Integrated Planning and Reinforcement Learning for Compositional Domains

—

Harsha Kokel
Research Scientist
IBM Research
Integrated Planning and Reinforcement Learning for Compositional Domains

Harsha Kokel
Research Scientist
IBM Research

Prof. Sriraam Natarajan (left)
Reinforcement Learning

Relational MDP
Planning

A: Pre: P
    Add: Q
    Del: P

B: Pre: P, Q, R
    Add: T
    Del: Q, R

C: Pre: P, Q
    Add: U, V
    Del: P, Q

D: Pre: P, T
    Add: U
    Del: P

E: Pre: S, T
    Add: V, W
    Del: S, T

Initial state

P, Q, R

Q, R, S

A

S, T

T, U

R, U, V

B

C

D

E

A state satisfying goals

X

Long and Fox (2002)
Planning

- Search through space of states
- Relies on an explicit model of the environment

RL

- Search through space of policies
- Relies on trial & error by interaction
Köhler (1948)
Integrating Planning and RL

- Knowledge
- Planner
- Environment
- Initial state & task
- State, Reward
- High-level plan
- Action
Integrating Planning and RL

knowledge

Planner

Environment

initial state & goal

State, Reward

high-level plan

action
Decomposing GRMDP

Goal-directed RMDP:

\[ \langle S, A, P, R, \gamma, G \rangle \]

Multiple subgoal RMDP:

\[ \langle S, A, P_1, R_1, \gamma \rangle \]
\[ \langle S, A, P_2, R_2, \gamma \rangle \]
\[ \ldots \]
\[ \langle S, A, P_n, R_n, \gamma \rangle \]

\[
R_o(s, a, s') = \begin{cases} 
    t_R + R(s, a, s') & \text{if } s' \in \beta(o) \text{ and } s \notin \beta(o) \\
    0 & \text{if } s' \in \beta(o) \text{ and } s \in \beta(o) \\
    R(s, a, s') & \text{otherwise}
\end{cases}
\]

\[
P_o(s, a, s') = \begin{cases} 
    0 & \text{if } s \in \beta(o) \text{ and } s' \notin \beta(o) \\
    1 & \text{if } s \in \beta(o) \text{ and } s' \in \beta(o) \\
    P(s, a, s') & \text{otherwise}
\end{cases}
\]
Irrelevant variables

Factored MDP represented as Dynamic Bayesian Network (DBN)
A **model-agnostic abstraction** $\phi(s)$ is such that for any action $a$ and an abstract state $\bar{s}$, $\phi(s_1)=\phi(s_2)$ if and only if

$$
\sum_{\{s'_1 | \phi(s'_1) = \bar{s}\}} R_0(s_1, a, s'_1) = \sum_{\{s'_2 | \phi(s'_2) = \bar{s}\}} R_0(s_2, a, s'_2)
$$

$$
\sum_{\{s'_1 | \phi(s'_1) = \bar{s}\}} P_0(s_1, a, s'_1) = \sum_{\{s'_2 | \phi(s'_2) = \bar{s}\}} P_0(s_2, a, s'_2)
$$

Dietterich NeurIPS 2000; Ravindran and Barto IJCAI 2003; Givan, Dean, and Greig AI 2003; Li, Walsh, and Littman ISAIM 2006
Graphical models

Bayesian Network (BN)

Dynamic BN

Dynamic PLM

Probabilistic Logic Models (PLM, PRM, BLP, LBN, DAPER)

Koller and Friedman 2009; Getoor and Tasker 2007; Raedt et al. 2016

OSU: AI Seminar
First Order Conditional Influence (FOCI) statements

if \langle condition \rangle \text{ then } \langle influences \rangle \text{ QINF } \langle resultant \rangle

Dynamic FOCI statements

\langle subgoal \rangle: \langle influences \rangle \xrightarrow{+1} \langle resultant \rangle

Natarajan, Tadepalli, Dietterich, and Fern 2008
D-FOCI

\{\text{action, taxi}_\text{at}(X)\} \xrightarrow{+1} \text{taxi}_\text{at}(X) \quad (3a)

\text{pick}(P) : \{\text{action, taxi}_\text{at}(X), \text{at}(P, Y), \}

\quad \text{in}_\text{taxi}(P) \xrightarrow{+1} \text{in}_\text{taxi}(P) \quad (3b)

\text{pick}(P) : \{\text{in}_\text{taxi}(P)\} \rightarrow \text{Reward} \quad (3c)

\text{drop}(P) : \{\text{at}_\text{dest}(P)\} \rightarrow \text{Reward} \quad (3d)

\text{drop}(P) : \{\text{at}(P, X), \text{dest}(P, D), \text{at}_\text{dest}(P)\}

\quad \rightarrow \text{at}_\text{dest}(P) \quad (3e)

\text{drop}(P) : \{\text{action, taxi}_\text{at}(X), \text{at}(P, Y), \}

\quad \text{in}_\text{taxi}(P) \xrightarrow{+1} \text{at}(P, K) \quad (3f)
Experiments

Domains
- Office World
- Minecraft World
- Relational Taxi
- Relational Box World
- Craft World
- Robotic Fetch domain

Baselines
- HRL (options framework)
- Tabular Q-learning
- Deep RL (DDQN, HDQN, SAC)
- Deep Relational RL
- Planning+RL (Taskable RL)
Experiments

Sample efficiency
Transfer across task

Office World

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>▲</td>
<td>Agent</td>
</tr>
<tr>
<td>⋆</td>
<td>Furniture</td>
</tr>
<tr>
<td>🔧</td>
<td>Coffee machine</td>
</tr>
<tr>
<td>📧</td>
<td>Mail room</td>
</tr>
<tr>
<td>🏡</td>
<td>Office</td>
</tr>
<tr>
<td>A, B, C, D</td>
<td>Marked locations</td>
</tr>
</tbody>
</table>

Kokel et al. ICAPS 2021
Deep Relational RL

![Image of a robot with multiple colored cubes on a block.

Bar chart showing success rate vs number of blocks.

Line chart showing average rewards vs steps in environment.

References:
1. Kokel et al. NCAA 2022a
2. Li et al. ICRA 2022
AI Planning + RL = Integrated Methods

AI Planning:
- Domain independent heuristics
- Domain knowledge
- Symbolic

RL:
- Domain dependent reward
- Simulator and Data
- Neural

Integrated Methods:
- Scalability
- Sample efficiency
- Neuro-Symbolic
Bridging the Gap
Planning & RL
Planning

- Search through space of states
- Relies on an explicit model of the environment
- PDDL Task

RL

- Search through space of policies
- Relies on trial & error by interaction
- MDP
PDDL Task
\( \langle \mathcal{L}, \mathcal{O}, I, G \rangle \)

Lifted Action Models \( \mathcal{O} \)

(:action move
:parameters (?curpos ?nextpos ?dir)
:precondition (and (place ?curpos)
  (place ?nextpos) (at-robot ?curpos)
:effect (and (at-robot ?nextpos)
  (not (at-robot ?curpos))))

move(c_1_1, c_1_2, east),
move(c_2_2, c_2_1, west), ...

MDP
\( \langle S, A, T, R, \gamma \rangle \)

Actions
[east, west, north, south]
"Two actions that are applicable in the same state cannot have the same label"
(\textbf{action} pickup

\textbf{parameters} (?k - key ?r - room)

\textbf{precondition} (and (at ?k ?r)
  (at-agent ?r)
  (empty-hand))

\textbf{effect} (and (not (at ?k ?r))
  (not (empty-hand))
  (carry ?k))

# of grounding = #of keys * # of rooms

Are all the parameters of LAM relevant?*

*Do they define different grounded actions that can be applied in a single state?
Relevant parameters

Know,
(at key1 room1) \( \oplus \) (at key1 room2)

So,
(pickup key1 room1) \( \oplus \)
(pickup key1 room2)
\( \oplus \): Mutually exclusive

(:action pickup
:parameters (?k - key ?r - room)
:precondition (and (at ?k ?r)
(at-agent ?r)
(empty-hand))
:effect (and (not (at ?k ?r))
(not (empty-hand))
(carry ?k))
)
Applicable Action Mutex Group (AAMG)

\[(\text{pickup key1 room1}), (\text{pickup key1 room2}), (\text{pickup key1 room3}), \ldots\]  
\[(\text{pickup key2 room1}), (\text{pickup key2 room2}), (\text{pickup key2 room3}), \ldots\]

\[(\text{pickup key1 })\]  
\[(\text{pickup key2 })\]

\[(\text{pickup ?k - key ?r - room })\]  
\[(\text{pickup ?k – key })\]

Parameter Seed Set of pickup  
\{?k – key \}
Action Space Reduction

Figure 2: Comparison of label set sizes on (a) 14 IPC STRIPS domains and (b) 7 HTG domains.
Impact on learning RL policies

Figure 3: Learning curve in the (a) ferry, (b) gripper, (c) blocks, and (d) logistics; with and without action label reduction.
Integrating Planning and RL

- How to represent this knowledge?
- Can this knowledge be refined?
- How to obtain this knowledge?
- How to represent the goal?
- How to customize plan for agent’s skill?
- When/how to replan?
Reference

Harsha Kokel, Arjun Manoharan, Sriraam Natarajan, Balaraman Ravindran, Prasad Tadepalli, RePReL: Integrating Relational Planning and Reinforcement Learning for Effective Abstraction, In ICAPS 2021a.


Harsha Kokel, Sriraam Natarajan, Balaraman Ravindran, Prasad Tadepalli, RePReL: A Unified Framework for Integrating Relational Planning and Reinforcement Learning for Effective Abstraction in Discrete and Continuous Domains, In NCAA 2022a.

Harsha Kokel, Nikhilesh Prabhakar, Sriraam Natarajan, Balaraman Ravindran, Prasad Tadepalli, Hybrid Deep RePReL: Integrating Relational Planning and Reinforcement Learning for Information Fusion, In FUSION 2022b.

Harsha Kokel, Mayukh Das, Rakibul Islam, Julia Bonn, Jon Cai, Soham Dan, Anjali Narayan-Chen, Prashant Jayannavar, Janardhan Rao Doppa, Julia Hockenmaier, Sriraam Natarajan, Martha Palmer, Dan Roth, Lara -- Human-guided collaborative problem solver: Effective integration of learning, reasoning and communication, In ACS 2022c.

Harsha Kokel, Junkyu Lee, Michael Katz, Kavitha Srinivas, Shirin Sohrabi, Action Space Reduction for Planning Domains, IJCAI 2023
Questions?
Backup Slides
D-FOCI

First Order Conditional Influence (FOCI) statements

\[
\text{if } \langle \text{condition} \rangle \quad \text{then } \langle \text{influents} \rangle \quad \text{QINF} \quad \langle \text{resultant} \rangle
\]

Dynamic FOCI statements

\[
[\langle \text{subgoal} \rangle]: \langle \text{influents} \rangle \xrightarrow{[+1]} \langle \text{resultant} \rangle
\]

Natarajan, Tadepalli, Dietterich, and Fern 2008
Definition 4 (Li, Walsh, and Littman (2006)). A model-agnostic abstraction $\phi(s)$ is such that for any action $a$ and abstract state $\bar{s}$, $\phi(s_1) = \phi(s_2)$ if and only if

$$
\sum_{\{s'_1 | \phi(s'_1) = \bar{s}\}} R_o(s_1, a, s'_1) = \sum_{\{s'_2 | \phi(s'_2) = \bar{s}\}} R_o(s_2, a, s'_2)
$$

$$
\sum_{\{s'_1 | \phi(s'_1) = \bar{s}\}} P_o(s_1, a, s'_1) = \sum_{\{s'_2 | \phi(s'_2) = \bar{s}\}} P_o(s_2, a, s'_2)
$$
Collaborative Problem Solving

Target structure

Build a red tower.

Of what size?

4

Minecraft

Kokel et al. ACS 2022c
Capabilities

Bidirectionally contentful
- ask for missing dimensions
- ask for reference points

Context-aware
- Understand the next instruction in context of current structure

Composable Vocabulary
- can plan for arbitrary combinations of primitive shape
- learn a new shape from single example
- use learnt shape in various combinations

Habitability
- actionable errors/questions

Robustness
- Undo which doesn’t lose context
Integrating Relational Planning and Reinforcement Learning

Plan the sequence of high level subgoals and learn to execute each subgoal at lower level

Advantage:
- Compositionality
- Task specific state representations

Dynamic First Order Conditional Influence (D-FOCI) statements to obtain task-specific abstract representations

Kokel et al. ICAPS 2021
Definition 3. The subgoal RMDP \( M_o \) for each operator \( o \) is defined by the tuple \( \langle S, A, P_o, R_o, \gamma \rangle \) consisting of states \( S \), actions \( A \), transition function \( P_o \), reward function \( R_o \), and discount factor \( \gamma \). State and Actions remain same as the original RMDP. The reward function \( R_o \) and transition probability distribution function \( P_o \) are defined as follows:

\[
R_o(s, a, s') = \begin{cases} 
  t_R + R(s, a, s') & \text{if } s' \in \beta(o) \text{ and } s \notin \beta(o) \\
  0 & \text{if } s' \in \beta(o) \text{ and } s \in \beta(o) \\
  R(s, a, s') & \text{otherwise}
\end{cases}
\]

\[
P_o(s, a, s') = \begin{cases} 
  0 & \text{if } s \in \beta(o) \text{ and } s' \notin \beta(o) \\
  1 & \text{if } s \in \beta(o) \text{ and } s' \in \beta(o) \\
  P(s, a, s') & \text{otherwise}
\end{cases}
\]

with \( R(s, a, s') \) indicating the reward function from the original GRMDP definition. \( t_R \) is a fixed terminal reward.
Abstraction

Given:
- D-FOCI statements from Equation 3
- state \( s = \{ \text{at}(p1,r), \text{taxi}_{\text{at}}(l3), \text{dest}(p1,d1), \neg \text{at}_{\text{dest}}(p1), \neg \text{in}_{\text{taxi}}(p1), \text{at}(p2,b), \neg \text{at}_{\text{dest}}(p2), \neg \text{in}_{\text{taxi}}(p2) \} \)
- grounded optiono\(\theta\): \(\text{pick}(P)\) \(\{P/p1\}\)

Output: A set of relevant state literals: \(\hat{s}\)

Depth 1 unrolling:
1. Find a substitution that grounds relevant D-FOCI statements that have a reward on RHS
   \[
   \begin{align*}
   \text{pick}(p1): & \quad \text{in}_{\text{taxi}}(p1) \rightarrow \text{Reward} \\
   \theta = & \{P/p1\}
   \end{align*}
   \]
2. Collect LHS in relevant literals set \(\hat{s}\)
   \[
   \hat{s} = \{\text{in}_{\text{taxi}}(p1)\}
   \]

Depth 2 unrolling:
1. Find a substitution that grounds relevant D-FOCI statements that have a relevant literal on RHS
   \[
   \begin{align*}
   \text{pick}(P): & \quad \{ \text{action}, \text{taxi}_{\text{at}}(l3), \text{at}(p1, r), \text{in}_{\text{taxi}}(p1) \} \rightarrow \text{in}_{\text{taxi}}(p1) \\
   \theta = & \{P/p1, X/l3, Y/r\}
   \end{align*}
   \]
2. Collect LHS in set \(\hat{s}\)
   \[
   \hat{s} = \{\text{in}_{\text{taxi}}(p1), \text{action}, \text{taxi}_{\text{at}}(l3), \text{at}(p1, r)\}
   \]

Depth 3 unrolling:
1. Ground applicable D-FOCI statements that have a relevant literal (\(\hat{s}\)) on RHS
   \[
   \begin{align*}
   \{\text{action}, \text{taxi}_{\text{at}}(l3)\} & \rightarrow^{+1} \text{taxi}_{\text{at}}(l3) \\
   \text{pick}(p1): & \quad \{ \text{action}, \text{taxi}_{\text{at}}(l3), \text{at}(p1, r), \text{in}_{\text{taxi}}(p1) \} \rightarrow \text{in}_{\text{taxi}}(p1) \\
   \theta = & \{P/p1, X/l3, Y/r\}
   \end{align*}
   \]
2. Collect LHS in set \(\hat{s}\)
   \[
   \hat{s} = \{\text{in}_{\text{taxi}}(p1), \text{action}, \text{taxi}_{\text{at}}(l3), \text{at}(p1, r)\}
   \]

recursive grounding and unrolling process
Lifted Successor Generation

Treats planning state as database and task of generating applicable action groundings as a database query.

\begin{align*}
\text{(at obj1 l1)} & \quad \text{(precondition} \quad \text{and (at ?X ?Y)} \\
\text{(at obj2 l1)} & \quad \text{(path ?Y ?W)} \\
\text{(at obj3 l3)} & \quad \text{)} \\
\text{(at obj4 l2)} & \quad \text{(path ?W ?Z))} \\
\text{(path l1 l2)} & \quad \text{at} \\
\text{(path l1 l3)} & \quad \text{obj1 l1} \\
\text{(path l3 l4)} & \quad \text{l1 l2} \\
\text{(path l2 l3)} & \quad \text{l1 l3} \\
\text{(path l2 l3)} & \quad \text{obj3 l3} \\
\text{(path l3 l4)} & \quad \text{l2 l3} \\
\text{(path l2 l3)} & \quad \text{obj4 l2} \\
\text{(path l3 l4)} & \quad \text{l3 l4} \\
\end{align*}

$\text{at}(X, Y) \bowtie \text{path}(Y, W) \bowtie \text{path}(W, Z)$

Augusto Corrêa et al. 2020
If hypergraph is cyclic, the query evaluation is exponential in the size of input and output.

With parameter seed set we can improve cyclic query evaluation time.

If query hypergraph has a join-tree, it is acyclic. Acyclic query evaluation is output-polynomial.
Query Evaluation w/ Seed Set

\[ Q(X, Y, W, Z) = \text{at}(X, Y) \bowtie \text{path}(Y, W) \bowtie \text{path}(W, Z) \bowtie \text{path}(Y, Z) \bowtie \text{path}(Z, X) \]

Treat the non-seed parameters as 
\textit{non-distinguishable} variable

\[ Q(X, W, Z) = \text{at}(X, Y) \bowtie \text{path}(Y, W) \bowtie \text{path}(W, Z) \bowtie \text{path}(Y, Z) \bowtie \text{path}(Z, X) \]

Modify the join order

\[ Q(X, Y, W, Z) = \text{path}(Y, Z) \bowtie \text{at}(X, Y) \bowtie \text{path}(Y, W) \bowtie \text{path}(W, Z) \bowtie \text{path}(Z, X) \]
Experiments

Sample efficiency
Transfer across task
Generalization across objects

Kokel et al. ICAPS 2021
Deep Relational RL

Kokel et al. NCAA 2022a
Extension to hybrid domain

Allow hybrid of structured and unstructured data
Neural predicates in D-FOCI

Kokel et al. FUSION 2022b
Extension to deep, relational RL

Allowed continuous state and action space

Batch learning allowed any off-policy RL agent (inc. deep relational RL)

Removed fixed depth unrolling limitation

Kokel et al. NCAA 2022a
Markov Decision Process

Semi-MDP

Hierarchical RL or Options

Planning with RL

\[ O = \langle I, \beta, \pi \rangle \]
Reinforcement Learning

Compositional and Relational Domains
Compositional and Relational Domains
Compositional and Relational Domains
Sequential decision making

AI Planning + RL = Integrated Methods

- Domain independent heuristics
- Domain knowledge
- Symbolic
- Domain dependent reward
- Simulator and Data
- Neural
- Scalability
- Sample efficiency
- Neuro-Symbolic
LMG to identify Seed Set

\[ \ell = \langle \{?k - \text{object}\}, \{?r - \text{location}\}, \\
\{\text{at } ?k - \text{object } ?r - \text{location}\} \rangle \]

Conditions

1. atom of LMG is part of precondition
2. variable types in LMG is super-type of variable type of action parameter

Then removing the counted variable of LMG from action parameter defines an AOMG.

So, counted variable is not a seed set.

Leverage multiple LMGs in sequence to further reduce the parameter set.

(:action pickup
 :parameters (?k - key ?r - room)
 :precondition (and (at ?k ?r)
   (at-agent ?r)
   (empty-hand))
 :effect (and (not (at ?k ?r))
   (not (empty-hand))
   (carry ?k))
)
Lifted Mutex Group (LMG)

- **Mutex Group**
  A set of facts, of which only one fact is true in any reachable state
  
  Example:
  
  \[
  \{ (at \text{key1 } \text{room1}), (at \text{key1 } \text{room2}),
  (at \text{key1 } \text{room3}), (at \text{key1 } \text{room4}) \}
  \]

- **Lifted Mutex Group**
  An invariant candidate, whose ground atom sets are mutex groups
  
  Example:
  
  \[
  \ell = \langle \{?k – \text{key}\}, \{?r – \text{room}\}, \{(at ?k – \text{key} \ ?r – \text{room})\} \rangle
  \]

  \[
  \ell (?k/\text{key1}) = \{ (at \text{key1 } \text{room1}), (at \text{key1 } \text{room2}),
  (at \text{key1 } \text{room3}), (at \text{key1 } \text{room4}) \}
  \]

  \[
  \ell (?k/\text{key2}) = \{ (at \text{key2 } \text{room1}), (at \text{key2 } \text{room2}),
  (at \text{key2 } \text{room3}), (at \text{key2 } \text{room4}) \}
  \]

\[^1\text{Dan Fiser 2020}\]
Meltzoff, Waismeyer, & Gopnik (2012)
Hierarchical Planning

Planning for *Tasks* instead of goals

*Methods* to decompose tasks

*Operators* for executing action
Hierarchical Planning

Domain has predicates (Q), operators (O) and methods (M)

Methods have preconditions and subtasks

Operators have preconditions and effects

\[ \mathcal{D} = \langle Q, O, M \rangle \]

Planning problem

\[ m = \langle \text{task}(m), \text{pre}(m), \text{subtasks}(m) \rangle \]

\[ o = \langle \text{task}(o), \text{pre}(o), \text{eff}(o) \rangle \]

\[ \mathcal{P} = \langle \mathcal{D}, \text{initial state } s_0, \text{task list } t_o \rangle \]
Example

Abstract state for task1(3):

task1(X): solid(Y, G) $\rightarrow^{+1}$ color(G, green)
task1(X): color(X, green) $\rightarrow^{+1}$ Reward
Example

```
task1(X): dashed(G, Z) \rightarrow color(G, green)
task1(X): solid(Y, G) \rightarrow color(G, green)
task1(X): color(X, green) \rightarrow Reward
```

Abstract state for task1(3):