

Hierarchies of Reward Machines

Daniel Furelos-Blanco

Mark Law

Anders Jonsson

Kryisia Broda

Alessandra Russo

**Imperial College
London**

ILASP
Learning Logically

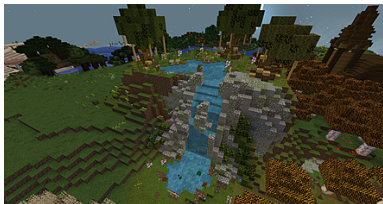
upf. **Universitat
Pompeu Fabra**
Barcelona

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

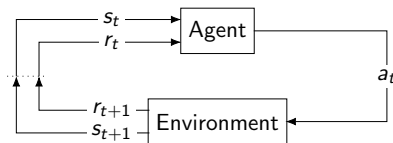


CreateVillageAnimalPen

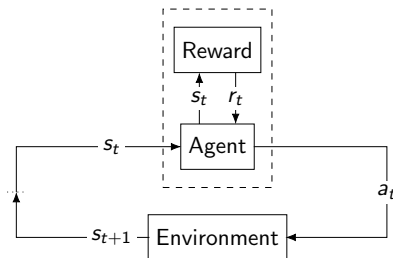


BuildVillageHouse

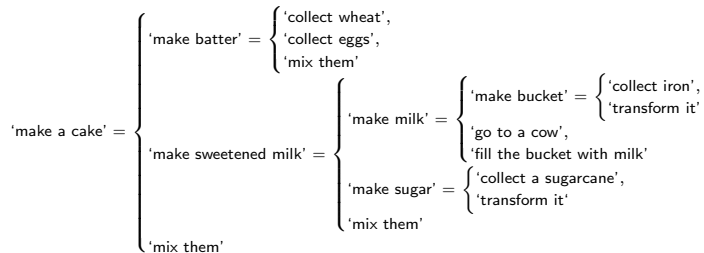
- The interaction hides the reward structure from the agent. . .



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

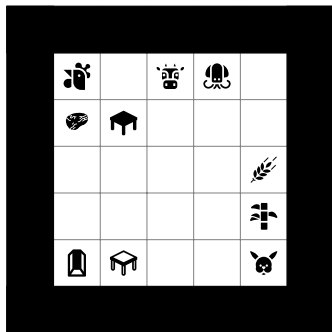


- Advantages & drawbacks:
 - + Interpretability.
 - + Enable task decomposition.
 - /+ Handcrafted – but we can learn them!
- **Our focus:** Express-Exploit-Learn descriptions of tasks that depend on other tasks.



Motivation

Reward Machines

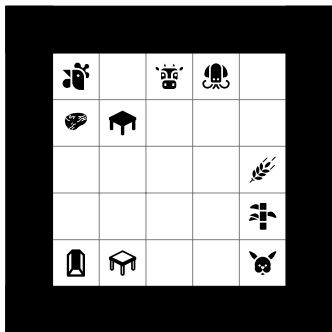


Events



Motivation

Reward Machines



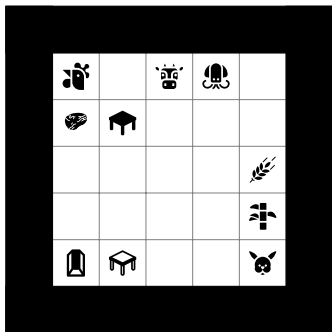
Events



Motivation

Reward Machines

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



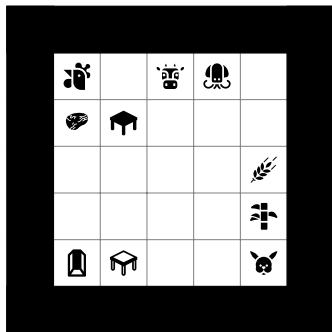
Events

{🏠, 🏠, 🍄, 🍄, 🌾, 🐛,
🍄, 🐛, 🐛, 🏠}

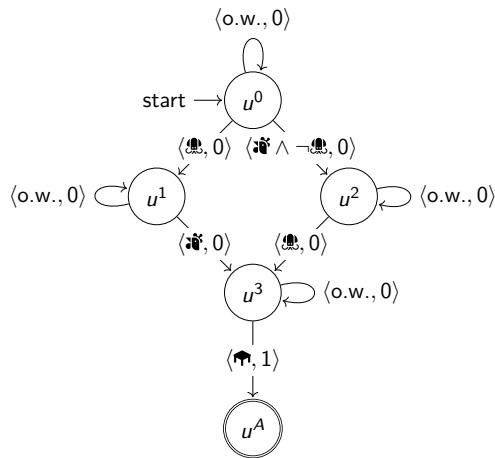
Motivation

Reward Machines

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



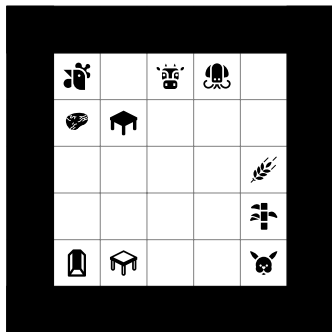
Events



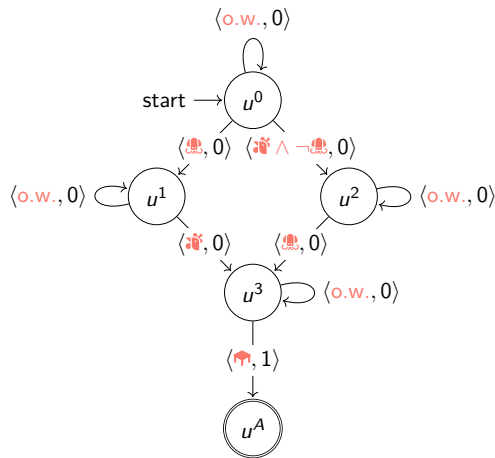
Motivation

Reward Machines

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



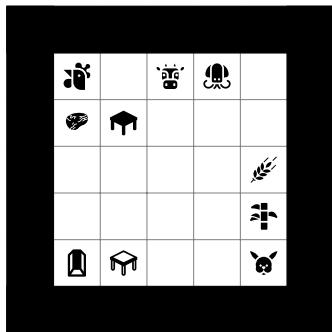
Events



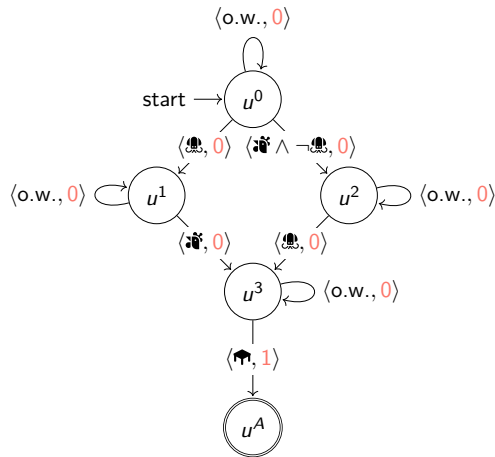
Motivation

Reward Machines

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



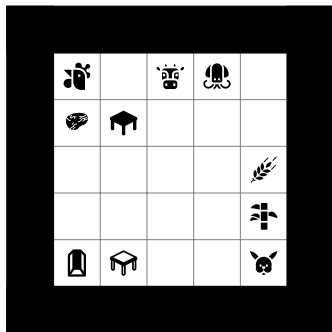
Events



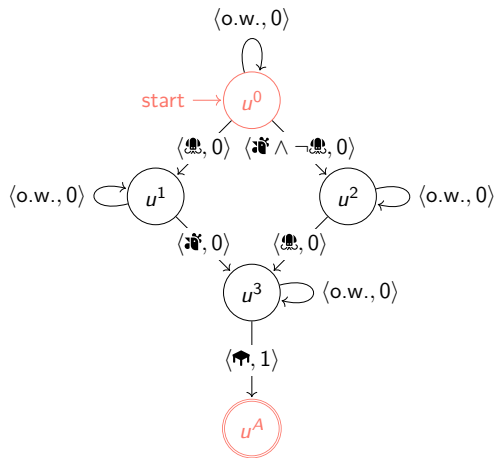
Motivation

Reward Machines

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



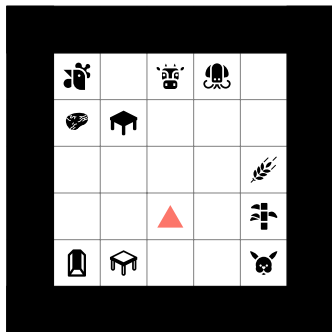
Events



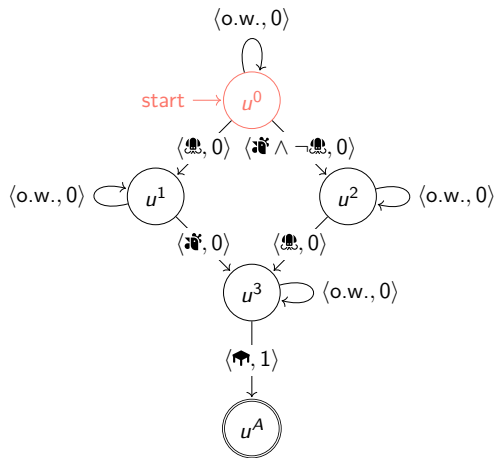
Motivation

Reward Machines

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



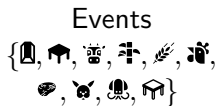
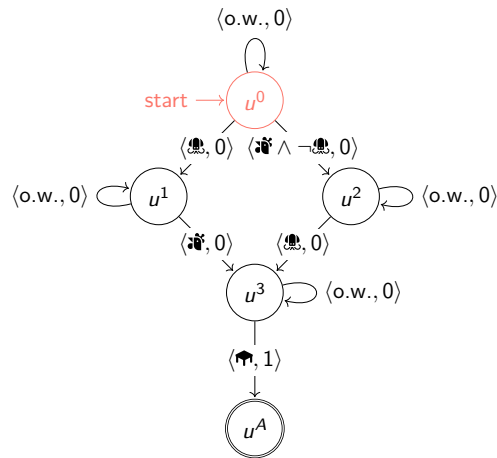
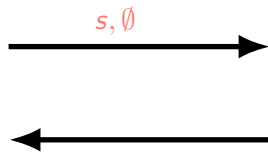
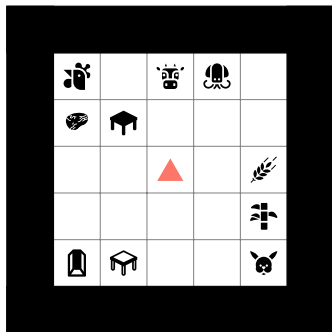
Events



Motivation

Reward Machines

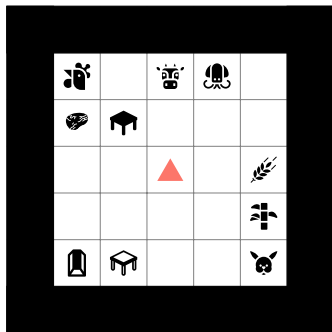
Task Collect 🐞 and 🐛 (in any order), then go to 🏠.



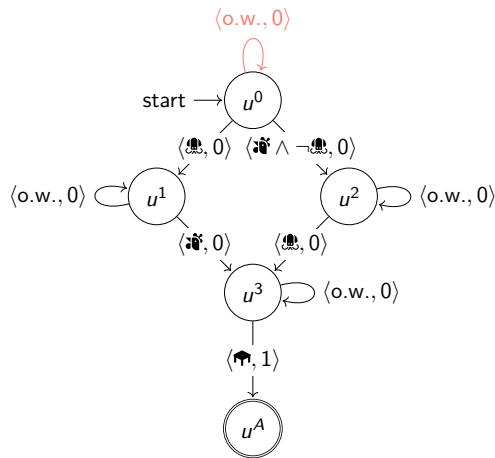
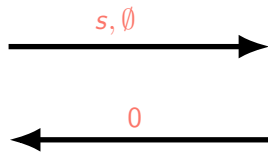
Motivation

Reward Machines

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



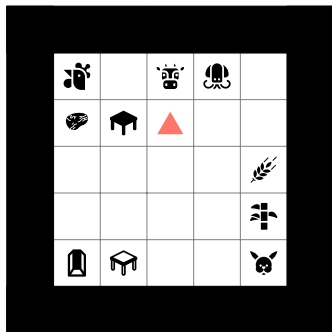
Events



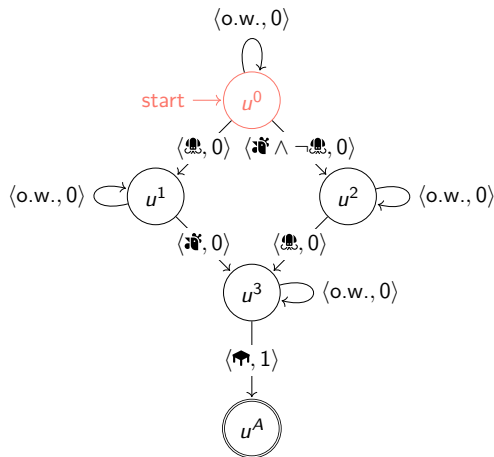
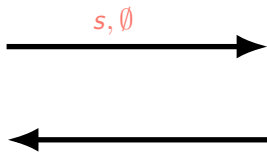
Motivation

Reward Machines

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



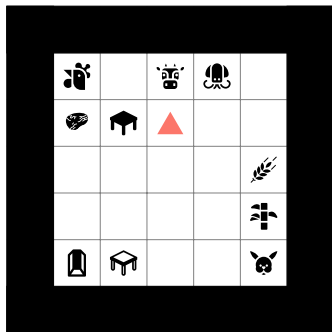
Events



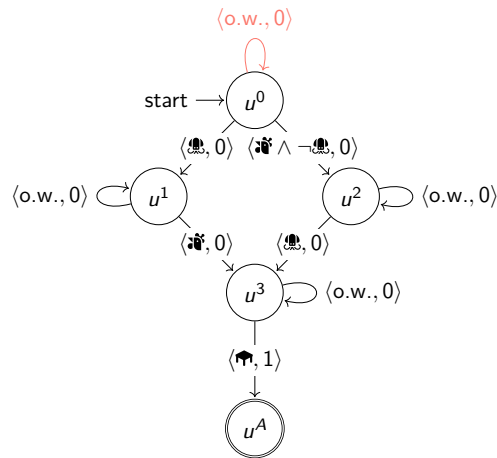
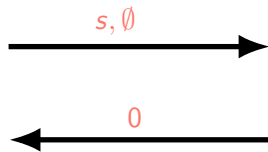
Motivation

Reward Machines

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



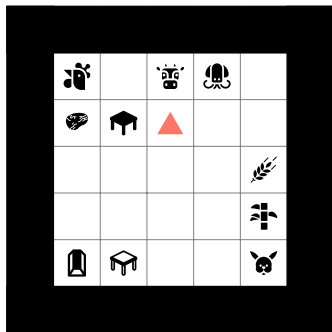
Events



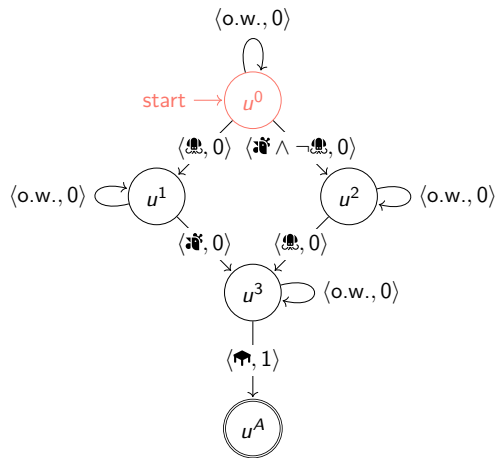
Motivation

Reward Machines

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



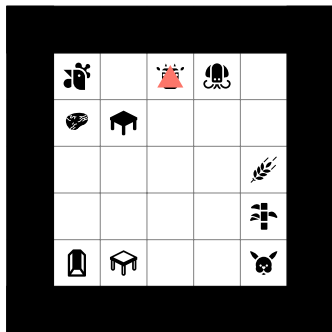
Events



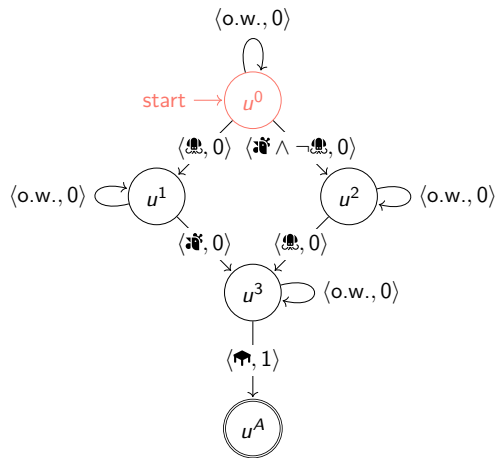
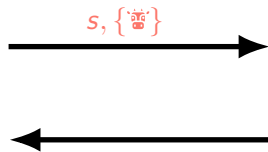
Motivation

Reward Machines

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



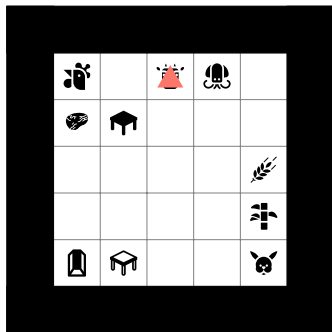
Events



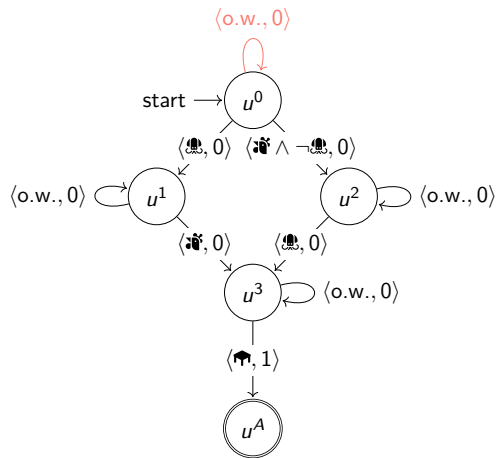
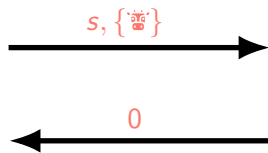
Motivation

Reward Machines

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



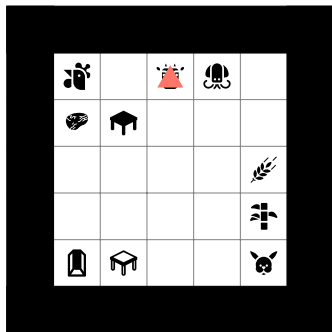
Events



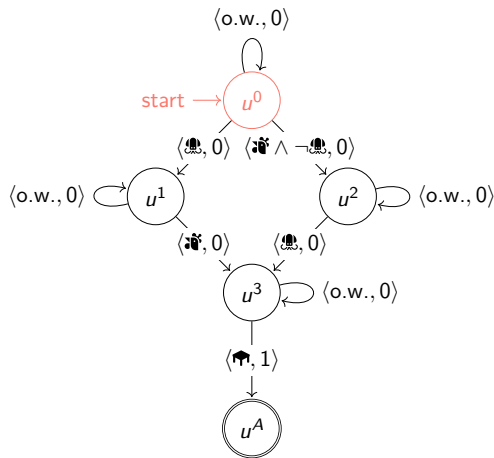
Motivation

Reward Machines

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



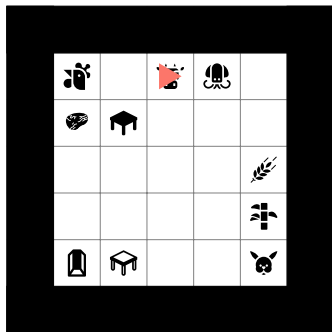
Events



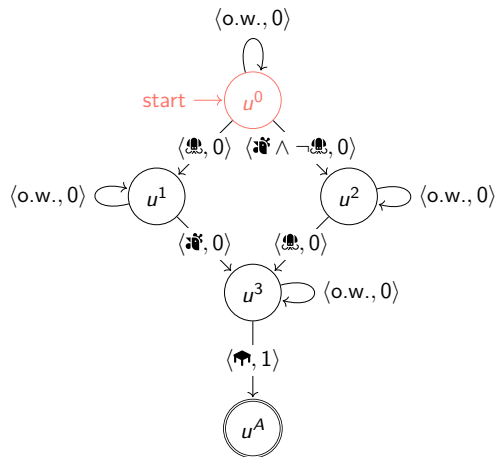
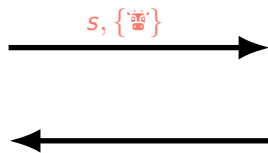
Motivation

Reward Machines

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



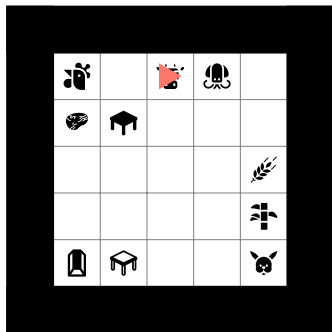
Events



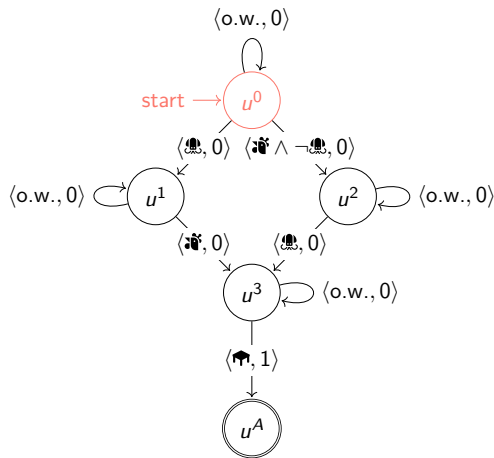
Motivation

Reward Machines

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



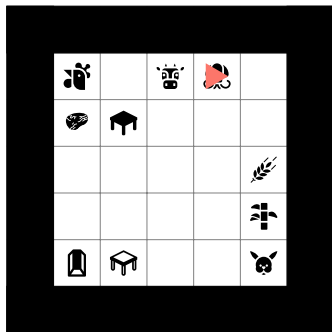
Events



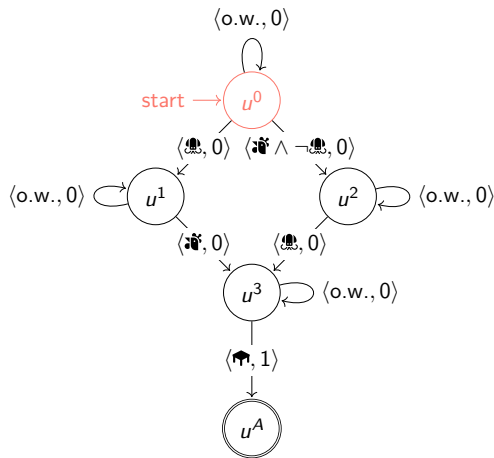
Motivation

Reward Machines

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



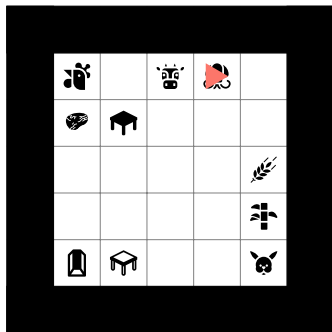
Events



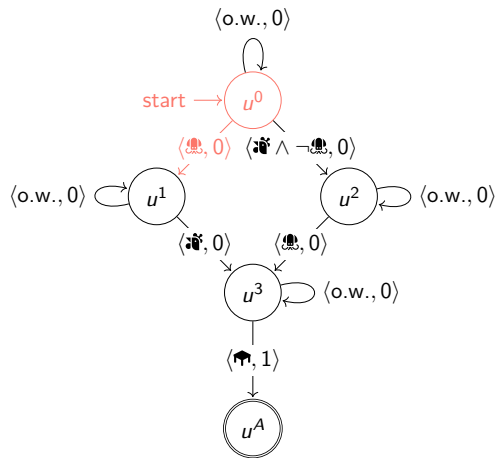
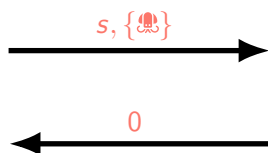
Motivation

Reward Machines

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



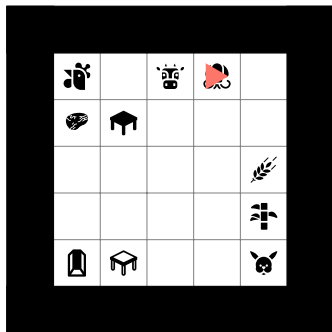
Events



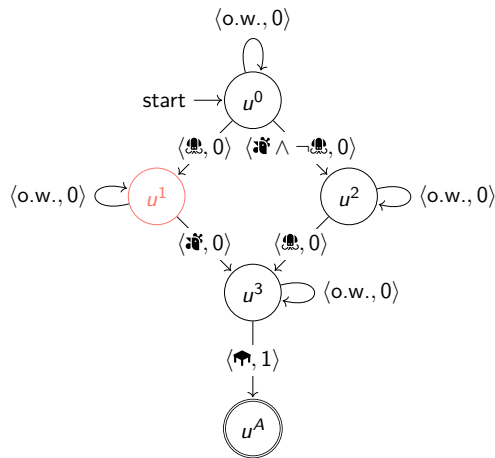
Motivation

Reward Machines

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



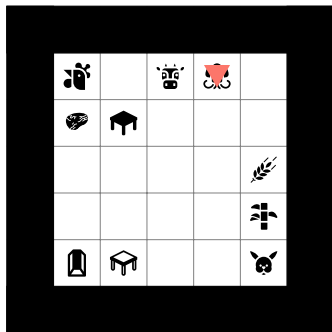
Events



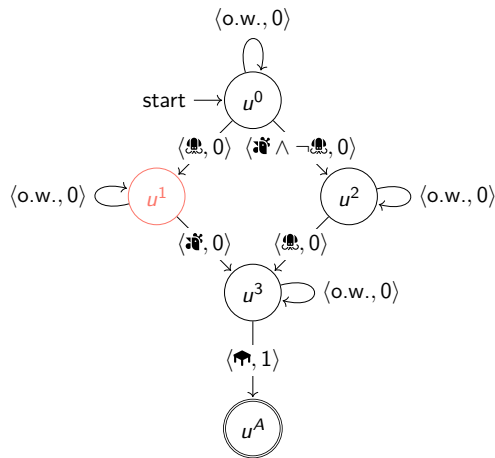
Motivation

Reward Machines

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



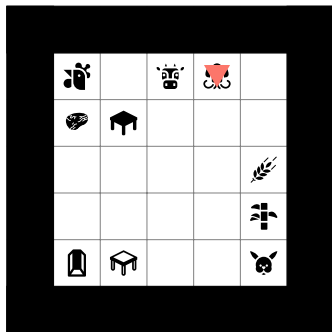
Events



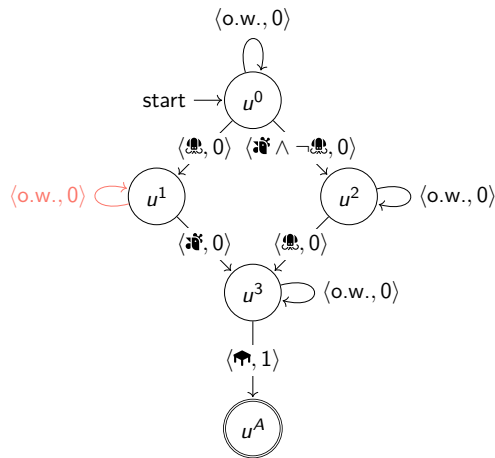
Motivation

Reward Machines

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



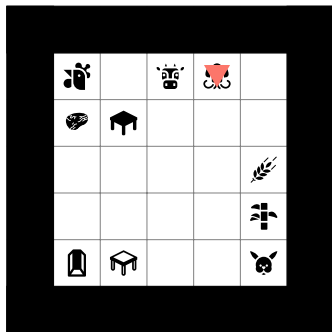
Events



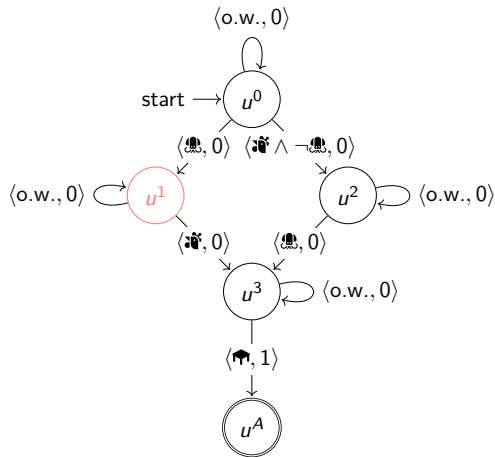
Motivation

Reward Machines

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



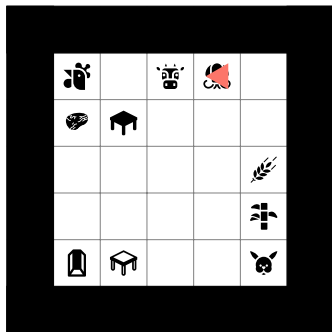
Events



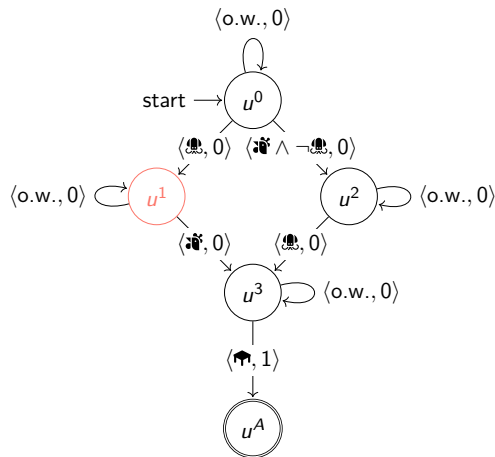
Motivation

Reward Machines

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



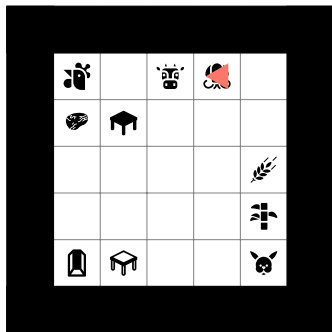
Events



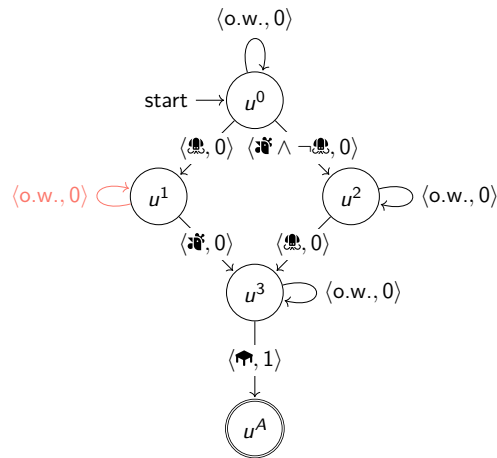
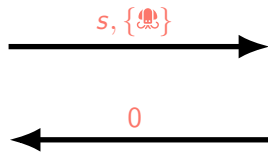
Motivation

Reward Machines

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



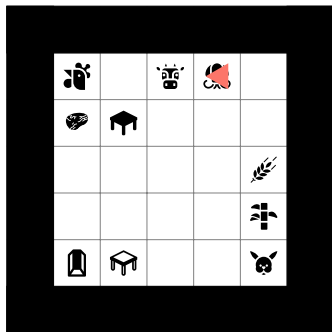
Events



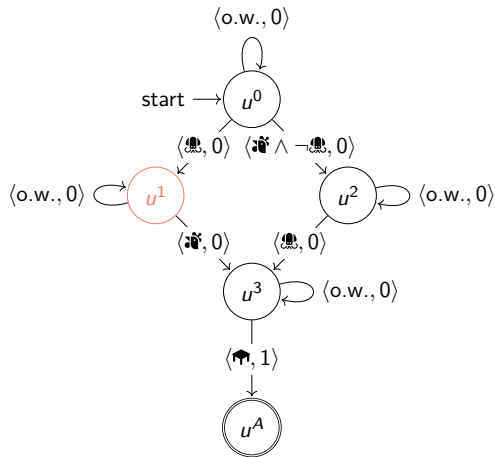
Motivation

Reward Machines

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



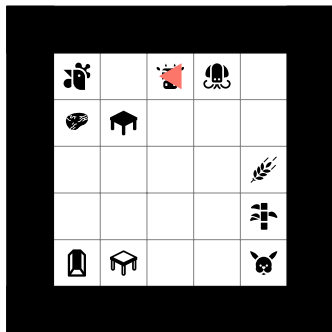
Events



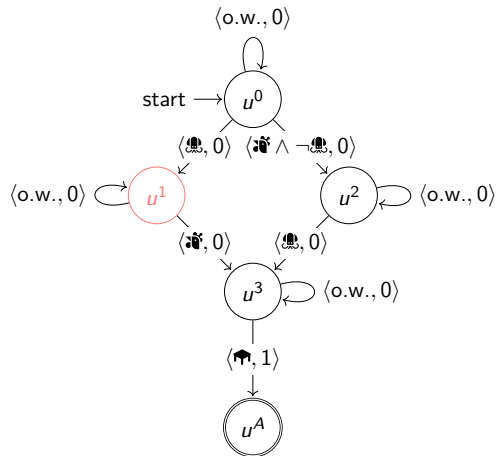
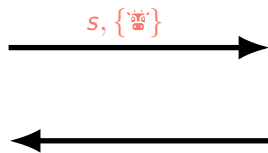
Motivation

Reward Machines

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



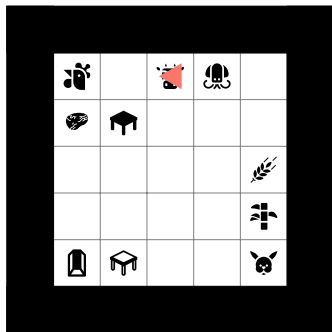
Events



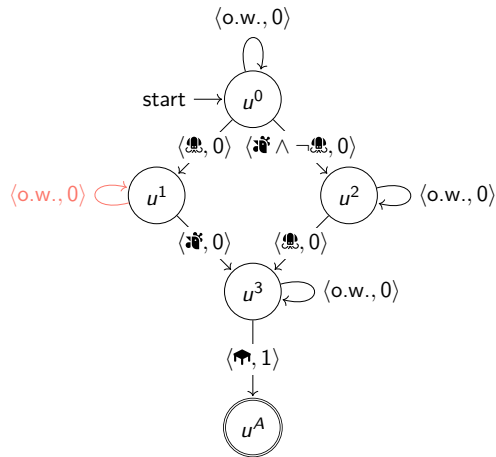
Motivation

Reward Machines

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



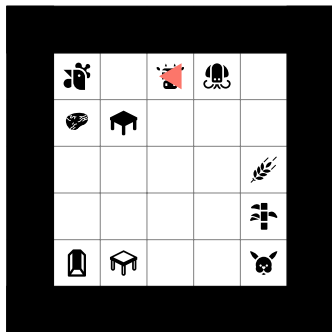
Events



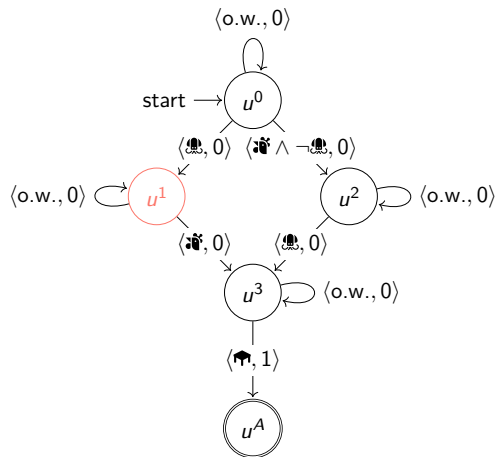
Motivation

Reward Machines

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



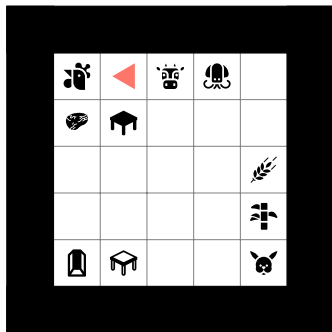
Events



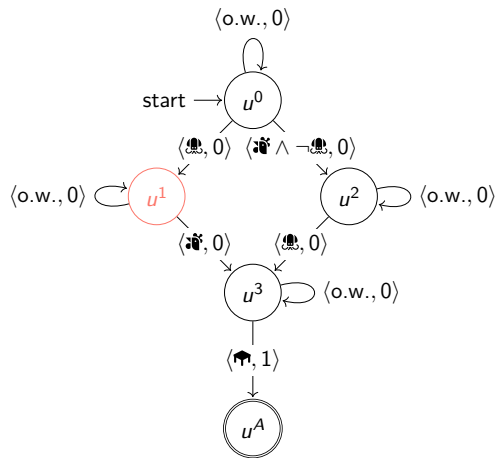
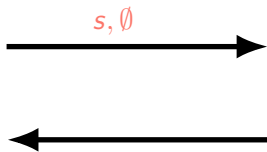
Motivation

Reward Machines

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



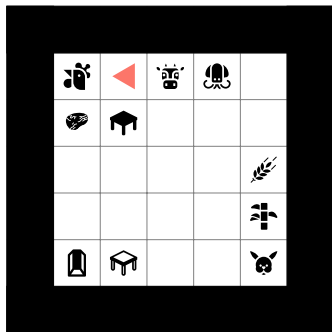
Events



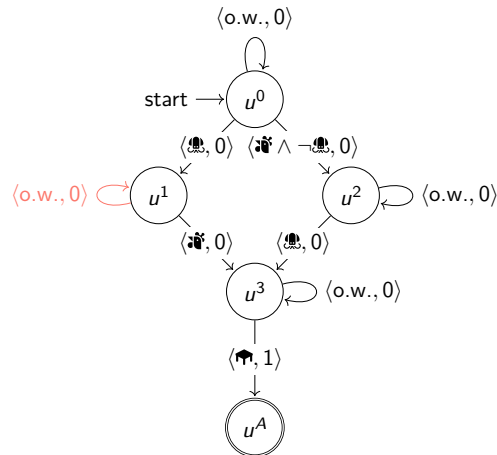
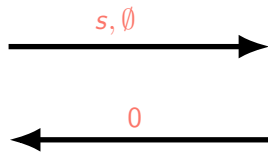
Motivation

Reward Machines

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



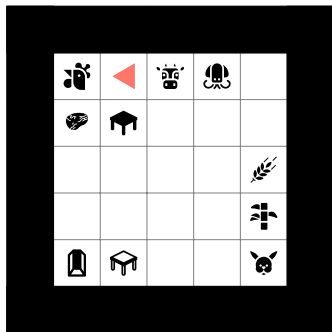
Events



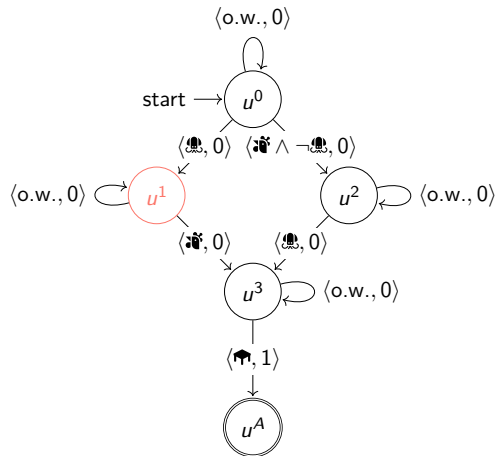
Motivation

Reward Machines

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



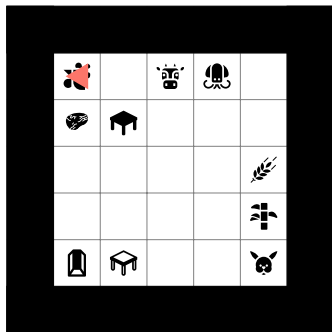
Events



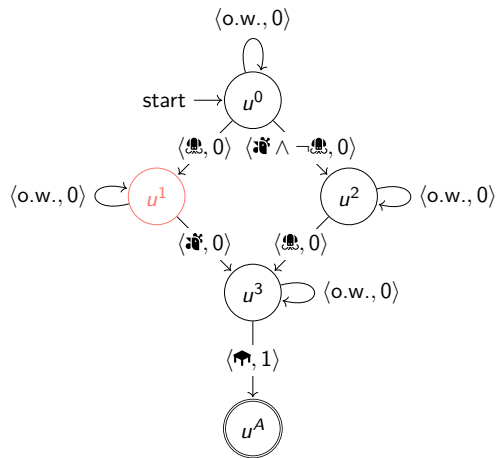
Motivation

Reward Machines

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



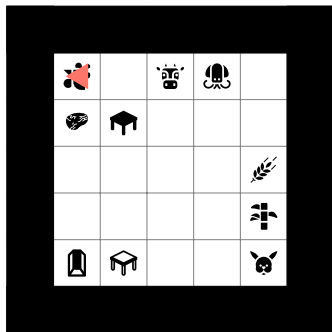
Events
{🏠, 🏠, 🍄, 🍄, 🍄, 🍄,
🍄, 🍄, 🍄, 🏠}



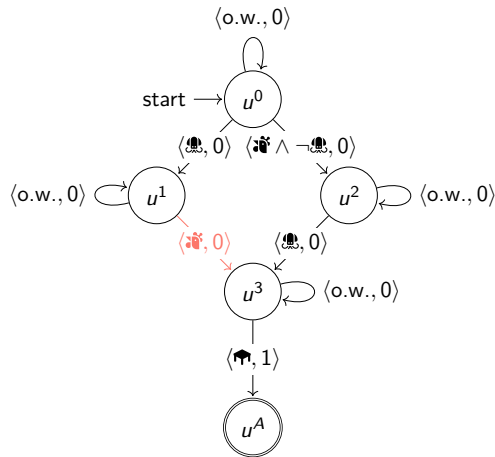
Motivation

Reward Machines

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



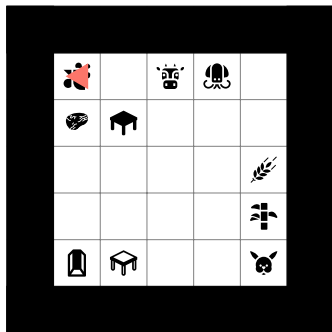
Events



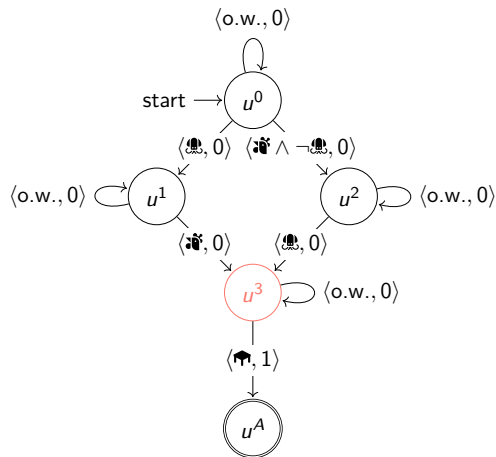
Motivation

Reward Machines

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



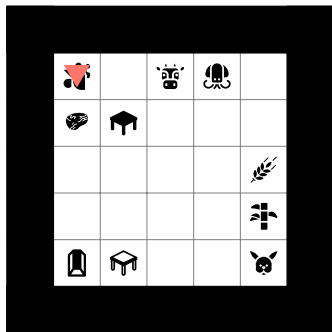
Events



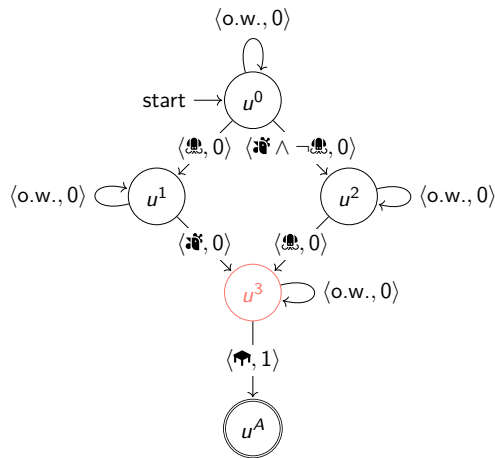
Motivation

Reward Machines

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



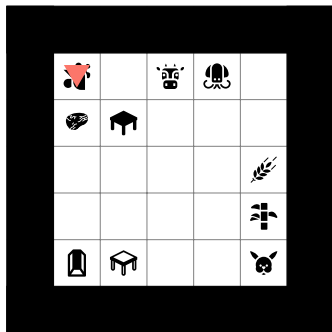
Events
{🏠, 🏠, 🍄, 🍄, 🍄, 🍄,
🍄, 🐞, 🐞, 🏠}



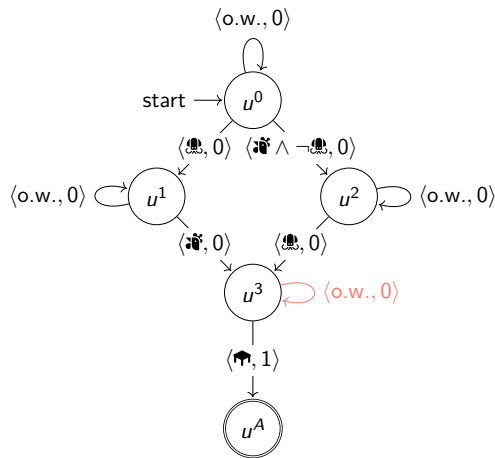
Motivation

Reward Machines

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



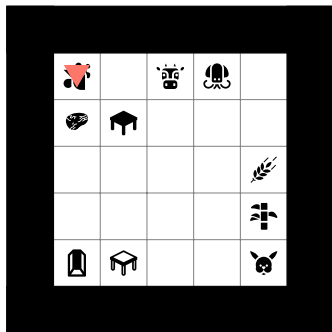
Events



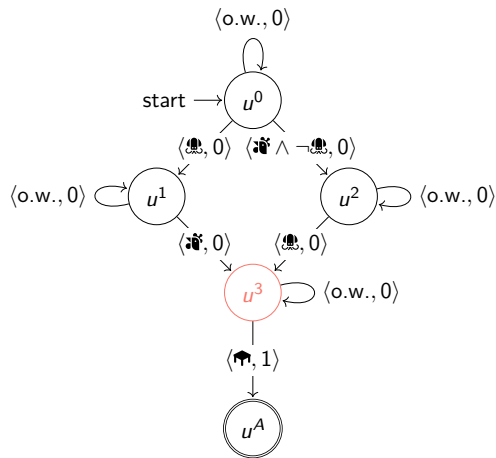
Motivation

Reward Machines

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



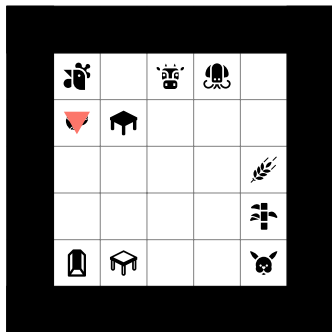
Events



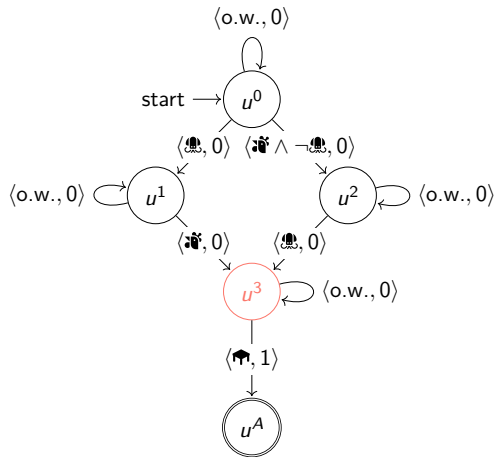
Motivation

Reward Machines

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



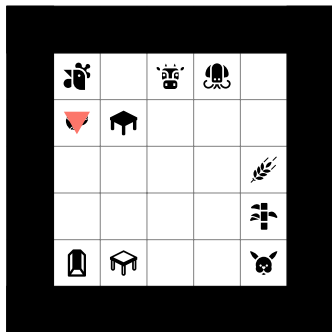
Events



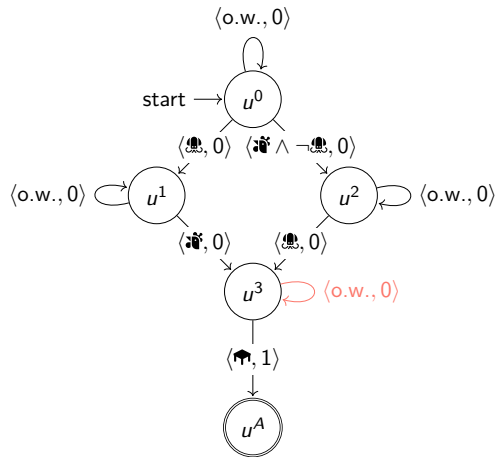
Motivation

Reward Machines

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



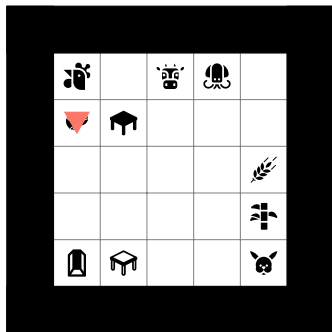
Events



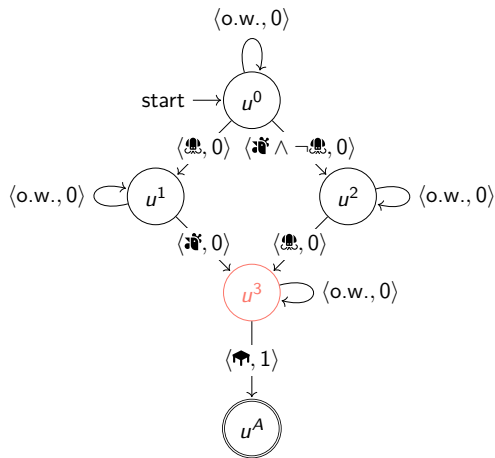
Motivation

Reward Machines

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



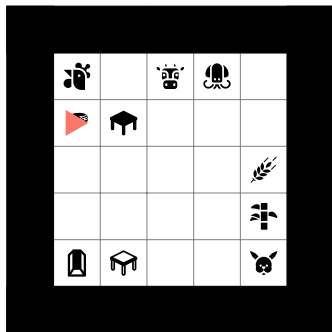
Events



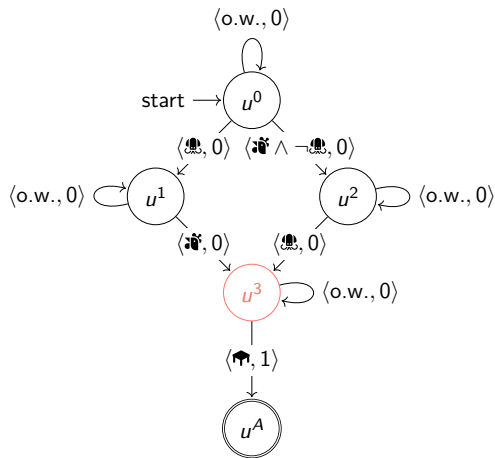
Motivation

Reward Machines

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



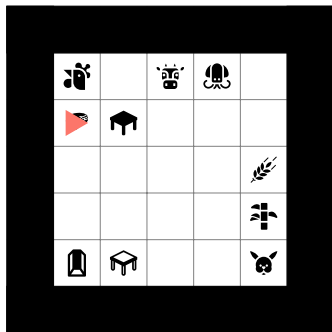
Events



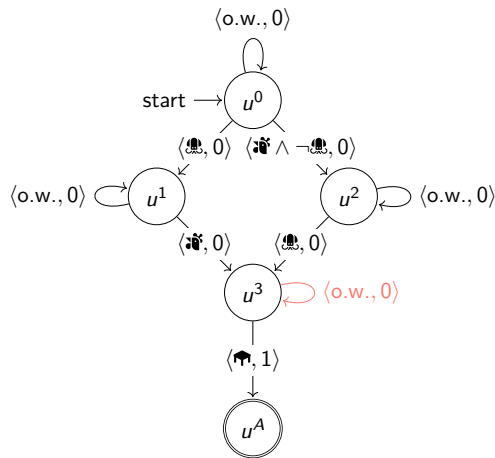
Motivation

Reward Machines

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



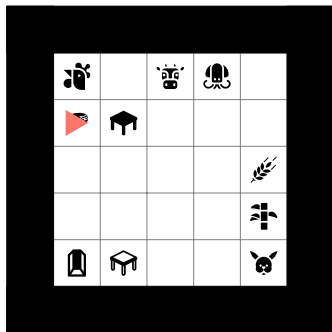
Events



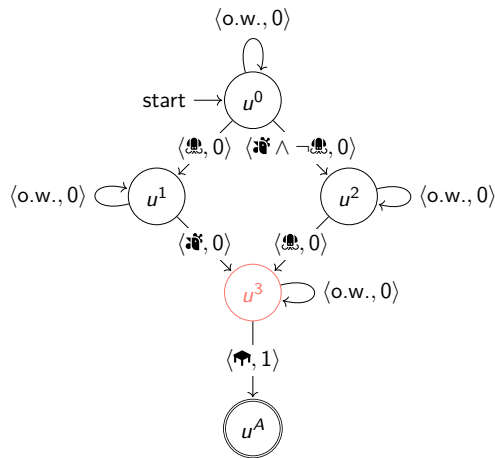
Motivation

Reward Machines

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



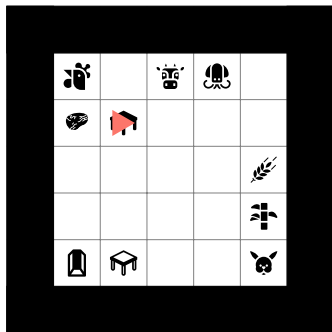
Events



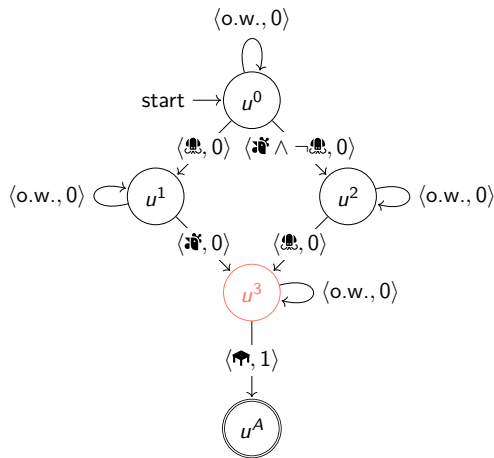
Motivation

Reward Machines

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



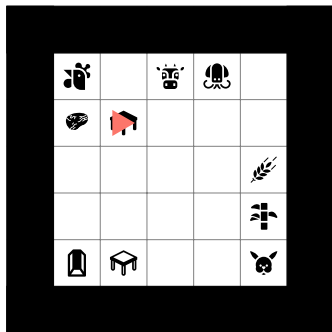
Events



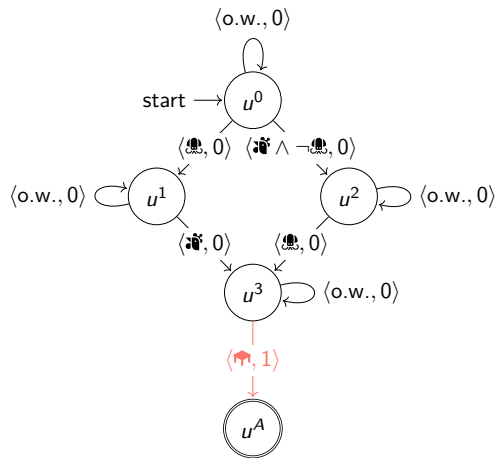
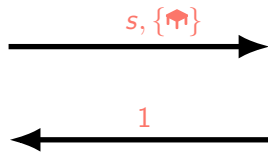
Motivation

Reward Machines

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



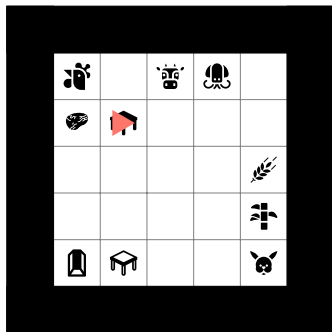
Events



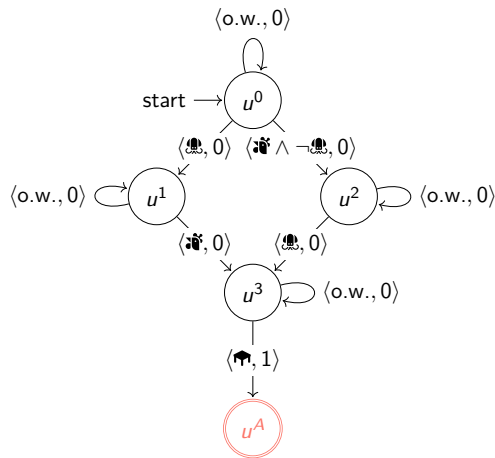
Motivation

Reward Machines

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



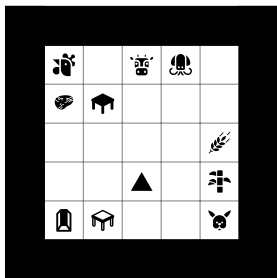
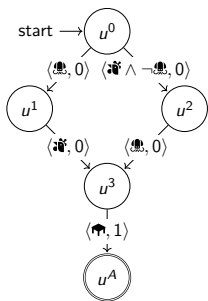
Events



Motivation

Reward Machines – Exploitation

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



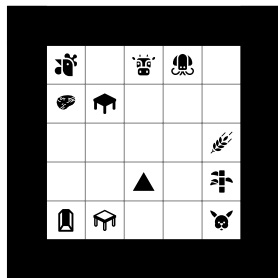
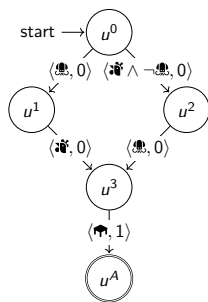
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

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

Motivation

Reward Machines – Exploitation

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



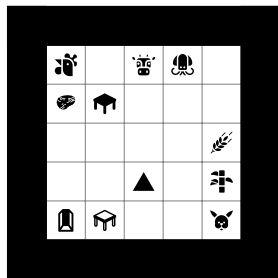
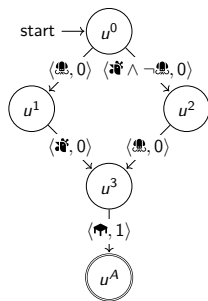
Fueros-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

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

Motivation

Reward Machines – Exploitation

- RMs 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 .



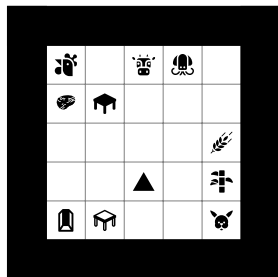
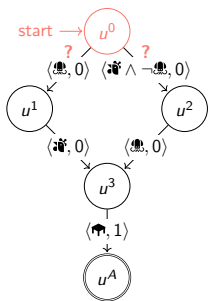
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

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

Motivation

Reward Machines – Exploitation

- RMs 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 .



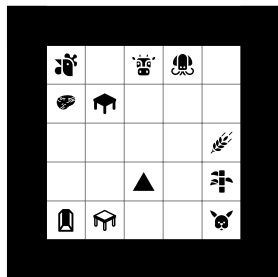
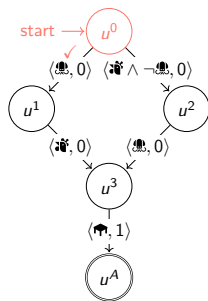
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

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

Motivation

Reward Machines – Exploitation

- RMs 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 .



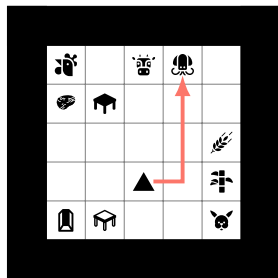
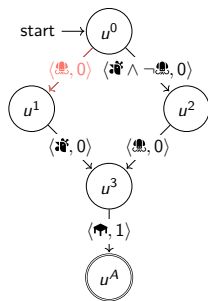
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

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

Motivation

Reward Machines – Exploitation

- RMs 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.



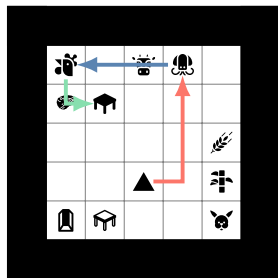
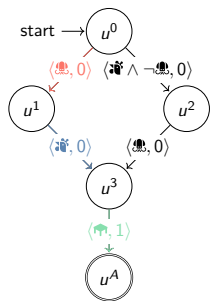
Furelos-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

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

Motivation

Reward Machines – Exploitation

- RMs 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.



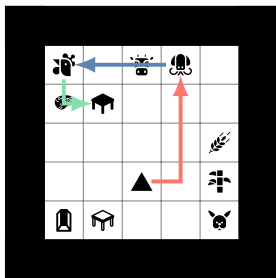
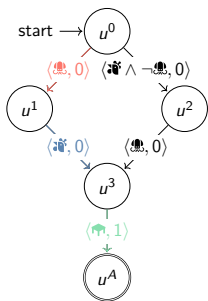
Fueiros-Blanco et al. "Induction and Exploitation of Subgoal Automata for Reinforcement Learning". JAIR, 2021.

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

Motivation

Reward Machines – Exploitation

- RMs 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)?

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.

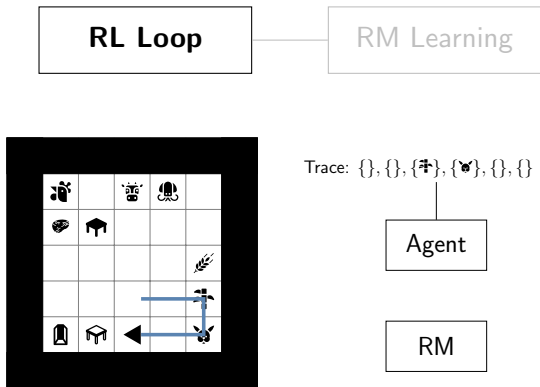
Gaon and Brafman. "Reinforcement Learning with Non-Markovian Rewards". AAAI, 2020.

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

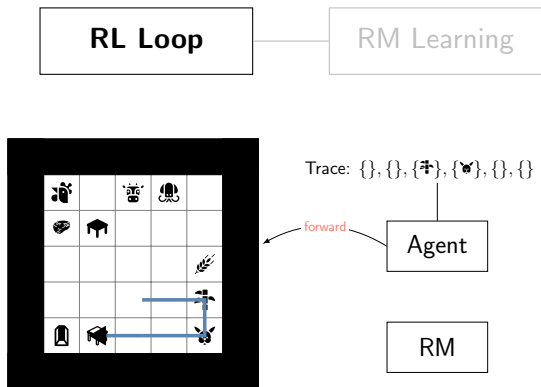
Reward Machines – Learning I



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

Motivation

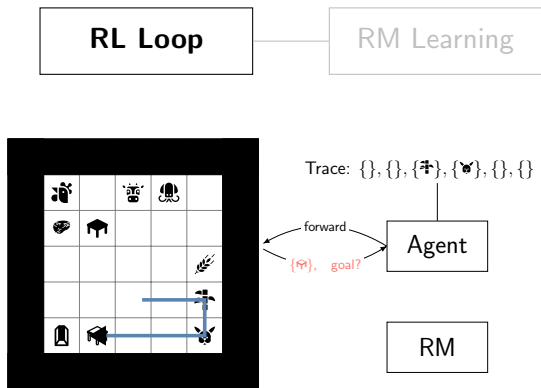
Reward Machines – Learning I



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

Motivation

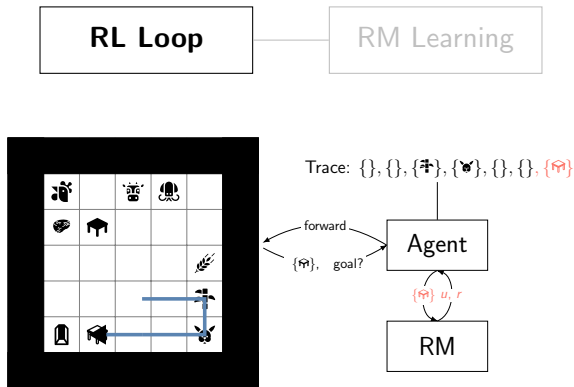
Reward Machines – Learning I



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

Motivation

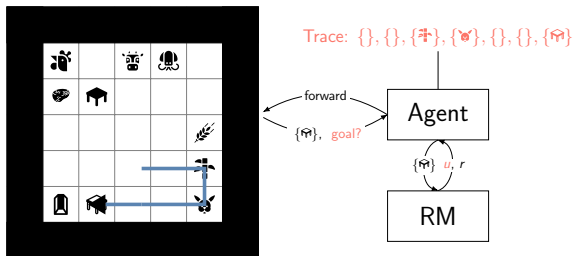
Reward Machines – Learning I



- 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

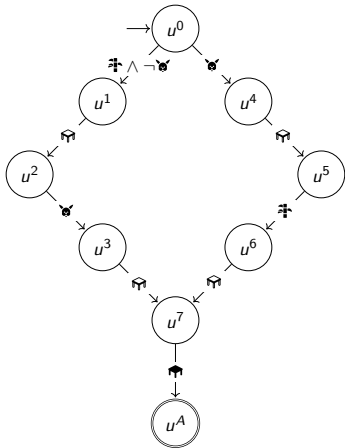


- A *new* RM is learned if the trace is a *counterexample* (e.g., reaches the task's goal but not the accepting state).

Motivation

Reward Machines – Learning II

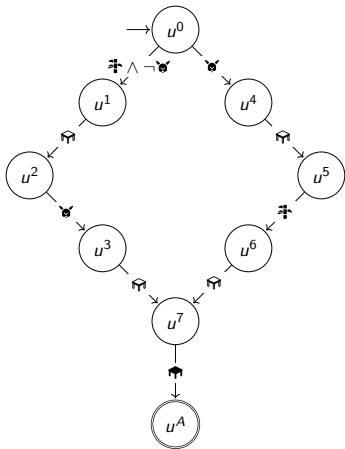
- Learning of *minimal* RMs (i.e., with the fewest possible states) scales poorly with the number of states.



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?

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?

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?

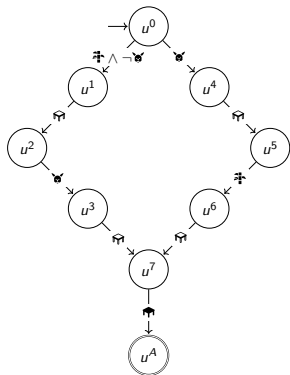
Our Approach

Construct hierarchies of reward machines!

Hierarchies of Reward Machines

Formalism I

RM

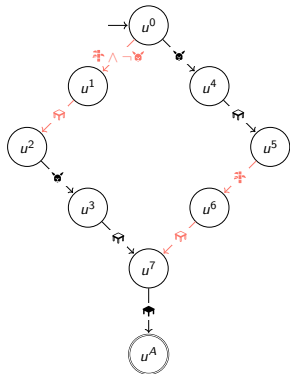


HRM

Hierarchies of Reward Machines

Formalism I

RM

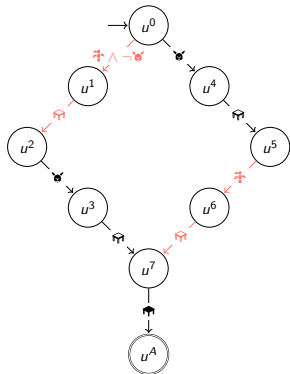


HRM

Hierarchies of Reward Machines

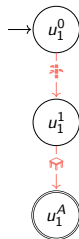
Formalism I

RM



HRM

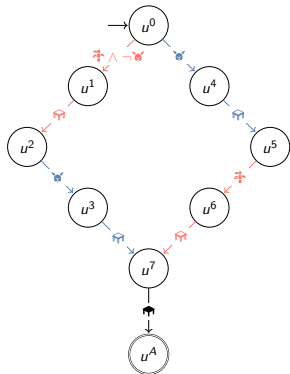
M_1



Hierarchies of Reward Machines

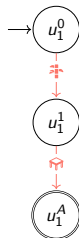
Formalism I

RM



HRM

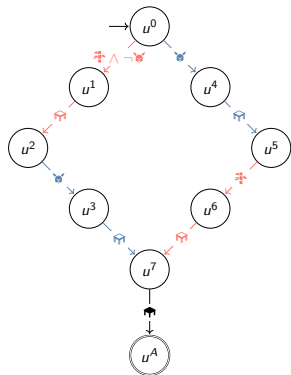
M_1



Hierarchies of Reward Machines

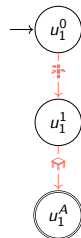
Formalism I

RM

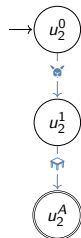


HRM

M_1



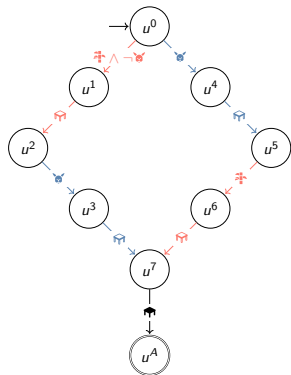
M_2



Hierarchies of Reward Machines

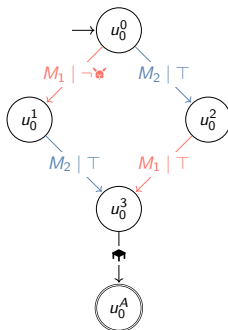
Formalism I

RM

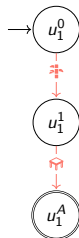


HRM

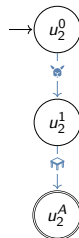
M_0 (root)



M_1



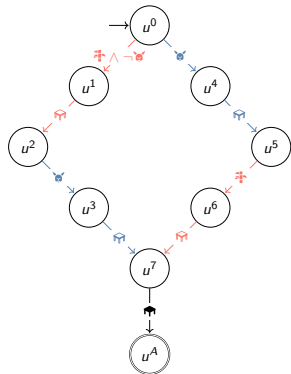
M_2



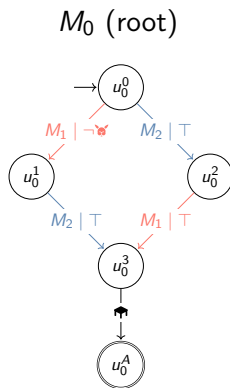
Hierarchies of Reward Machines

Formalism I

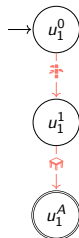
RM



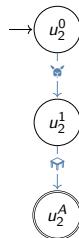
HRM



M_1



M_2

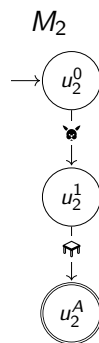
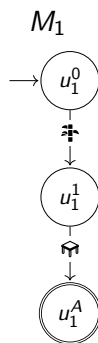
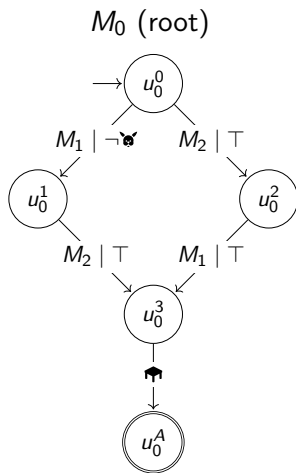


Properties

- 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

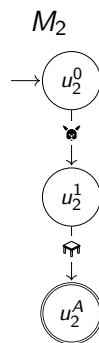
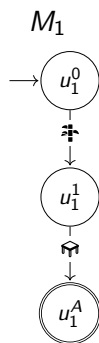
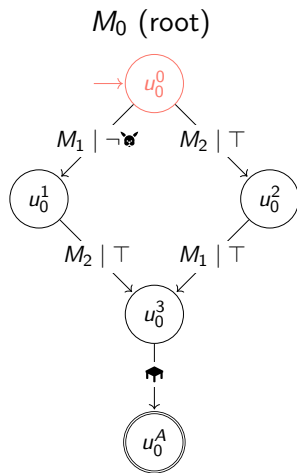


Trace = \langle

Stack = \square

Hierarchies of Reward Machines

Formalism II

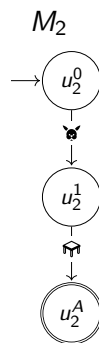
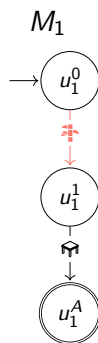
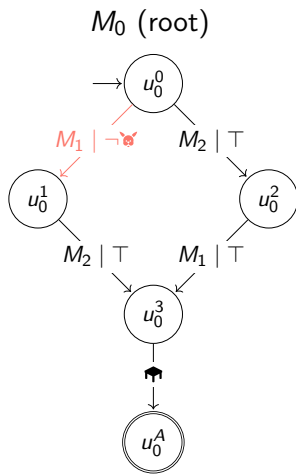


Trace = \langle

Stack = \square

Hierarchies of Reward Machines

Formalism II

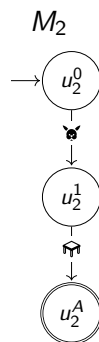
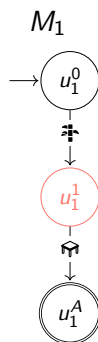
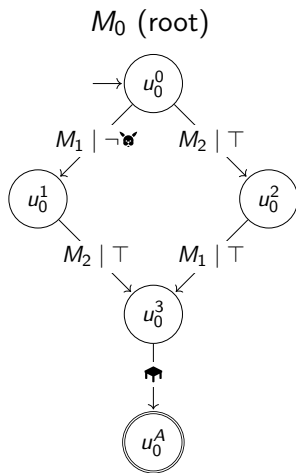


Trace = $\langle \{ \text{X} \} \rangle$,

Stack = $\langle \rangle$

Hierarchies of Reward Machines

Formalism II

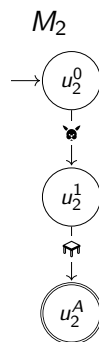
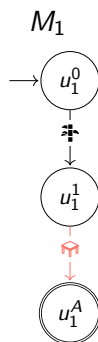
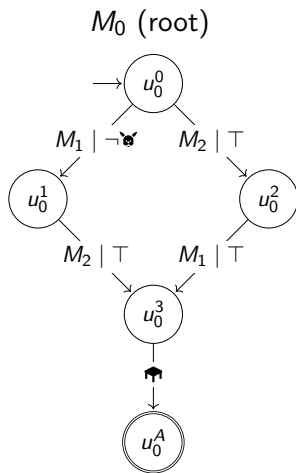


Trace = $\langle \{\text{bug}\} \rangle$,

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

Hierarchies of Reward Machines

Formalism II

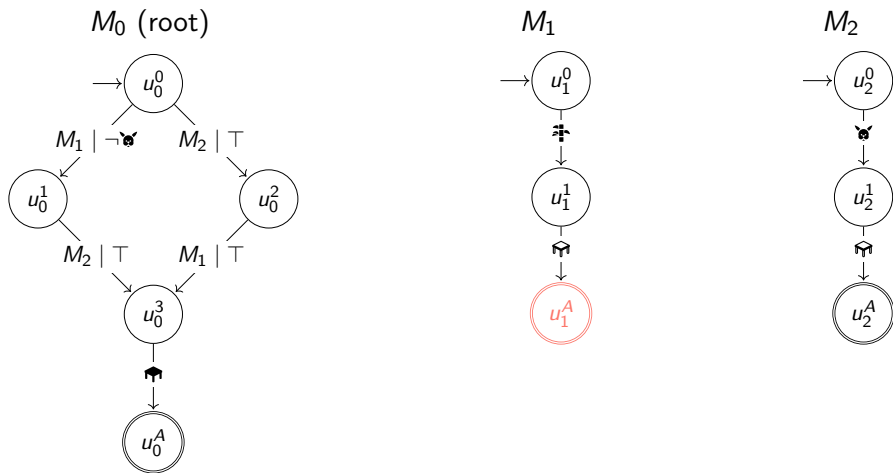


Trace = $\langle \{\text{bug}\}, \{\text{house}\} \rangle$,

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

Hierarchies of Reward Machines

Formalism II

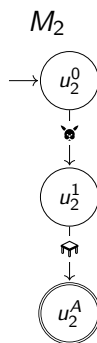
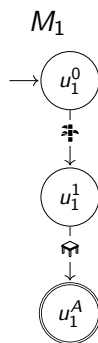
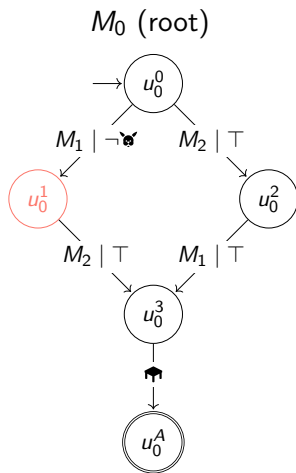


Trace = $\langle \{\ddagger\}, \{\text{house}\},$

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

Hierarchies of Reward Machines

Formalism II

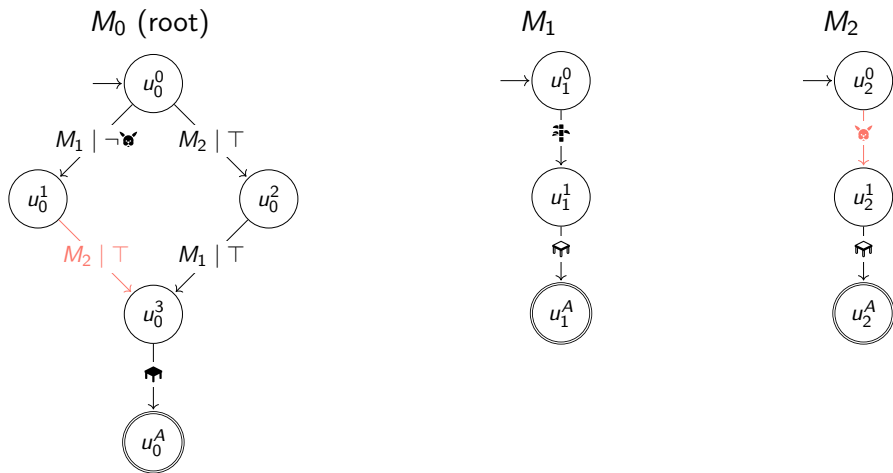


Trace = $\langle \{\text{eye-crossed-out}\}, \{\text{house}\} \rangle$,

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

Hierarchies of Reward Machines

Formalism II

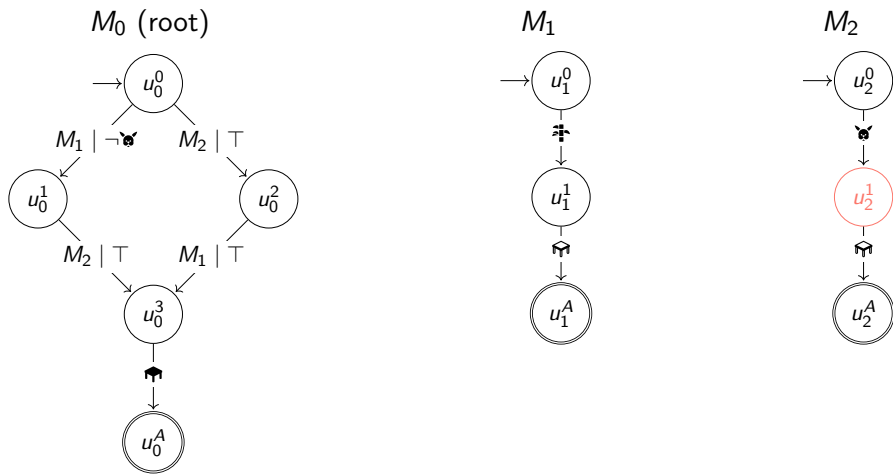


Trace = $\langle \{\text{house with roof}\}, \{\text{house}\}, \{\text{red cat face}\} \rangle$

Stack = $\langle \rangle$

Hierarchies of Reward Machines

Formalism II

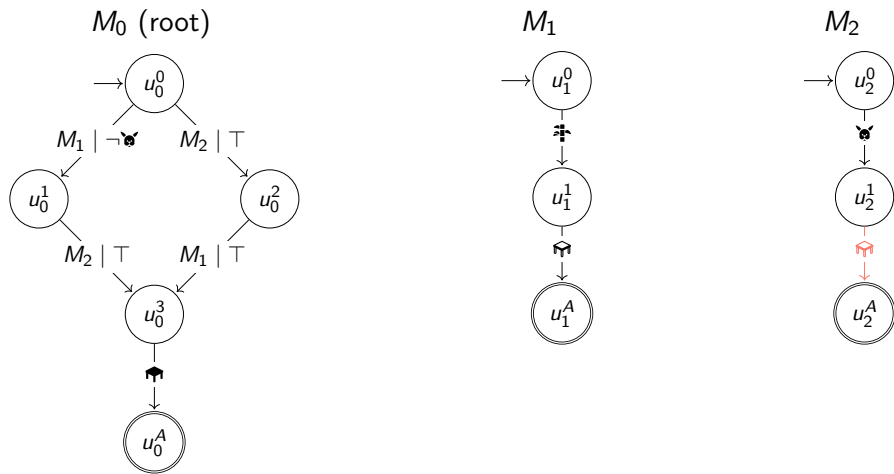


Trace = $\langle \{\ddagger\}, \{\text{house}\}, \{\text{cat}\} \rangle$,

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

Hierarchies of Reward Machines

Formalism II

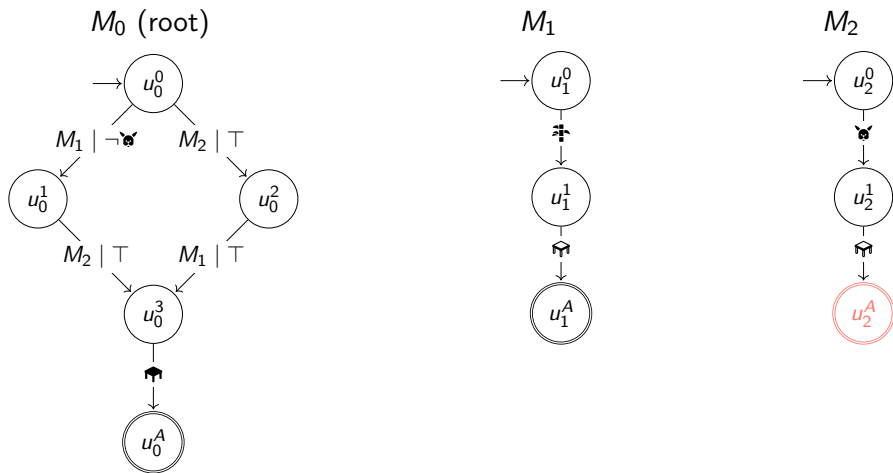


Trace = $\langle \{\text{cat}\}, \{\text{house}\}, \{\text{cat}\}, \{\text{house}\} \rangle$

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

Hierarchies of Reward Machines

Formalism II

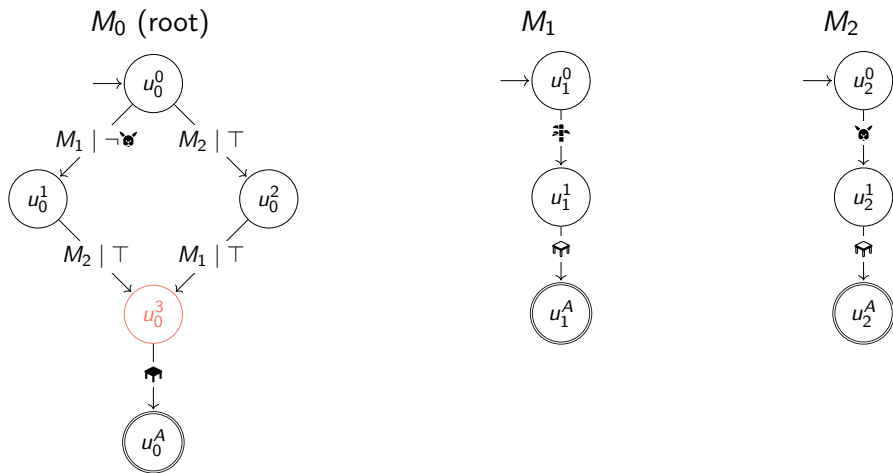


Trace = $\langle \{\text{bird}\}, \{\text{house}\}, \{\text{bird}\}, \{\text{house}\} \rangle$,

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

Hierarchies of Reward Machines

Formalism II

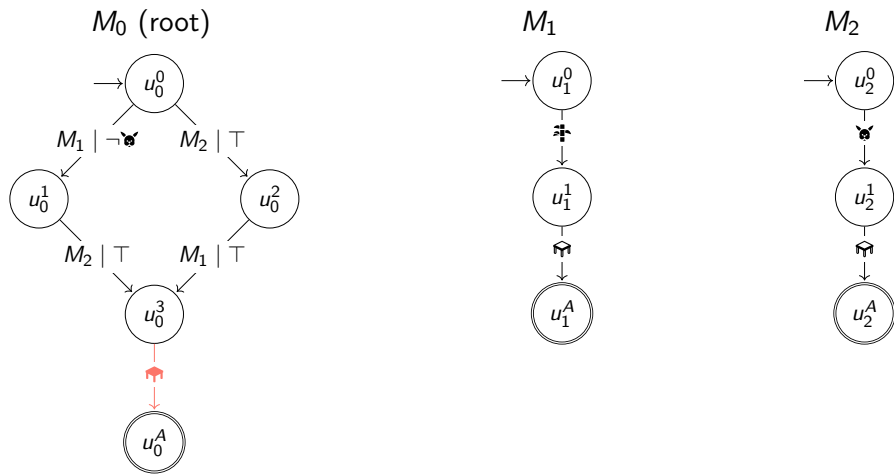


Trace = $\langle \{\ddagger\}, \{\uparrow\}, \{\blacktriangleright\}, \{\uparrow\} \rangle$,

Stack = $\langle \langle \cancel{M_0}, u_0^3 \rangle \rangle$

Hierarchies of Reward Machines

Formalism II

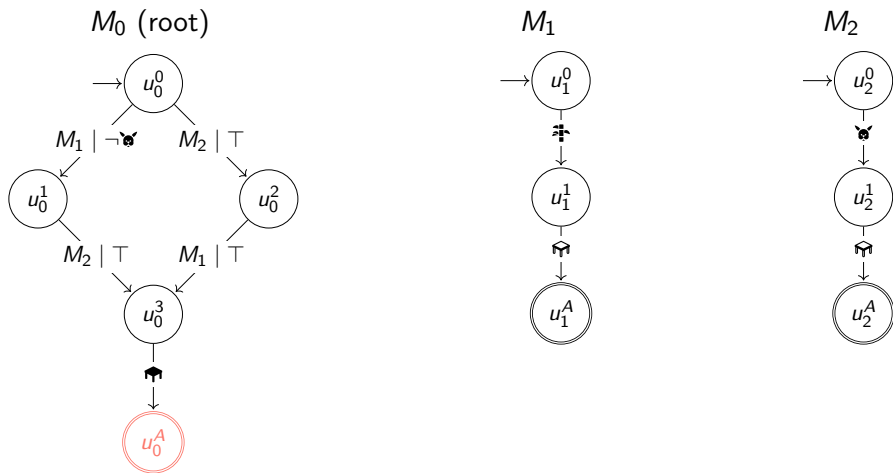


Trace = $\langle \{\text{cross}\}, \{\text{house}\}, \{\text{cat}\}, \{\text{house}\}, \{\text{house}\} \rangle$

Stack = \square

Hierarchies of Reward Machines

Formalism II



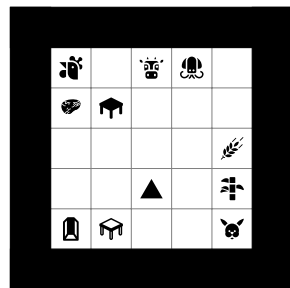
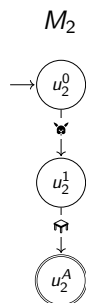
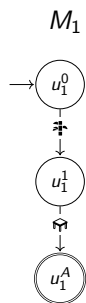
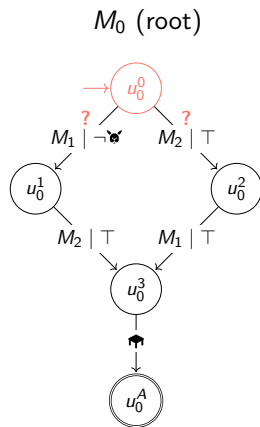
Trace = $\langle \{ \text{cat} \}, \{ \text{house} \}, \{ \text{cat} \}, \{ \text{house} \}, \{ \text{house} \} \rangle$

Stack = \square

Hierarchies of Reward Machines

Exploitation

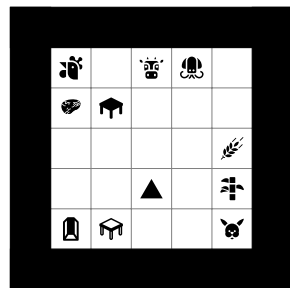
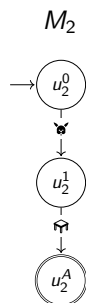
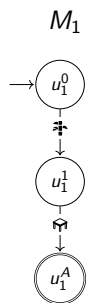
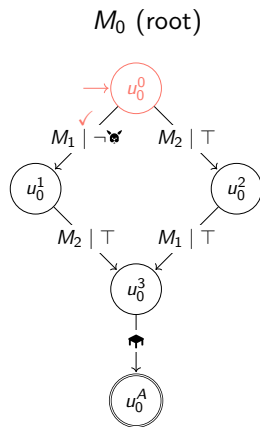
- The structure of an HRM can be exploited *hierarchically*:
 - RM policies** – Choose formulas or calls to (eventually) reach an accepting state.
 - Formula policies** – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



Hierarchies of Reward Machines

Exploitation

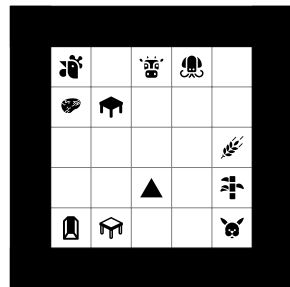
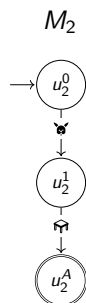
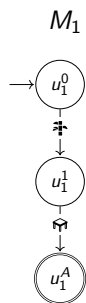
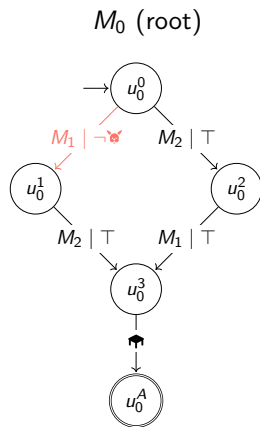
- The structure of an HRM can be exploited *hierarchically*:
 - RM policies** – Choose formulas or calls to (eventually) reach an accepting state.
 - Formula policies** – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



Hierarchies of Reward Machines

Exploitation

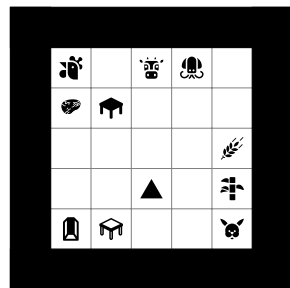
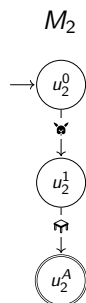
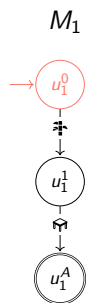
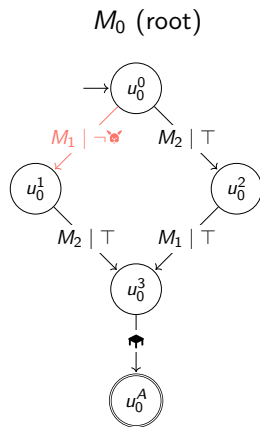
- The structure of an HRM can be exploited *hierarchically*:
 - RM policies** – Choose formulas or calls to (eventually) reach an accepting state.
 - Formula policies** – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



Hierarchies of Reward Machines

Exploitation

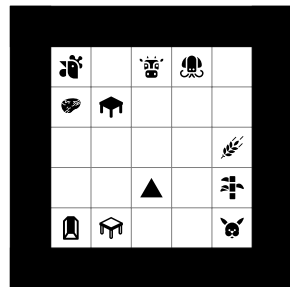
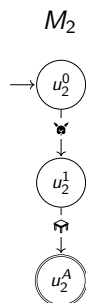
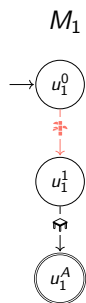
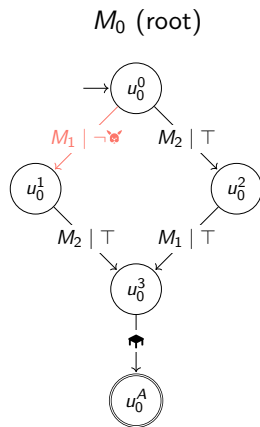
- The structure of an HRM can be exploited *hierarchically*:
 - RM policies** – Choose formulas or calls to (eventually) reach an accepting state.
 - Formula policies** – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



Hierarchies of Reward Machines

Exploitation

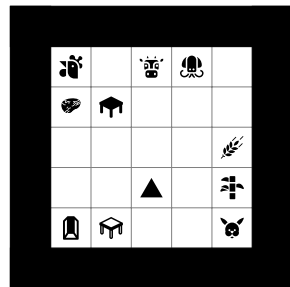
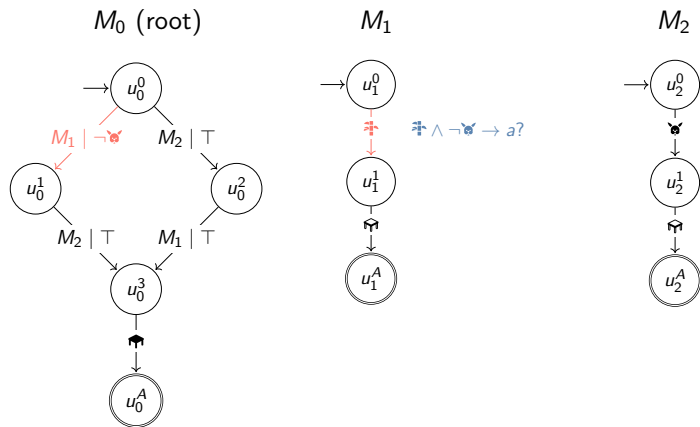
- The structure of an HRM can be exploited *hierarchically*:
 - RM policies** – Choose formulas or calls to (eventually) reach an accepting state.
 - Formula policies** – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



Hierarchies of Reward Machines

Exploitation

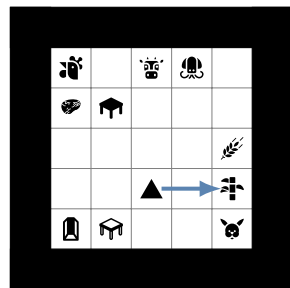
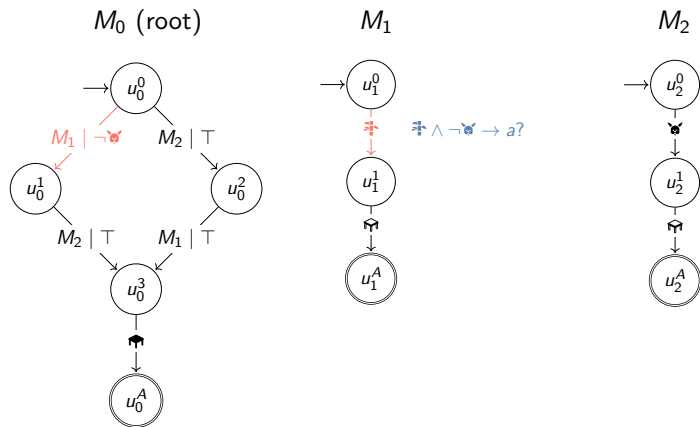
- The structure of an HRM can be exploited *hierarchically*:
 - RM policies** – Choose formulas or calls to (eventually) reach an accepting state.
 - Formula policies** – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



Hierarchies of Reward Machines

Exploitation

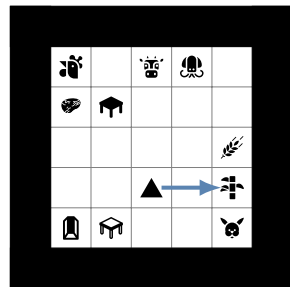
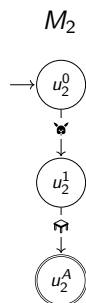
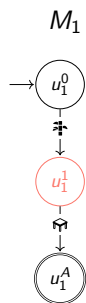
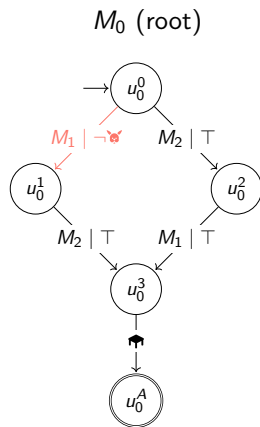
- The structure of an HRM can be exploited *hierarchically*:
 - RM policies** – Choose formulas or calls to (eventually) reach an accepting state.
 - Formula policies** – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



Hierarchies of Reward Machines

Exploitation

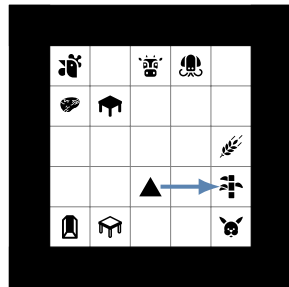
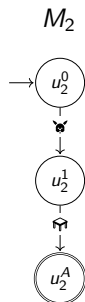
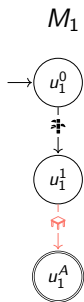
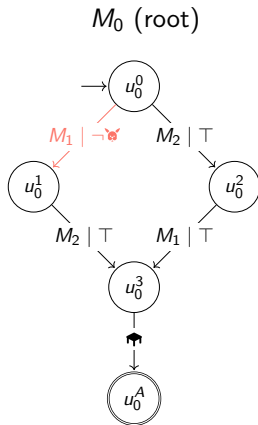
- The structure of an HRM can be exploited *hierarchically*:
 - RM policies** – Choose formulas or calls to (eventually) reach an accepting state.
 - Formula policies** – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



Hierarchies of Reward Machines

Exploitation

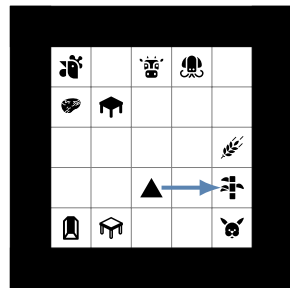
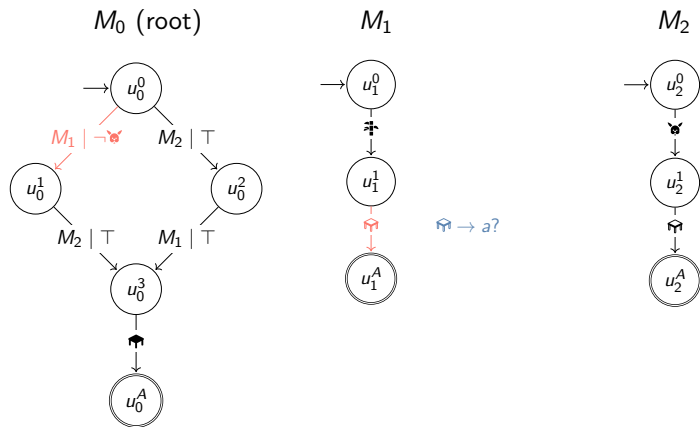
- The structure of an HRM can be exploited *hierarchically*:
 - RM policies** – Choose formulas or calls to (eventually) reach an accepting state.
 - Formula policies** – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



Hierarchies of Reward Machines

Exploitation

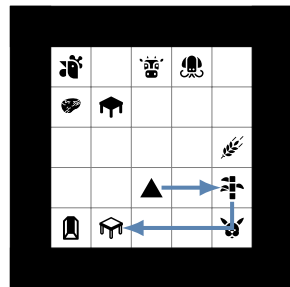
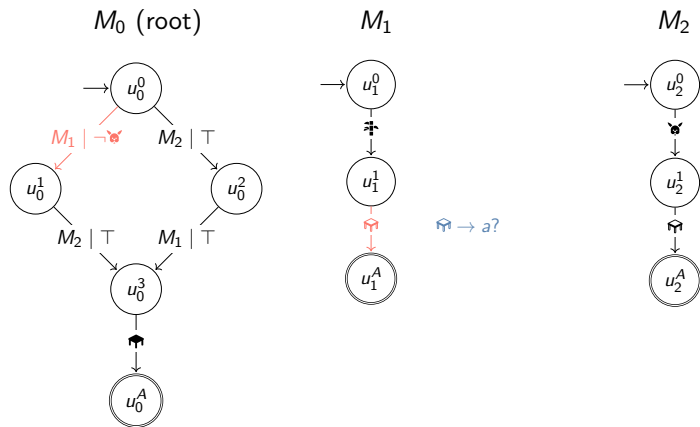
- The structure of an HRM can be exploited *hierarchically*:
 - RM policies** – Choose formulas or calls to (eventually) reach an accepting state.
 - Formula policies** – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



Hierarchies of Reward Machines

Exploitation

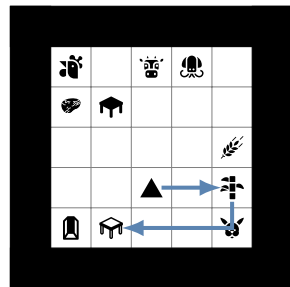
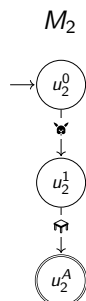
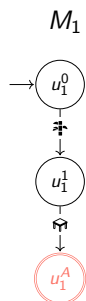
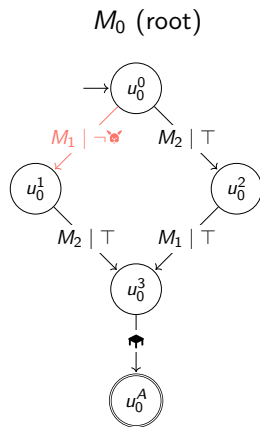
- The structure of an HRM can be exploited *hierarchically*:
 - RM policies** – Choose formulas or calls to (eventually) reach an accepting state.
 - Formula policies** – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



Hierarchies of Reward Machines

Exploitation

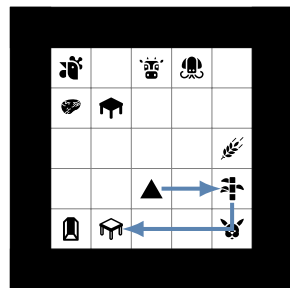
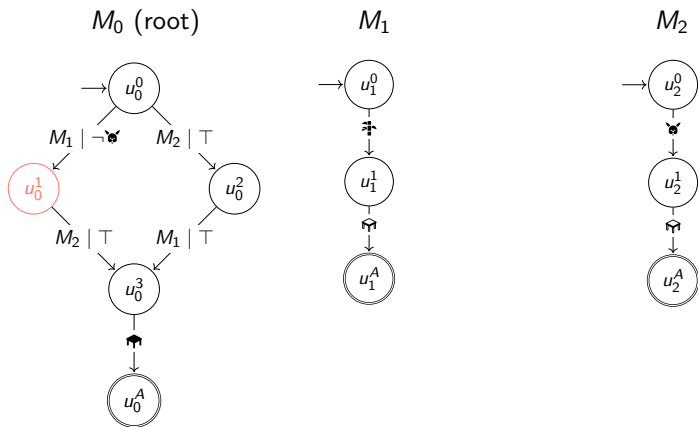
- The structure of an HRM can be exploited *hierarchically*:
 - RM policies** – Choose formulas or calls to (eventually) reach an accepting state.
 - Formula policies** – Choose actions to (eventually) satisfy a formula.
- Subgoals are selected top-down the hierarchical structure until a formula is chosen.



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

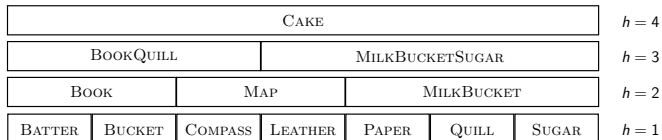


Hierarchies of Reward Machines

Learning I

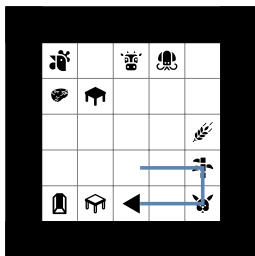


- An HRM is learned for each task.
- Each task has a *level* h .
- Learning proceeds from lower to higher levels.
- Level is increased when the average performance surpasses a *threshold*.

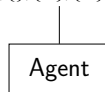


Hierarchies of Reward Machines

Learning I



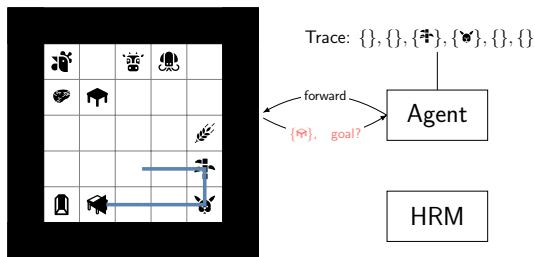
Trace: {}, {}, {+}, {*}, {}, {}



- 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

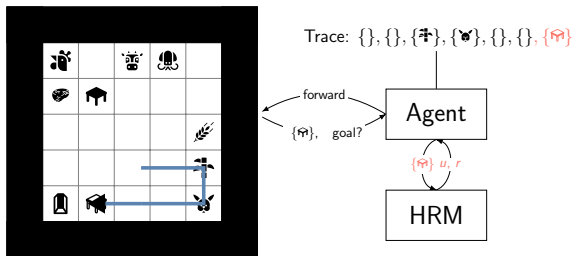
Learning I



- 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

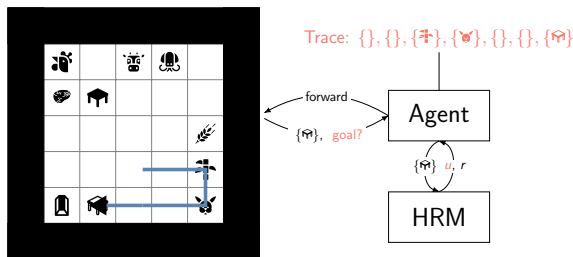
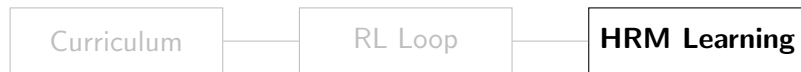
Learning I



- 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

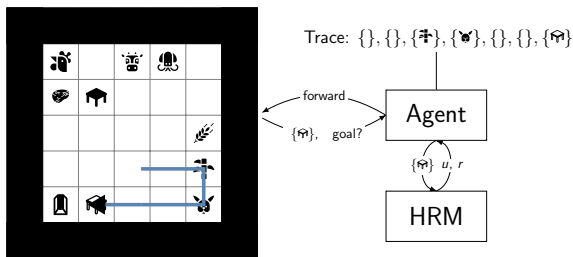
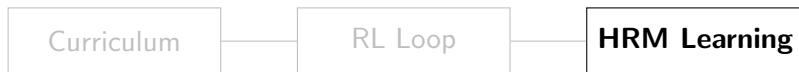
Learning I



- 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

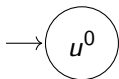


- HRMs for lower-level tasks may be called.
- Lower-level task policies can be used for *exploration*: observing goal traces becomes easier!

Hierarchies of Reward Machines



Learning II

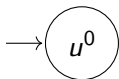
Example: Sequence of RMs learned for the task “Collect 📦 then go to 🏠”.



Hierarchies of Reward Machines

Learning II



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



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

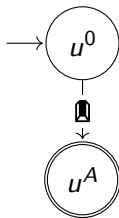
Hierarchies of Reward Machines

Learning II

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





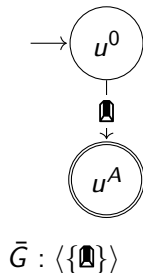
$G : \langle \{\text{key}\}, \{\text{house}\} \rangle$



Hierarchies of Reward Machines



Learning II

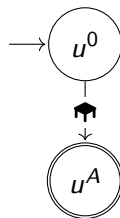
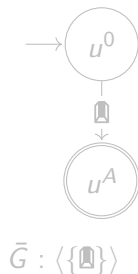
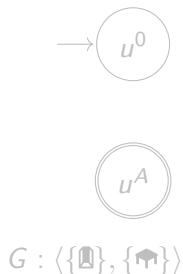
Example: Sequence of RMs learned for the task “Collect  then go to ”.



Hierarchies of Reward Machines



Learning II

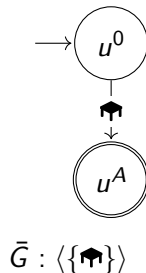
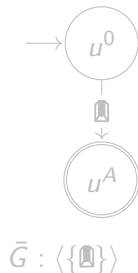
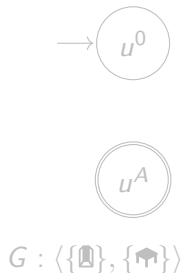
Example: Sequence of RMs learned for the task “Collect  then go to ”.



Hierarchies of Reward Machines


Learning II

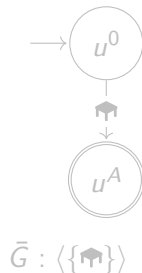
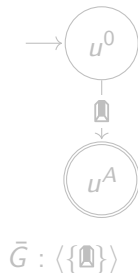
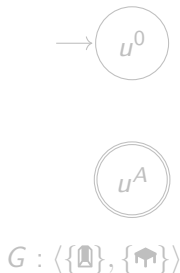
Example: Sequence of RMs learned for the task “Collect  then go to ”.



Hierarchies of Reward Machines

Learning II



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



UNSATISFIABLE,
Increment the number
of states!

Hierarchies of Reward Machines

Learning II

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



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

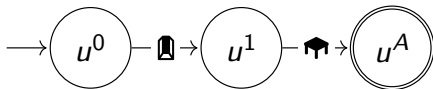


$\bar{G} : \langle \{ \text{key} \} \rangle$





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

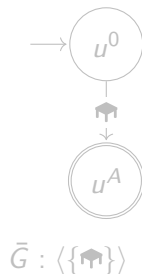
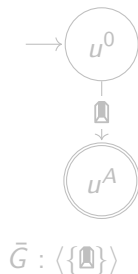
UNSATISFIABLE,
Increment the number
of states!



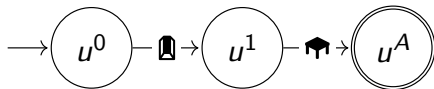
Hierarchies of Reward Machines

Learning II

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



UNSATISFIABLE,
Increment the number
of states!



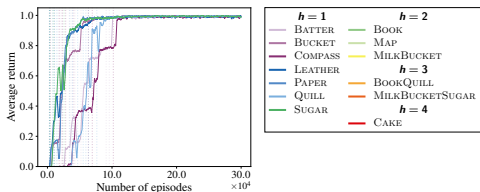
In higher-level tasks, lower-level RMs can be called.

Evaluation

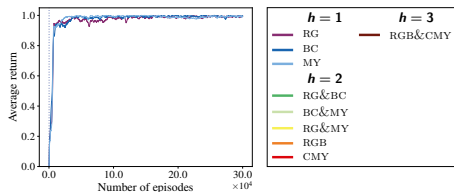
Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD

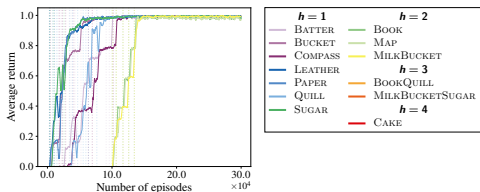


Evaluation

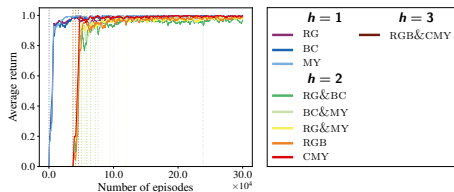
Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD

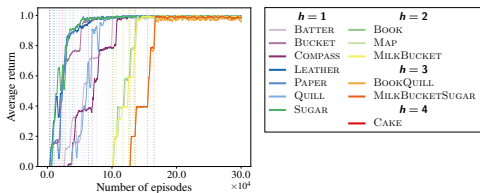


Evaluation

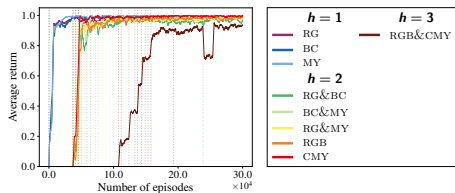
Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD

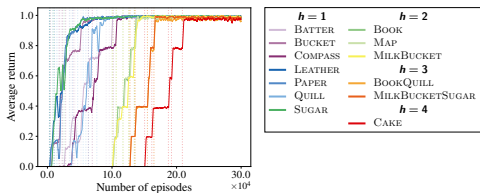


Evaluation

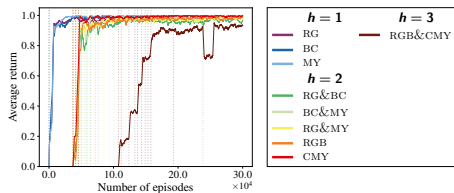
Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD

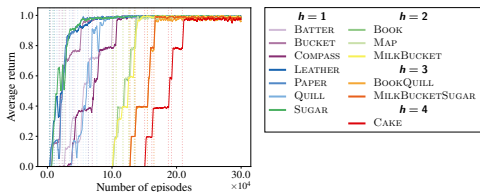


Evaluation

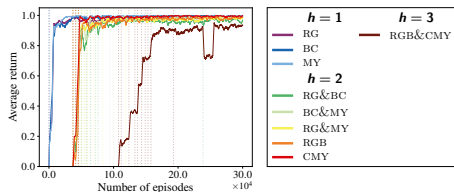
Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD



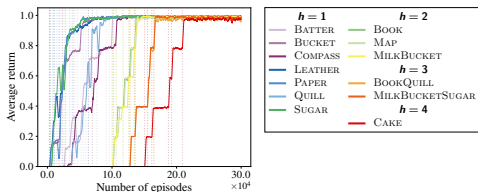
Insights:

Evaluation

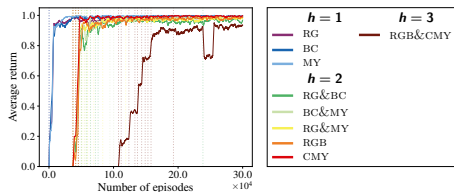
Learning of HRMs

HRM learning is feasible in two different domains.

CRAFTWORLD



WATERWORLD



Insights:

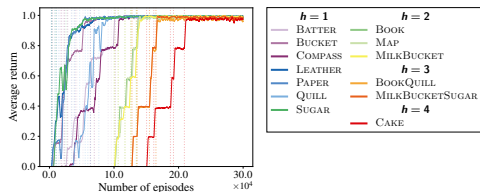
- 1 HRM learning becomes less scalable as the number of tasks and levels grows.

Evaluation

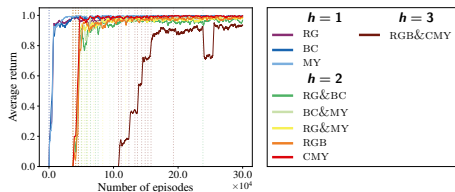
Learning of HRMs

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.

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:

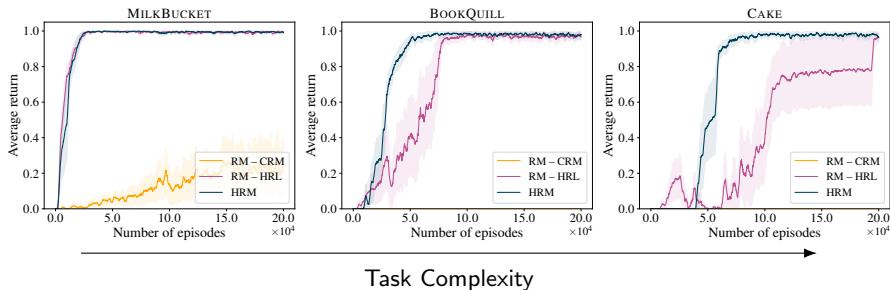
- ① Minimal RM learning methods poorly scale as the number of states increases.
- ② DeepSynth and LRM tend to overfit to the observed traces.
- ③ DeepSynth, JIRP and LRM need exponentially more edges in `WATERWORLD` since they do not use formulas.

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

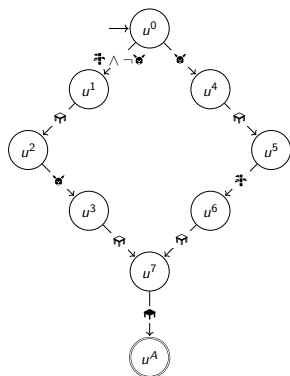
- Build RM learning methods that adapt to changing environments or agent capabilities (e.g., traces that achieved the goal but later do not).

- 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

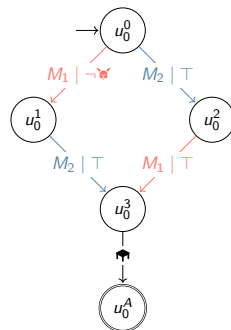
- 1 HRMs, a formalism for hierarchically composing RMs.

RM

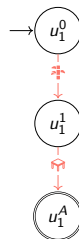


HRM

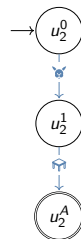
M_0 (root)



M_1

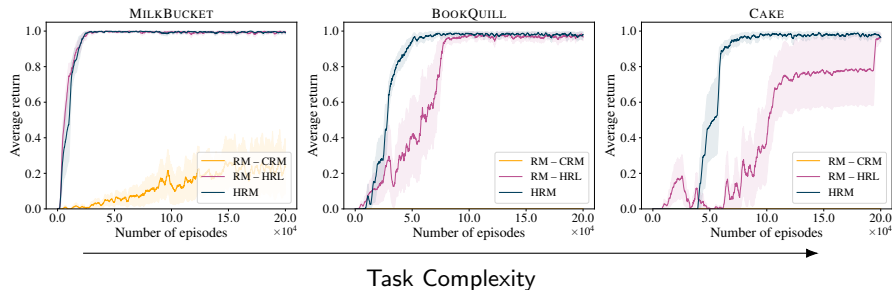


M_2



Conclusions

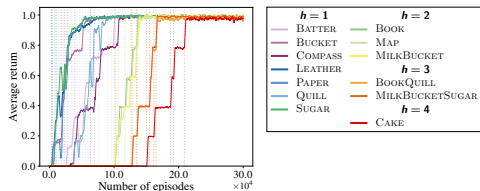
- ① HRMs, a formalism for hierarchically composing RMs.
- ② A method that *exploits* the structure of an HRM.



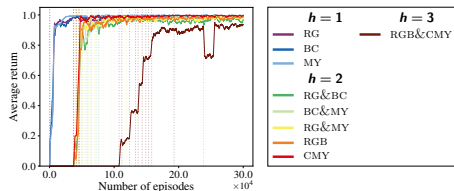
Conclusions

- 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?