Transfer Learning Using Soar

Joseph Xu
with John Laird and Sam Wintermute
5/27/2007

Outline

- Year 2 Transfer Learning Project
  - General Game Player
  - Games
- Our approach to transfer learning
- Nuggets and coals

Year 2 Transfer Learning

- General Game Player platform as exclusive simulation/evaluation environment
- Focus on higher levels of transfer (composition, abstraction, generalization, reformulation, differing)
- Source-Target evaluation paradigm remains the same
  1. Spend time studying and solving some Source problems
  2. Demonstrate improved learning performance on the Target problem

General Game Player

- Maintained by Stanford Logic Group
- Game Description Language (GDL) + Game Manager
  - GDL description gives you everything you need to know to play a game
    - All game rules + Initial conditions
  - Game Manager is an Internet server that hosts gameplay
    - Maintain state of a game

Characteristics of GDL Games

- Perfect information
  - Agent “has” all the knowledge it can ever want about the game
  - Cuts down on potentially transferrable knowledge
- Deterministic
  - Internal simulation is straightforward
- Finite
  - Limited number of symbols, states

Example GDL Rules

Define legal moves:

```
(<= (legal explorer (move ?dir))
    (direction ?dir))
```

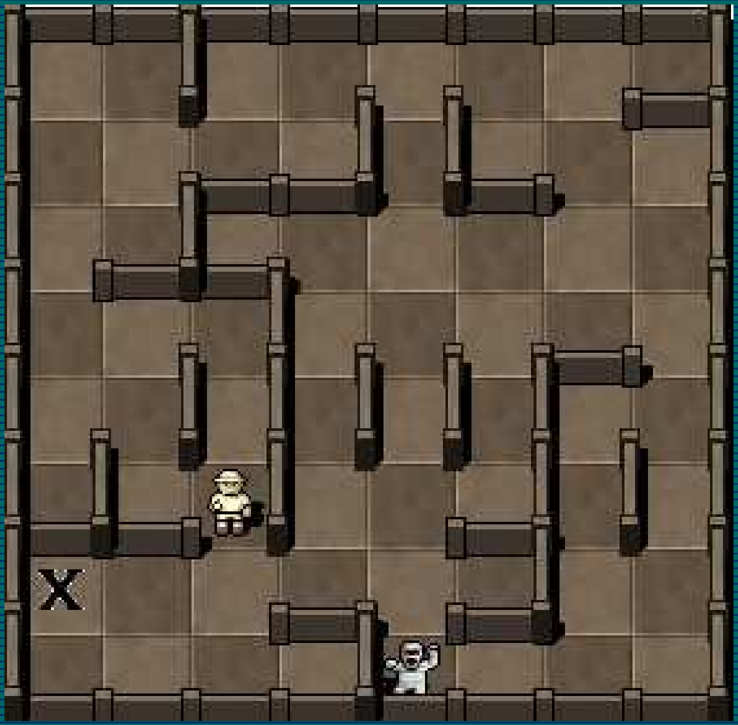
Update game states:

```
(<= (next (location explorer ?x2 ?y2))
    (true (location explorer ?x1 ?y1))
    (does explorer (move ?dir))
    (newlocation1 ?x1 ?y1 ?dir ?x2 ?y2 explorer))
```

Entailments:

```
(<= (newlocation1 ?x ?y1 north ?x ?y2 ?obj)
    (true (location ?obj ?x ?y1))
    (succ ?y1 ?y2)
    (not (true (wall ?x ?y1 north)))
    (distinct ?obj exit))
```

Year 2 Games

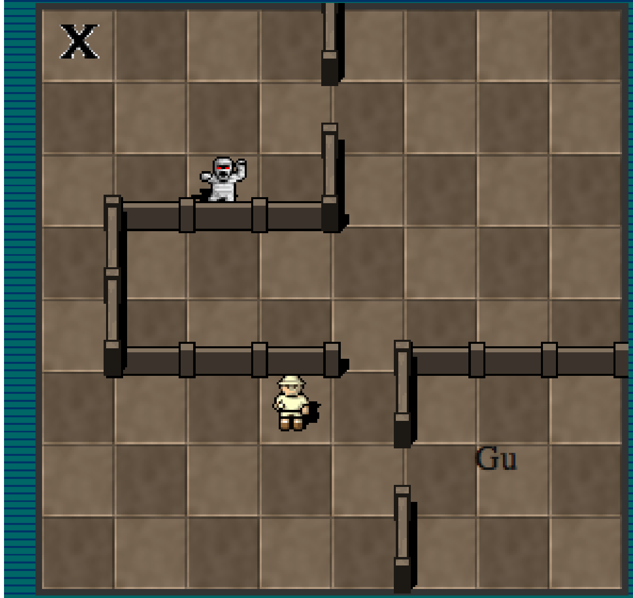


- Mummy Maze
  - Explorer has to get to exit
  - Mummy chases explorer 2 steps for every step explorer takes
  - Mummy’s choice of moves deterministic
    - Always move toward explorer
  - Enriched game structure and interactions
    - Weapons and items

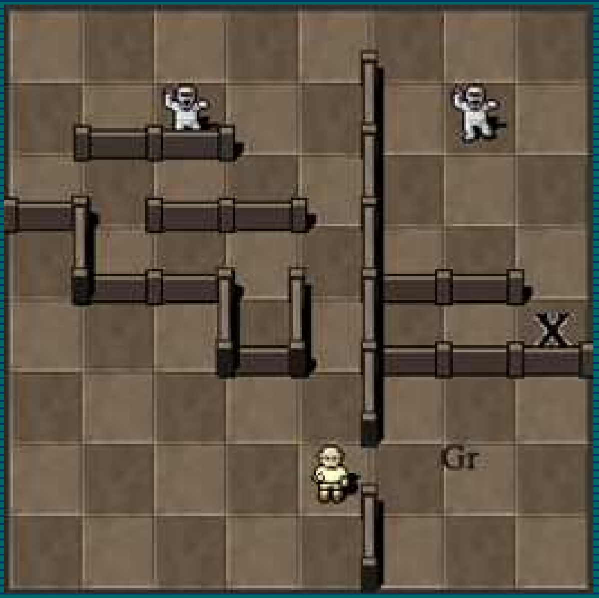
Year 2 Games

- Escape (MM variant)
  - No mummy, instead there are obstacles
  - Explorer must obtain items to overcome obstacles
  - Items can be combined
- Rogue (MM variant)
  - Loosely based on Nethack
  - Navigate a set of rooms looking for a prize
  - Weapons, armor, items, monsters
- Build
  - Build a structure as high as possible with various materials
  - Focuses on nonlinear interactions between combined materials
  - Not situated in a 2d grid world like others

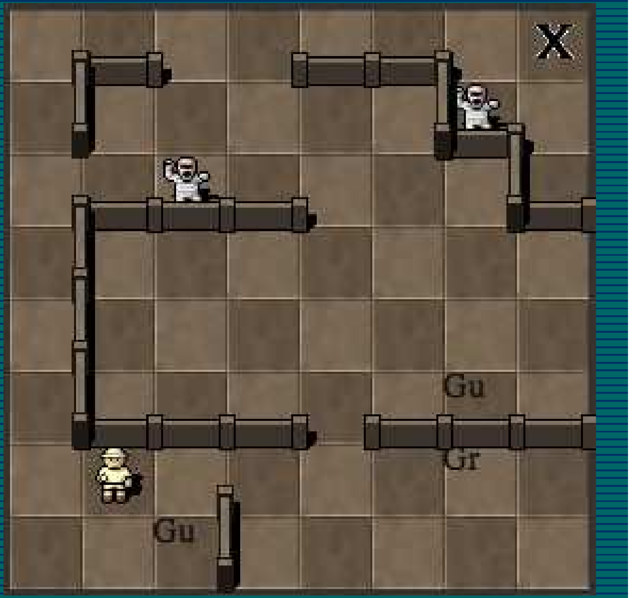
Transfer Level 6 Example



Source 1: Explorer must kill mummy with gun



Source 2: Explorer must kill mummies with grenade

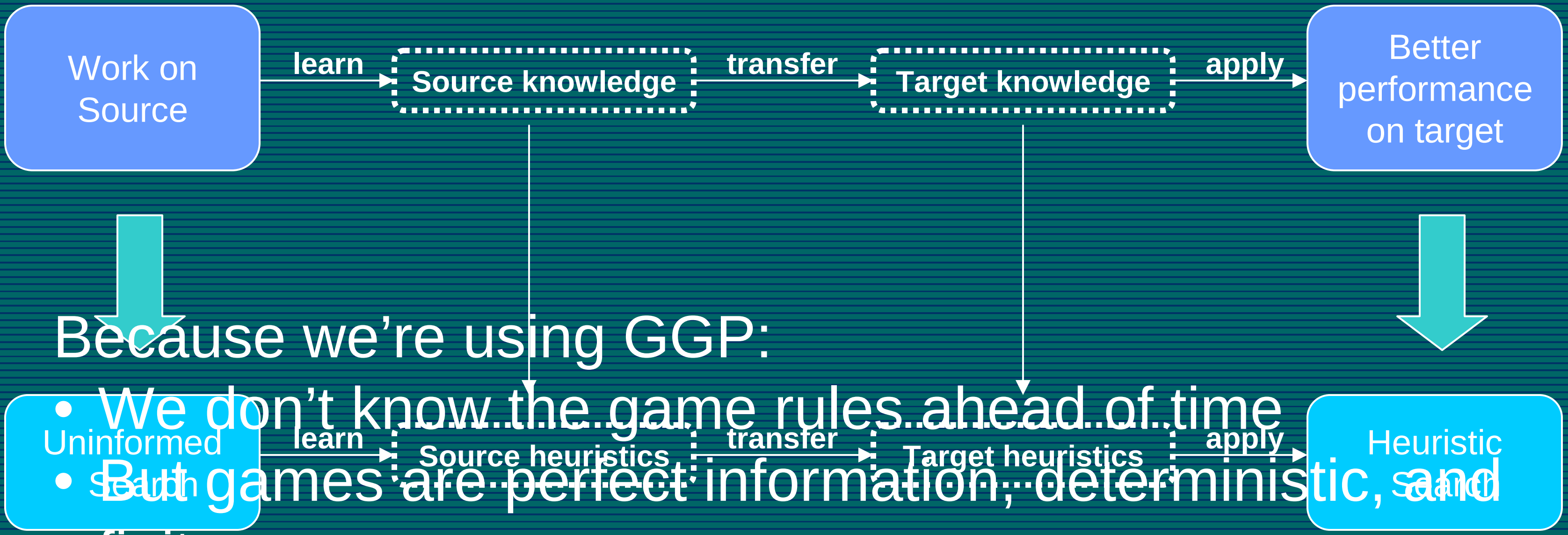


Target: Explorer must use guns or grenade to kill mummies

Outline

- Year 2 Transfer Learning Project
  - General Game Player
  - Games
- Our approach to transfer learning
- Nuggets and coals

Transfer Paradigm



Because we’re using GGP:

- We don’t know the game rules ahead of time
- But games are perfect information, deterministic, and finite

How do we do this in Soar?

Uninformed Search in Soar

- Automatically translate GDL rules into Soar productions
  - ↳ Allows Soar to simulate game completely internally
  - Use the selection space mechanism to do internal look-ahead
    - Equivalent to depth-first search of game tree
  - Modify to use iterative deepening for improved performance

Learning Heuristics

- Chunking
  - What the agent should do in a particular state
- Rule relaxation
  - Distance to goal estimates
    - Manhattan distance
- Solution trace analysis
  - Evaluate accuracy of other heuristics
  - Learn positive and negative indicators for how good a state is
    - Positive: picking up the gun, killing the mummy
    - Negative: losing health, wasting a bullet

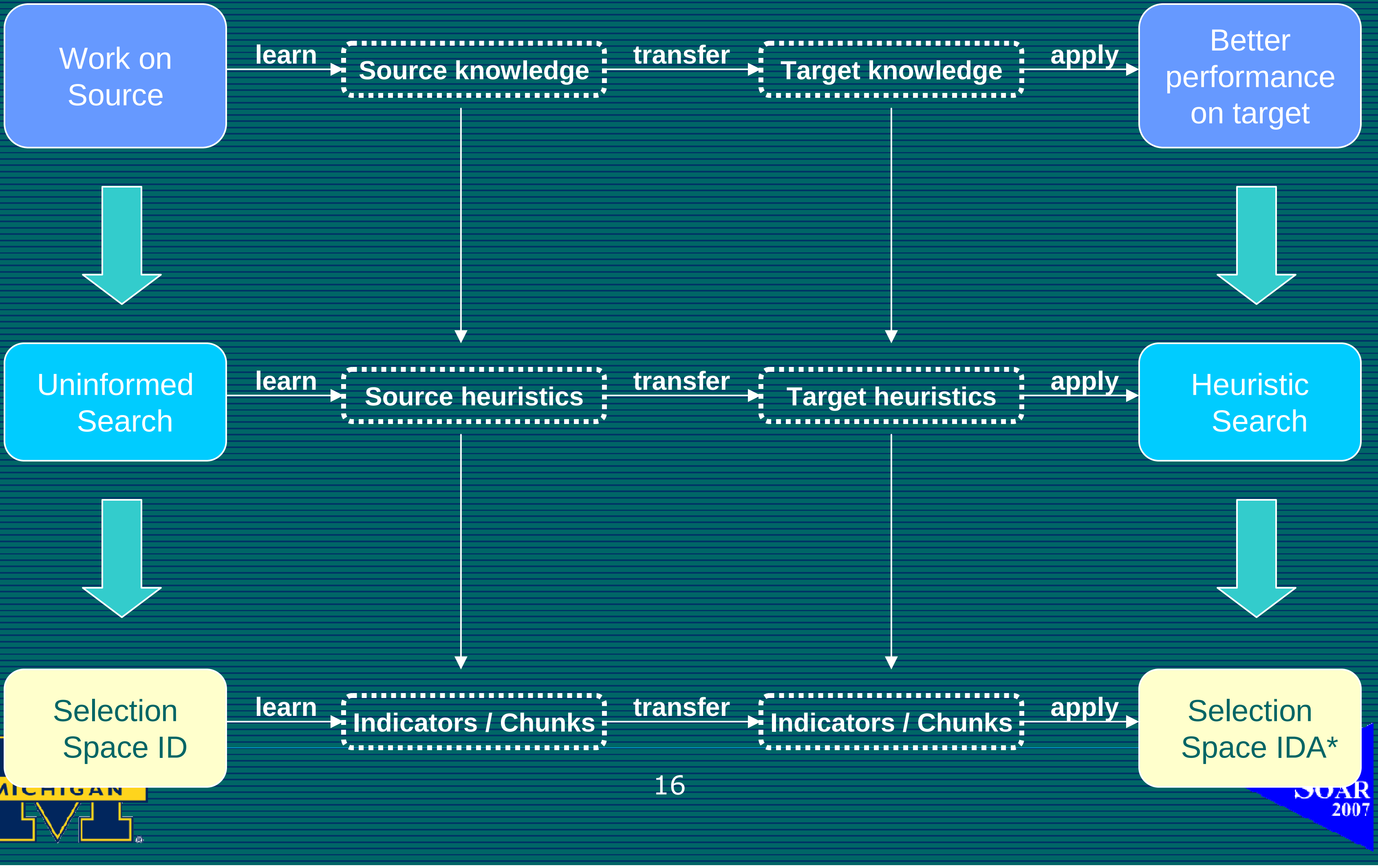
Heuristic Searching in Soar

- A* is an optimal heuristic search
  - f(n) = g(n) + h(n)
  - Uses exponential amount of memory
    - Abuses Soar’s working memory
- Use iterative deepening A*
  - Only requires substate stack as memory
  - Cut off by f(n) rather than depth

h(n) heuristic

- Combination of relaxed distance estimates and state indicators
  - Encourages searching further on more promising paths
- Might not be admissible, so not guaranteed optimal solutions

Transfer Paradigm



Work on Source

learn

Source knowledge

transfer

Target knowledge

apply

Better performance on target

Uninformed Search

learn

Source heuristics

transfer

Target heuristics

apply

Heuristic Search

Selection Space ID

learn

Indicators / Chunks

transfer

Indicators / Chunks

apply

Selection Space IDA*

Transferring Knowledge

- All knowledge has implicit assumptions about their native domain
- If those assumptions are violated, the knowledge is no longer useful
- Transfer is the attempt to modify or generalize knowledge to apply to different domains

Transferring Knowledge

- The knowledge we will be transferring are the heuristics learned in the source
- Some heuristics are general enough to transfer for free
  - Most game scenarios are set in a 2d grid world
    - Manhattan distance

Transferring Knowledge

- Others require making analogies between the source and target games
  - Rule level analogies
    - Try to map source GDL rules onto target GDL rules
    - Transfers relaxations, distance estimates
  - State level analogies
    - Try to map a game state in source to a game state in target
    - Transfers indicators, chunks

Nuggets and Coals

- Nuggets
  - Have a general paradigm for doing transfer in Soar
  - Should be able to handle a large variety of games as-is
- Coals
  - Still have a lot of work to do to demonstrate transfer
  - Transfer methods not proven/comprehensive