

Interactively Learning a Blend of Goal-Based and Procedural Tasks

Aaron Mininger and John E. Laird

University of Michigan

Published in AAI 18

May 17, 2018

2018 Soar Workshop

Interactive Task Learning

What are different ways an agent can formulate a task?

- Achieving a goal
- Maximizing a reward or objective function
- Following a procedure
- Keeping within some constraints

Goal-Based Formulation

Formulate the task as achieving a goal

Use planning and search to select actions



Discard the soda.

What is the goal?

The goal is that the
soda is in the garbage



Goal-Based Formulation

Explanation-Based Learning Approach

Explanation of why actions led to the goal -> policy

- + Works over complex, relational task structures
- + Exploits rich domain knowledge to learn from few examples
- + Flexible to variations and novel conditions
- Requires extensive domain knowledge
- All or nothing approach
- Hard to describe a goal for some tasks

Procedural Formulation

Formulate the task as following a procedure

Rely on the procedure to execute the task in the future



Lead a tour.

Go to the copy room.

Say 'Here is where you can make copies.'

Go to the kitchen.

Say 'Here is where you can eat lunch.'

⋮



Procedural Formulation

Formulate the task as following a procedure

Directly represent the steps needed to perform the task

- + Does not require specialized domain knowledge
- + Can incorporate complex control flow
- + Easy to correct, modify, and extend
- Difficult to adapt to problems that arise
- Requires the instructor to know the actions the agent can take

Goal-Based vs Procedural

Both formulations have unique tradeoffs

Which is more suitable may depend on:

- The specific task being taught
- The preferences of the instructor
- The capabilities and knowledge of the agent

Interactive Task Learning

Extended Rosie to support learning procedural tasks

Can learn tasks:

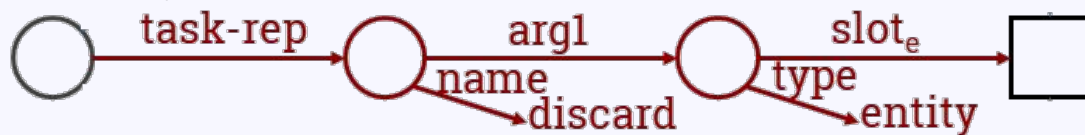
- Where the goal is not easily expressed
- When the agent lacks domain knowledge
- Which blend both formulations

Previous Approach: Learning Goal-Based Tasks

Learning Goal-Based Form.

1. Extract task structure from command

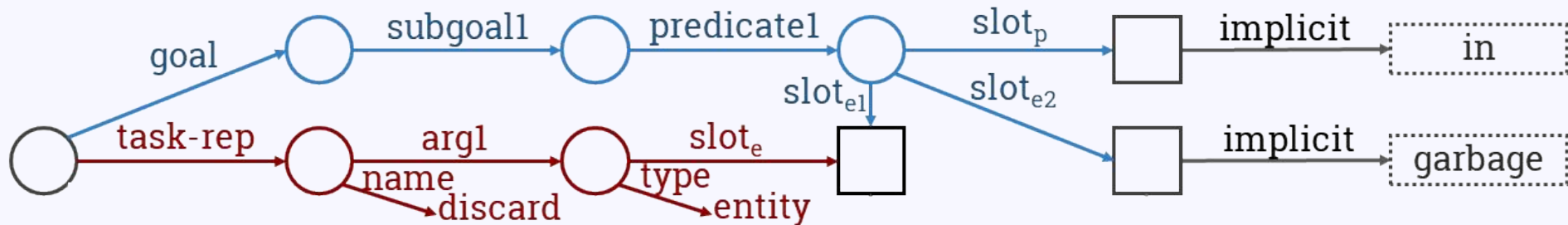
Discard the soda.



Learning Goal-Based Form.

2. Store a representation of the goal

The goal is that the
soda is in the garbage



Learning Goal-Based Form.

3. Learn a goal elaboration rule

```
if
  task = discard( $e_1$ )
  garbage( $e_2$ )
then
  goal = in( $e_1$ ,  $e_2$ )
```

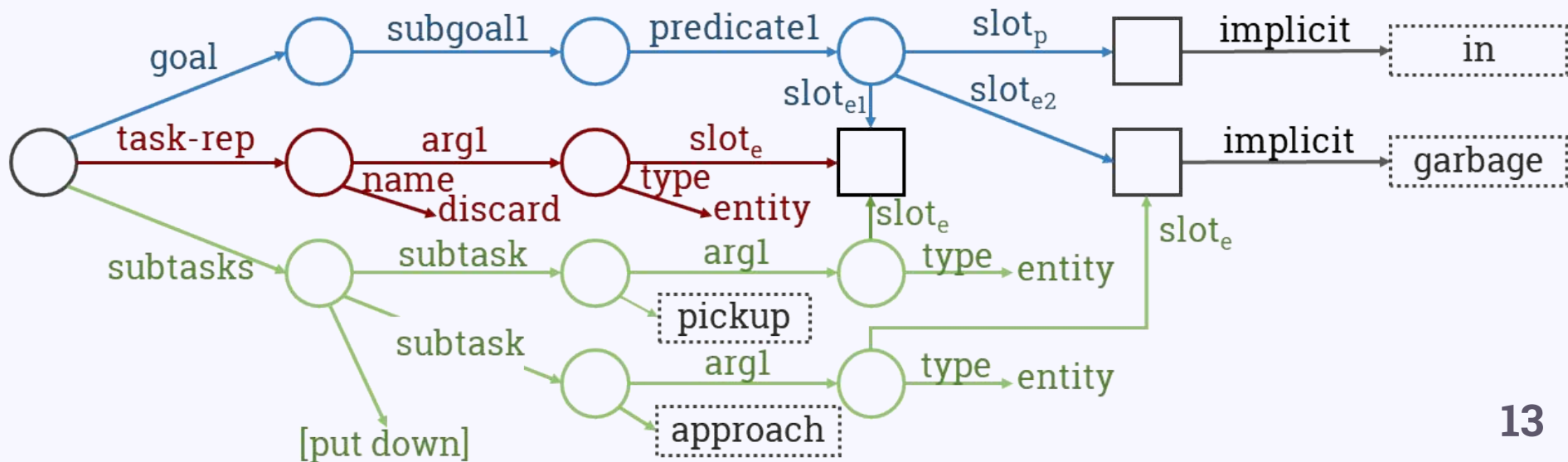
Learning Goal-Based Form.

4. Perform actions to achieve goal

Pick up the soda

Approach the garbage

Put the soda in the garbage



Learning Goal-Based Form.

5. Use EBG to learn policy rules

```
if
  task = discard( $e_1$ )
  grabbed( $e_1$ )
  garbage( $e_2$ )
  near( $e_{self}$ ,  $e_2$ )
then
  perform put-down( $e_1$ ,  $e_2$ )
```

Learning Goal-Based Form.

What if the goal cannot be easily expressed?

Learning Procedural Formulation

Learning Procedural Form.

Teach the task of giving a tour



Lead a tour.

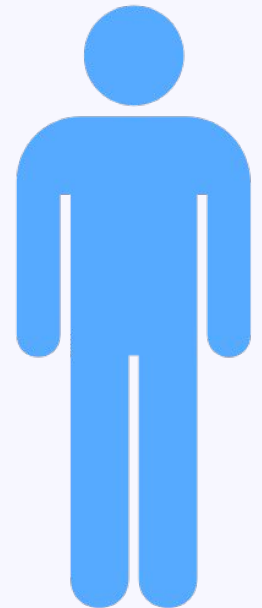
Go to the main office.

Say 'This is the main office.'

Go to the kitchen.

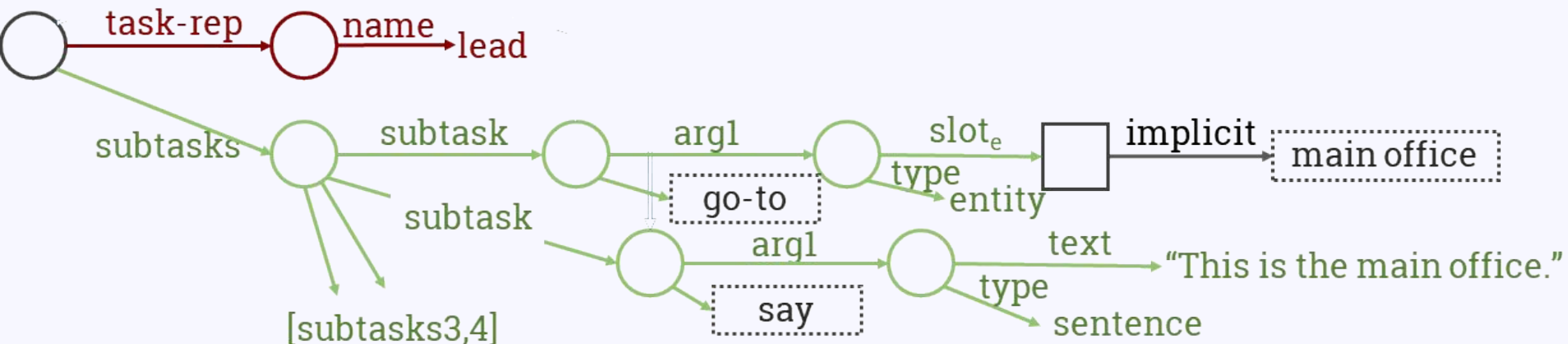
Say 'Here is where you can eat lunch.'

You are done.



Learning Procedural Form.

Teach the task of giving a tour



Learning Procedural Form.

Retrospective Learning: With Goal

initial state: discard

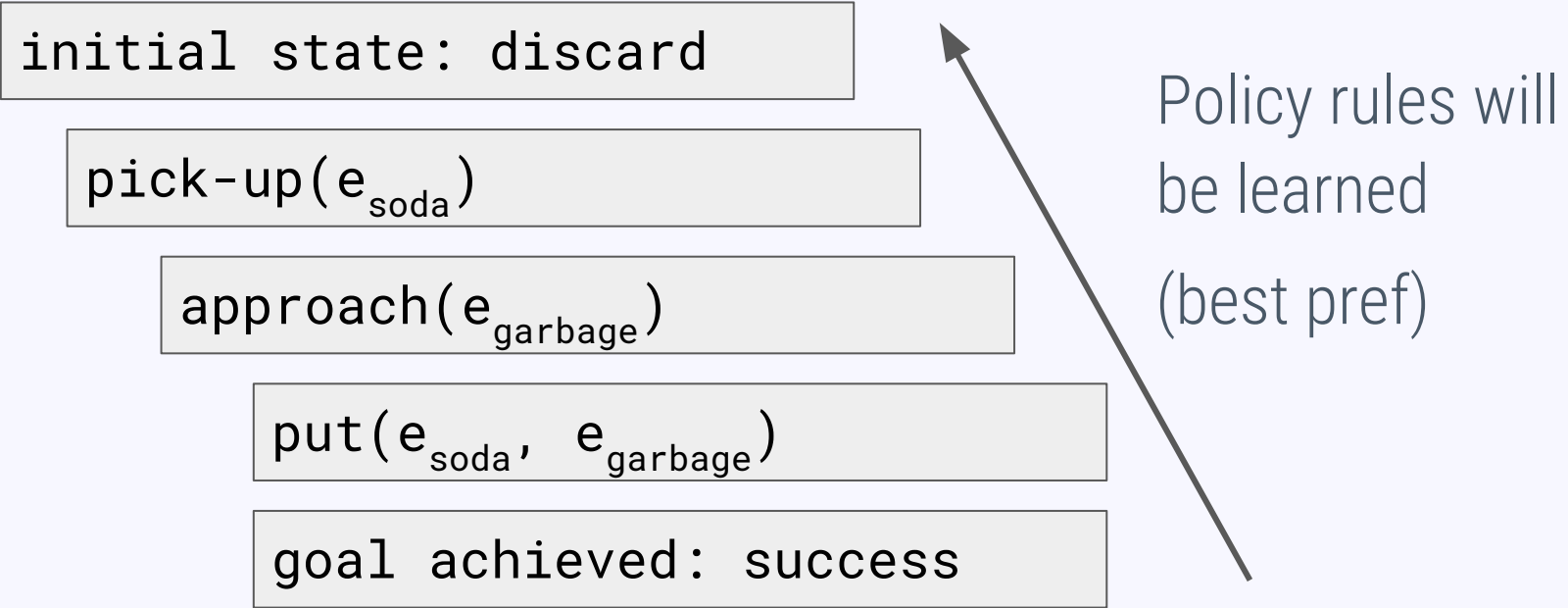
pick-up(e_{soda})

approach(e_{garbage})

put(e_{soda} , e_{garbage})

goal achieved: success

Policy rules will
be learned
(best pref)



Learning Procedural Form.

Retrospective Learning will fail

```
initial state: lead-tour
```

```
go-to( $e_{\text{main\_office}}$ )
```

```
say("This is ...")
```

```
go-to( $e_{\text{kitchen}}$ )
```

```
say("Here is where...")
```

```
no goal
```

Learning Procedural Form.

Change 1: Break the retrospective learning into individual episodes


Work backwards from the last subtask first

state3 of discard

`put(esoda, egarbage)`

goal achieved: success

learn best
preference rule
for put



Learning Procedural Form.

Change 1: Break the retrospective learning into individual episodes


Work backwards from the last subtask first

state2 of discard

approach(e_{garbage})

put(e_{soad} , e_{garbage})

goal achieved: success



learn best
preference rule
for approach

Learning Procedural Form.

Now we do the same thing with lead-tour

```
state4 lead-tour
```

```
say("Here is where...")
```

```
no goal
```

nothing is learned

Learning Procedural Form.

Change 2: Create a new kind of goal for performing an action

```
state4 lead-tour
```

```
say("Here is where...")
```

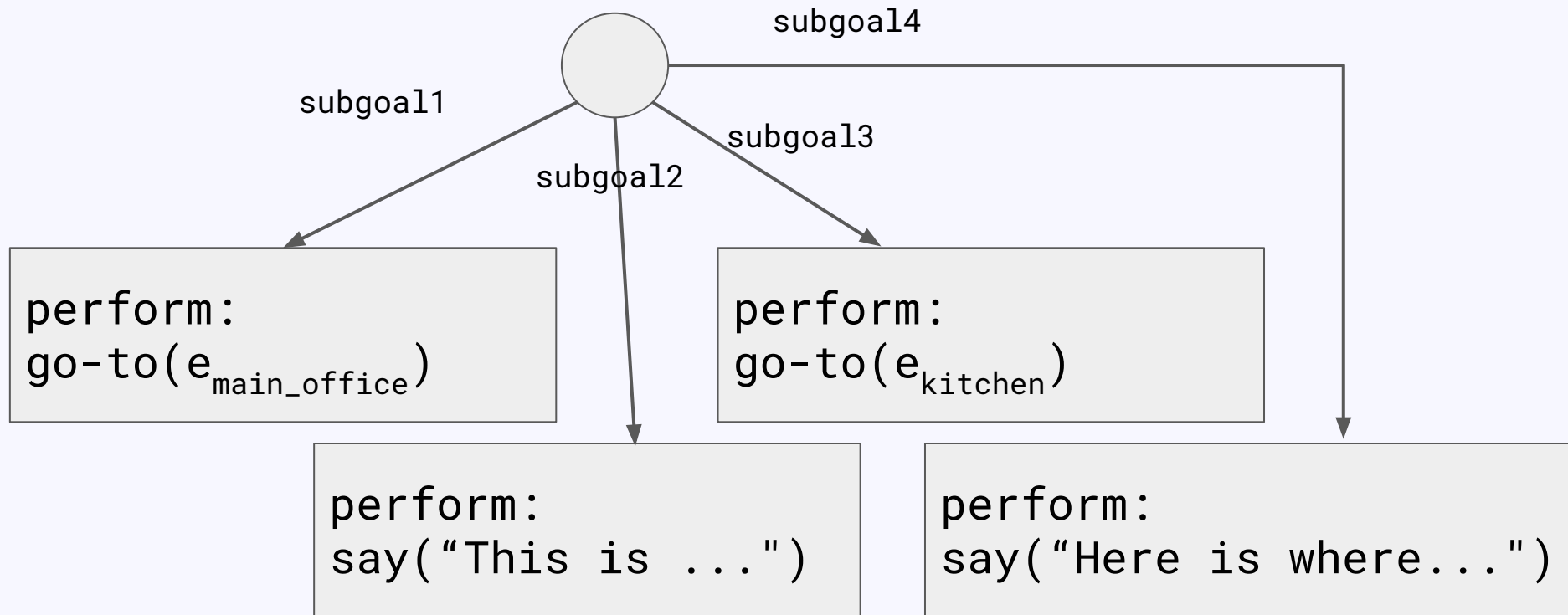
```
no goal
```

nothing is learned

push new goal of performing say

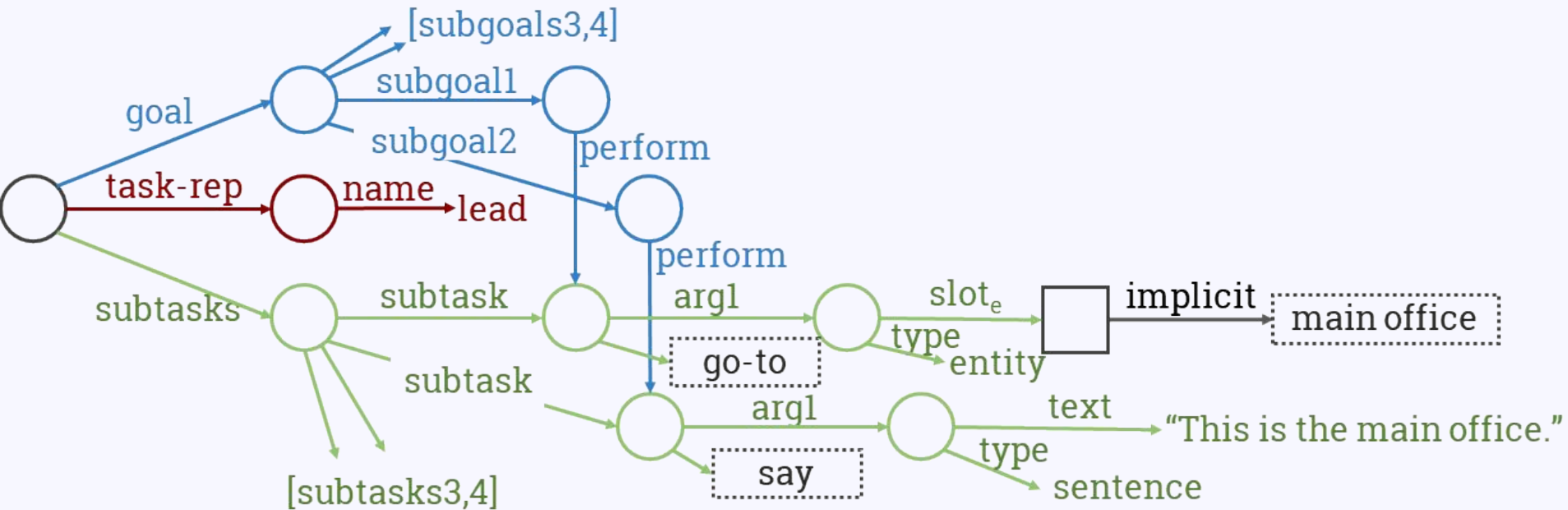
Learning Procedural Form.

Change 3: Allow an ordered sequence of subgoals



Learning Procedural Form.

Change 3: Allow an ordered sequence of subgoals



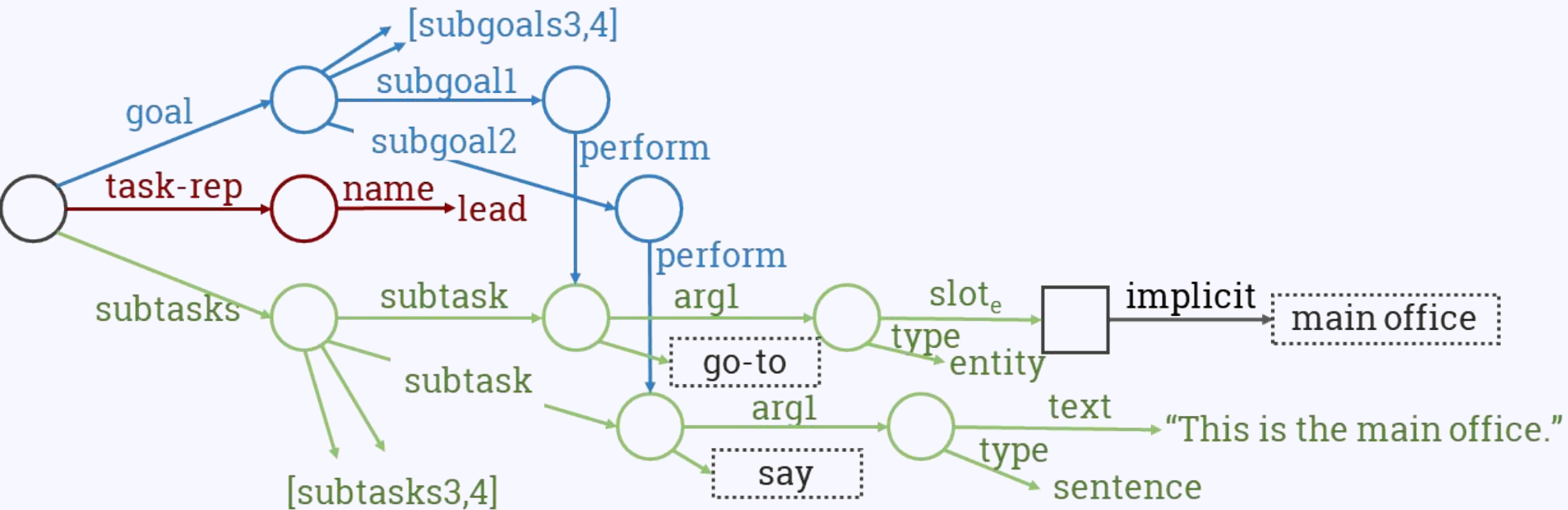
Learning Procedural Form.

Change 4: Keep track of current subgoal during task execution

- State has the id of the current subgoal
- When the current subgoal is satisfied:
 - Either retrieve the next one from smem
 - Or if it is the last, report task success

Learning Procedural Form.

Next time, will perform the same actions in the same order



Blended Formulation Example

Learning Blended Formulation

Teach the task of deliver,
Without the knowledge of what pickup does



Deliver the package to David.

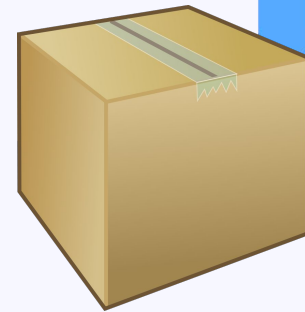
The goal is that David is holding the package.

What do I do next?

Pick up the package.

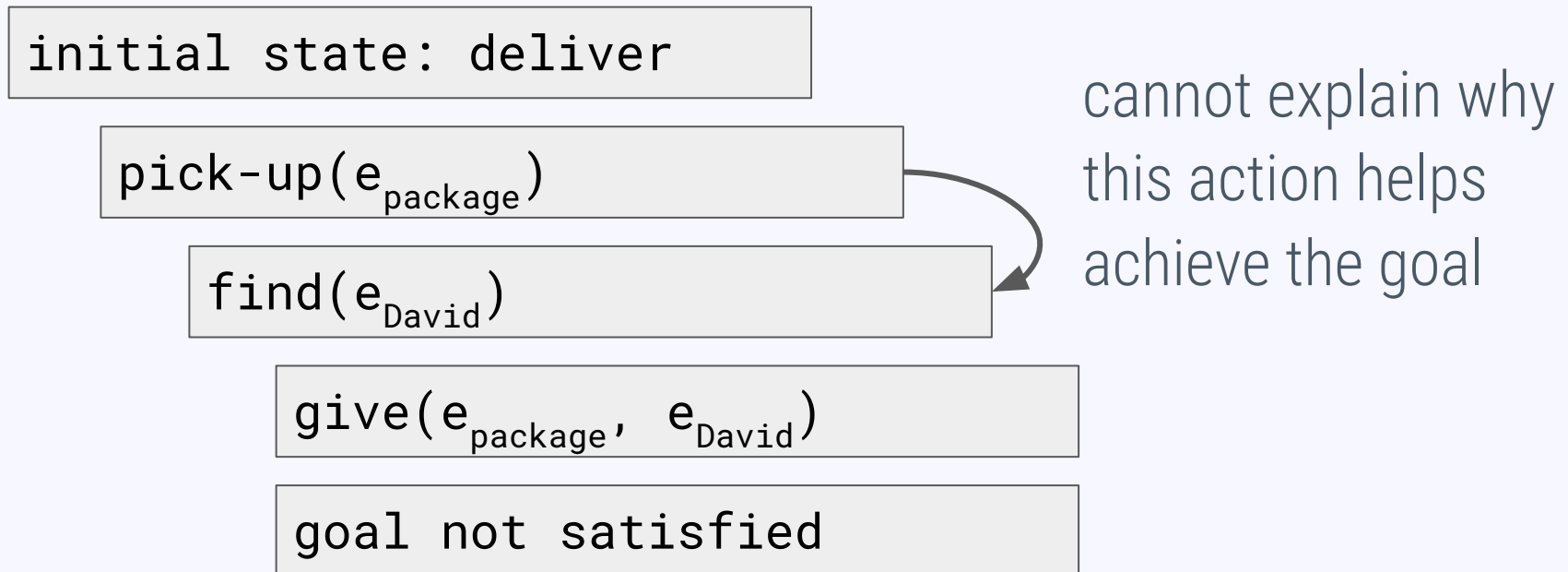
Rosie finds David

Rosie gives David the package



Learning Blended Formulation

Teach the task of deliver,
Without the knowledge of what pickup does



Learning Blended Formulation

Now, we can learn 2 policy rules

state3 deliver

give(e_{package} , e_{David})

goal satisfied

state2 deliver

find(e_{David})

give(e_{package} , e_{David})

goal satisfied

Learning Blended Formulation

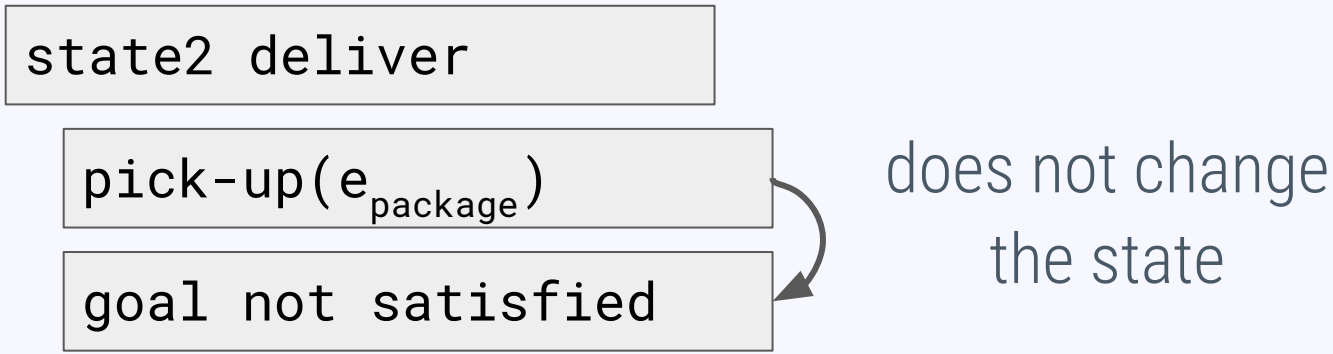
Learning rule for pickup still fails

state2 deliver

pick-up(e_{package})

goal not satisfied

does not change
the state



Learning Blended Formulation

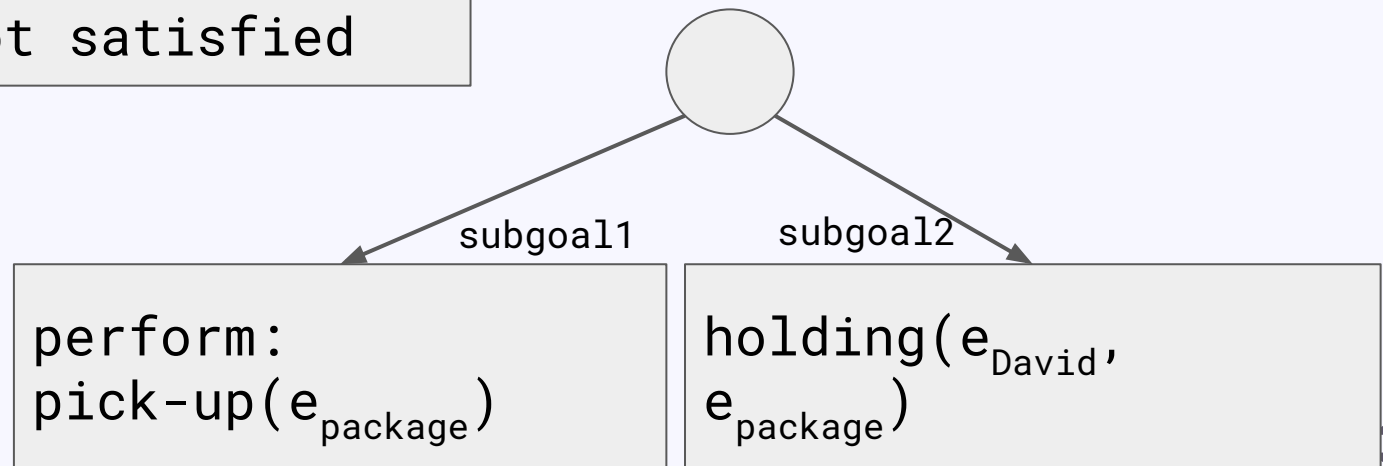
Learning rule for pickup still fails

So we push a new procedural subgoal

state2 deliver

pick-up(e_{package})

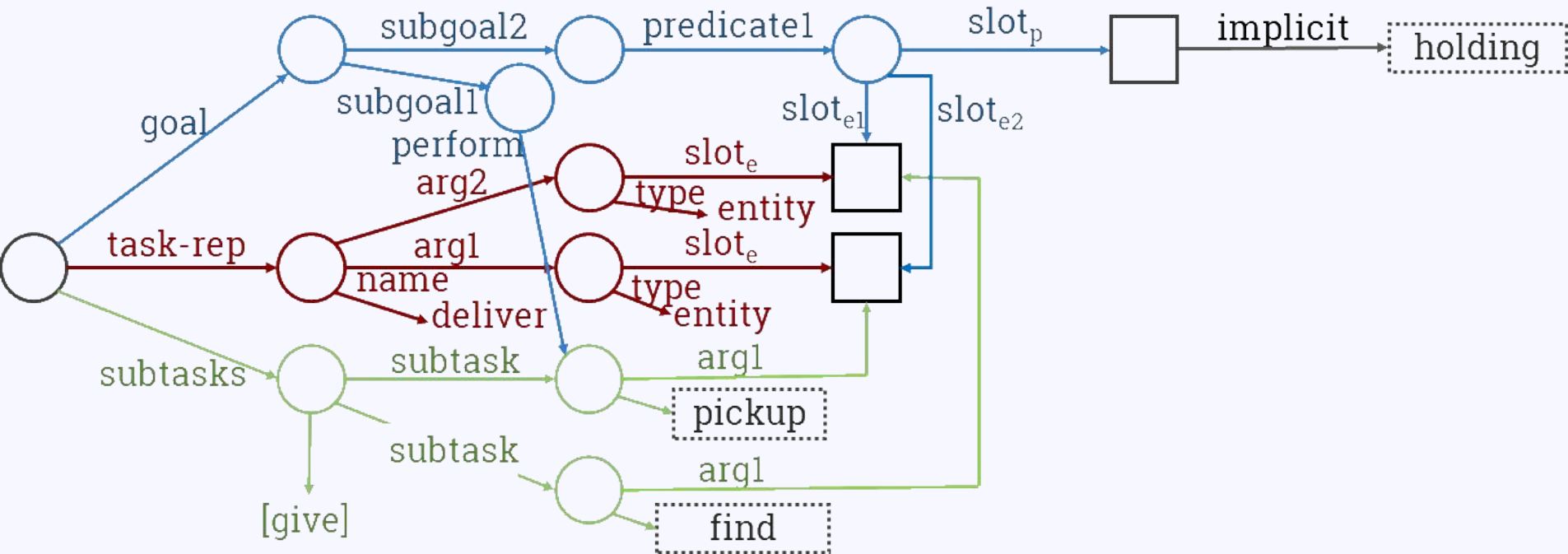
goal not satisfied



Learning Blended Formulation

Learning rule for pickup still fails

So we push a new procedural subgoal



Learning Blended Formulation



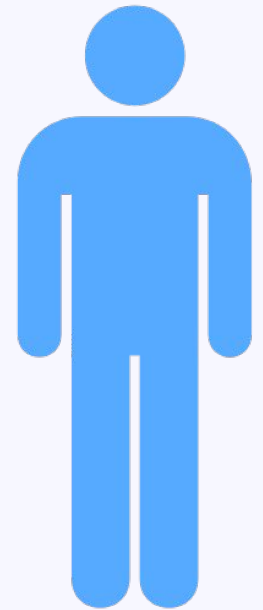
Deliver a water to Emily.

Rosie finds the water

Rosie picks up the water

Rosie finds Emily

Rosie gives Emily the water



Nuggets and Coal

Nuggets

- Improves the space of tasks Rosie can learn
- Able to learn when full explanation fails
- Integrated approach allows interesting blending of formulations
- Gives more options to the instructor

Coal

- Limited to singular actions
- No complex control flow
- Limited ability to learn task variations
- Will not change goals as more knowledge is gained

Questions?