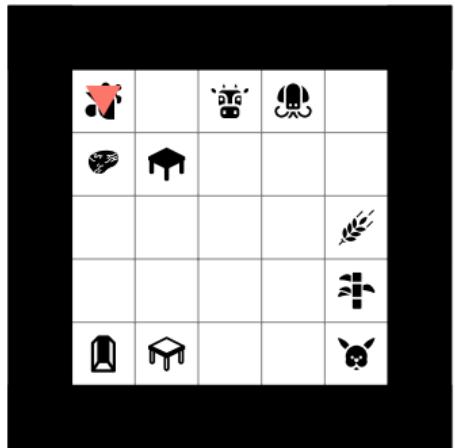
$\{\text{apple}, \text{house}, \text{table}, \text{book}, \text{chest}, \text{mushroom}, \text{person}\}$



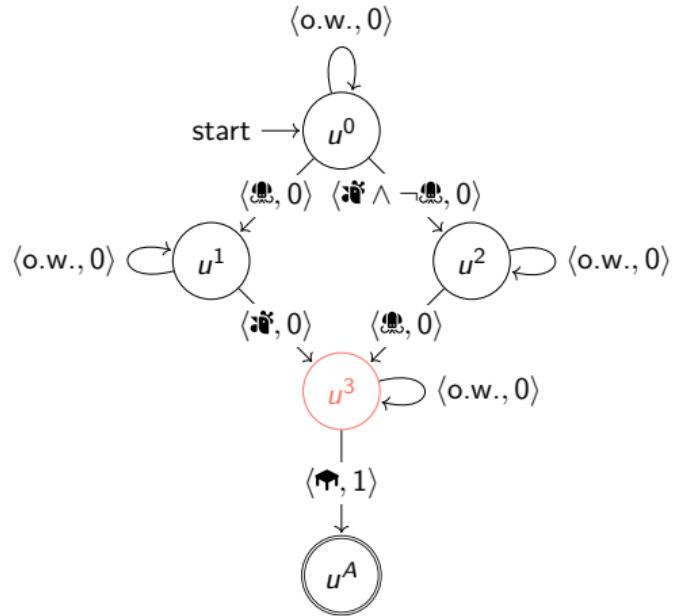
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .

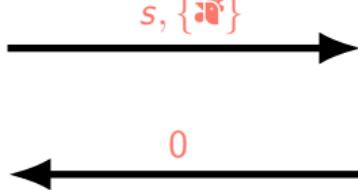


Events  
 $\{\text{apple}, \text{house}, \text{book}, \text{chair}, \text{flower}, \text{mushroom}, \text{leaf}\}$

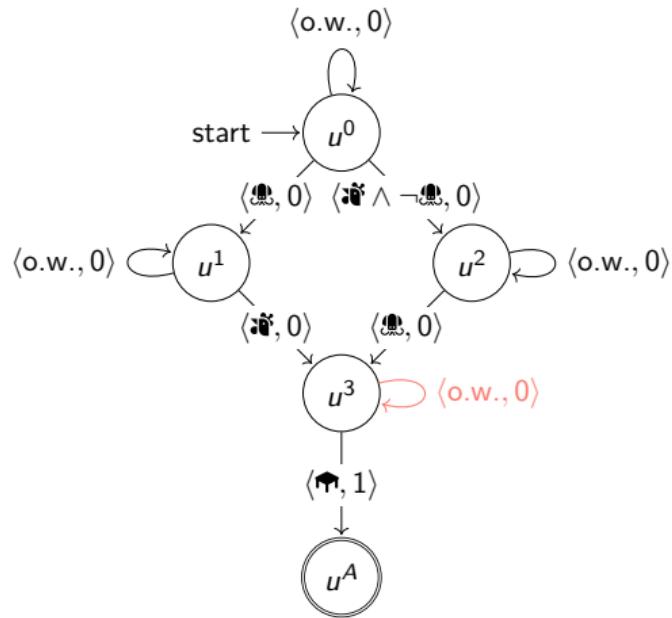


# Motivation

## Reward Machines



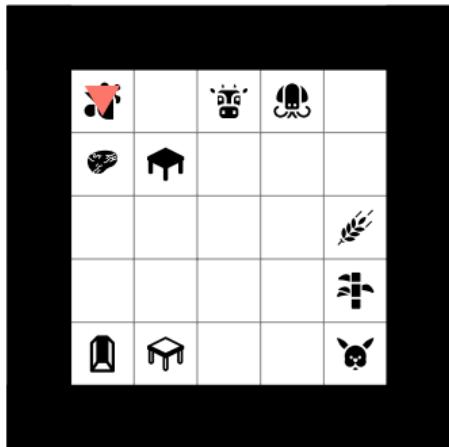
Events  
 $\{\ddagger, \text{house}, \text{book}, \text{flower}, \text{sword}\}$



# Motivation

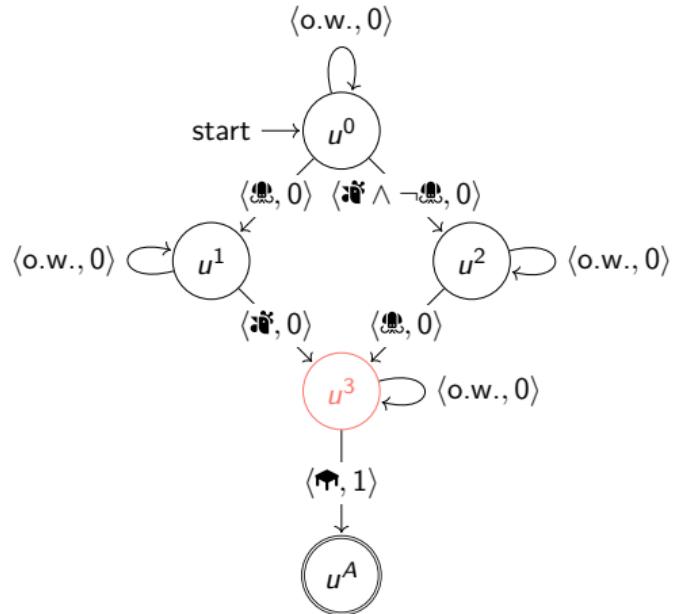
## Reward Machines

**Task** Collect and (in any order), then go to .



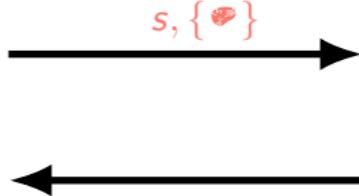
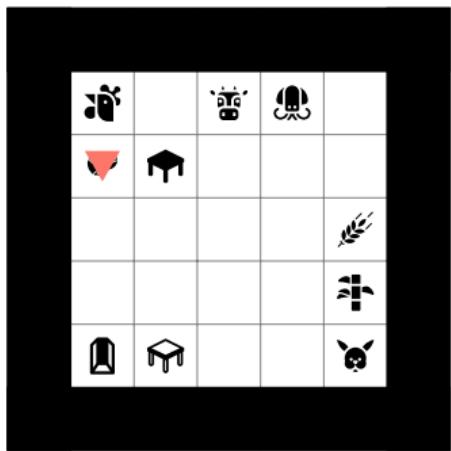
Events

$\{\text{apple}, \text{house}, \text{chest}, \text{mushroom}, \text{wheat}, \text{key}, \text{book}, \text{star}\}$

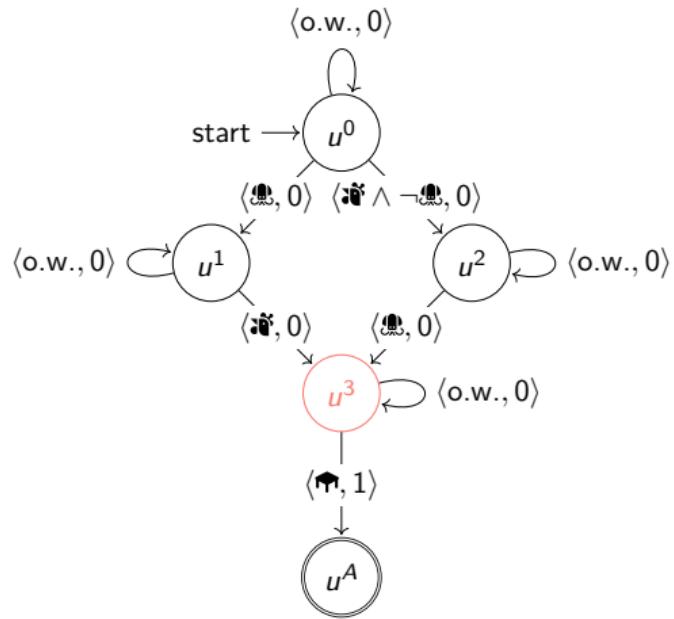


# Motivation

## Reward Machines

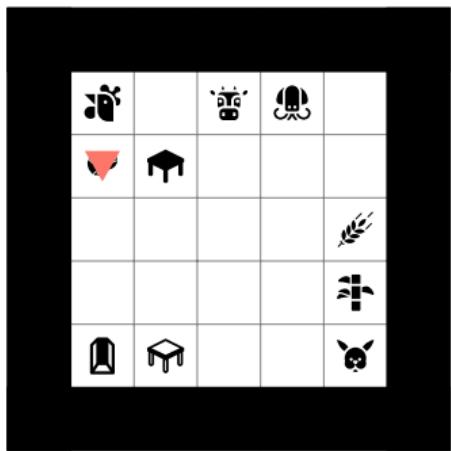


Events

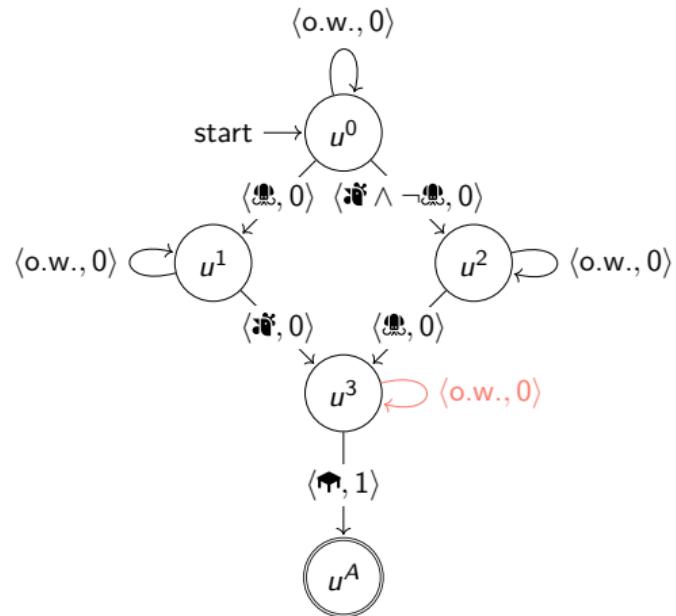
$$\{\text{shield}, \text{castle}, \text{blue sword}, \text{green sword}, \text{yellow sword}, \text{purple sword}, \text{brown sword}, \text{grey sword}, \text{white sword}, \text{black sword}\}$$


# Motivation

## Reward Machines



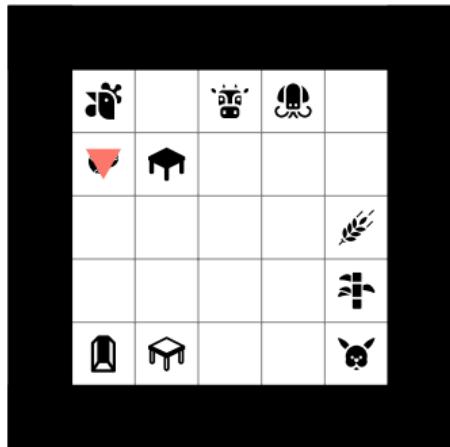
Events  
 $\{\text{house}, \text{chest}, \text{flower}, \text{cross}, \text{wheat}, \text{key}, \text{apple}, \text{skull}, \text{map}\}$



# Motivation

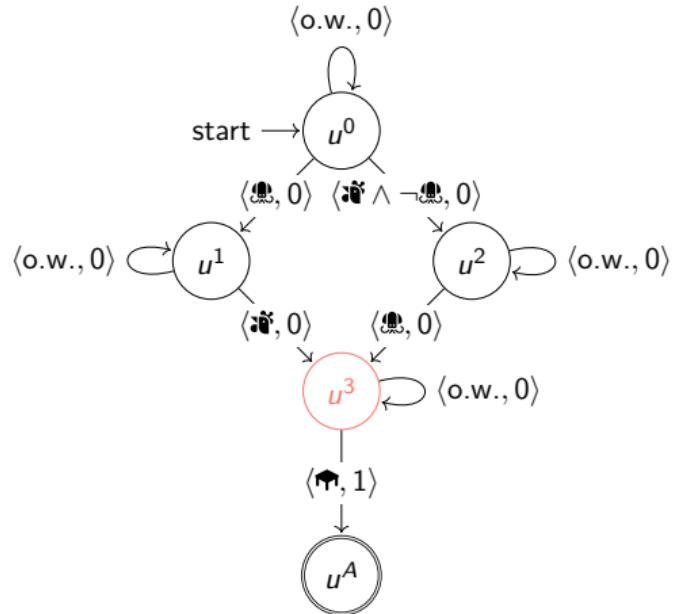
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

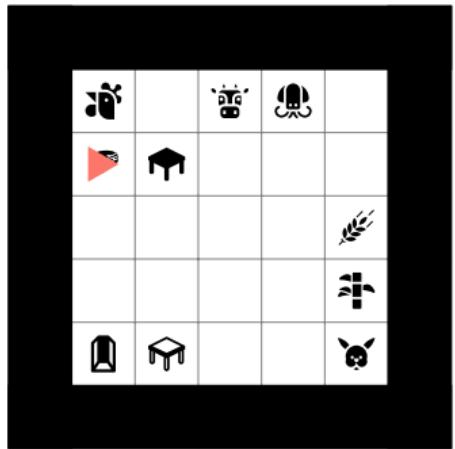
$\{\text{apple}, \text{house}, \text{key}, \text{chest}, \text{wheat}, \text{cross}, \text{book}\}$



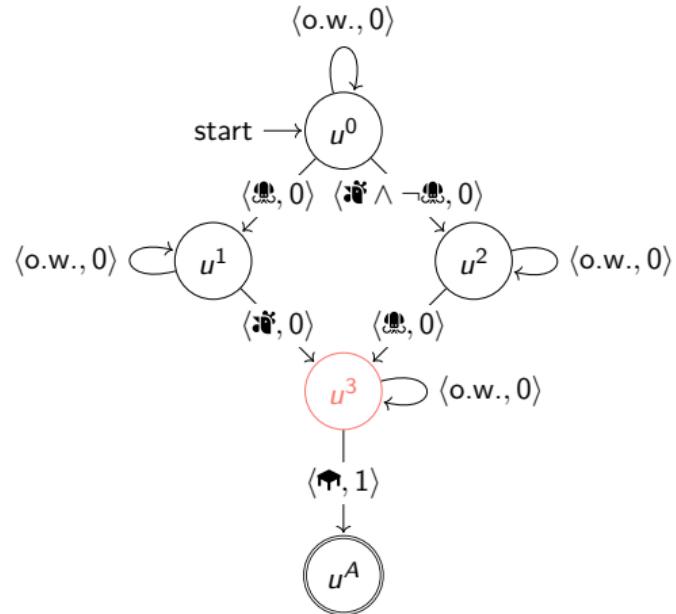
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



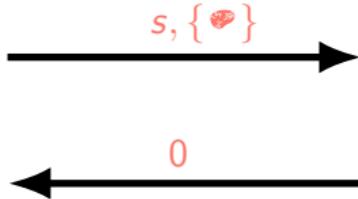
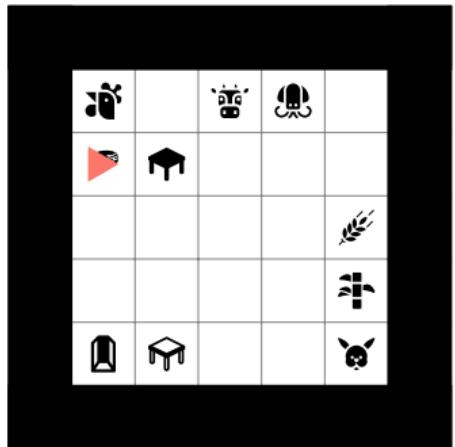
Events  
 $\{\text{key}, \text{house}, \text{chest}, \text{table}, \text{plant}, \text{coin}, \text{skull}\}$



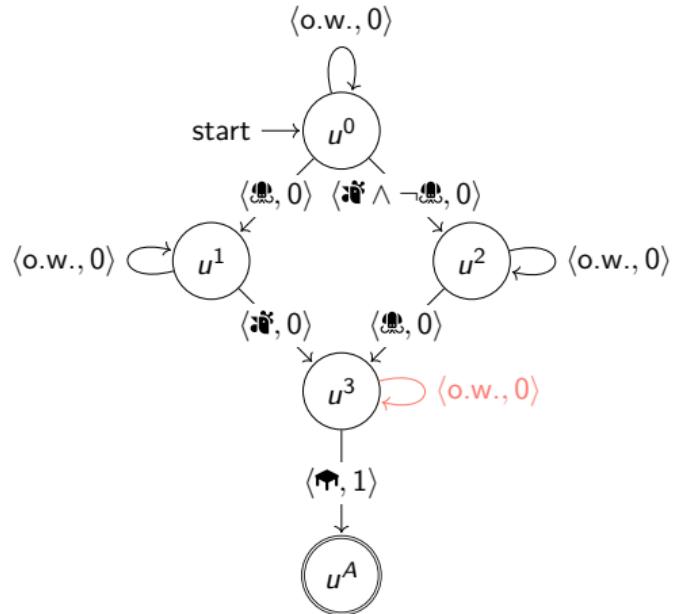
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



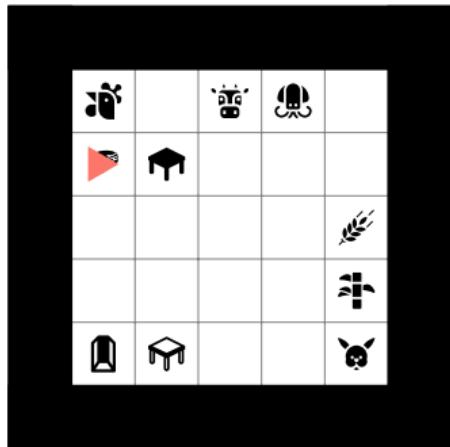
Events  
 $\{\text{key}, \text{house}, \text{table}, \text{wheat}, \text{coin}, \text{key}, \text{coin}, \text{house}, \text{table}\}$



# Motivation

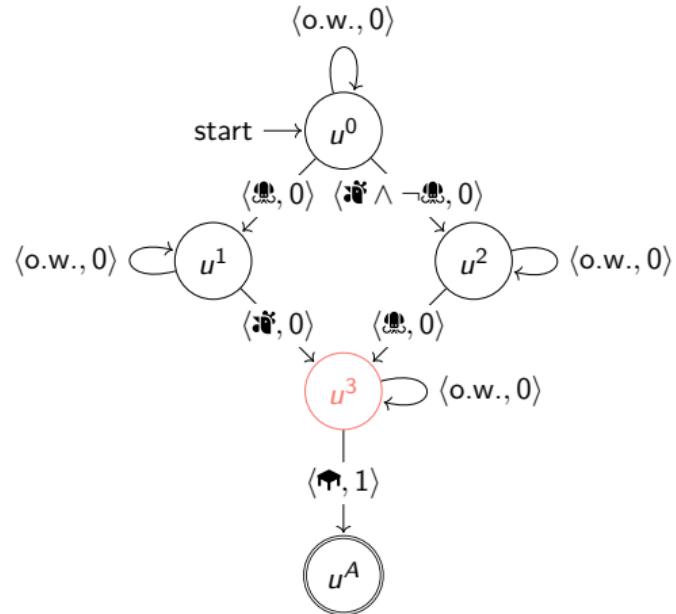
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

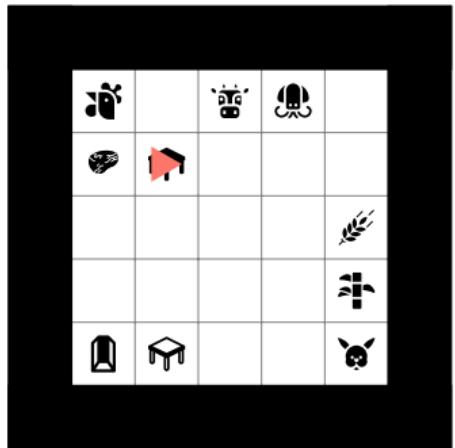
$\{\text{apple}, \text{key}, \text{house}, \text{wheat}, \text{key}, \text{apple}, \text{house}, \text{house}\}$



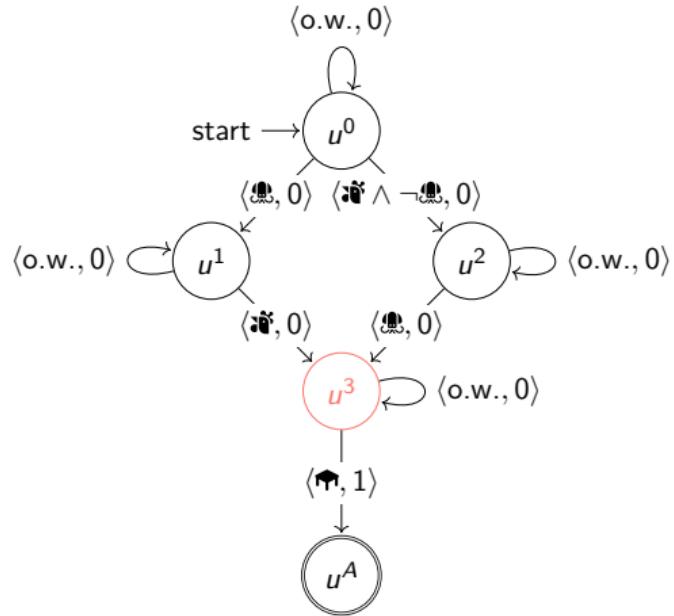
# Motivation

## Reward Machines

**Task** Collect and (in any order), then go to .



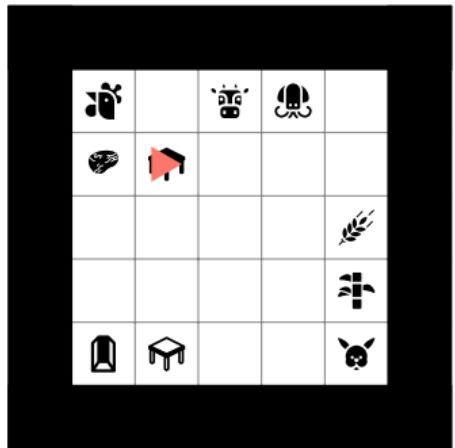
Events  
 $\{\text{Apple}, \text{House}, \text{Mushroom}, \text{Table}, \text{Wheat}, \text{Book}, \text{Skull}, \text{Red Arrow}\}$



# Motivation

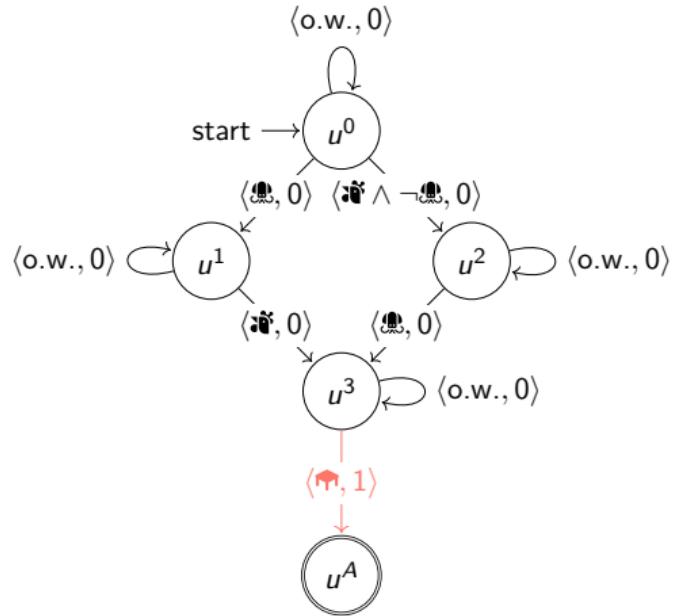
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

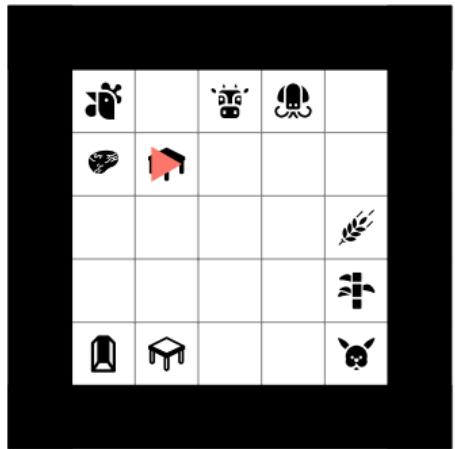
$\{\text{Apple}, \text{House}, \text{Table}, \text{Book}, \text{Wheat}, \text{Apple},$   
 $\text{House}, \text{Skull}, \text{Book}, \text{House}\}$



# Motivation

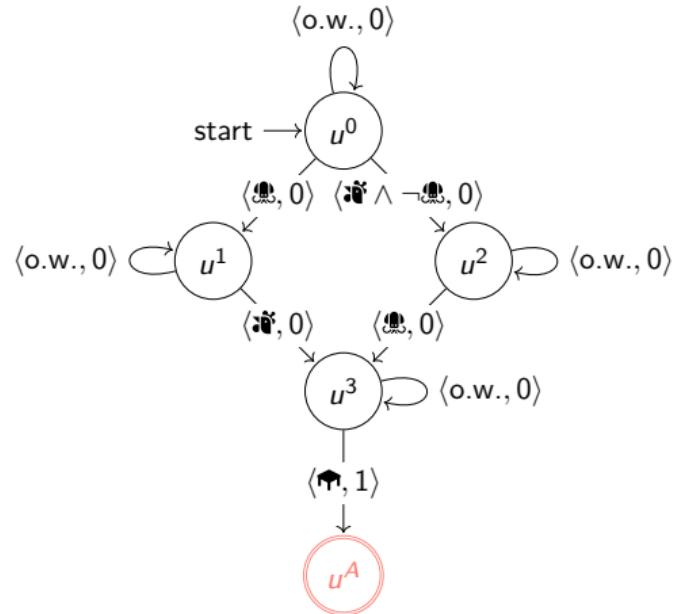
## Reward Machines

**Task** Collect and (in any order), then go to .



Events

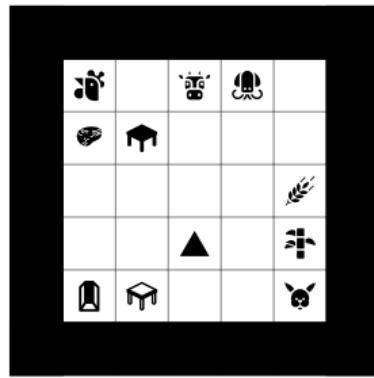
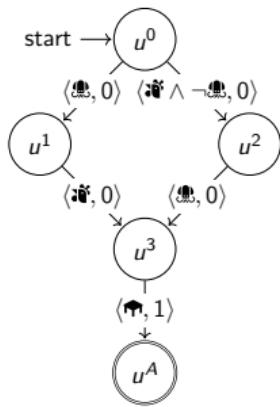
{, , , , , , , }



# Motivation

## Reward Machines – Exploitation

- RMs enable *task decomposition*: each formula is an independently solvable subtask.



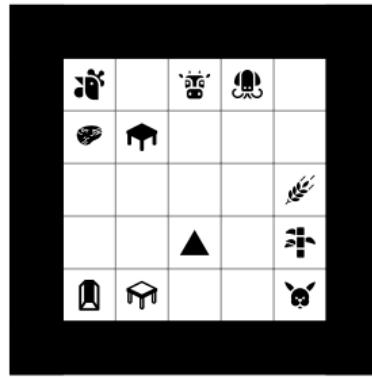
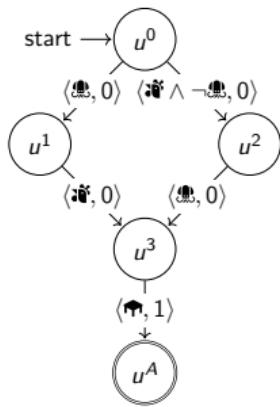
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

Toro Icarte et al. "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning". ICML, 2018.

# Motivation

## Reward Machines – Exploitation

- RM enable *task decomposition*: each formula is an independently solvable subtask.
- Decision-making can happen at two *hierarchical levels*:



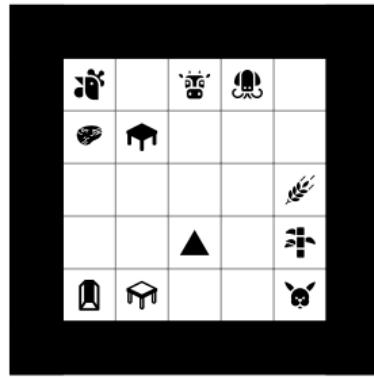
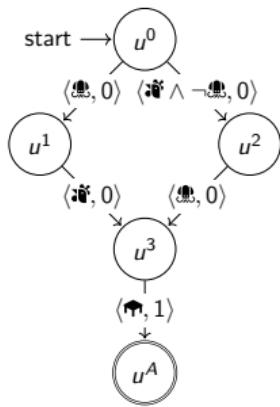
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

Toro Icarte et al. "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning". ICML, 2018.

# Motivation

## Reward Machines – Exploitation

- RM enable *task decomposition*: each formula is an independently solvable subtask.
- Decision-making can happen at two *hierarchical levels*:
  - ① From a state, choose a formula to (eventually) reach  $u^A$ .



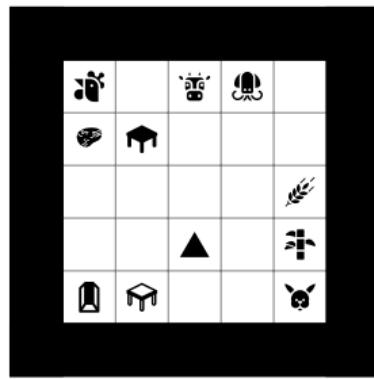
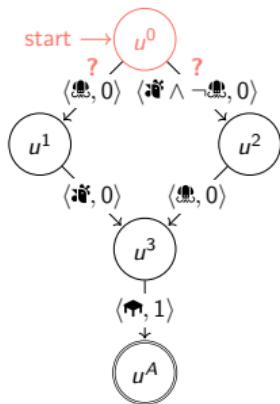
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

Toro Icarte et al. "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning". ICML, 2018.

# Motivation

## Reward Machines – Exploitation

- RM enable *task decomposition*: each formula is an independently solvable subtask.
- Decision-making can happen at two *hierarchical levels*:
  - ① From a state, choose a formula to (eventually) reach  $u^A$ .



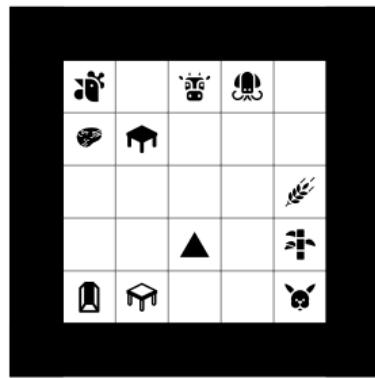
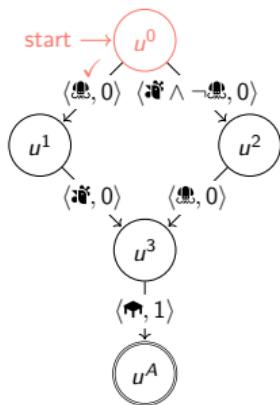
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

Toro Icarte et al. "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning". ICML, 2018.

# Motivation

## Reward Machines – Exploitation

- RM enable *task decomposition*: each formula is an independently solvable subtask.
- Decision-making can happen at two *hierarchical levels*:
  - ① From a state, choose a formula to (eventually) reach  $u^A$ .



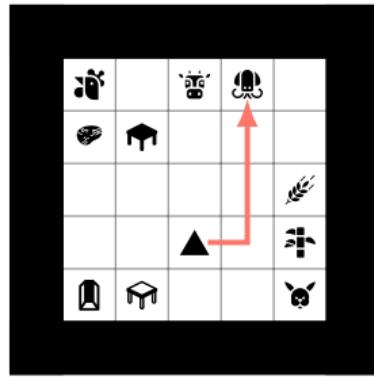
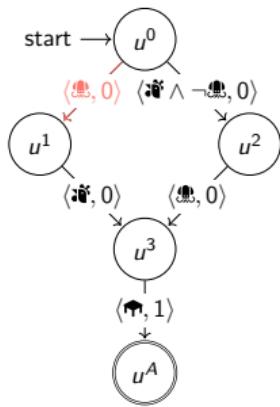
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

Toro Icarte et al. "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning". ICML, 2018.

# Motivation

## Reward Machines – Exploitation

- RM enable *task decomposition*: each formula is an independently solvable subtask.
- Decision-making can happen at two *hierarchical levels*:
  - ① From a state, choose a formula to (eventually) reach  $u^A$ .
  - ② Given a formula, choose an action to (eventually) satisfy it.



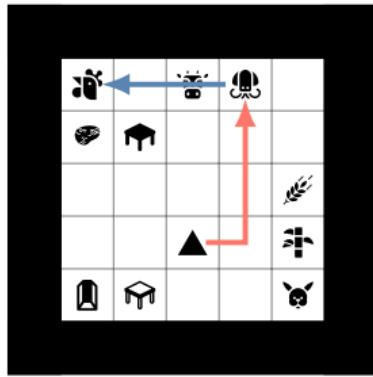
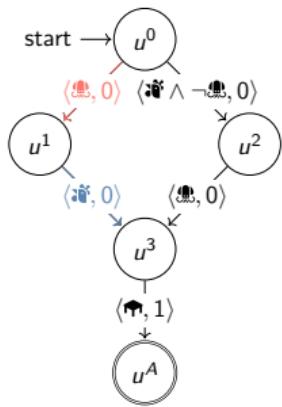
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

Toro Icarte et al. "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning". ICML, 2018.

# Motivation

## Reward Machines – Exploitation

- RM enable *task decomposition*: each formula is an independently solvable subtask.
- Decision-making can happen at two *hierarchical levels*:
  - ① From a state, choose a formula to (eventually) reach  $u^A$ .
  - ② Given a formula, choose an action to (eventually) satisfy it.



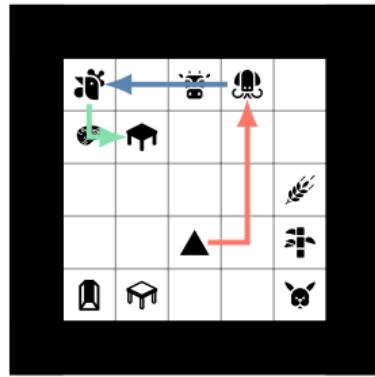
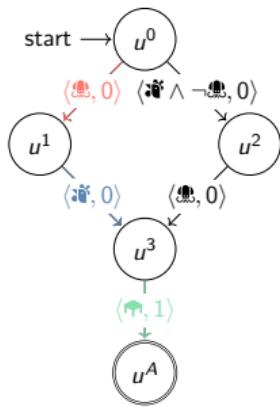
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

Toro Icarte et al. "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning". ICML, 2018.

# Motivation

## Reward Machines – Exploitation

- RM enable *task decomposition*: each formula is an independently solvable subtask.
- Decision-making can happen at two *hierarchical levels*:
  - ① From a state, choose a formula to (eventually) reach  $u^A$ .
  - ② Given a formula, choose an action to (eventually) satisfy it.



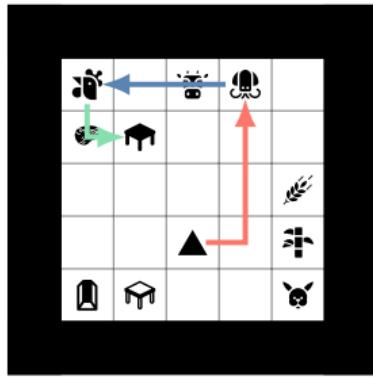
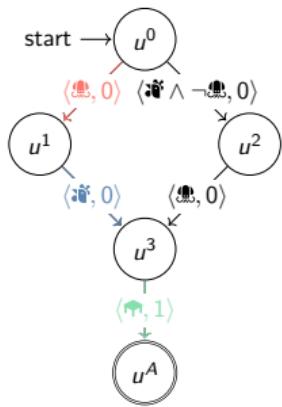
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

Toro Icarte et al. "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning". ICML, 2018.

# Motivation

## Reward Machines – Exploitation

- RM enable *task decomposition*: each formula is an independently solvable subtask.
- Decision-making can happen at two *hierarchical levels*:
  - ① From a state, choose a formula to (eventually) reach  $u^A$ .
  - ② Given a formula, choose an action to (eventually) satisfy it.



How can we make RMs **reusable**  
(i.e., independently solvable  
subtasks)?

Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

Toro Icarte et al. "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning". ICML, 2018.

# Motivation

## Reward Machines – Learning I



Furelos-Blanco et al. "Induction of Subgoal Automata for Reinforcement Learning". AAAI, 2020.

Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

Toro Icarte et al. "Learning Reward Machines for Partially Observable Reinforcement Learning". NeurIPS, 2019.

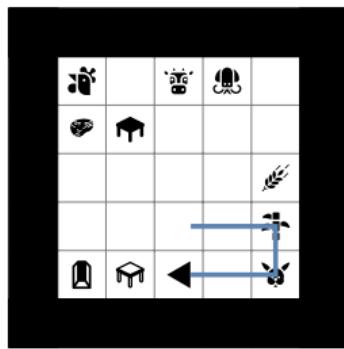
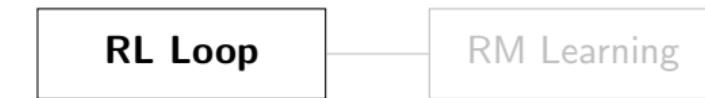
Gaon and Brafman. "Reinforcement Learning with Non-Markovian Rewards". AAAI, 2020.

Xu et al. "Joint Inference of Reward Machines and Policies for Reinforcement Learning". ICAPS, 2020.

Hasanbeig et al. "DeepSynth: Automata Synthesis for Automatic Task Segmentation in Deep Reinforcement Learning". AAAI, 2021.

# Motivation

## Reward Machines – Learning I



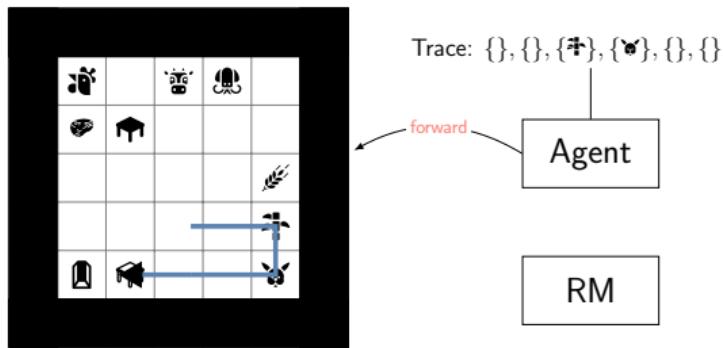
Trace: {}, {}, {⊕}, {⊗}, {}, {}, {}



- The agent attempts to achieve the task's goal.
- The agent maintains a *trace* of the events observed so far.

# Motivation

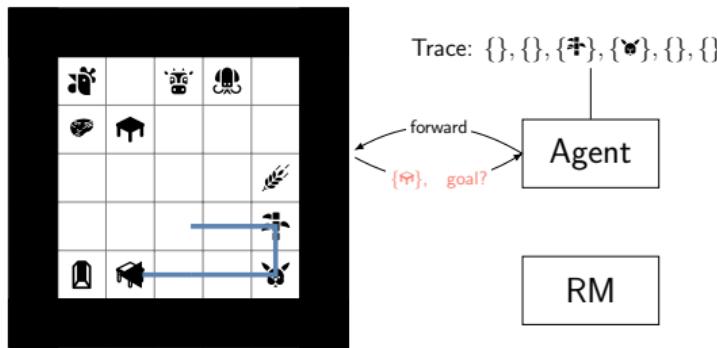
## Reward Machines – Learning I



- The agent attempts to achieve the task's goal.
- The agent maintains a *trace* of the events observed so far.

# Motivation

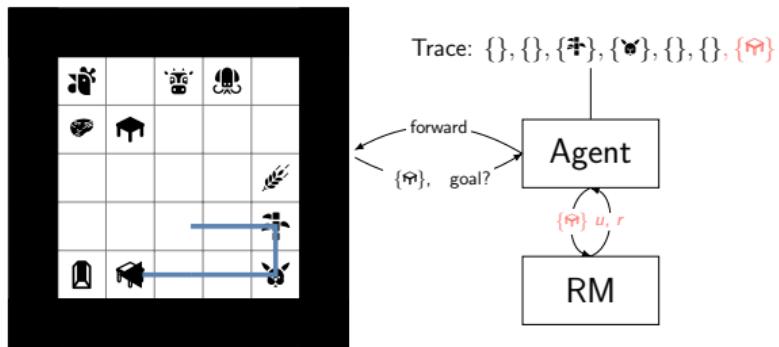
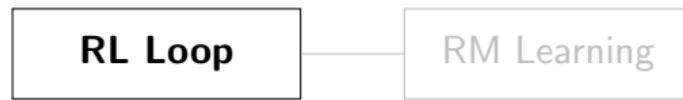
## Reward Machines – Learning I



- The agent attempts to achieve the task's goal.
- The agent maintains a *trace* of the events observed so far.

# Motivation

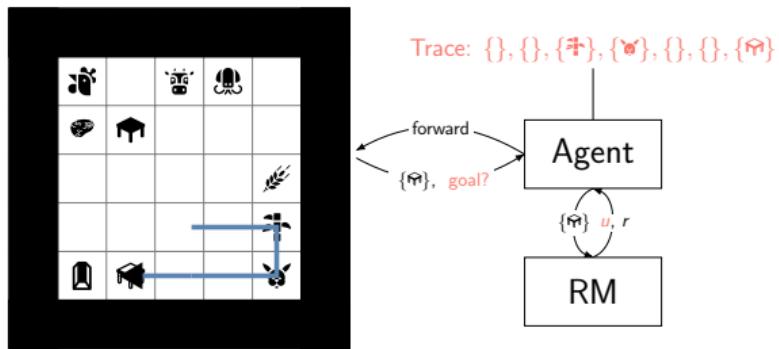
## Reward Machines – Learning I



- The agent attempts to achieve the task's goal.
- The agent maintains a *trace* of the events observed so far.

# Motivation

## Reward Machines – Learning I

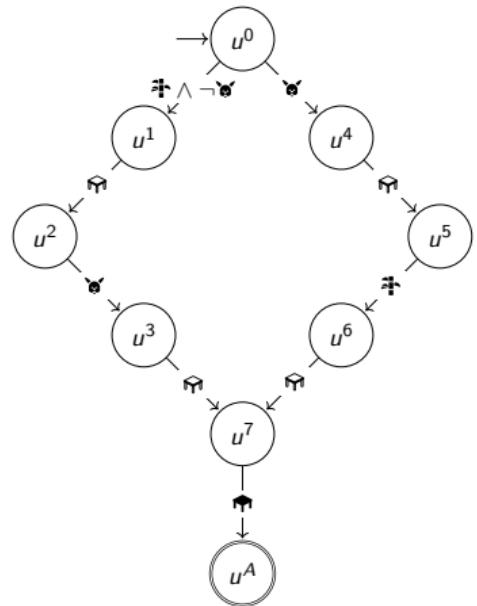


- A *new* RM is learned if the trace is a *counterexample* (e.g., reaches the task's goal but not the accepting state).

# Motivation

## Reward Machines – Learning II

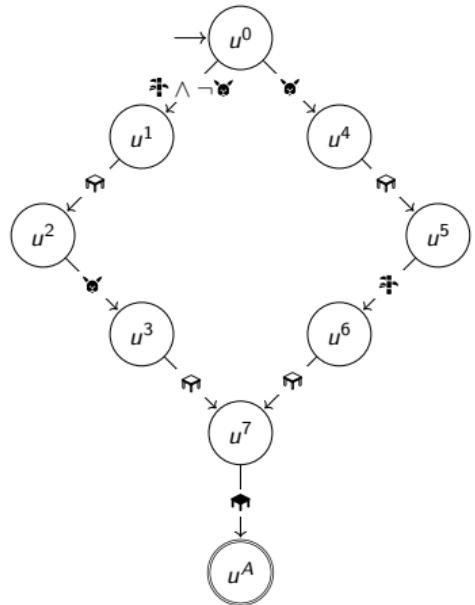
- Learning of *minimal* RM (i.e., with the fewest possible states) scales poorly with the number of states.



# Motivation

## Reward Machines – Learning II

- Learning of *minimal* RMs (i.e., with the fewest possible states) scales poorly with the number of states.



Can we build large RMs by composing small but easier to learn RMs?

# Motivation

## Reward Machines – Research Questions

### Question #1

How can we make RMs reusable (i.e., independently solvable subtasks)?

### Question #2

Can we build large RMs by composing small but easier to learn RMs?

# Motivation

## Reward Machines – Research Questions

### Question #1

How can we make RMs reusable (i.e., independently solvable subtasks)?

### Question #2

Can we build large RMs by composing small but easier to learn RMs?

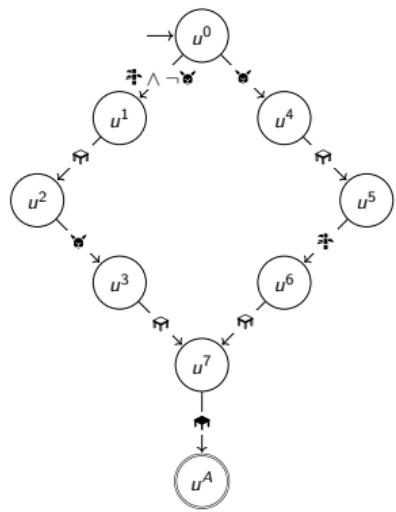
### Our Approach

Construct hierarchies of reward machines!

# Hierarchies of Reward Machines

## Formalism I

RM

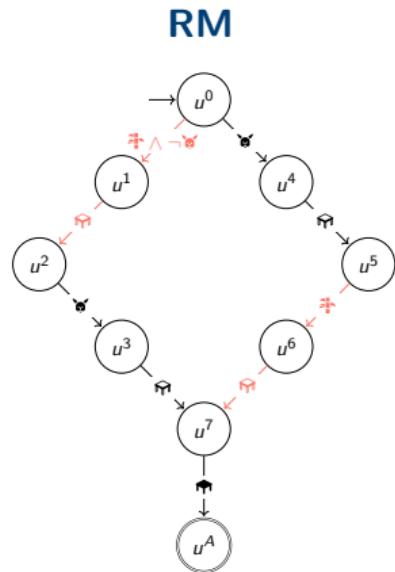


HRM

HRM

# Hierarchies of Reward Machines

## Formalism I

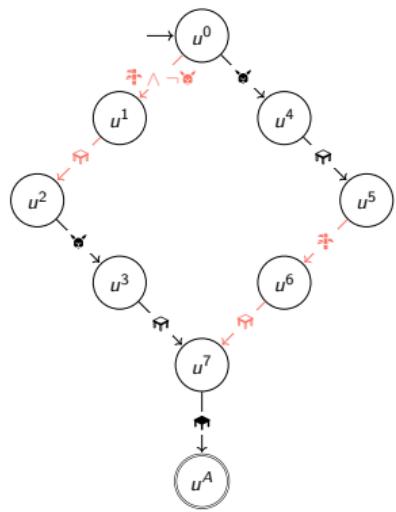


## HRM

# Hierarchies of Reward Machines

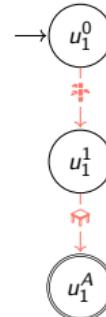
## Formalism I

RM



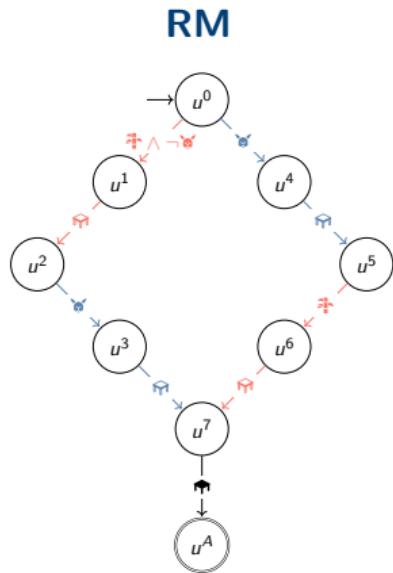
HRM

$M_1$

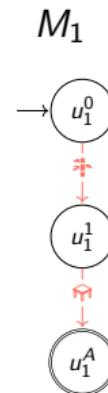


# Hierarchies of Reward Machines

## Formalism I

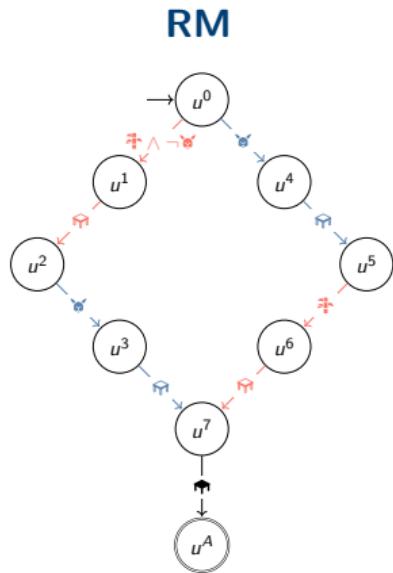


## HRM



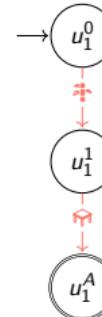
# Hierarchies of Reward Machines

## Formalism I

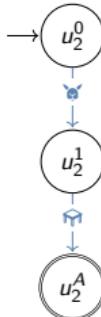


**HRM**

$M_1$

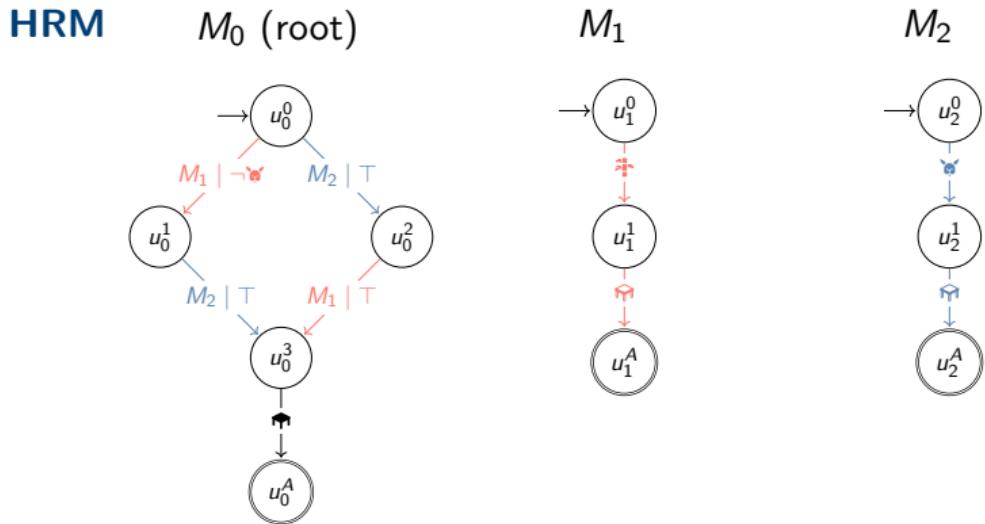
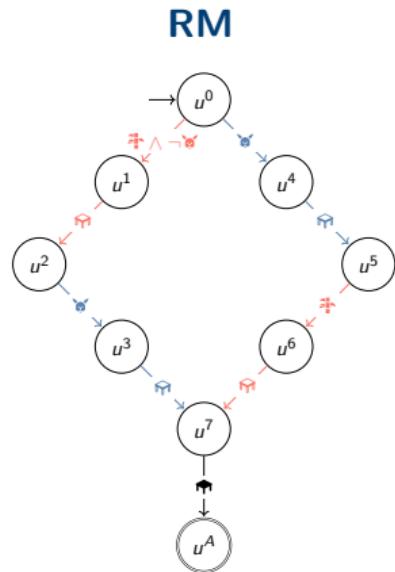


$M_2$



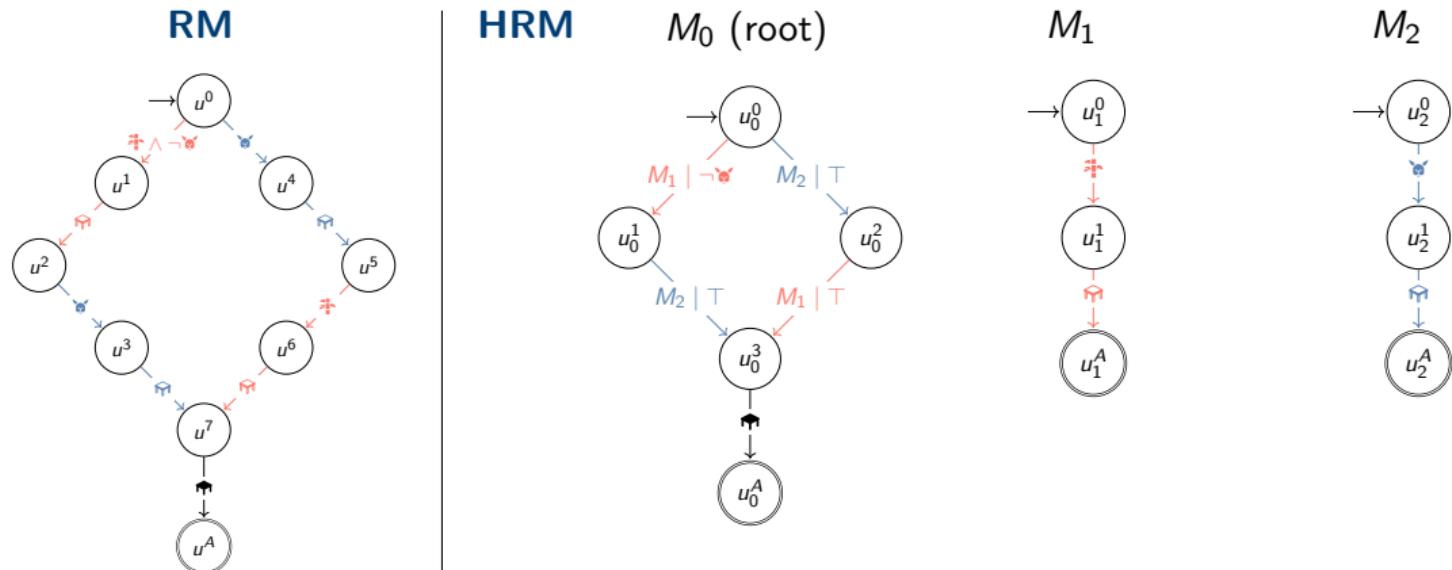
# Hierarchies of Reward Machines

## Formalism I



# Hierarchies of Reward Machines

## Formalism I

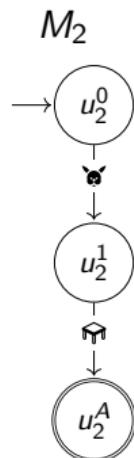
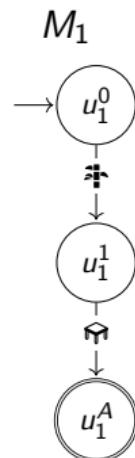
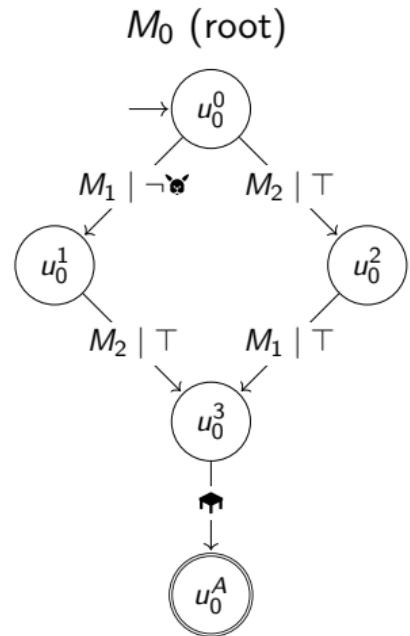


## Properties

- ① Given an HRM, there exists an *equivalent* RM.
- ② Given an HRM, an equivalent RM *may* have *exponentially* more states and edges.

# Hierarchies of Reward Machines

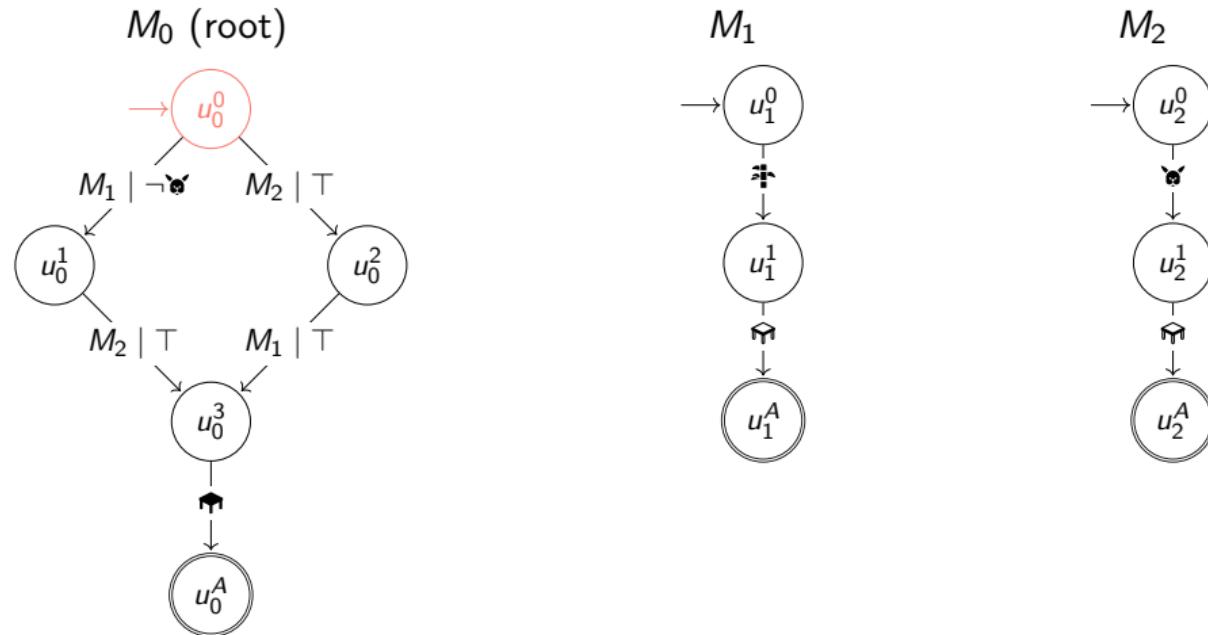
## Formalism II



Trace =  $\langle$   
Stack = []

# Hierarchies of Reward Machines

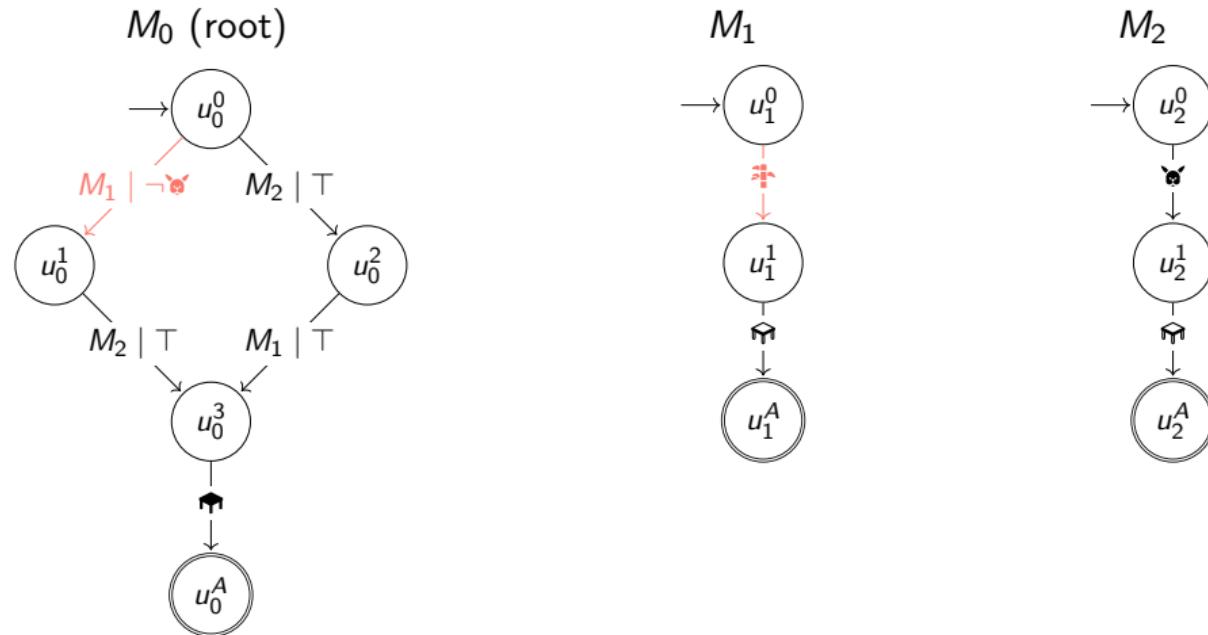
## Formalism II



Trace =  $\langle$   
Stack = []

# Hierarchies of Reward Machines

## Formalism II

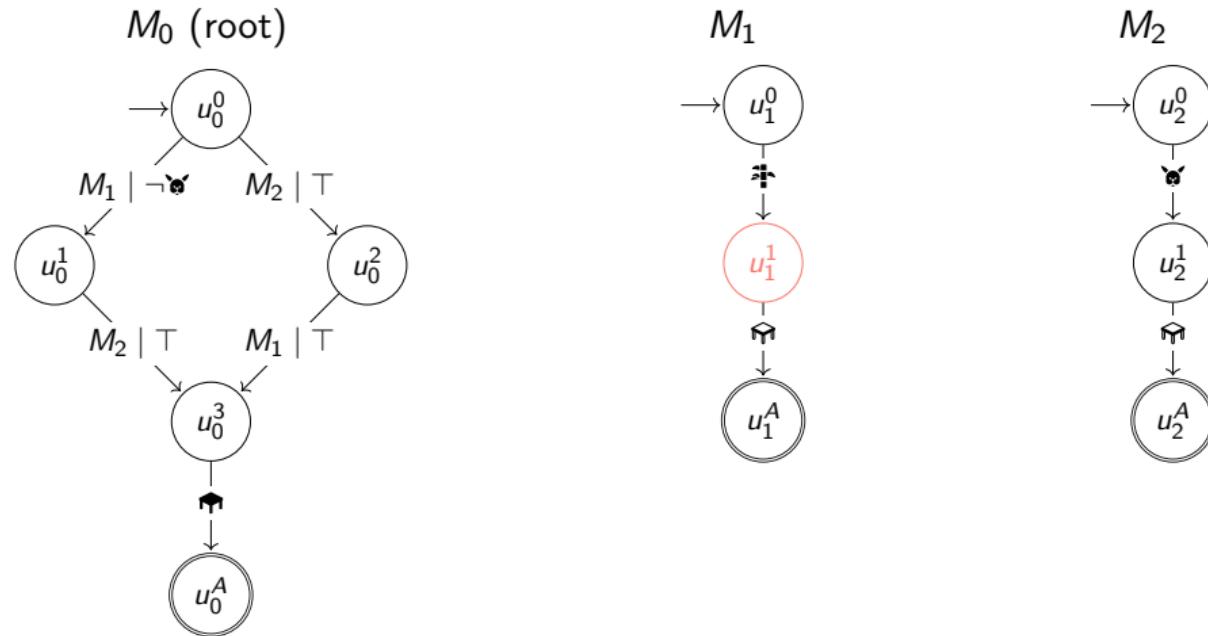


Trace =  $\langle \{\oplus\} \rangle$ ,

Stack = []

# Hierarchies of Reward Machines

## Formalism II

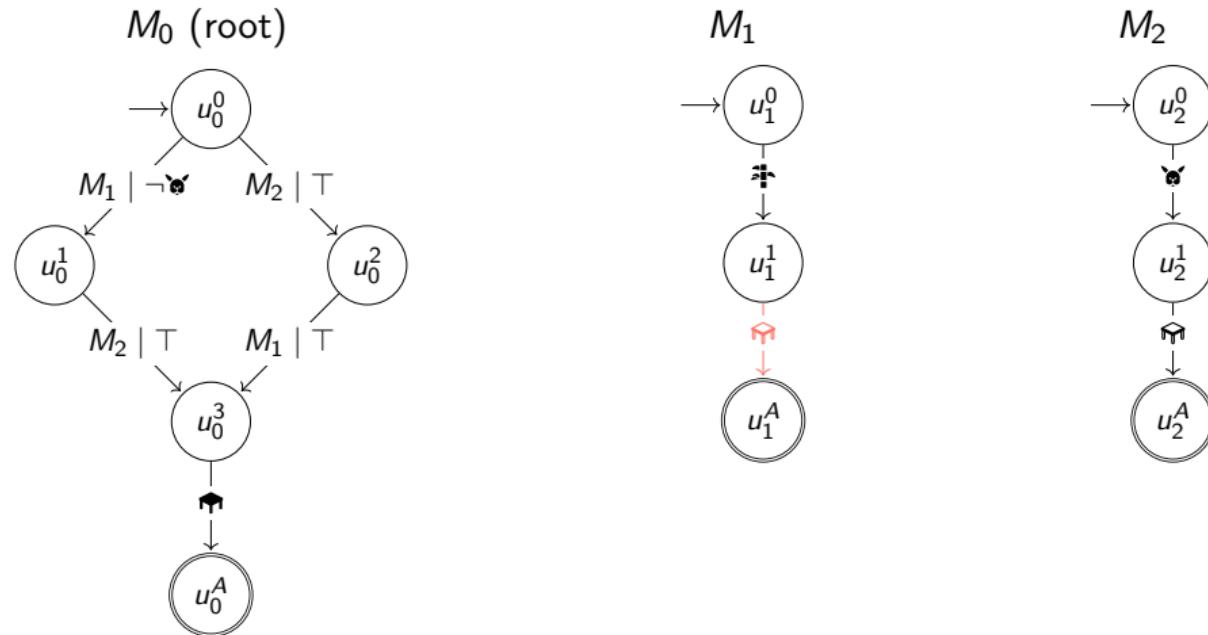


Trace =  $\langle \{\oplus\} \rangle$ ,

Stack =  $[\langle M_0, u_0^1 \rangle]$

# Hierarchies of Reward Machines

## Formalism II

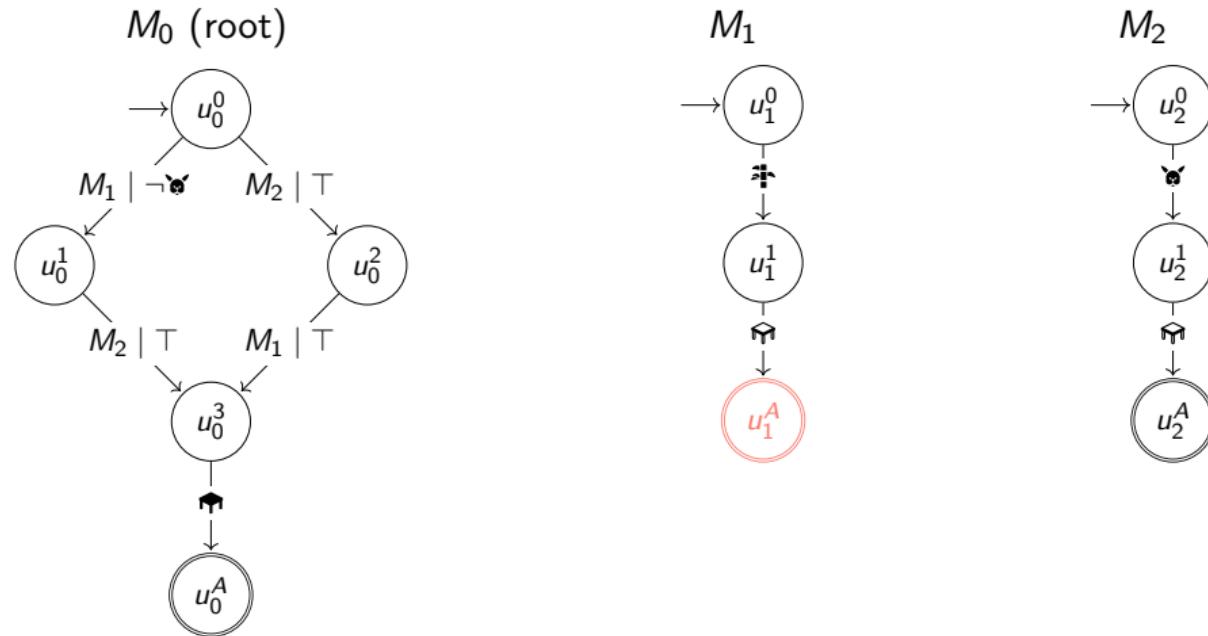


Trace =  $\langle \{\text{↑}\}, \{\text{↑}\},$

Stack =  $[\langle M_0, u_0^1 \rangle]$

# Hierarchies of Reward Machines

## Formalism II

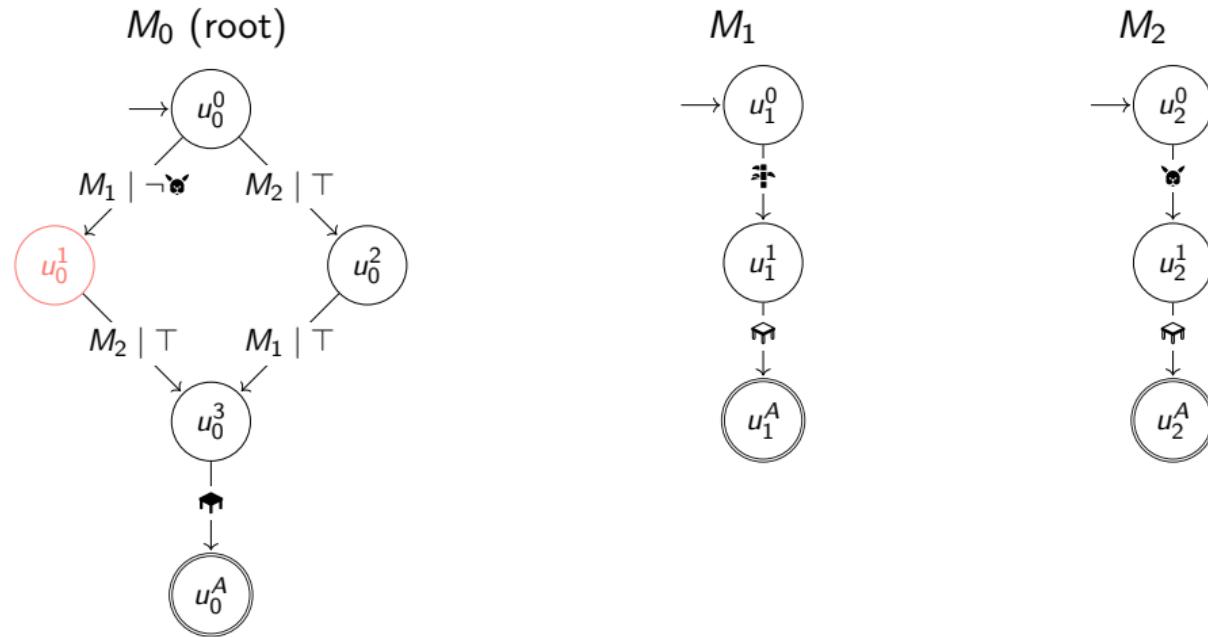


Trace =  $\langle \{\bowtie\}, \{\bowtie\},$

Stack =  $[\langle M_0, u_0^1 \rangle]$

# Hierarchies of Reward Machines

## Formalism II

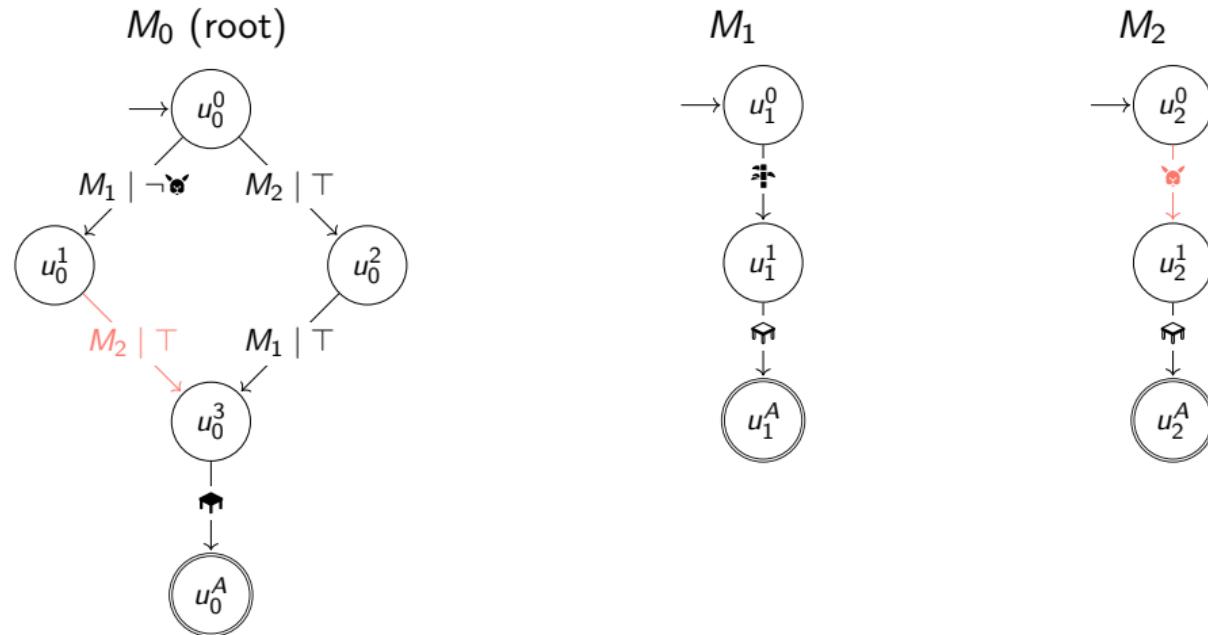


$$\text{Trace} = \langle \{\bowtie\}, \{\bowtie\},$$

$$\text{Stack} = [(\cancel{M_0}, \cancel{u_0^1})]$$

# Hierarchies of Reward Machines

## Formalism II

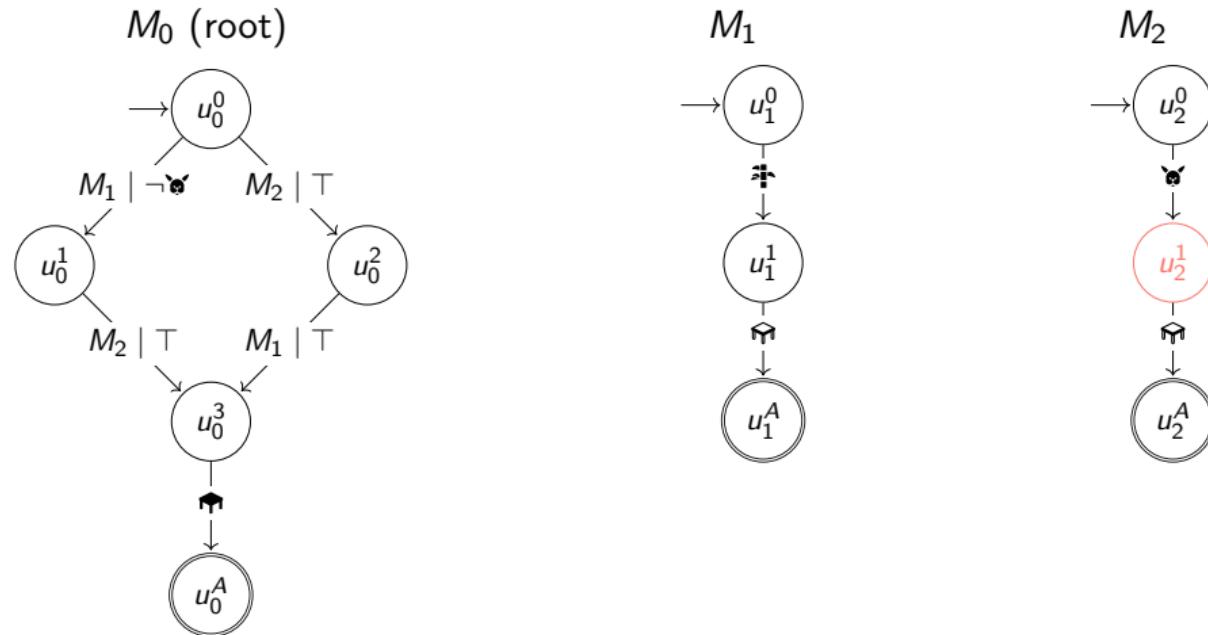


$\text{Trace} = \langle \{\pitchfork\}, \{\pitchfork\}, \{\bullet\},$

$\text{Stack} = []$

# Hierarchies of Reward Machines

## Formalism II

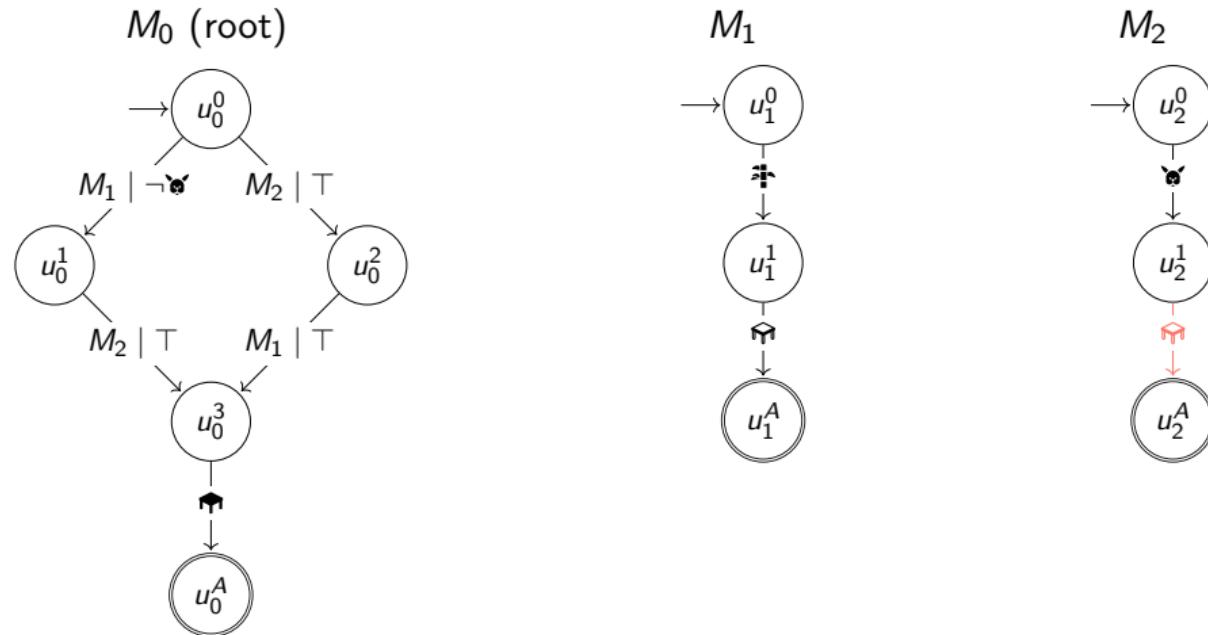


Trace =  $\langle \{\oplus\}, \{\sqcap\}, \{\bullet\},$

Stack = [ $\langle M_0, u_0^3 \rangle$ ]

# Hierarchies of Reward Machines

## Formalism II

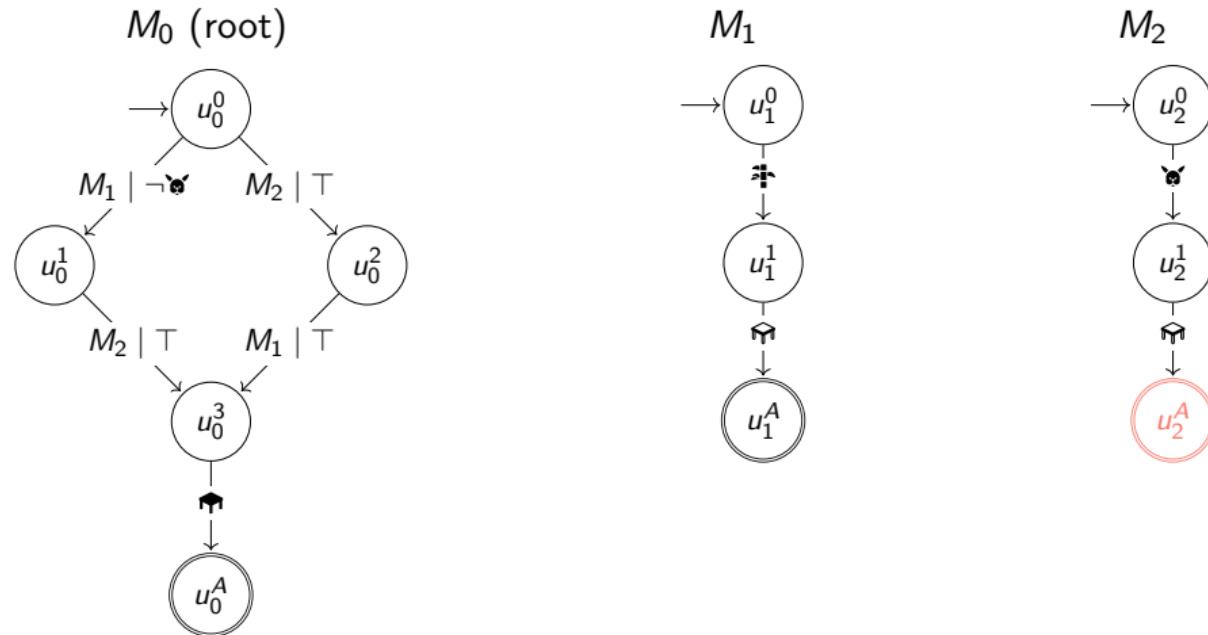


Trace =  $\langle \{\oplus\}, \{\sqcap\}, \{\bullet\}, \{\sqcap\},$

Stack =  $[\langle M_0, u_0^3 \rangle]$

# Hierarchies of Reward Machines

## Formalism II

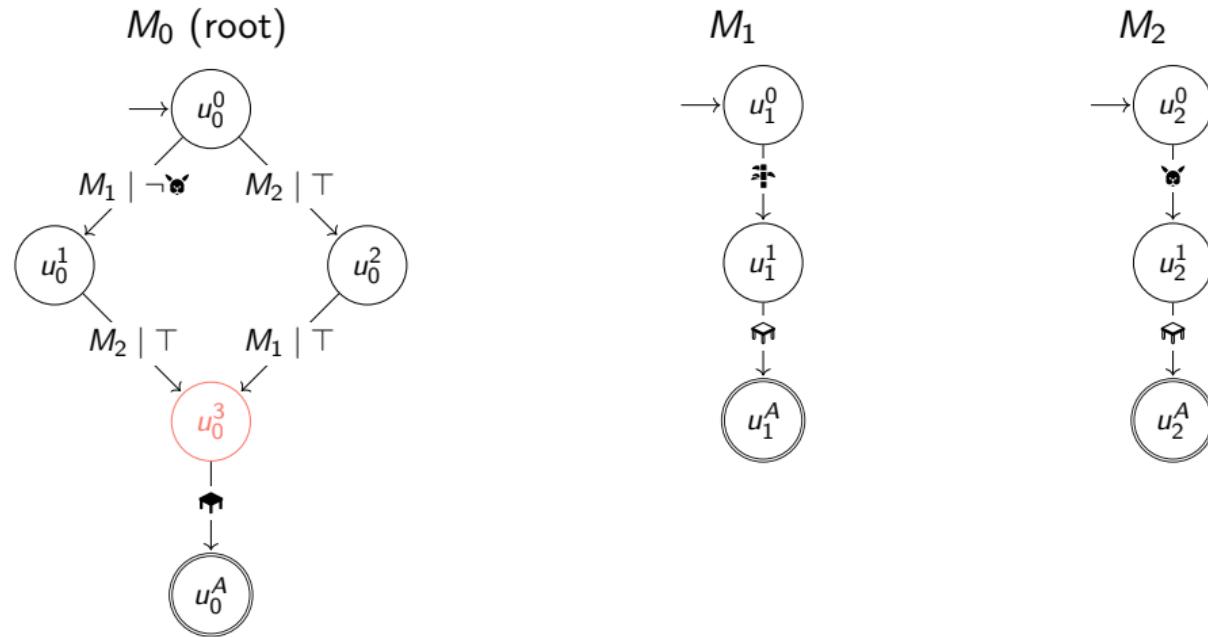


$\text{Trace} = \langle \{\oplus\}, \{\ominus\}, \{\bullet\}, \{\ominus\},$

$\text{Stack} = [\langle M_0, u_0^3 \rangle]$

# Hierarchies of Reward Machines

## Formalism II

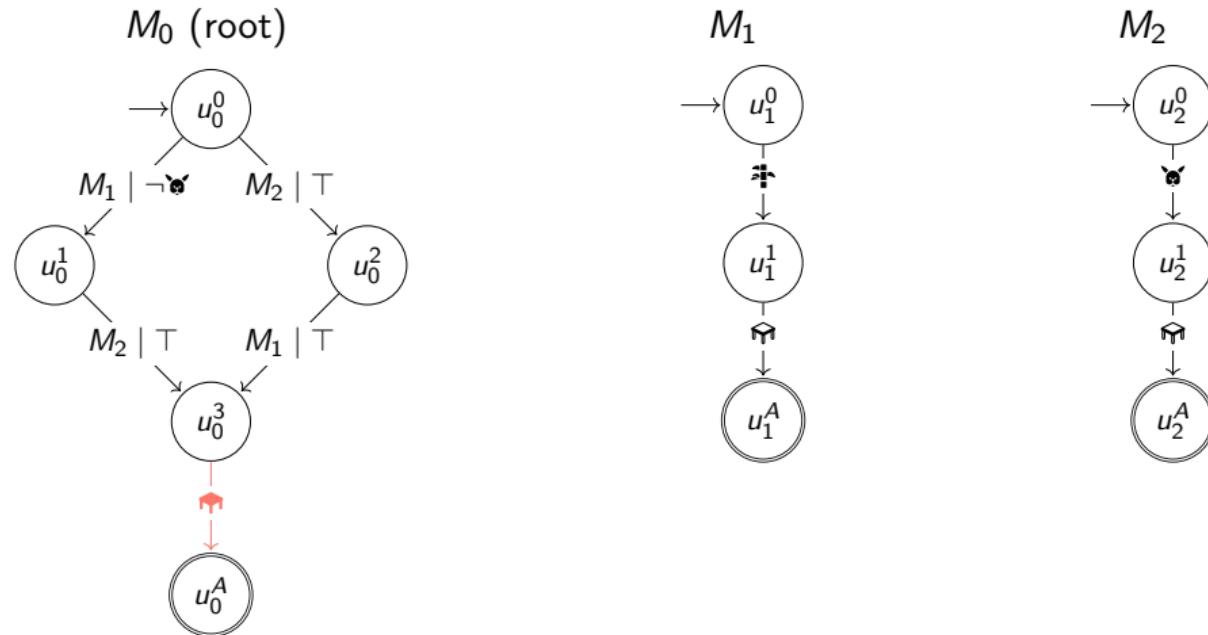


Trace =  $\langle \{\oplus\}, \{\sqcap\}, \{\bullet\}, \{\sqcap\} \rangle$ ,

Stack =  $[(M_0, u_0^3)]$

# Hierarchies of Reward Machines

## Formalism II

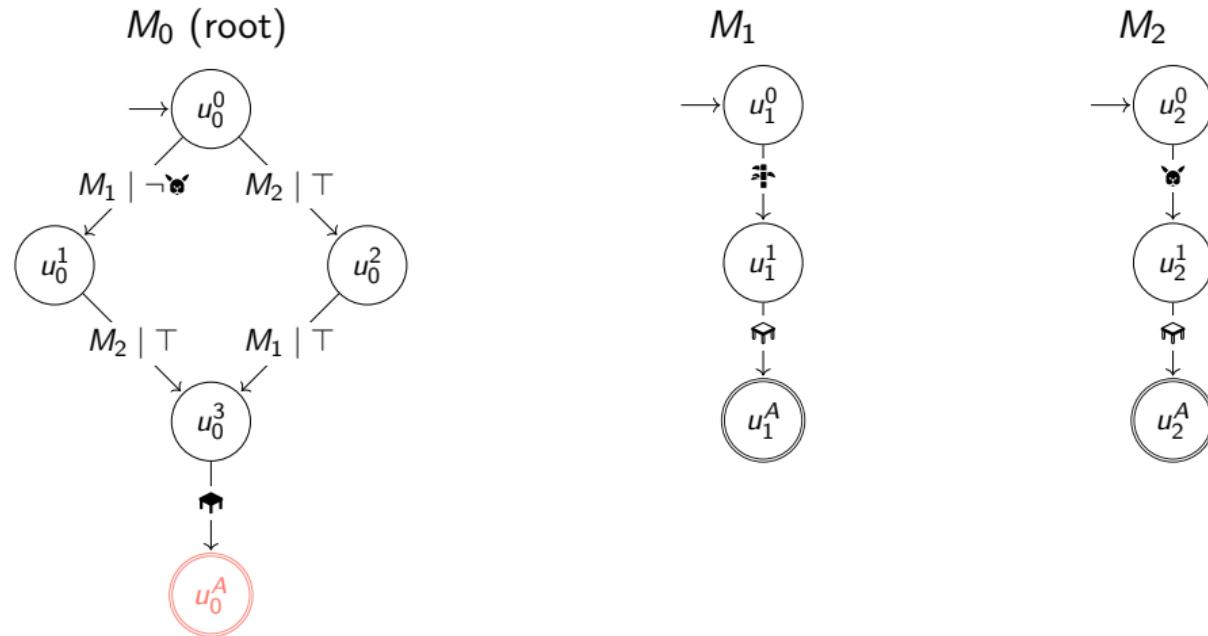


Trace =  $\langle \{\oplus\}, \{\wedge\}, \{\bullet\}, \{\wedge\}, \{\text{red}\uparrow\} \rangle$

Stack = []

# Hierarchies of Reward Machines

## Formalism II



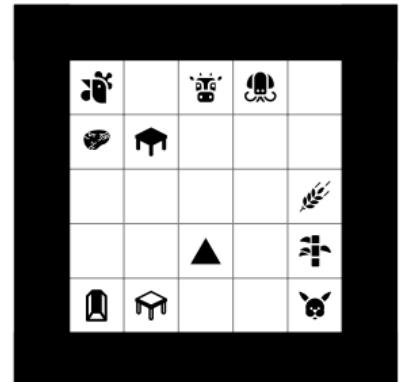
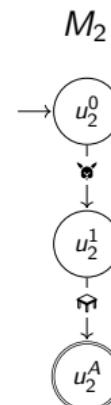
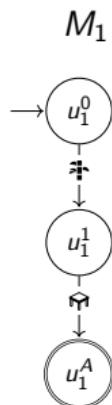
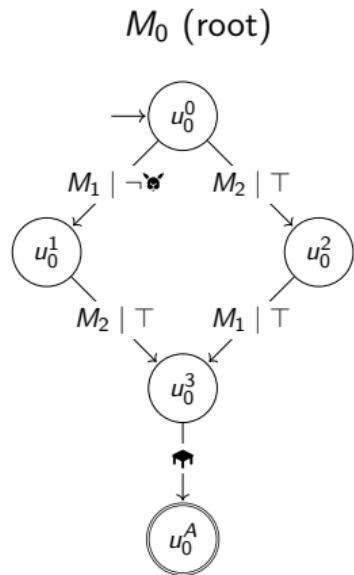
Trace =  $\langle \{\oplus\}, \{\ominus\}, \{\bullet\}, \{\triangleright\}, \{\blacktriangleleft\} \rangle$

Stack = []

# Hierarchies of Reward Machines

## Exploitation

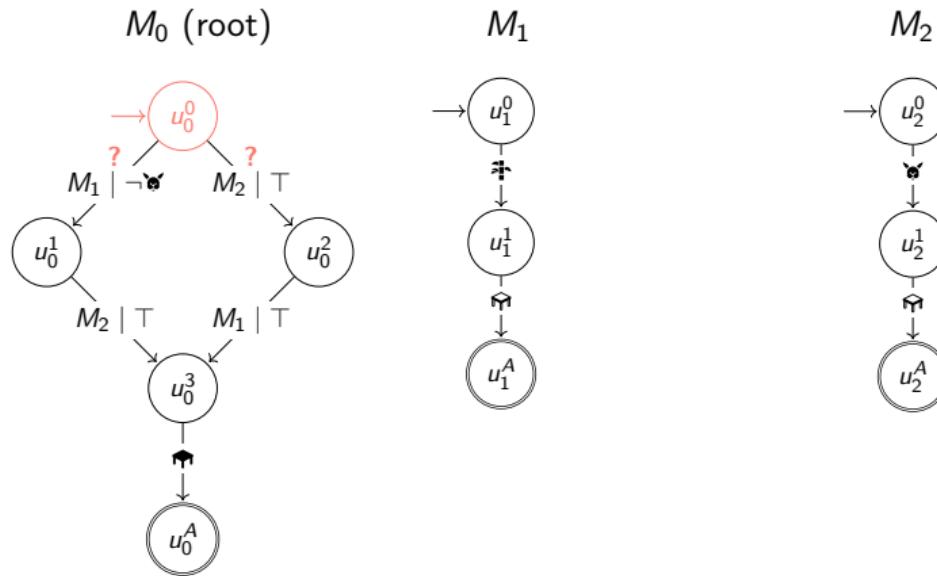
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

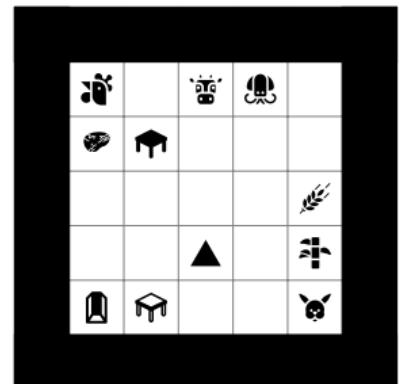
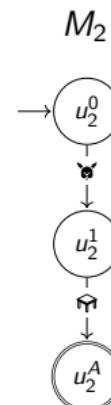
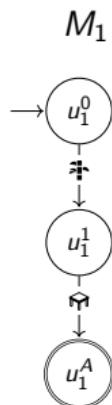
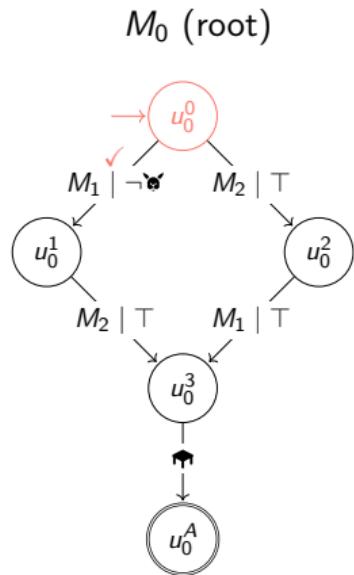
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

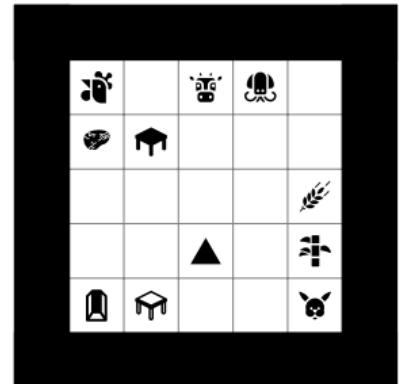
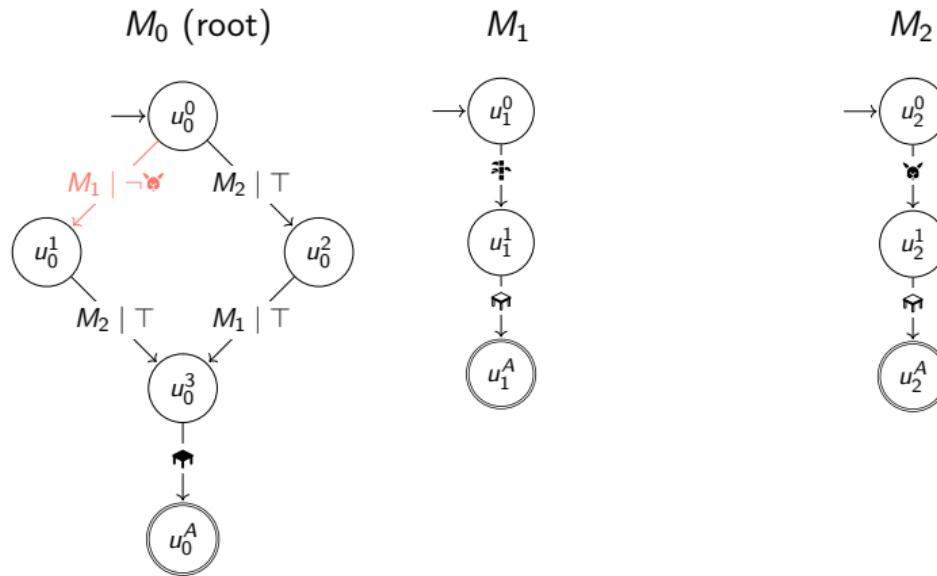
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

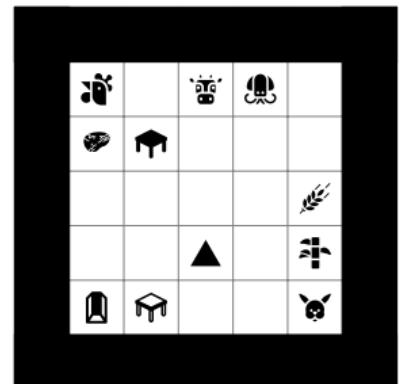
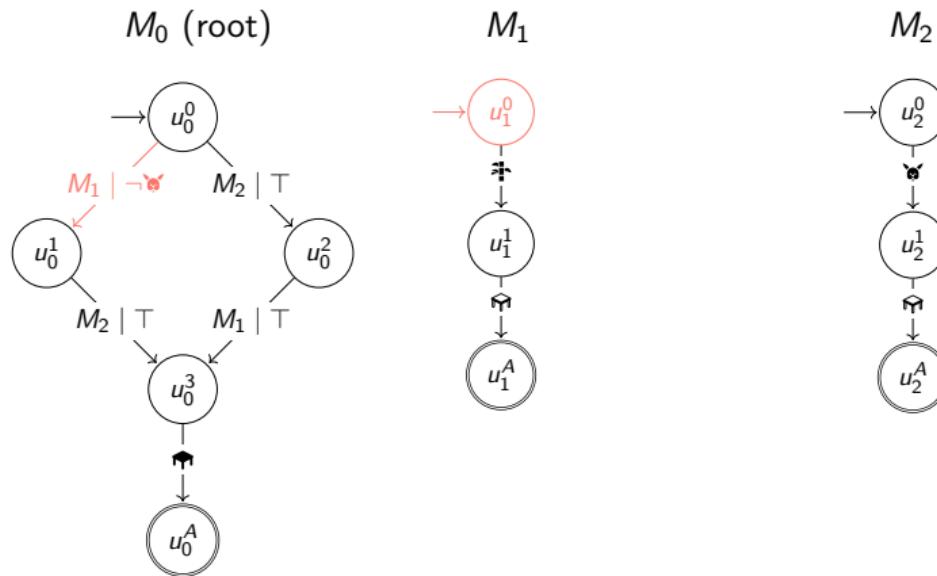
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

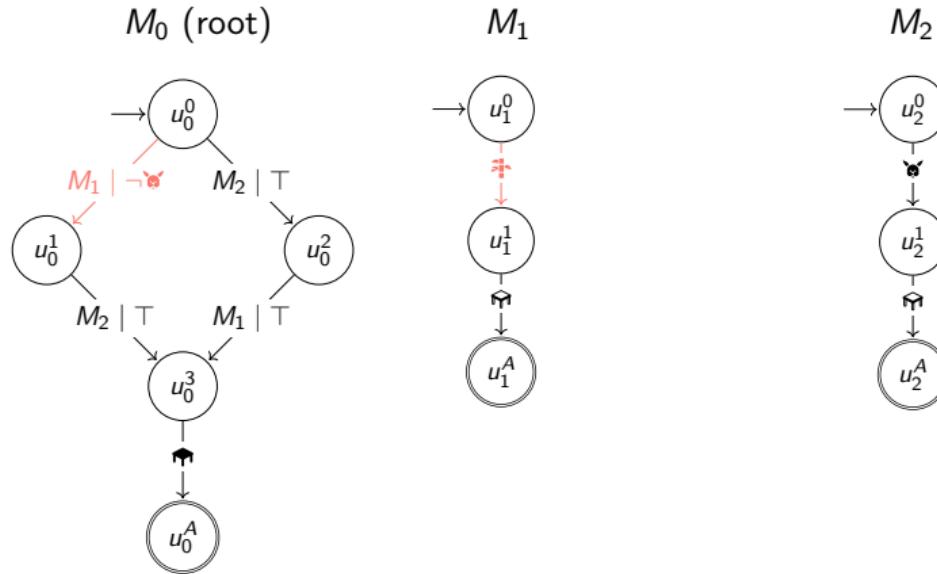
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

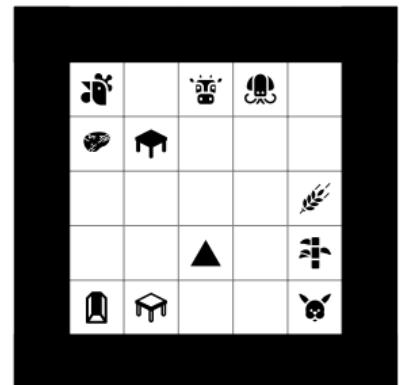
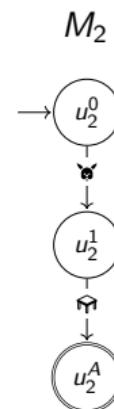
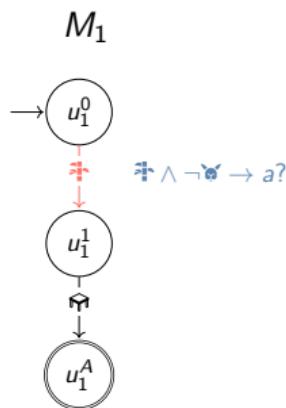
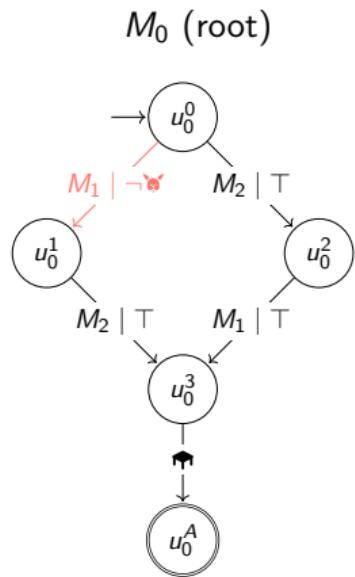
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

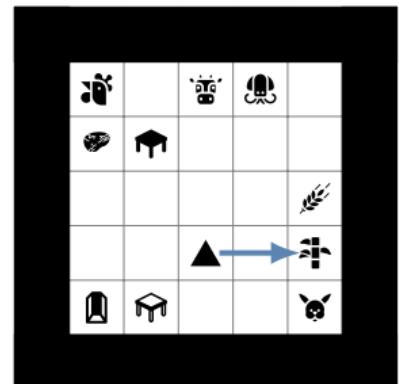
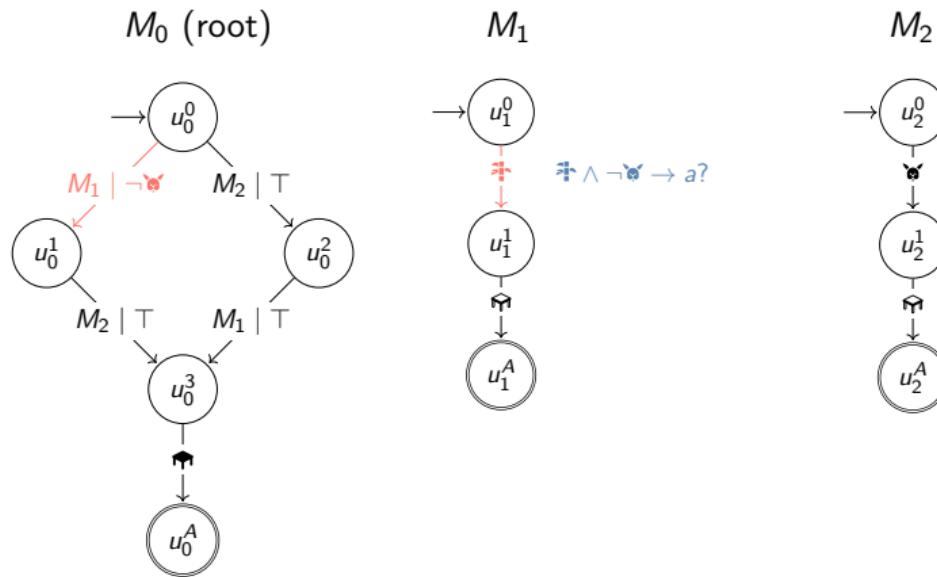
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

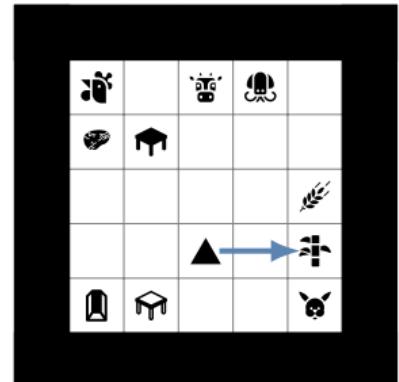
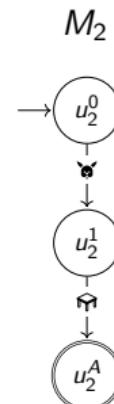
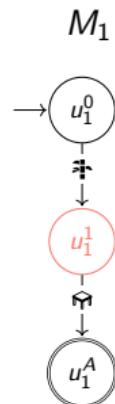
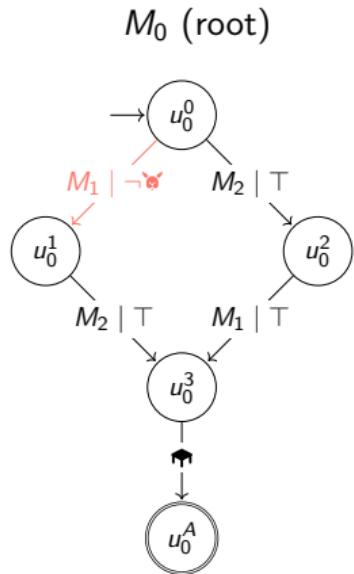
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

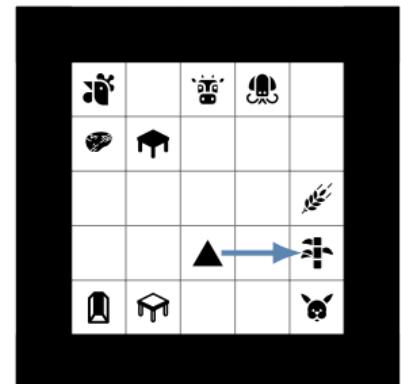
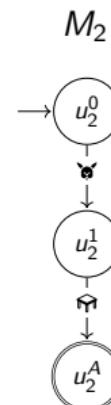
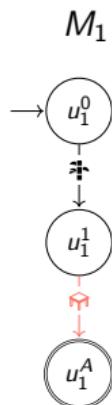
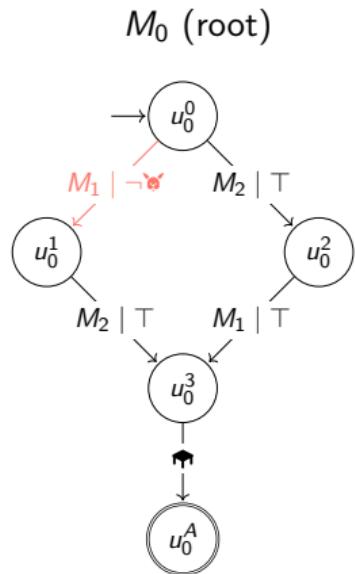
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

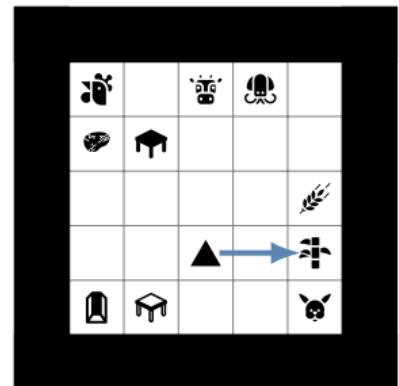
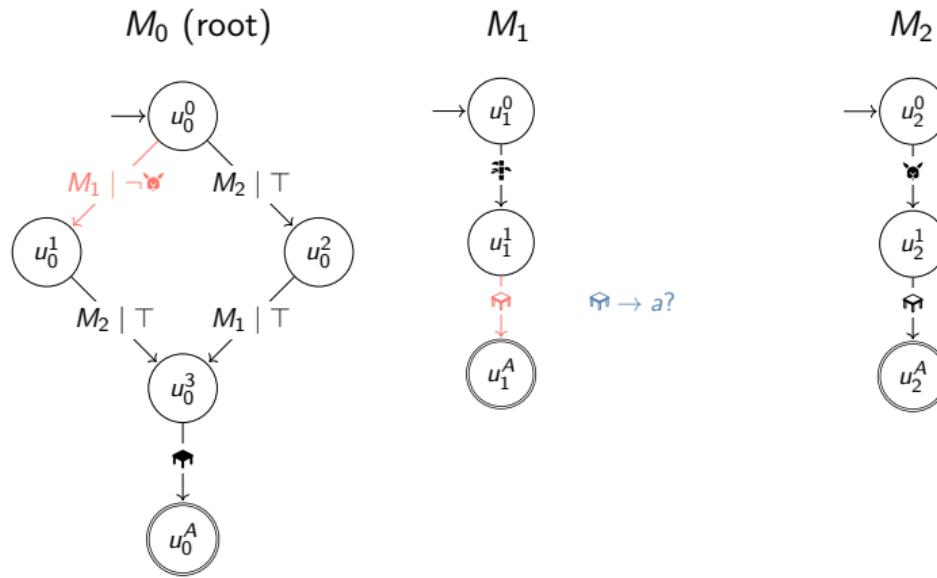
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

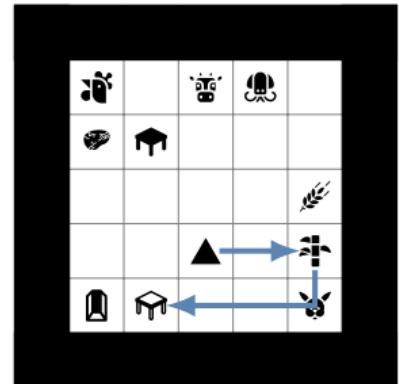
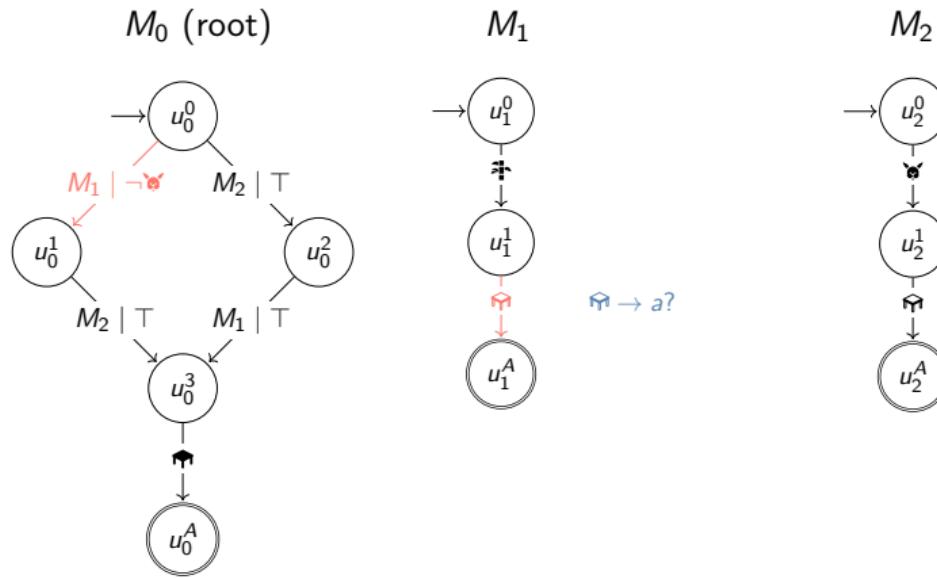
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

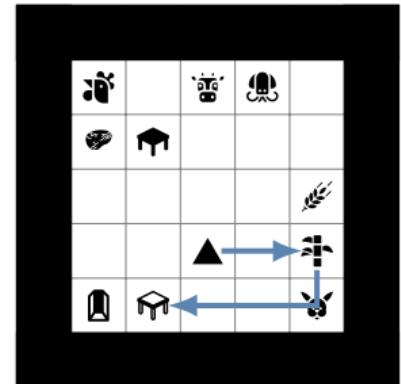
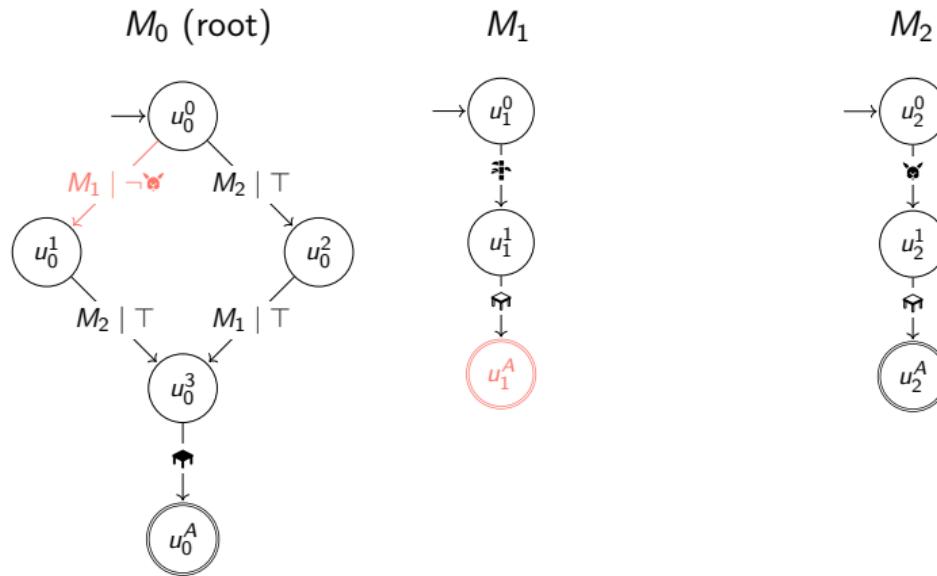
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

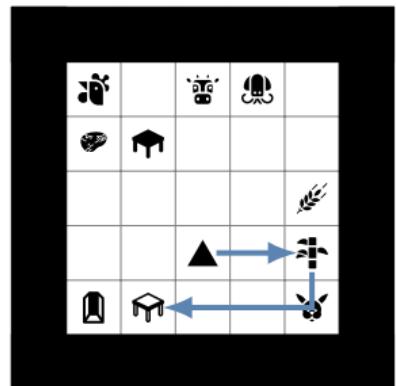
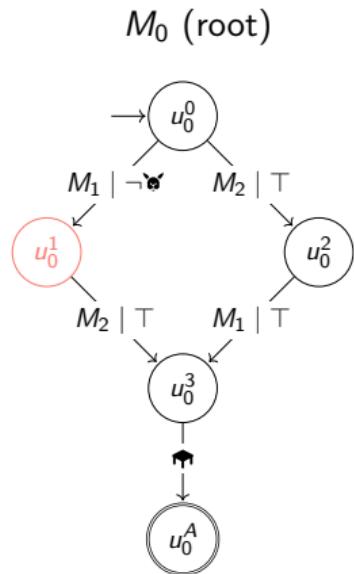
- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



# Hierarchies of Reward Machines

## Exploitation

- The structure of an HRM can be exploited *hierarchically*:
  - RM policies – Choose formulas or calls to (eventually) reach an accepting state.
  - Formula policies – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



continues...

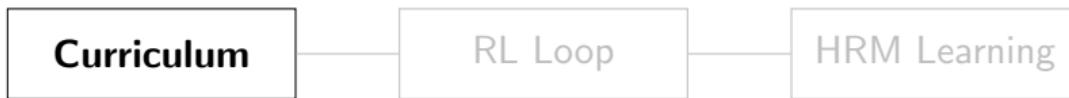
# Hierarchies of Reward Machines

## Learning I



# Hierarchies of Reward Machines

## Learning I

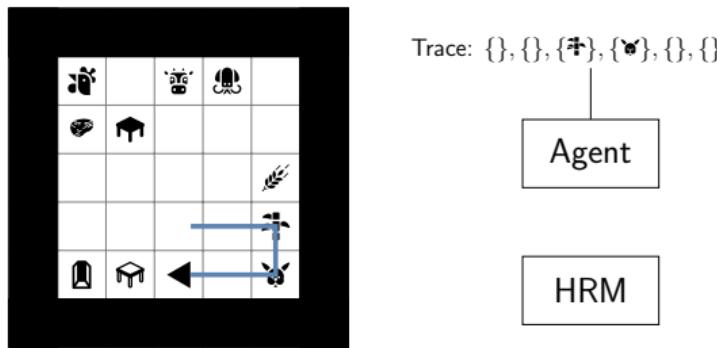


- An HRM is learned for each task.
- Each task has a *level h*.
- Learning proceeds from lower to higher levels.
- Level is increased when the average performance surpasses a *threshold*.

CAKE							$h = 4$
BOOKQUILL				MILKBUCKETSUGAR			
BOOK		MAP			MILKBUCKET		
BATTER	BUCKET	COMPASS	LEATHER	PAPER	QUILL	SUGAR	$h = 1$

# Hierarchies of Reward Machines

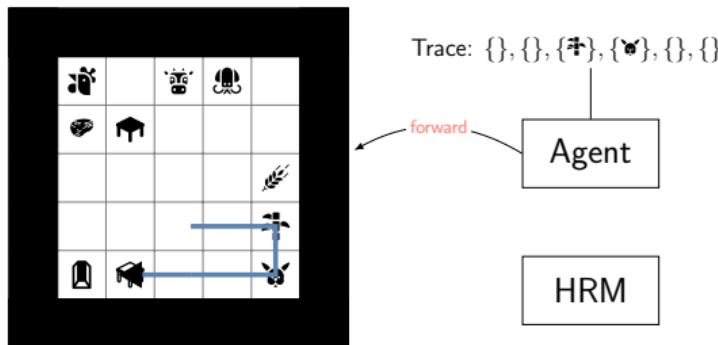
## Learning I



- The agent *selects a task* at the beginning of each episode and attempts to complete it.
- The agent maintains a *trace* of the events observed so far.

# Hierarchies of Reward Machines

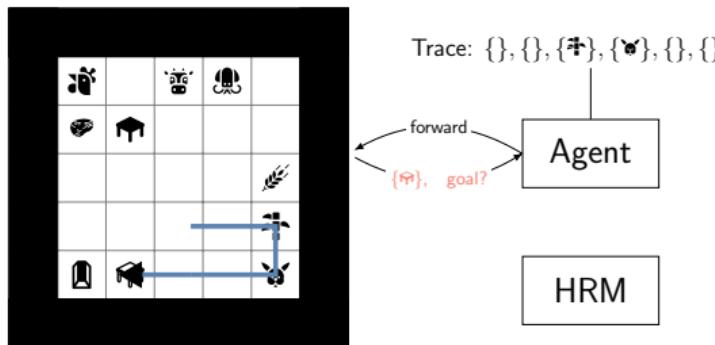
## Learning I



- The agent *selects a task* at the beginning of each episode and attempts to complete it.
- The agent maintains a *trace* of the events observed so far.

# Hierarchies of Reward Machines

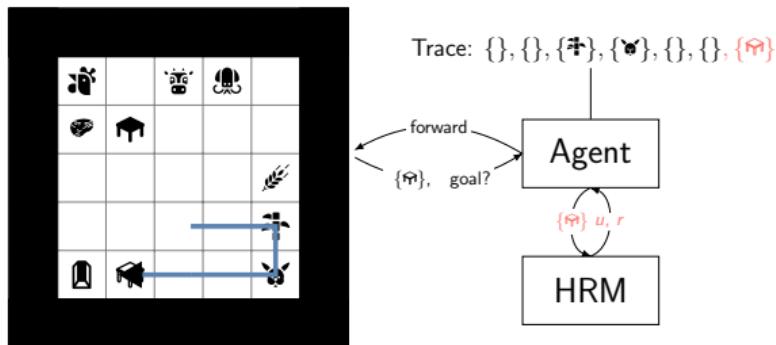
## Learning I



- The agent *selects a task* at the beginning of each episode and attempts to complete it.
- The agent maintains a *trace* of the events observed so far.

# Hierarchies of Reward Machines

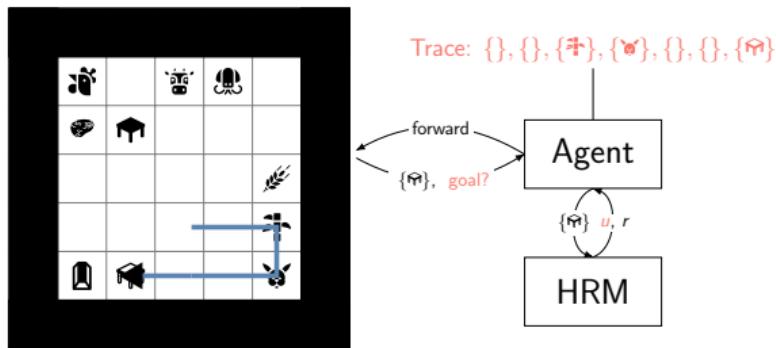
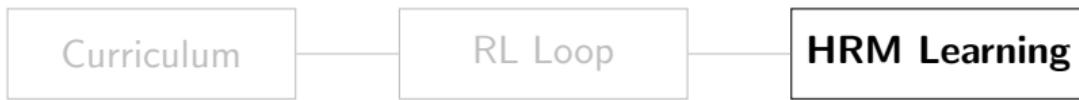
## Learning I



- The agent *selects a task* at the beginning of each episode and attempts to complete it.
- The agent maintains a *trace* of the events observed so far.

# Hierarchies of Reward Machines

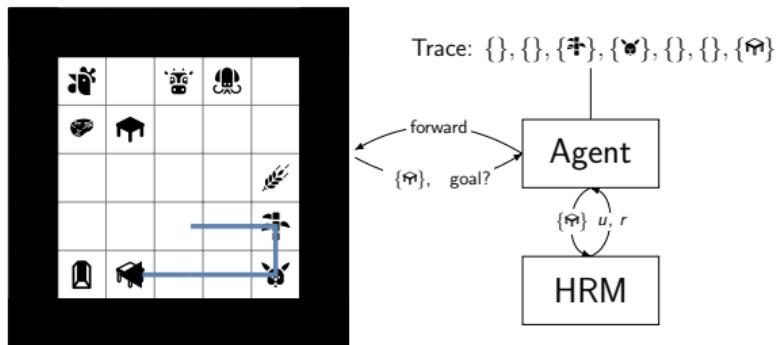
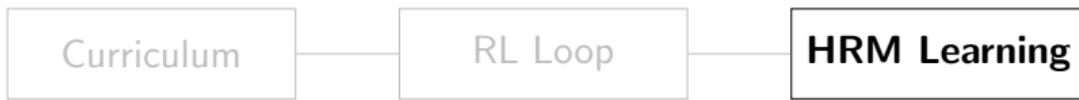
## Learning I



- A *new* HRM is learned if the trace is a *counterexample* (e.g., reaches the task's goal but not the *root*'s accepting state).
- HRMs are learned using *ILASP*, an inductive logic programming system.

# Hierarchies of Reward Machines

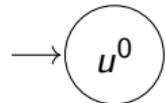
## Learning I



- HRMs for lower-level tasks may be called.
- Lower-level task policies can be used for *exploration*: observing goal traces becomes easier!

# Hierarchies of Reward Machines

## Learning II

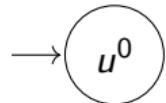
**Example:** Sequence of RMs learned for the task “Collect  then go to 

A circular diagram representing a reward machine. An arrow points to the left side of the circle, indicating the input or start state. Inside the circle, the symbol  $u^0$  is centered.



# Hierarchies of Reward Machines

## Learning II

**Example:** Sequence of RMs learned for the task “Collect  then go to 

A circular diagram representing a reward machine. An arrow points to the left side of the circle, indicating the start state. Inside the circle, the label  $u^0$  is centered.



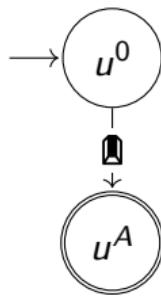
$G : \langle \{\text{key}\}, \{\text{door}\} \rangle$

# Hierarchies of Reward Machines

## Learning II

**Example:** Sequence of RMs learned for the task “Collect  then go to 

A circular state diagram labeled  $u^0$  with an incoming arrow from the left, representing the initial state of the reward machine.



$$G : \langle \{\text{book}\}, \{\text{house}\} \rangle$$

# Hierarchies of Reward Machines

## Learning II

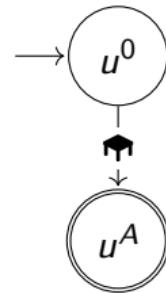
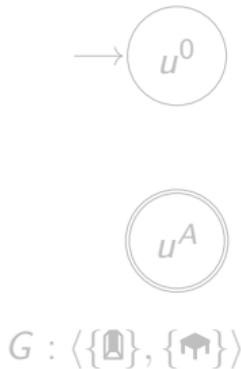
**Example:** Sequence of RMs learned for the task “Collect then go to ”.



# Hierarchies of Reward Machines

## Learning II

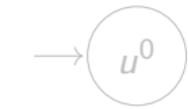
**Example:** Sequence of RMs learned for the task “Collect then go to ”.



# Hierarchies of Reward Machines

## Learning II

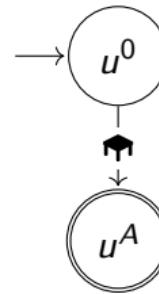
**Example:** Sequence of RMs learned for the task “Collect then go to ”.



$$G : \langle \{\text{book}\}, \{\text{person}\} \rangle$$



$$\bar{G} : \langle \{\text{book}\} \rangle$$



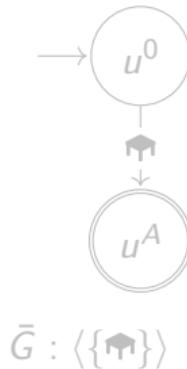
$$\bar{G} : \langle \{\text{person}\} \rangle$$

# Hierarchies of Reward Machines

## Learning II

**Example:** Sequence of RMs learned for the task “Collect  then go to 

A diagram illustrating a Reward Machine  $G$ . It consists of two states: an initial state  $u^0$  at the top and a goal state  $u^A$  at the bottom. An arrow points from  $u^0$  to  $u^A$ . Below the states, the expression  $G : \langle \{\text{book}\}, \{\text{house}\} \rangle$  is given, indicating the reward symbols associated with each state.

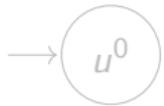


UNSATISFIABLE,  
Increment the number  
of states!

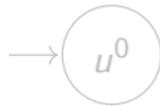
# Hierarchies of Reward Machines

## Learning II

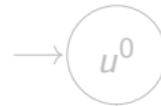
**Example:** Sequence of RMs learned for the task “Collect then go to ”.



$$G : \langle \{\text{key}\}, \{\text{house}\} \rangle$$

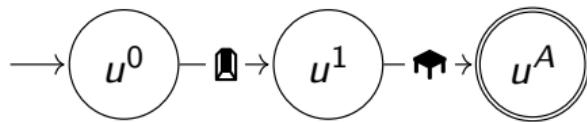


$$\bar{G} : \langle \{\text{key}\} \rangle$$



$$\bar{G} : \langle \{\text{house}\} \rangle$$

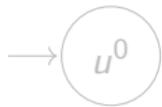
UNSATISFIABLE,  
Increment the number  
of states!



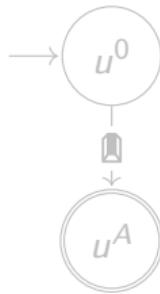
# Hierarchies of Reward Machines

## Learning II

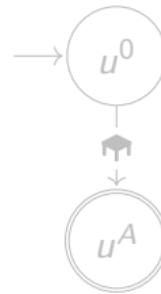
**Example:** Sequence of RMs learned for the task “Collect then go to ”.



$$G : \langle \{\text{book}\}, \{\text{house}\} \rangle$$

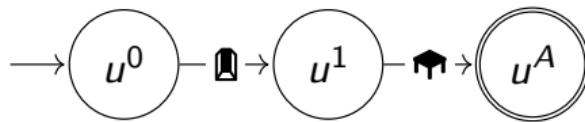


$$\bar{G} : \langle \{\text{book}\} \rangle$$



$$\bar{G} : \langle \{\text{house}\} \rangle$$

UNSATISFIABLE,  
Increment the number  
of states!



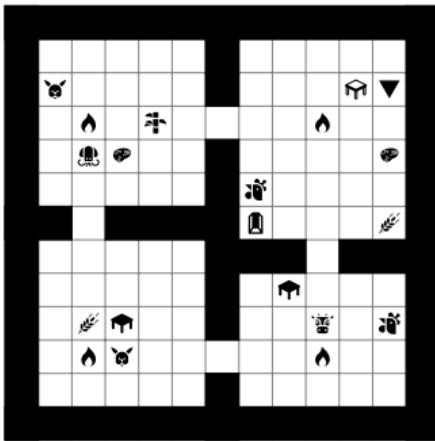
In higher-level tasks, lower-level RMs can be called.

## Evaluation

## Domains

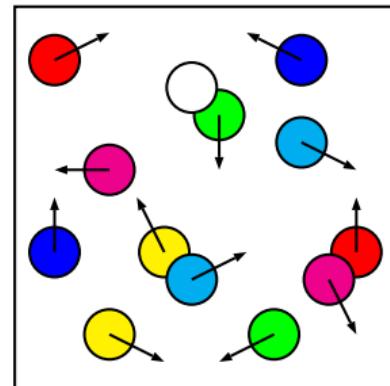
CRAFTWORLD

<b>Environment</b>	Based on Minigrid [Chevalier-Boisvert et al., 2023]. Image-like grid observations, discrete actions.
<b>Tasks</b>	Visit a sequence of landmark locations.



WATERWORLD [Toro Icarte et al., 2018]

<b>Environment</b>	Continuous observations (positions, velocities), discrete actions.
<b>Tasks</b>	Observe a sequence of colors.



Toro Icarte et al. "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning". ICML, 2018.

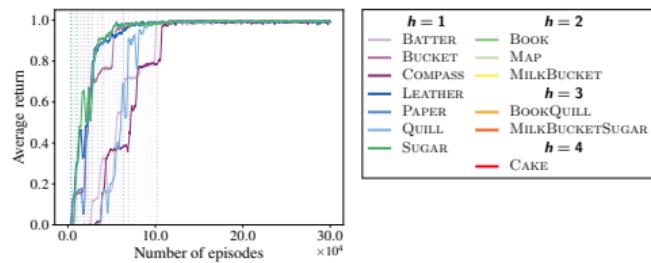
Chevalier-Boisvert et al. "Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks". arXiv, 2023.

# Evaluation

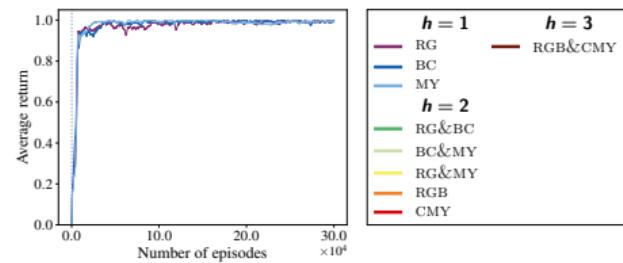
## Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD

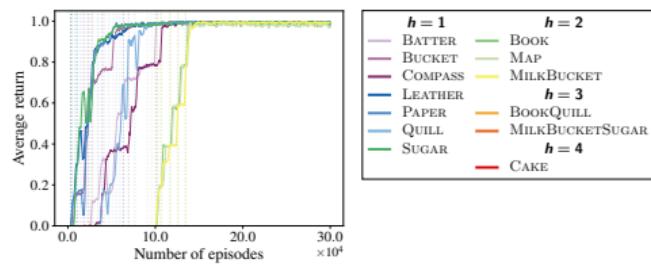


# Evaluation

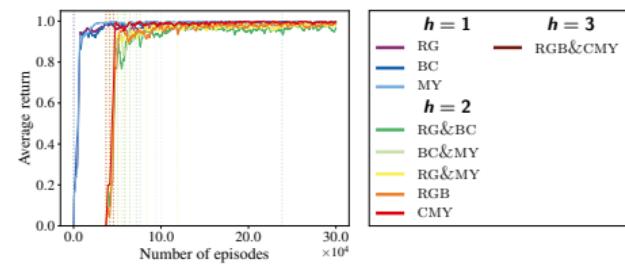
## Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD

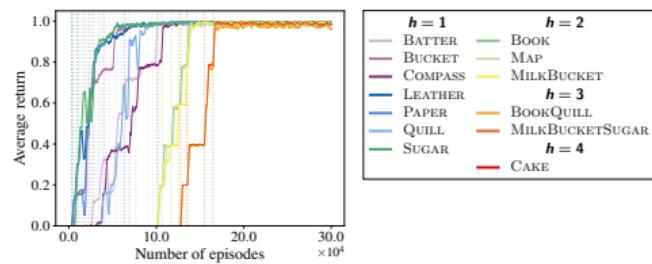


# Evaluation

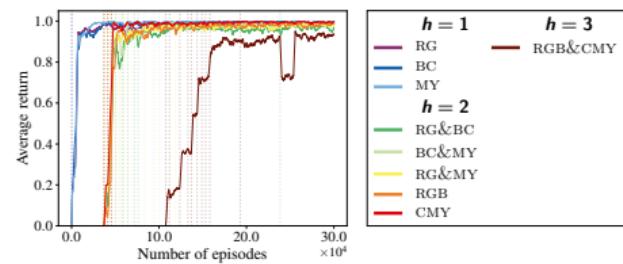
## Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD

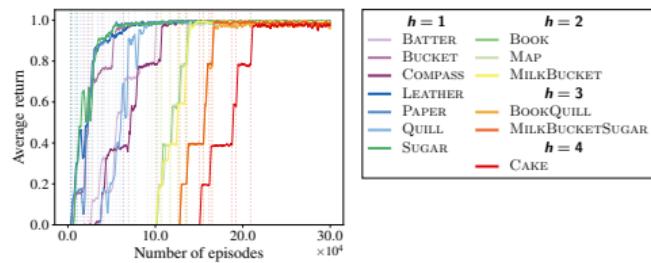


# Evaluation

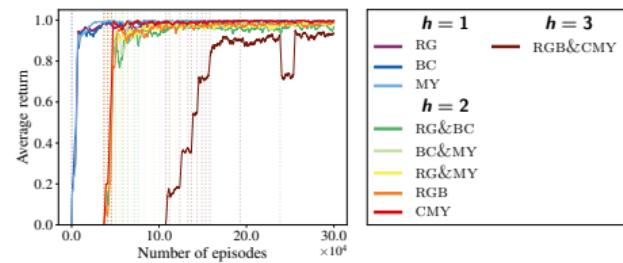
## Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD

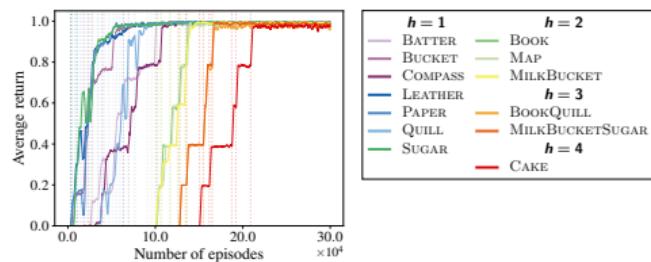


# Evaluation

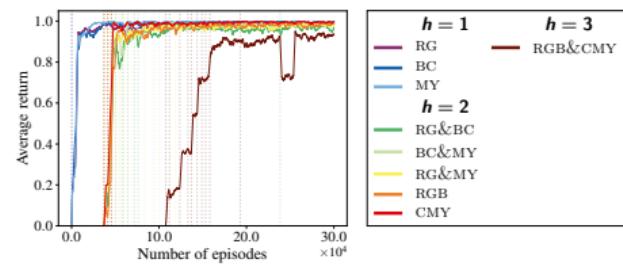
## Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD



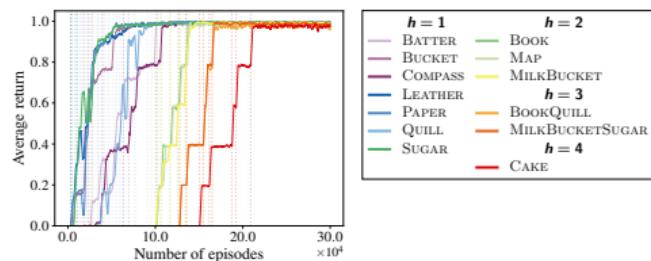
## Insights:

# Evaluation

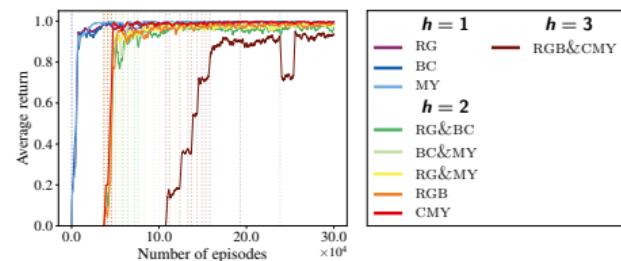
## Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD



## Insights:

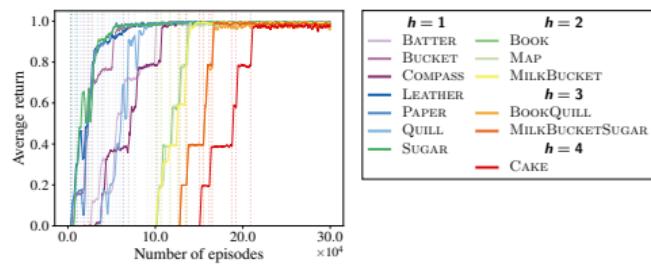
- ① HRM learning becomes less scalable as the number of tasks and levels grows.

# Evaluation

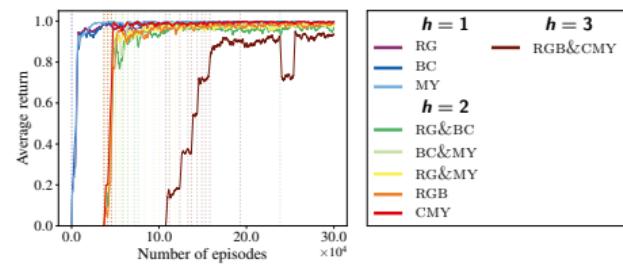
## Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD



## Insights:

- ① HRM learning becomes less scalable as the number of tasks and levels grows.
- ② Exploration with low-level policies enables observing goal trace examples faster.

# Evaluation

## HRM Learning vs RM Learning

### RM Learning Baselines:

- Minimal RMs: Ours (but learning a flat HRM) and JIRP [Xu et al., 2020].
- RMs that predict the next event accurately: DeepSynth [Hasanbeig et al., 2021] and LRM [Toro Icarte et al., 2019].

### Observations:

- ① Minimal RM learning methods poorly scale as the number of states increases.
- ② DeepSynth and LRM tend to overfit to the observed traces.
- ③ DeepSynth, JIRP and LRM need exponentially more edges in WATERWORLD since they do not use formulas.

# Evaluation

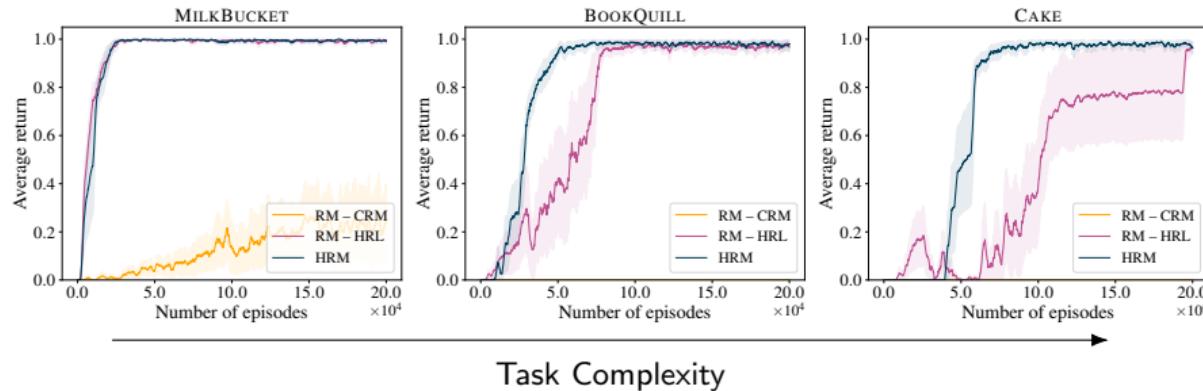
## Policy Learning

### Baselines:

- Hierarchical method on an RM (i.e., flat HRM).
- CRM [Toro Icarte et al., 2022] – Learns a global policy over an RM (i.e., not hierarchical).

### Observations:

- Hierarchical policy learning can be faster in **HRMs** than in **RMs**.
- Convergence is faster w.r.t. **CRM**, which does not independently solve the subtasks.



## Remove Handcrafted Event Set

- The agent learns its own mapping from observations to propositional events.
- Need for supporting noisy events (i.e., the mapping might make mistakes).
- Loss of interpretability.

## Remove Known Task Set

- The agent makes its own set of tasks over the event set.
- Autocurricula: start from simpler tasks and build upon them to perform high-level behaviors.

## Continual Learning

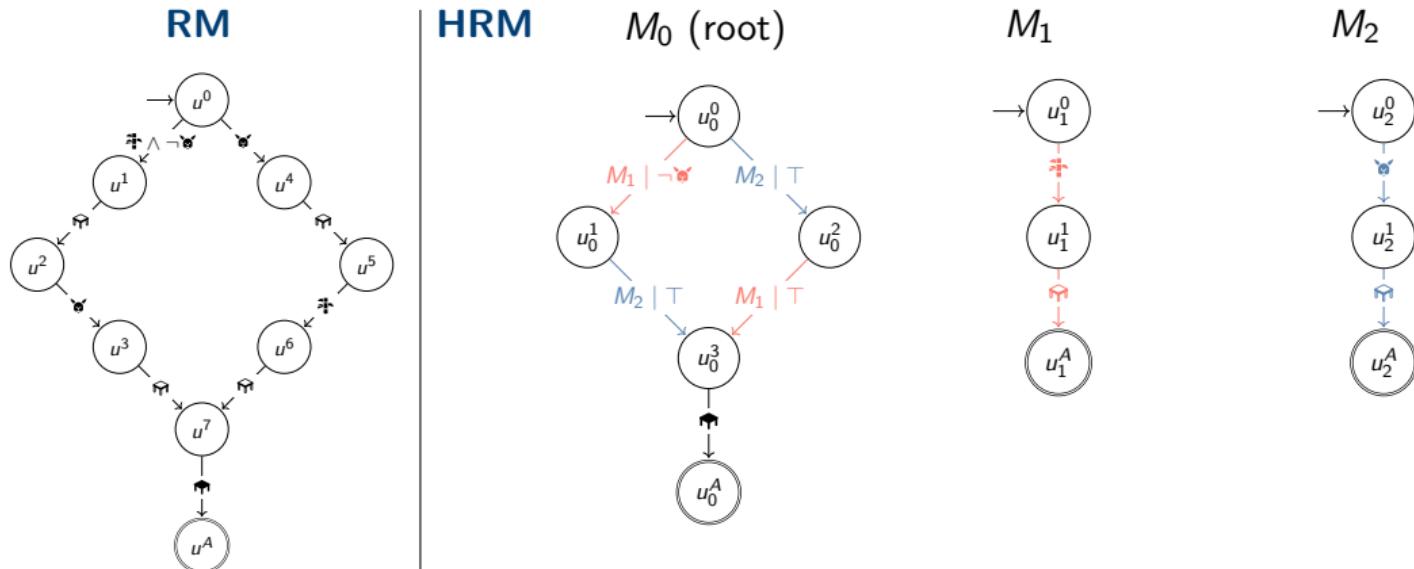
- Build RM learning methods that adapt to changing environments or agent capabilities (e.g., traces that achieved the goal but later do not).

# Key Insights

- Reveal the task structure to the agent.
- Learn reusable policies and task structures.
- Learning the structures alleviates human intervention, but does not remove it.

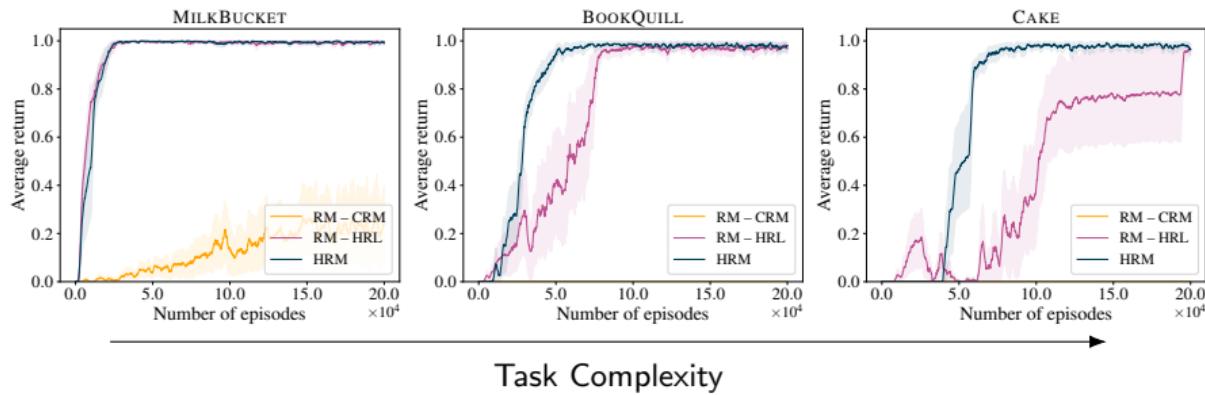
# Conclusions

- ① HRMs, a formalism for hierarchically composing RMs.



# Conclusions

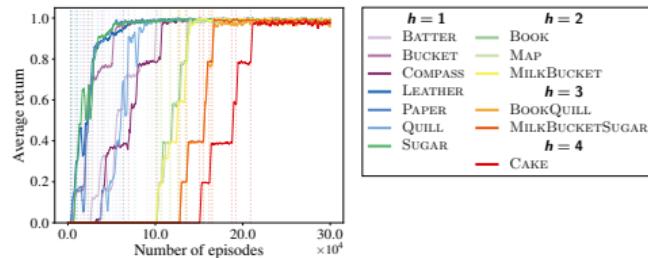
- ① HRMs, a formalism for hierarchically composing RMs.
- ② A method that *exploits* the structure of an HRM.



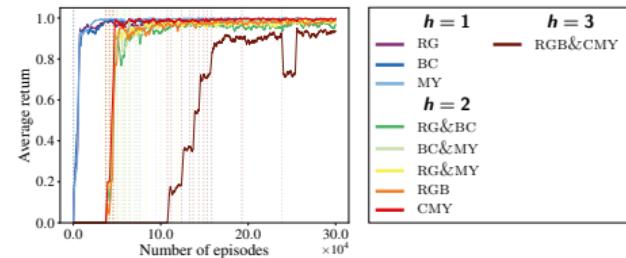
# Conclusions

- ① HRMs, a formalism for hierarchically composing RMs.
- ② A method that *exploits* the structure of an HRM.
- ③ A method for *learning a collection of HRMs* from traces.

CRAFTWORLD



WATERWORLD



Questions?