Knowledge Representation for Systems with General Intelligence

Michael Thielscher, IJCAI 2021, Montreal

Artificial General Intelligence

Intelligence is the ability to

- Perceive or infer information, and to
- Retain it as knowledge to be applied towards adaptive behaviours within an environment or context

Wikipedia

Artificial general intelligence enables systems to

- Perceive or infer information about new tasks
- Adapt to these tasks without human intervention

Systems with General Intelligence

A system with general intelligence can

- Take representations of new problems as input
- Solve these problems without human intervention

Disclaimer

This isn't just Newell and Simon's General Problem Solver:

- They only used symbols to represent problems
- They only used symbol manipulation to solve them

Systems with General Intelligence

A system with general intelligence can

- Take representations of new problems as input
- Solve these problems without human intervention

Claim

Systems with general intelligence require to integrate different AI methods

Outline

1. General game-playing programs
    - KR for describing new games
    - + Inference + Search + Deep Learning

2. General collaborative problem-solving robots
    - Hybrid robot control
    - + Epistemic planning for collaboration

3. Agent programming for interactive artworks
4. A new challenge – Escape Room Robot

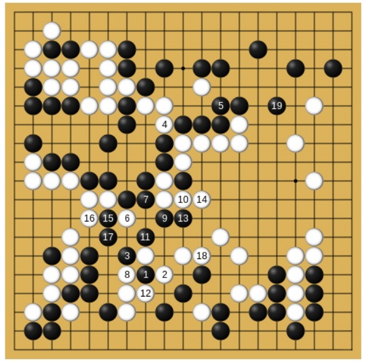
General Game Playing

IBM

Jim Gardner

S.M.S.I., Inc. - Owen Williams, The Kasparov Agency

AlphaGo



Sugitaro

Cyberoro ORO

A general game playing system can

- Understand descriptions of new games
- Learn to play these new games effectively without human intervention

Defining General Problem-Solving Tasks

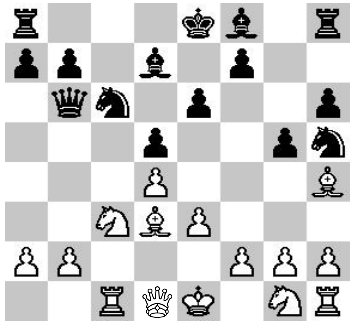
Define:

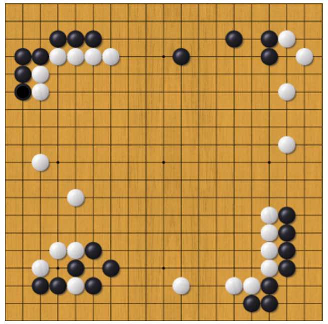
1. A problem class C
2. How to present problems p ∈ C to a system

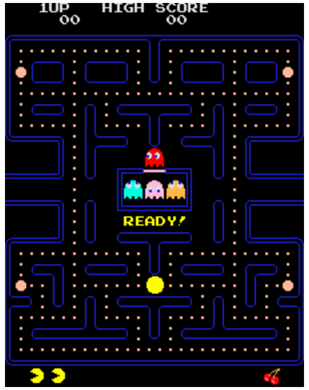
General Game Playing (2005, 1st AAAI Competition)

Finite, n-player games with

- Synchronous moves
- Complete state information



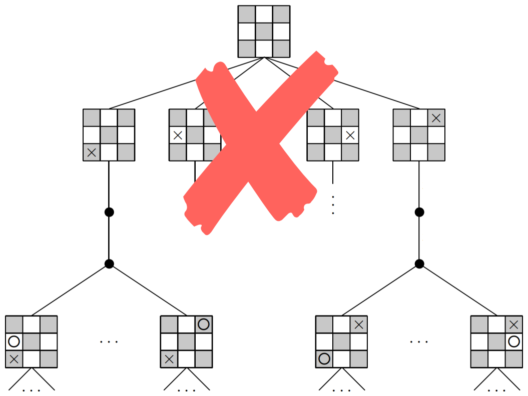


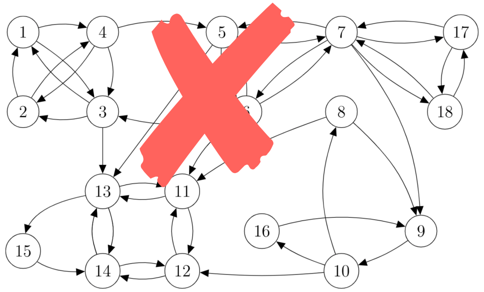


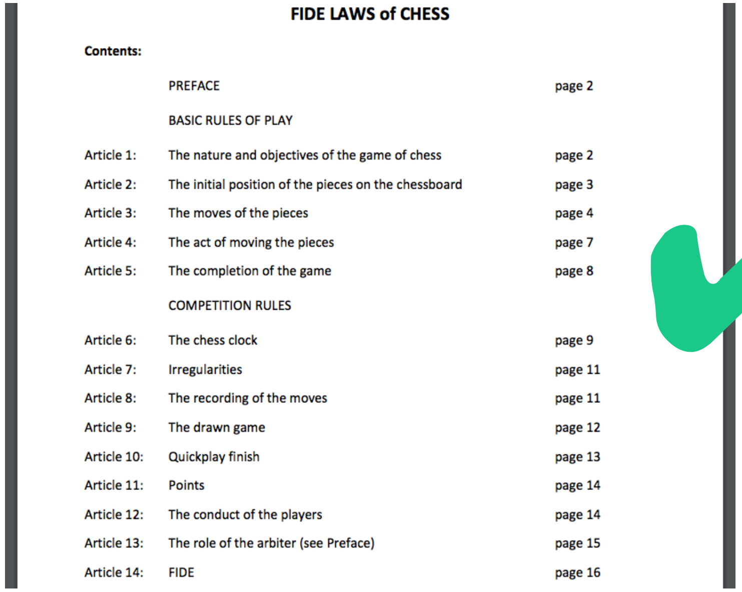
Defining General Problem-Solving Tasks

Define:

1. A problem class C
2. How to present problems p ∈ C to a system







FIDE LAWS of CHESS

Contents:

Outline

1. General game-playing programs
   - KR for describing new games
   - + Inference + Search + Deep Learning

2. General collaborative problem-solving robots
   - Hybrid robot control
   - + Epistemic planning for collaboration

3. Agent programming for interactive artworks
4. A new challenge – Escape Room Robot

Knowledge Representation Formalisms for Dynamic Domains

J. McCarthy's Situation Calculus (1968)

STRIPS (1972)

↪ PDDL (1998)

Game Description Language GDL (2005)

Video Game Description Language (2014)

Ludii (2016)

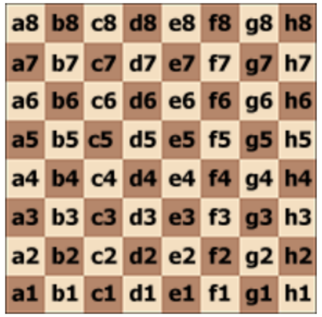
Game Descriptions

```
(init (cell a 1 white-rook))
  …
(init (cell h 8 black-rook))

(<= (legal white (castle ?direction))
    (true (control white))
    (true (cell e 1 white-king))
    … )

(<= (next (cell c 1 white-king))
    (does white (castle c-side)))

(<= (goal (?player 100))
    (checkmate)
    (not (true (control ?player))))
```



UNSW SYDNEY

Outline

1. General game-playing programs
   - KR for describing new games
   - + Inference + Search + Deep Learning

2. General collaborative problem-solving robots
   - Hybrid robot control
   - + Epistemic planning for collaboration

3. Agent programming for interactive artworks
4. A new challenge – Escape Room Robot

UNSW SYDNEY

Inferring New Knowledge

```
(init (cell a 1 white-rook))
 …
(<= (legal (white castle ?direction))
      … )
 …
(<= (goal (?player 100))
    (checkmate)
    (not (true (control ?player))))
```

Answer Set Program

Solver for single-player games

M.T., ICLP'09

Inferring New Knowledge

```
(init (cell a 1 white-rook))
 …
(<= (legal (white castle ?direction))
      … )
 …
(<= (goal (?player 100))
    (checkmate)
    (not (true (control ?player))))
```

Answer Set Program

Solver for single-player games

Stochastic Constraint Program

General search-based player

Koriche, Lagrue, Piette, & Tabary, *Constraints* 2016

Inferring New Knowledge

```
(init (cell a 1 white-rook))
 …
(<= (legal (white castle ?direction))
      … )
 …
(<= (goal (?player 100))
    (checkmate)
    (not (true (control ?player))))
```

Answer Set Program

Solver for single-player games

Stochastic Constraint Program

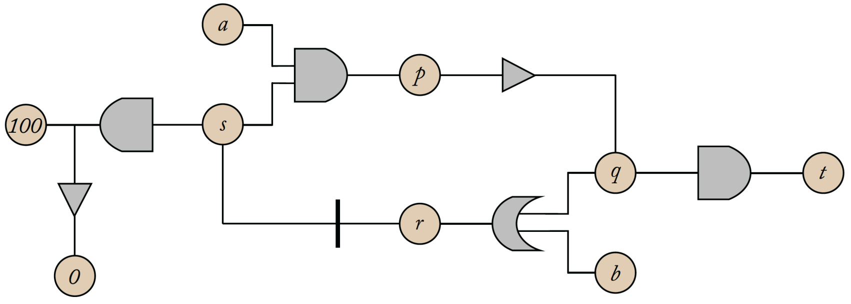
General search-based player

Quantified Boolean Formulas

Solver for two-player games

Learning to Play New Games

```
(init (cell a 1 white-rook))
 …
(<= (legal (white castle ?direction))
      … )
 …
(<= (goal (?player 100))
    (checkmate)
    (not (true (control ?player))))
```



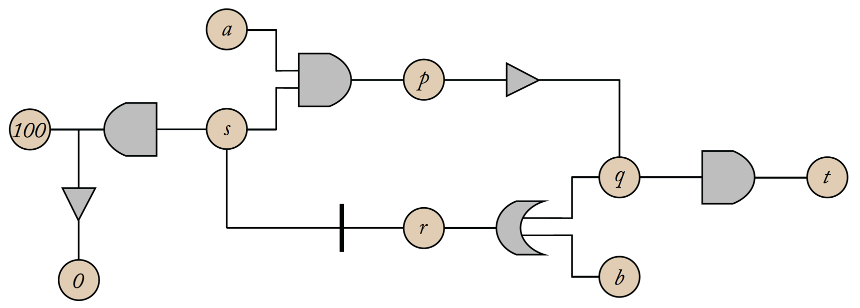
Propositional Network

Schkufza, Love, & Genesereth, AI'08

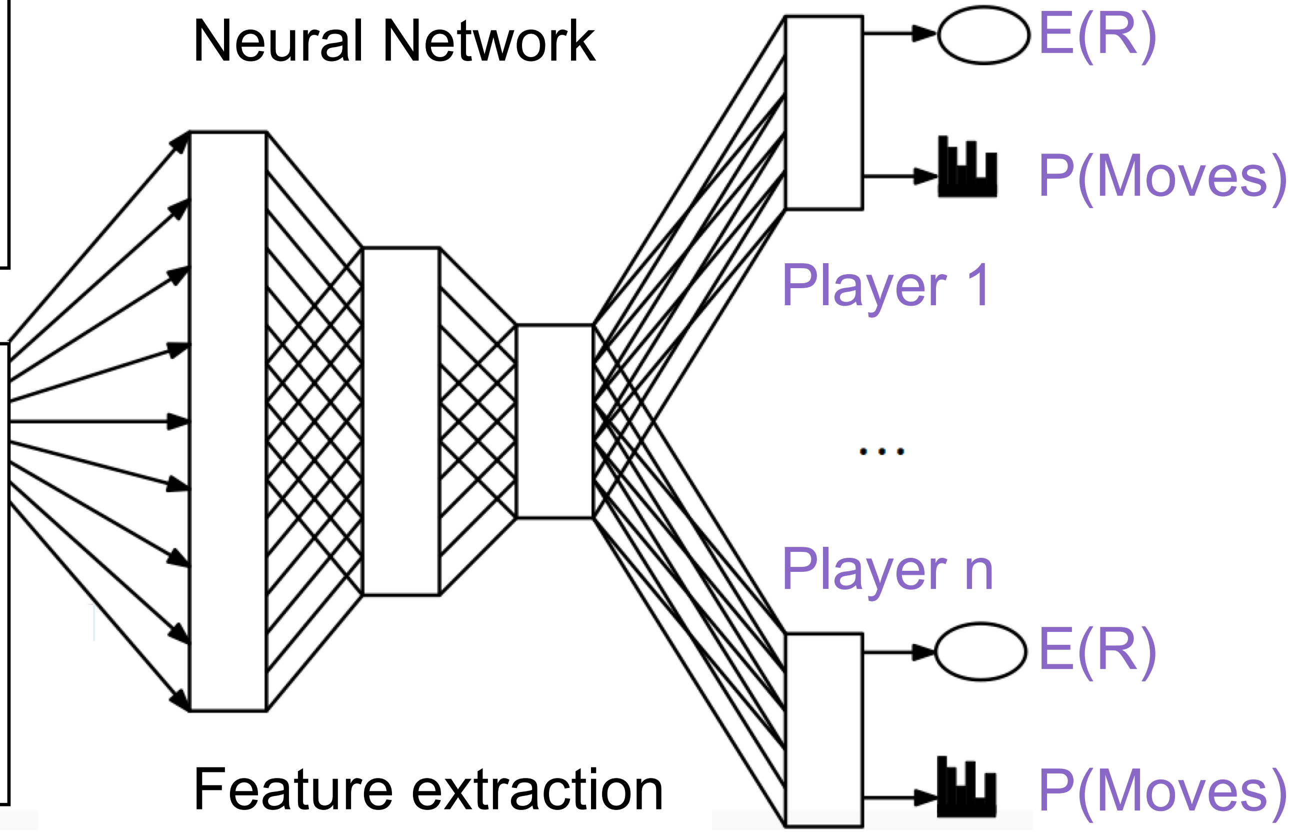
Learning to Play New Games

Goldwaser & M.T., AAAI'20

```
(init (cell a 1 white-rook))
 …
(<= (legal (white castle ?direction))
      … )
 …
(<= (goal (?player 100))
    (checkmate)
    (not (true (control ?player))))
```



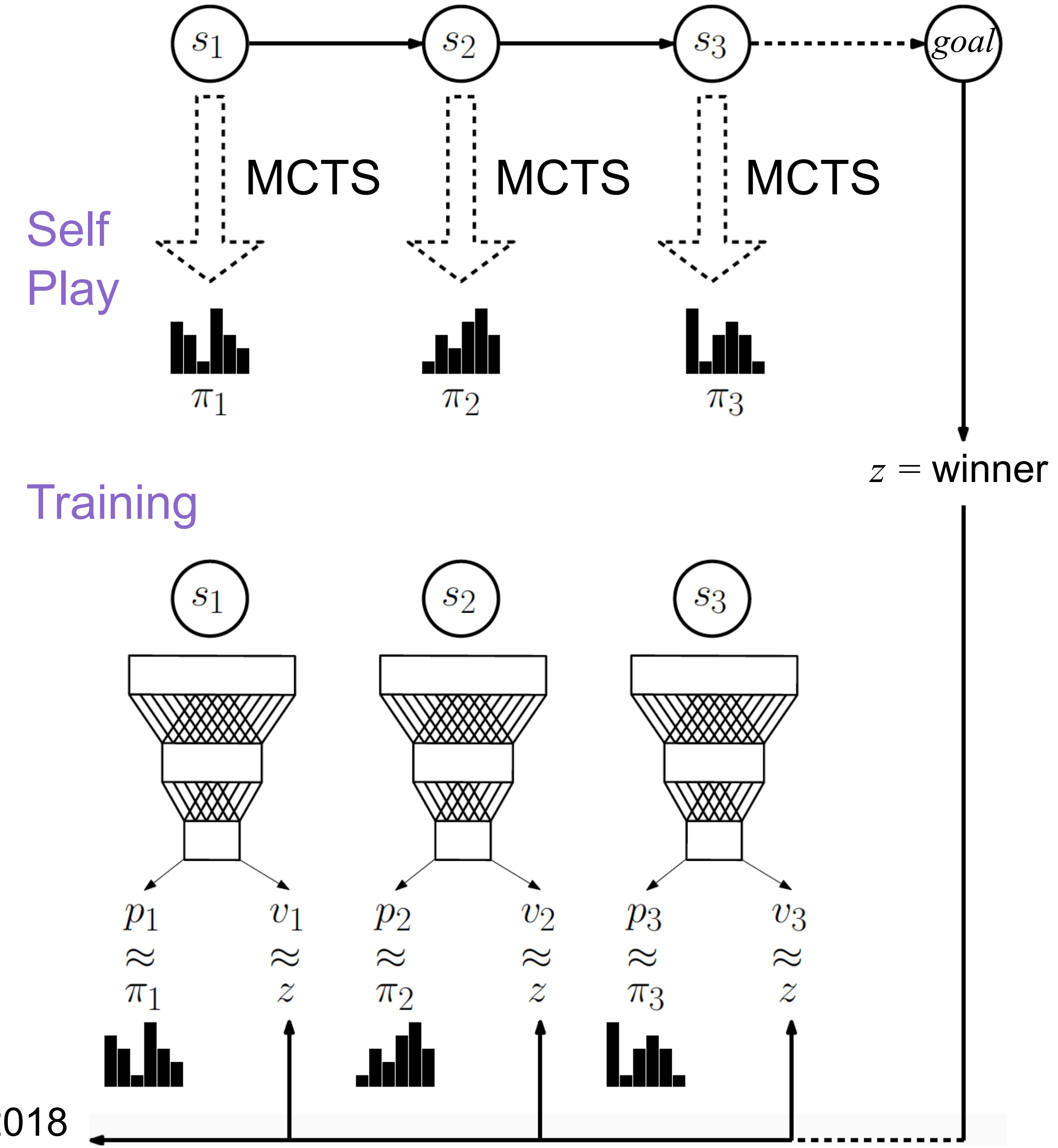
Propositional Network



The AlphaZero Method

- Self-play using Monte Carlo Tree Search (MCTS) at each state, guided by Neural Network

- Train Neural Network with self-play result



Silver et al., Science 2018

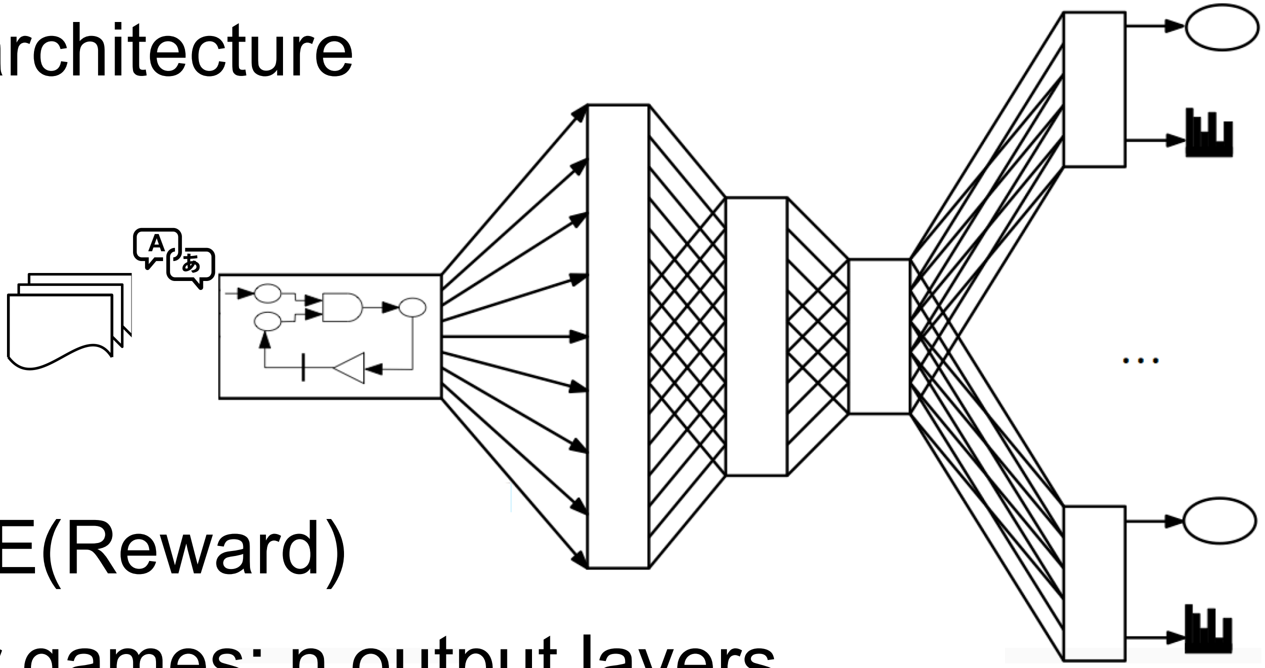
UNSW SYDNEY

Generalising AlphaZero to Full GDL

AlphaZero

- Two-player, zero-sum, symmetric, turn-based, board games
- Hand-crafted network architecture for each game

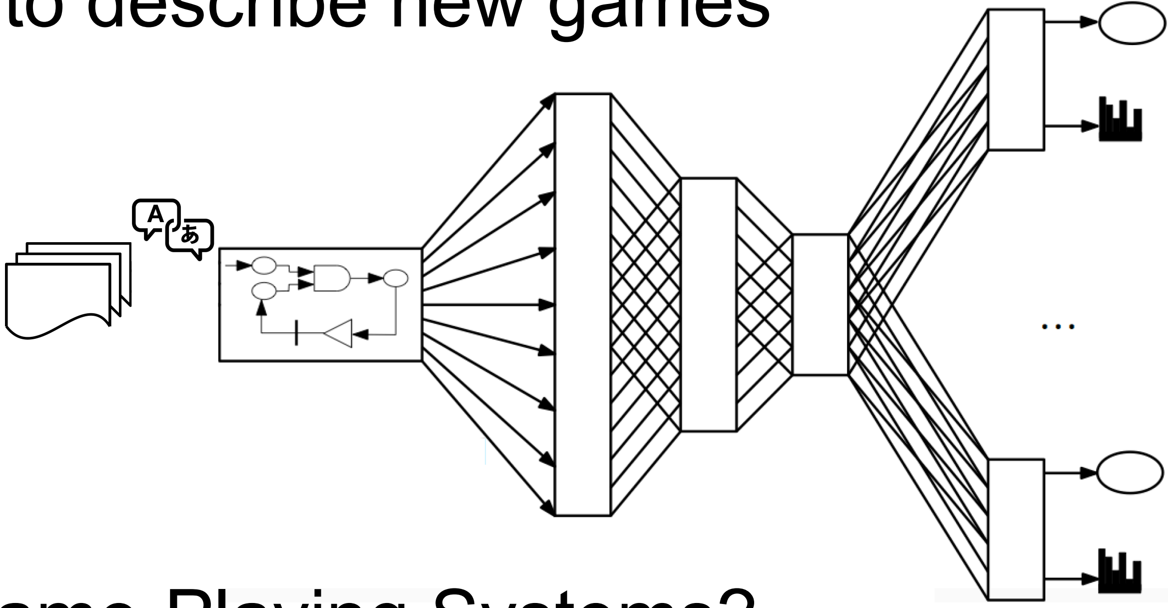
GGPZero



- Arbitrary GDL games
- Non-zero-sum games: E(Reward)
- Asymmetric multiplayer games: n output layers

GGP: KR + Inference + Search + Learning

- Knowledge representation to describe new games
- Reasoning to understand the rules
- Search enhanced by Learning to play well



Can we learn from General Game-Playing Systems?

- “Maybe our conception of chess has been too limited.”

  (Demis Hassabis)

- “Deeper understanding is very difficult to make explicit.”

  (Judit Polgár)

Outline

1. General game-playing programs
   - KR for describing new games
   - + Inference + Search + Deep Learning

2. General collaborative problem-solving robots
   - Hybrid robot control
   - + Epistemic planning for collaboration

3. Agent programming for interactive artworks
4. A new challenge – Escape Room Robot

General Problem-Solving Robot (2016)

Goals

- High-level planning, execution monitoring
- Parallel actions, execution constraints
- Handle human interference

2x speed

↪ General blocksworld problem-solving robot

Outline

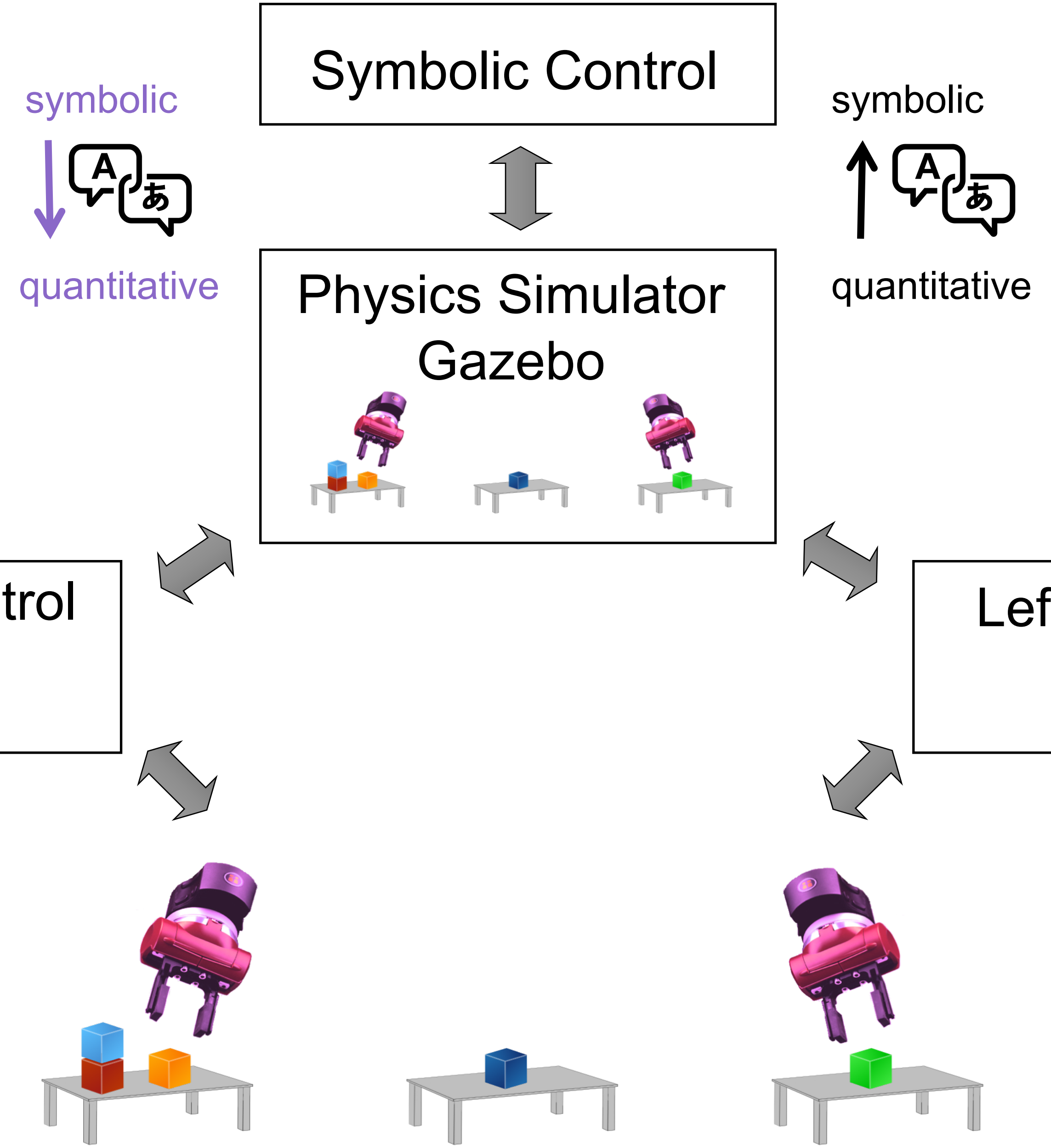
1. General game-playing programs
   - KR for describing new games
   - + Inference + Search + Deep Learning

2. General collaborative problem-solving robots
   - Hybrid robot control
   - + Epistemic planning for collaboration

3. Agent programming for interactive artworks
4. A new challenge – Escape Room Robot

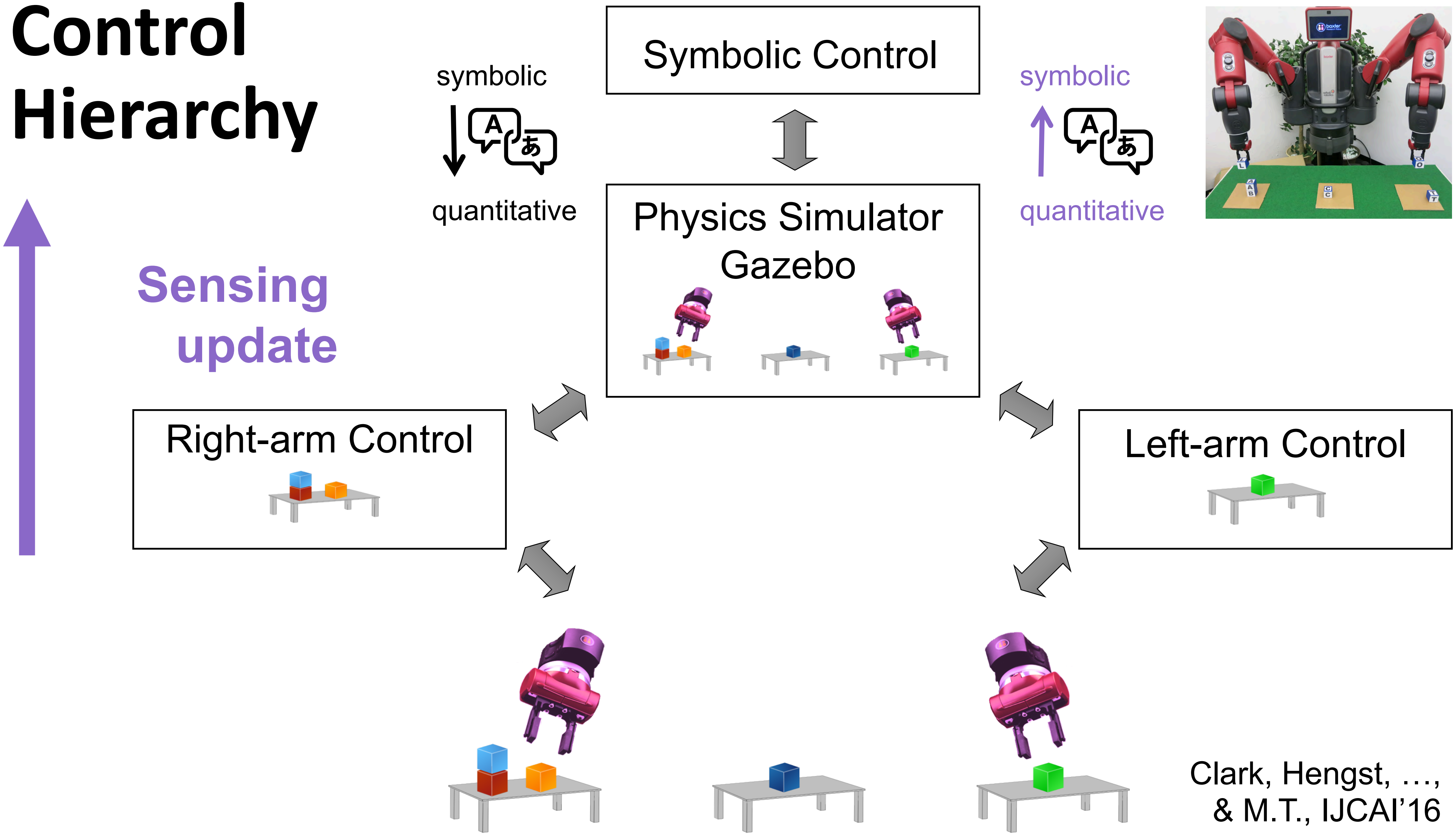
Control Hierarchy

Action update



Clark, Hengst, …,
& M.T., IJCAI'16

Control Hierarchy



Clark, Hengst, …,
& M.T., IJCAI'16

Outline

1. General game-playing programs
   - KR for describing new games
   - + Inference + Search + Deep Learning

2. General collaborative problem-solving robots
   - Hybrid robot control
   - + Epistemic planning for collaboration

3. Agent programming for interactive artworks
4. A new challenge – Escape Room Robot

UNSW SYDNEY

Describing Epistemic Planning Problems

- Robot R grabs a block (b) to check if it's labelled "A"
- Robot S may observe this

pre: `label(b,A)`

eff: `has(R,b)`

pre: `not label(b,A)`

eff: `has(R,b)`

Describing Epistemic Planning Problems

- Robot R grabs a block (b) to check if it's labelled "A"
- Robot S may observe this

```
pre:  label(b,A)
      looking(S)
eff:  has(R,b)
```

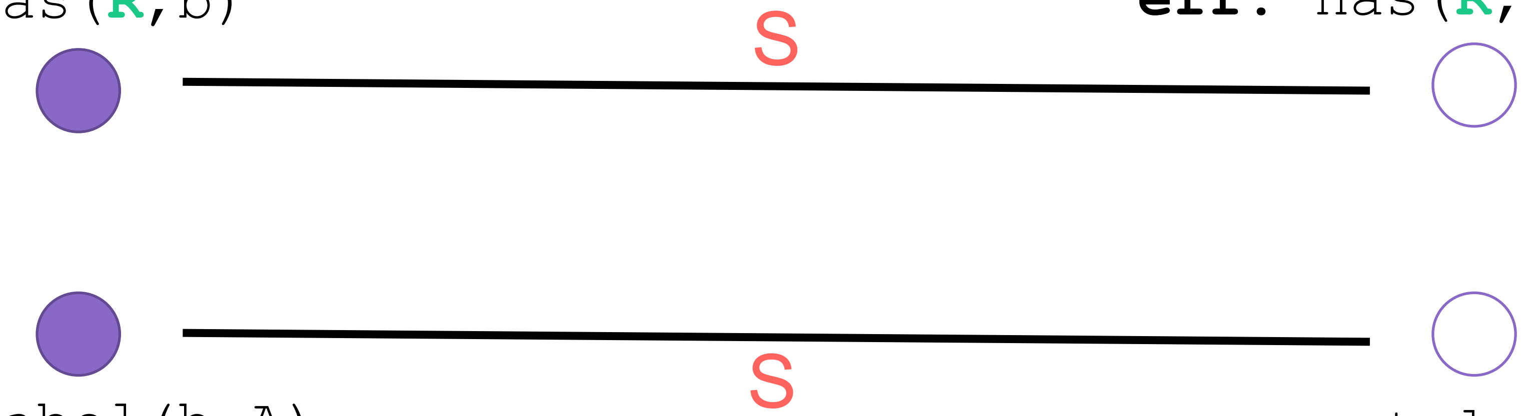
```
pre:  not label(b,A)
      looking(S)
eff:  has(R,b)
```

Describing Epistemic Planning Problems

- Robot R grabs a block (b) to check if it's labelled "A"
- Robot S may observe this

```
pre:  label(b,A)
      looking(S)
eff:  has(R,b)
```

```
pre:  not label(b,A)
      looking(S)
eff:  has(R,b)
```




```
pre:  label(b,A)
      not looking(S)
eff:  has(R,b)
```

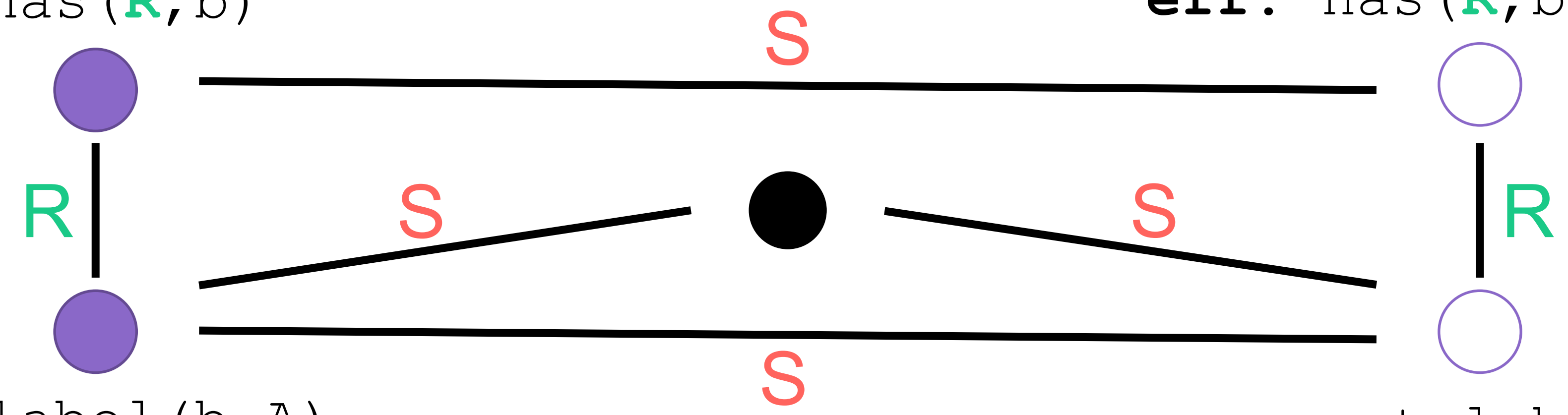
```
pre:  not label(b,A)
      not looking(S)
eff:  has(R,b)
```

Describing Epistemic Planning Problems

- Robot R grabs a block (b) to check if it's labelled "A"
- Robot S may observe this

```
pre: label(b,A)
     looking(S)
eff: has(R,b)
```

```
pre: not label(b,A)
     looking(S)
eff: has(R,b)
```




```
pre: label(b,A)
     not looking(S)
eff: has(R,b)
```

```
pre: not label(b,A)
     not looking(S)
eff: has(R,b)
```

A New Event Model Description Language

```
Grab(x,b)        causes has(x,b)
Grab(x,b)        causes obs(x,label(A))   if label(b,A)
Grab(x,b)        causes obs(y,grab(x,b)) if looking(y)

Signal(x,y)      causes looking(y)
Distract(x,y)    causes not looking(y)
```

It follows that, after Signal(R,S), Distract(R,T), Grab(R,b),

- R knows whether block b is labelled “A”
- S knows that R knows whether block b is labelled “A”
- T doesn’t know that S knows that R knows …

A New Event Model Description Language

```
Grab(x,b)       causes has(x,b)
Grab(x,b)       causes obs(x,label(A))   if label(b,A)
Grab(x,b)       causes obs(y,grab(x,b))  if looking(y)

Signal(x,y)     causes looking(y)
Distract(x,y)   causes not looking(y)
```

Theorem.

Rajaratnam & M.T., KR'21

- Every event model can be described in the language (DER)
- The DER description can be exponentially smaller
- The canonical DER description is always polynomial in size

General Epistemic BW Problem-Solving Robot (Work in Progress)

Goals

- High-level epistemic planning, execution monitoring
- Parallel actions, execution constraints
- Handle human/robot collaboration

Outline

1. General game-playing programs
   - KR for describing new games
   - + Inference + Search + Deep Learning

2. General collaborative problem-solving robots
   - Hybrid robot control
   - + Epistemic planning for collaboration

3. Agent programming for interactive artworks
4. A new challenge – Escape Room Robot

UNSW SYDNEY

Hierarchical Control for Virtual Agents

Animated characters

- Act and react to human gestures and movements

Hierarchical control combines

- High-level BDI-based agent program (Jason)
- Low-level control engine for characters (Unity)

Hierarchical Control for Virtual Agents

Intraspace
UNSW / Academy of Fine Arts Vienna, Christina Jauernek

Outline

1. General game-playing programs
   - KR for describing new games
   - + Inference + Search + Deep Learning

2. General collaborative problem-solving robots
   - Hybrid robot control
   - + Epistemic planning for collaboration

3. Agent programming for interactive artworks
4. A new challenge – Escape Room Robot

The Escape Room Robot Challenge

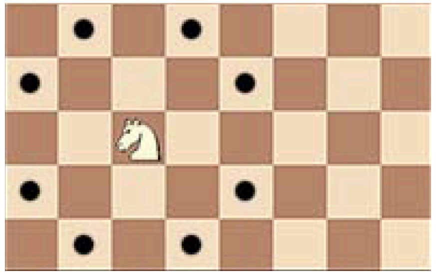
Goal

- Test the ability of robots to solve new problems in new environments

Tasks

- Scan and map room
- Identify physical objects
- Understand and solve symbolic puzzles
- Relate solutions to objects and actions

PHOCO pho-co.com



Summary: KR to Leverage AI Systems

Goal

- Leverage our AI systems by enhancing their ability to understand and adapt to new problems

KR helps systems to

- Understand and reason about problem descriptions

Examples

- KR + Search + Deep RL to build a general game player
- KR + Hybrid control architectures for robots, virtual agents

Thanks

Dennis Del Favero
Adrian Goldwaser
Som Guan
Bernhard Hengst
Maurice Pagnucco
David Rajaratnam
Ji Ruan
Claude Sammut

Keith Clark
Thorsten Engesser
Christian Freude
Alvaro Gunawan
Christina Jauernik
Robert Mattmüller
Ajit Narayanan
Bernhard Nebel
Peter Robinson
Wolfgang Tschapeller

UNSW SYDNEY