# B(E)4M36PUI – Artificial Intelligence Planning
# Solving universe-sized puzzles with human-sized patience

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Antonín Komenda
Artificial Intelligence Center (AIC)

AI CENTER FEE CTU

# Course Overview

- https://cw.fel.cvut.cz/wiki/courses/pui
- Lectures
  - 2 parts (Antonin Komenda and Stefan Edelkamp)
  - Invited lecture (Schlumberger – practically used automated planning)
- Seminars
  - 2 parts (Michaela Urbanovská and Jan Mrkos)
  - Synchronized topics with lectures
  - Two assignment projects with multiple parts
- Exam
  - Primarily written form (theory + exercises)
  - Points from the seminars
    - Maximum 50 points
    - 25 points required for the credit (zápočet)

# Lectures & Seminars Overview (Part 1)

AI CENTER FEE CTU

1. Introduction
2. Representations (for Classical Planning)
3. Search (for Classical Planning)
4. Automated Planning in Practice (invited lecture from Schlumberger)
5. Heuristics (for Classical Planning) I – Relaxations
6. Heuristics (for Classical Planning) II – Landmarks & Potentials
7. Heuristics (for Classical Planning) III – Abstractions

# Assignments

- Classical planning
  - PDDL modeling (5 points) deadline: week 3
  - Grounding implementation (10 points) deadline: week 4
  - Search algorithm + heuristic implementation (15 points) deadline: week 8
  - 30 points total
- Probabilistic planning
  - Implementation of probabilistic planning algorithm (up to 20 points)

Both assignments have to be submitted.

50 points in total → 25 required for the credit (zápočet)

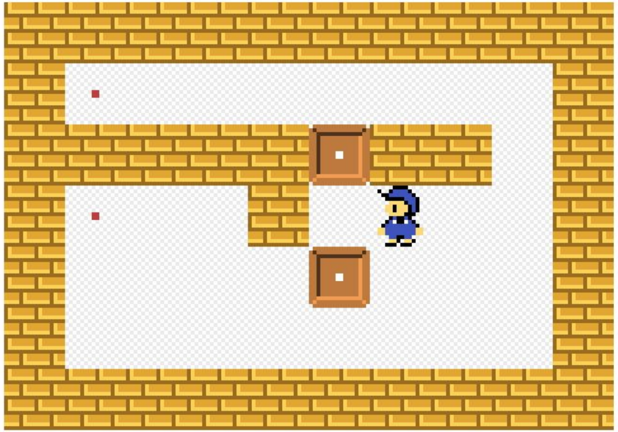
# Communication

- Email is the primary form of communication
- Ask at the lectures/tutorials!

Email addresses

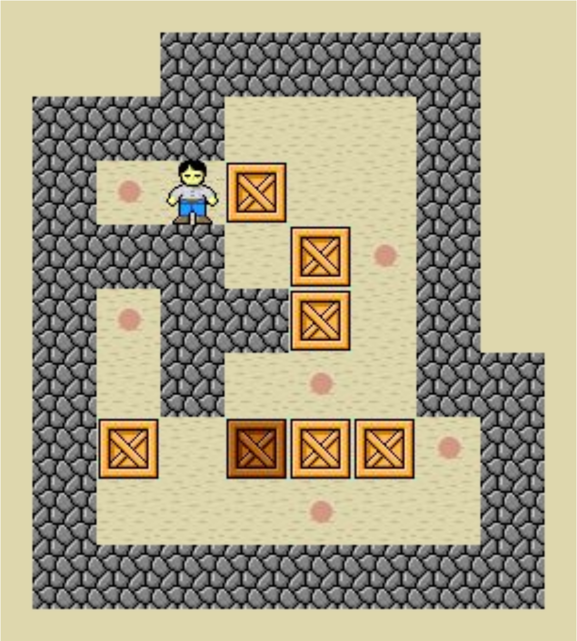
- Classical Planning (Part 1)
  - Antonín Komenda antonin.komenda@fel.cvut.cz
  - Michaela Urbanovská urbanm30@fel.cvut.cz
- Stochastic Planning (Part 2)
  - Stefan Edelkamp
  - Jan Mrkos

# Freshman's Sokoban Example



https://www.sokobanonline.com/play/community/experiment/127458_practice

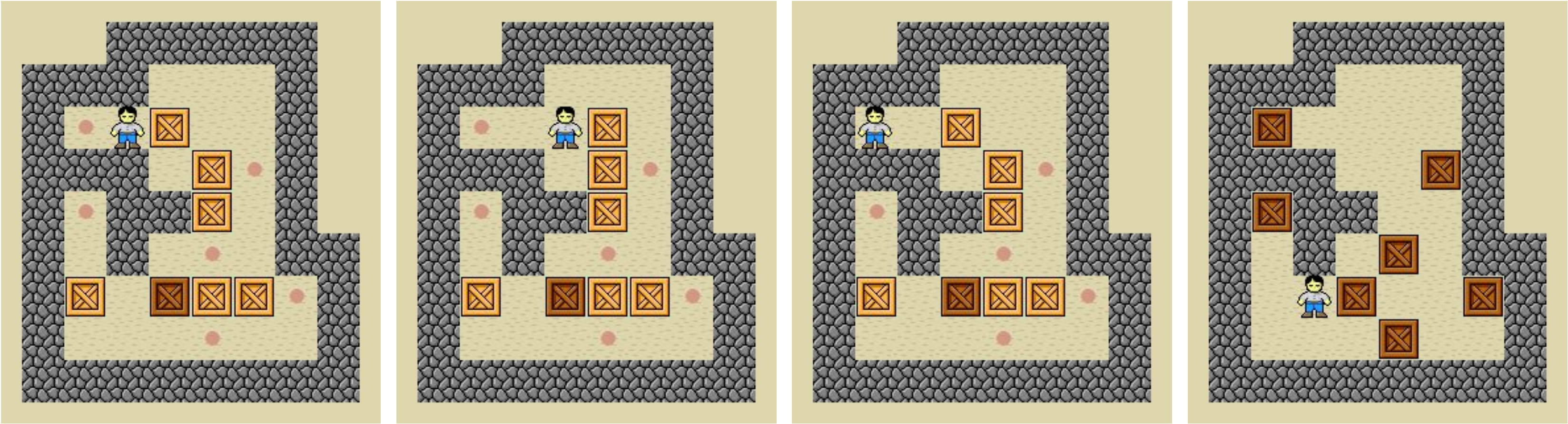
# Pondering Sokoban Example



[2 035 800]

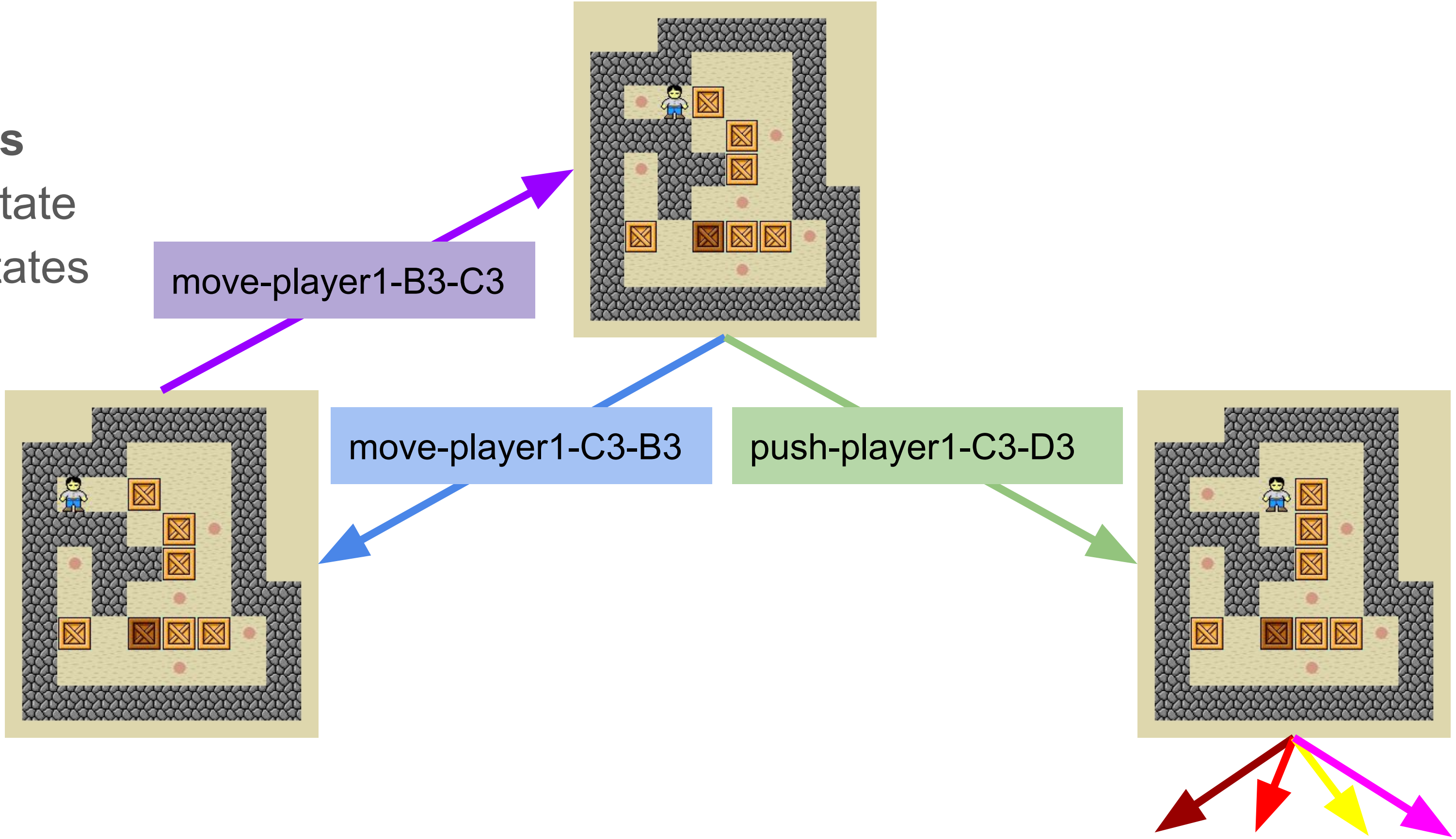
# Classical Planning Elements

- **States**
- Actions
- Initial state
- Goal states



# Classical Planning Elements

- States
- **Actions**
- Initial state
- Goal states

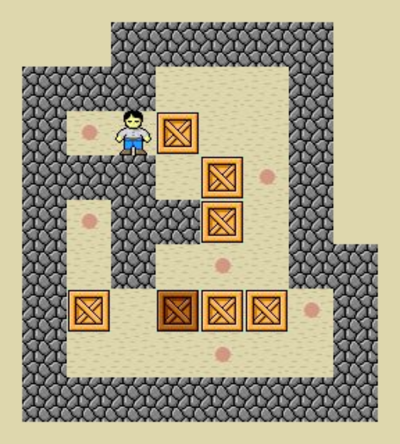


# Classical Planning Elements

- States
- Actions
- **Initial state**
- **Goal states**

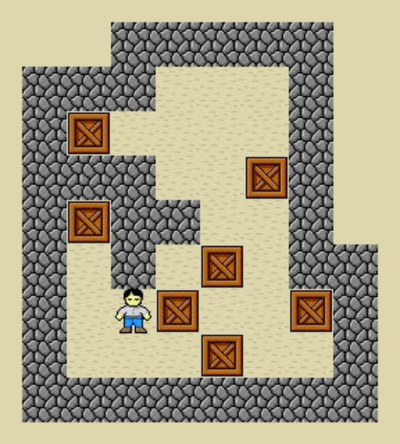
- **Problem:**

the initial state



$\rightsquigarrow$

a goal state



- **Solution: a plan**

  π=(move-player1-C3-B3, move-player1-B3-C3, push-player1-C3-D3, ...)

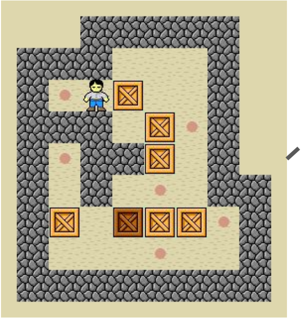
(under the regular Sokoban rules)

AI CENTER
FEE CTU

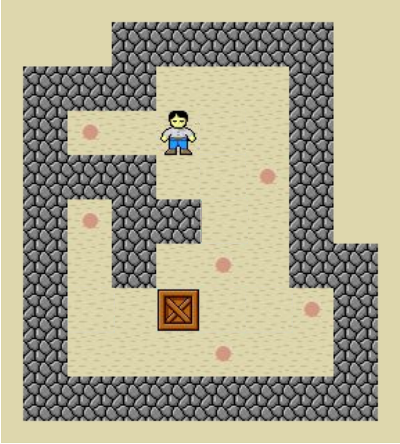
# Classical Planning Elements

- **States**
- Actions
- Initial state
- Goal states

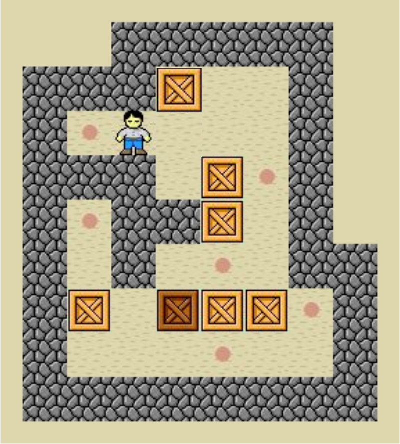
the initial state



an unreachable state



a dead-end state



(under the regular Sokoban rules)

# What is Automated Planning?

- Artificial Intelligence (sub-field)
  - (general) problem solving
- Decision Theory meets Computer Science
  - sequential decision making
  - various forms of combinatorial optimization problems
- Three approaches in AI to the problems of action selection or control
  - Learning: learn control from experience
  - Programming: specify control by hand
  - Planning: specify problem by hand, derive control automatically

AI CENTER
FEE CTU

# Automated Planning is Hard

- just search in the space of states using actions?

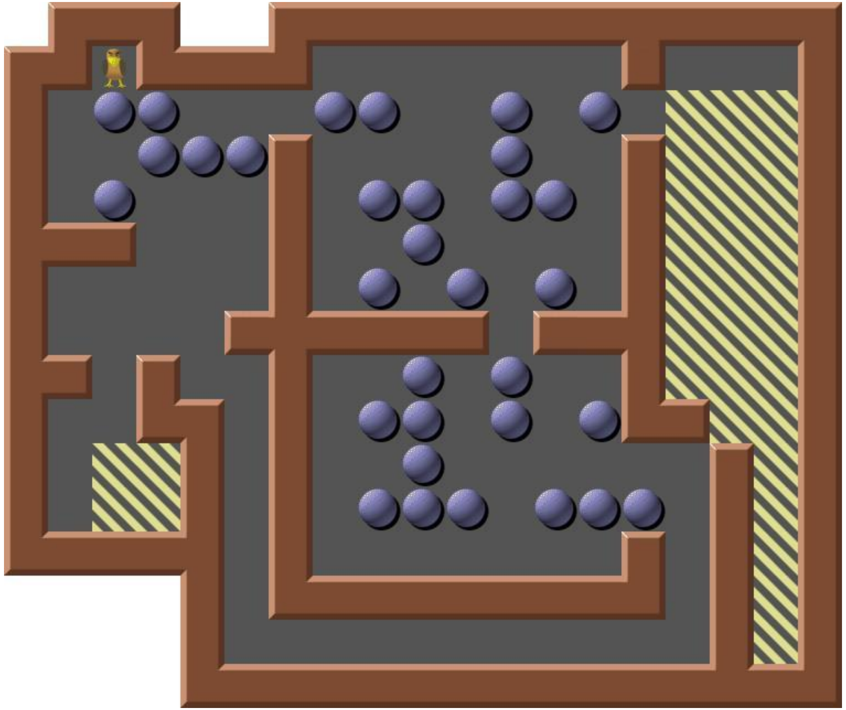
- **exponential** dependence on size of the problem
- **exponentially** long plans

- planning with “sane” length of plans is NP-complete
- classical planning is PSPACE-complete
- multi-agent variants are NEXP-complete

```
number of cells
 n | configs for (n/2) boxes

 2 | 2
 4 | 6
 6 | 20
 8 | 70
10 | 252
12 | 924
14 | 3432
16 | 12870
18 | 48620
20 | 184756
22 | 705432
24 | 2704156   | ~10^6
26 | 10400600  | ~10^7
28 | 40116600  | ~10^7
30 | 155117520 | ~10^8
...
```

# Serious Sokoban Example



[90 596 769 259 248 050 560 796 925 567 405 729]

~$10^{34}$ configurations of boxes

~$10^{70}$ +colored boxes

~$10^{81}$ +5 simultaneous players

estimated number of atoms in the observable universe: $10^{78}$–$10^{82}$, ups ;)

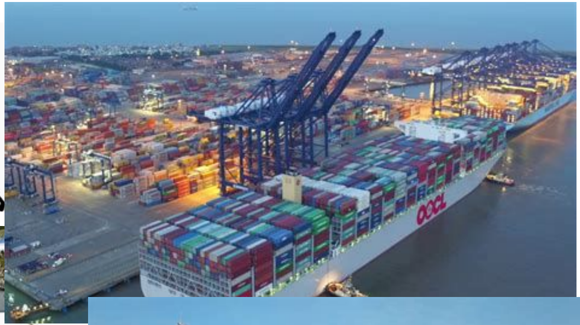
# Automated Planning is Hard

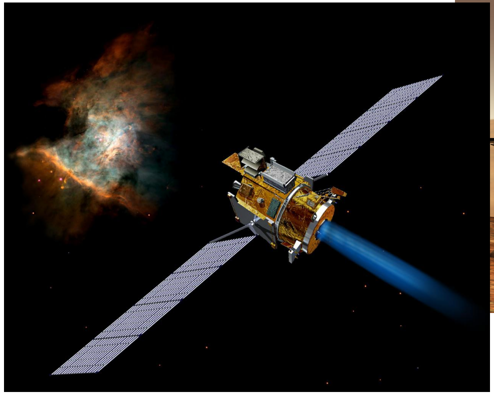
- computational hardness tells us nothing about the complexity of **individual problem instances**
- what about a problem, where each box is next to its goal cell and the player is not blocked to get to all of them?
- what about the Serious Sokoban example?
  solving such a problem is not hopeless!
  (it needs only ~$2 \cdot 10^7$ states to search through)

- → problem structure ($P \stackrel{?}{=} NP$)
- → clever representations
- → clever simplifications

# Domains

puzzles; computer, board, card games; production planning offshore drilling, and logistics; humanitarian and military missions; various-scale robotics; space missions



# Domain-independent and Domain-specific

Domain-independent:

- fundamental
- flexible
- reusable

Domain-specific:

- rigid
- efficient
- specialized

# Domain-independent and Domain-specific

Domain-independent:

- **fundamental**
- flexible
- reusable

Domain-specific:

- rigid
- efficient
- specialized


Until we get the fundamental principles ...

# Domain-independent and Domain-specific

AI CENTER
FEE CTU

Domain-independent:

- fundamental
- **flexible**
- **reusable**

Domain-specific:

- rigid
- efficient
- specialized

… we cannot be flexible and we cannot reuse ...

# Domain-independent and Domain-specific

Domain-independent:

- fundamental
- flexible
- reusable

Domain-specific:

- **rigid**
- **efficient**
- specialized

… we cannot optimize or ...

# Domain-independent and Domain-specific

Domain-independent:

- fundamental
- flexible
- reusable

Domain-specific:

- rigid
- efficient
- **specialized**

… specialize.

# Domain-independent and Domain-specific

Domain-independent:

- general structural properties and general algorithms
- automatically derived heuristics
- graph theory or probability theory, optimization theory, logic, algebra

Domain-specific:

- specific problems or specific structural families
- hand-crafted heuristics
- data structures, algorithmization, code efficiency

# Wanna plan?

## Representation → Search + Heuristics

- Representation (Lecture 2)
  - structurally analyze and **compactly represent** the problem
  - **deduce information** helping with solution of the problem
- Search (Lecture 3)
  - do not enumerate all states and actions
  - **find path** through the **implicit graph**
- Heuristics (Lectures 5, 6, 7)
  - **navigate the search** using simplified variant of the problem
  - how? relaxation, abstraction, structural information (e.g., landmarks or potentials)
  - machine learned heuristics

# Wanna plan?

**Representation** → Search + Heuristics

# **Representation** → Search + Heuristics

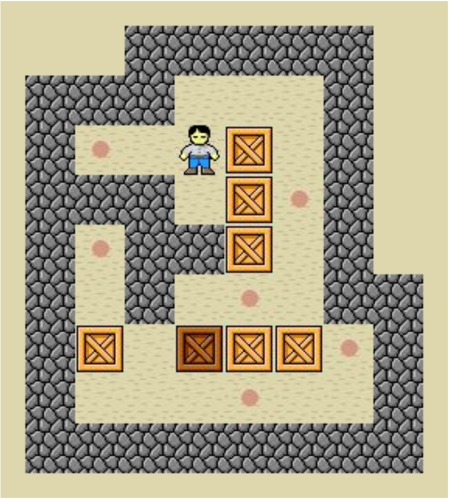
- factorized representation

```
{player1-at ∈ {A1, A2, …, B1, B2, ...},
 box1-at    ∈ {A1, A2, …, B1, B2, ...}, …}
```

- compact representation (+grounding)

```
move(who, from, to) → {move-player1-C3-B3, …}
```

- grounding of actions usable in reachable states only
- grounding of actions not leading to dead-ends only
- structural reductions
- prepare usable information for the search and heuristics

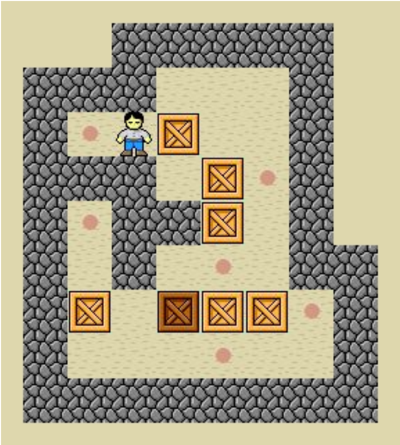


# Wanna plan?

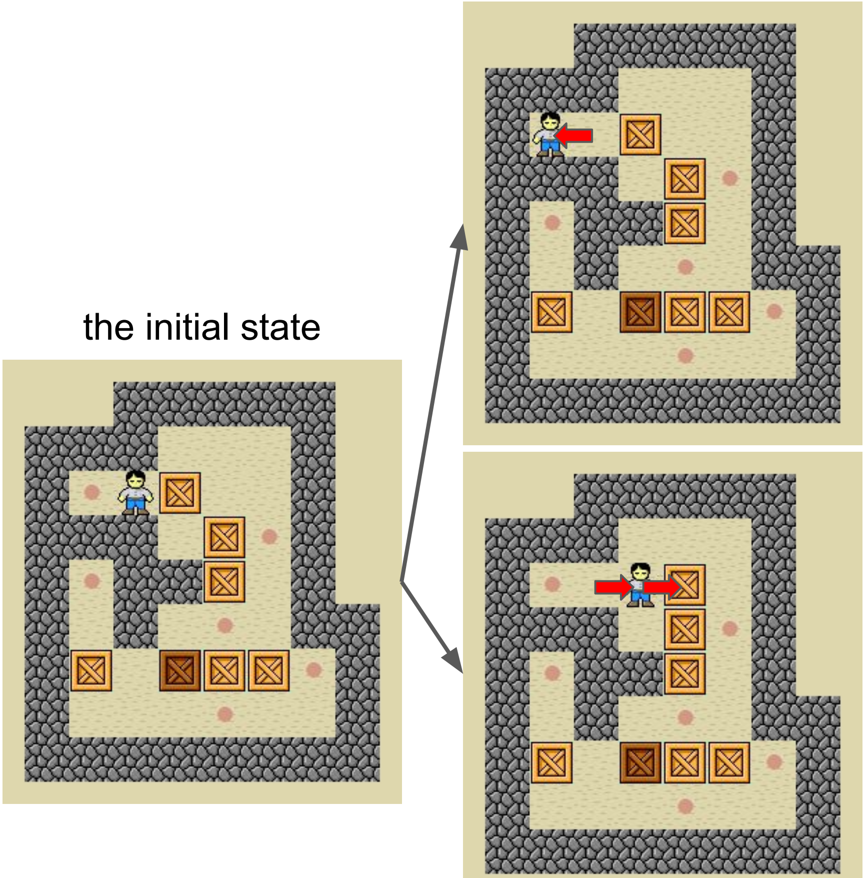
Representation → **Search** + Heuristics

# Representation → **Search** + Heuristics

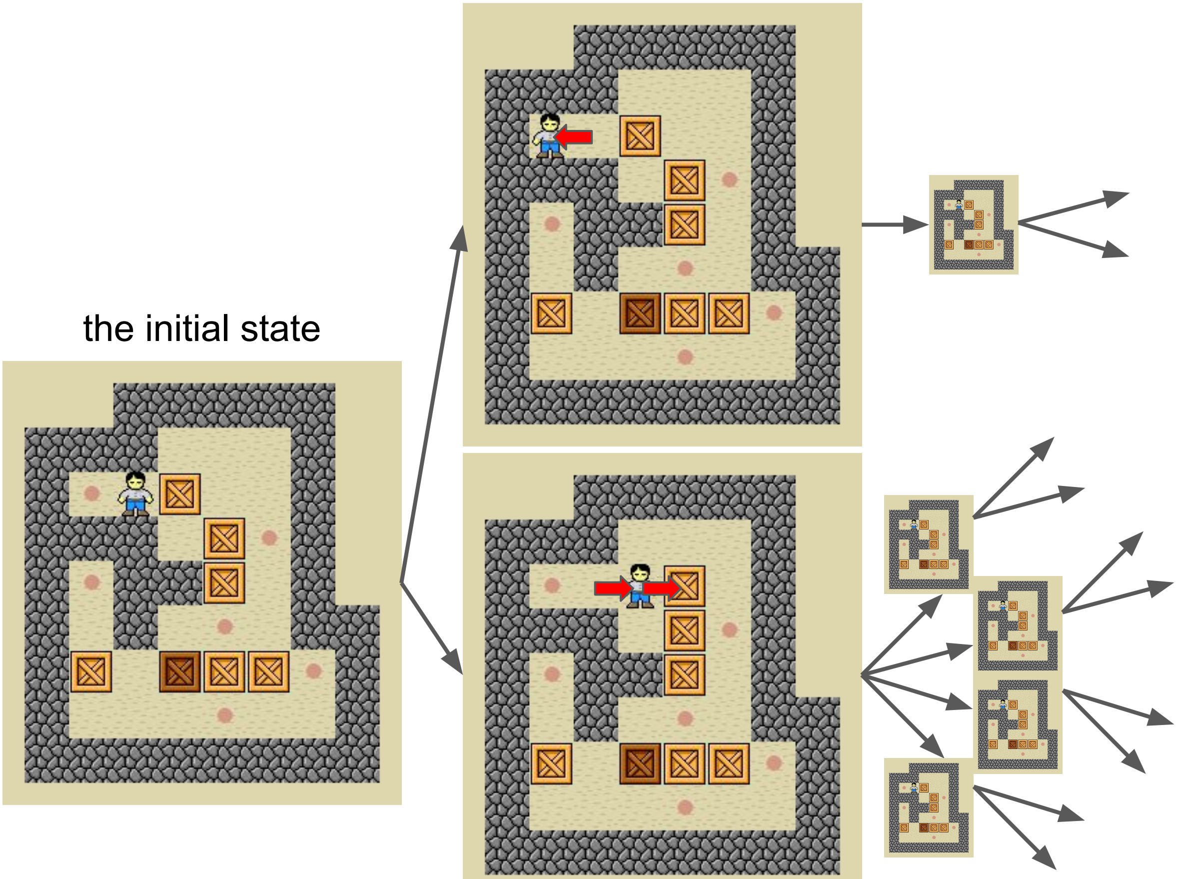
the initial state



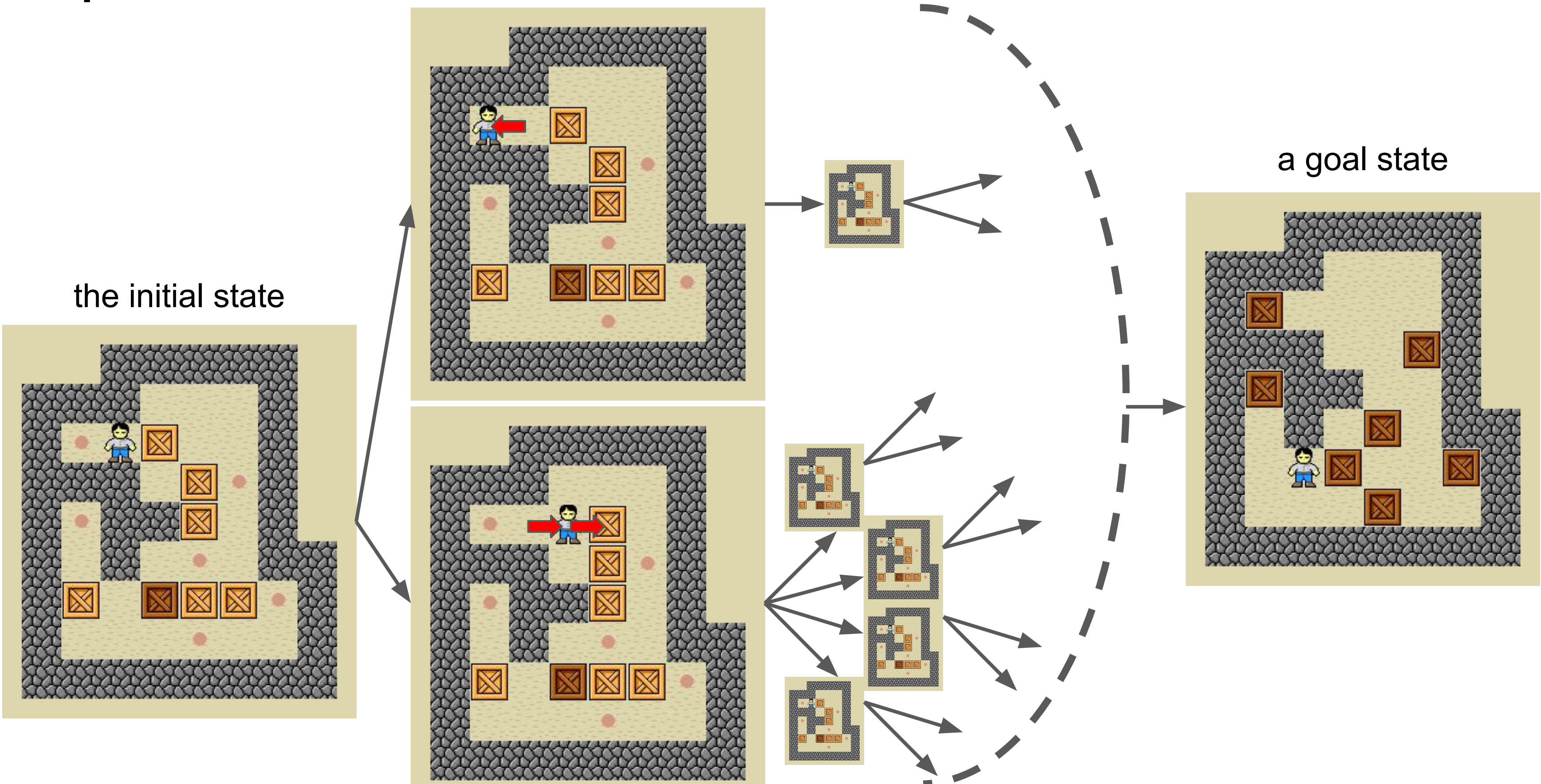
# Representation → **Search** + Heuristics



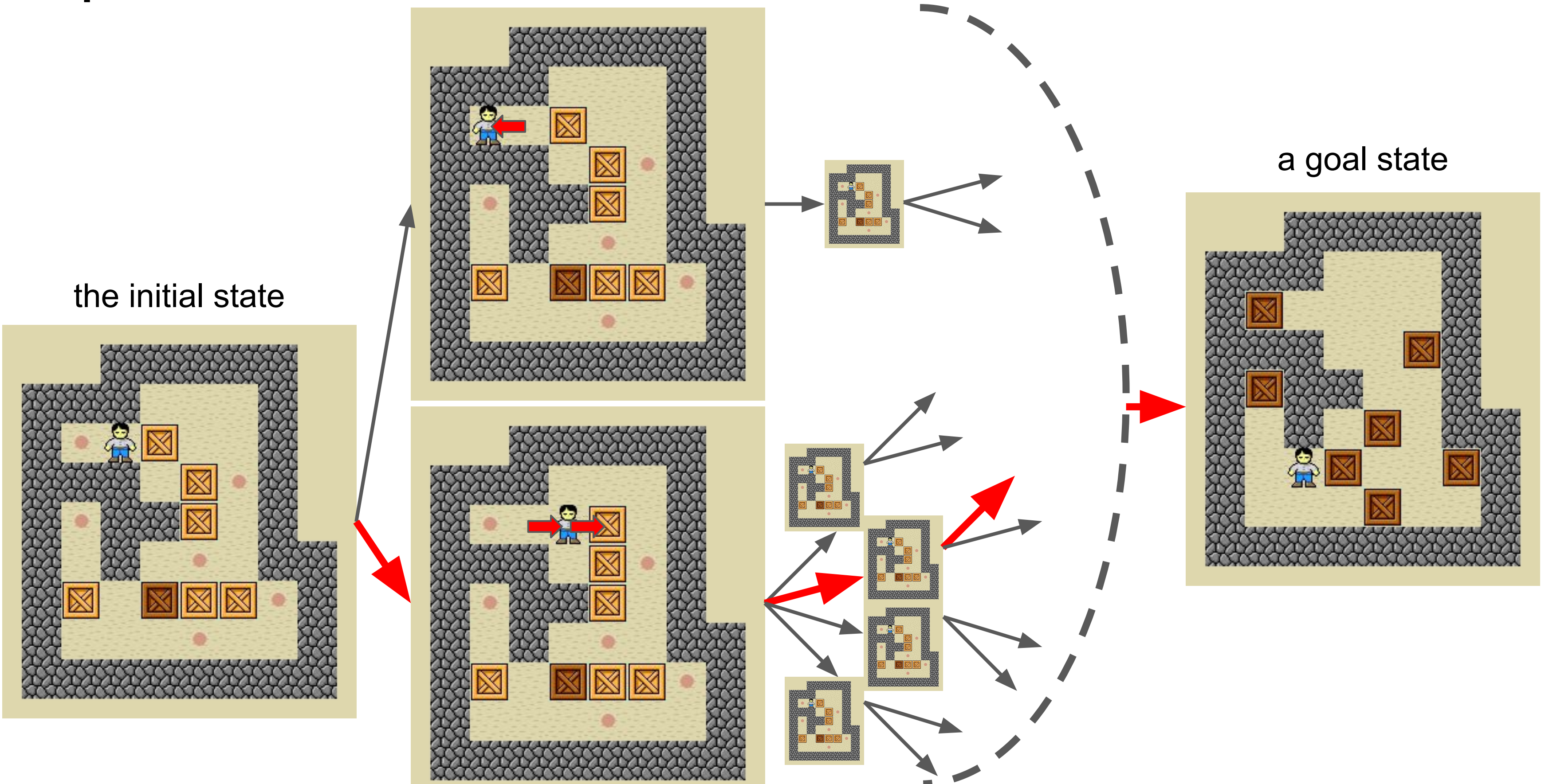
# Representation → **Search** + Heuristics



# Representation → **Search** + Heuristics



# Representation → **Search** + Heuristics



# Wanna plan?

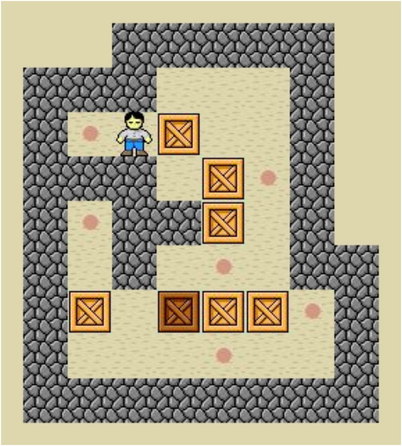
AI CENTER
FEE CTU

Representation → Search + **Heuristics**

Heuristics are strategies using readily accessible, though loosely applicable, information to control problem solving in human beings and machines. (J. Pearl)

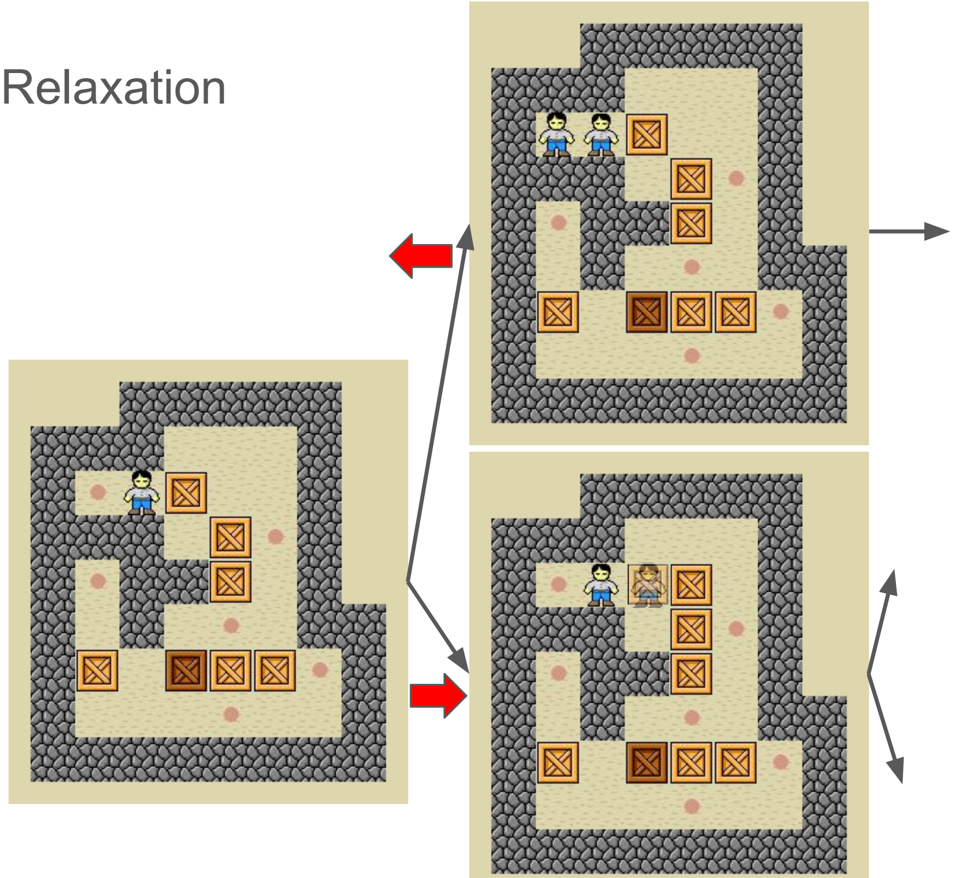
# Representation → Search + **Heuristics**

Relaxation



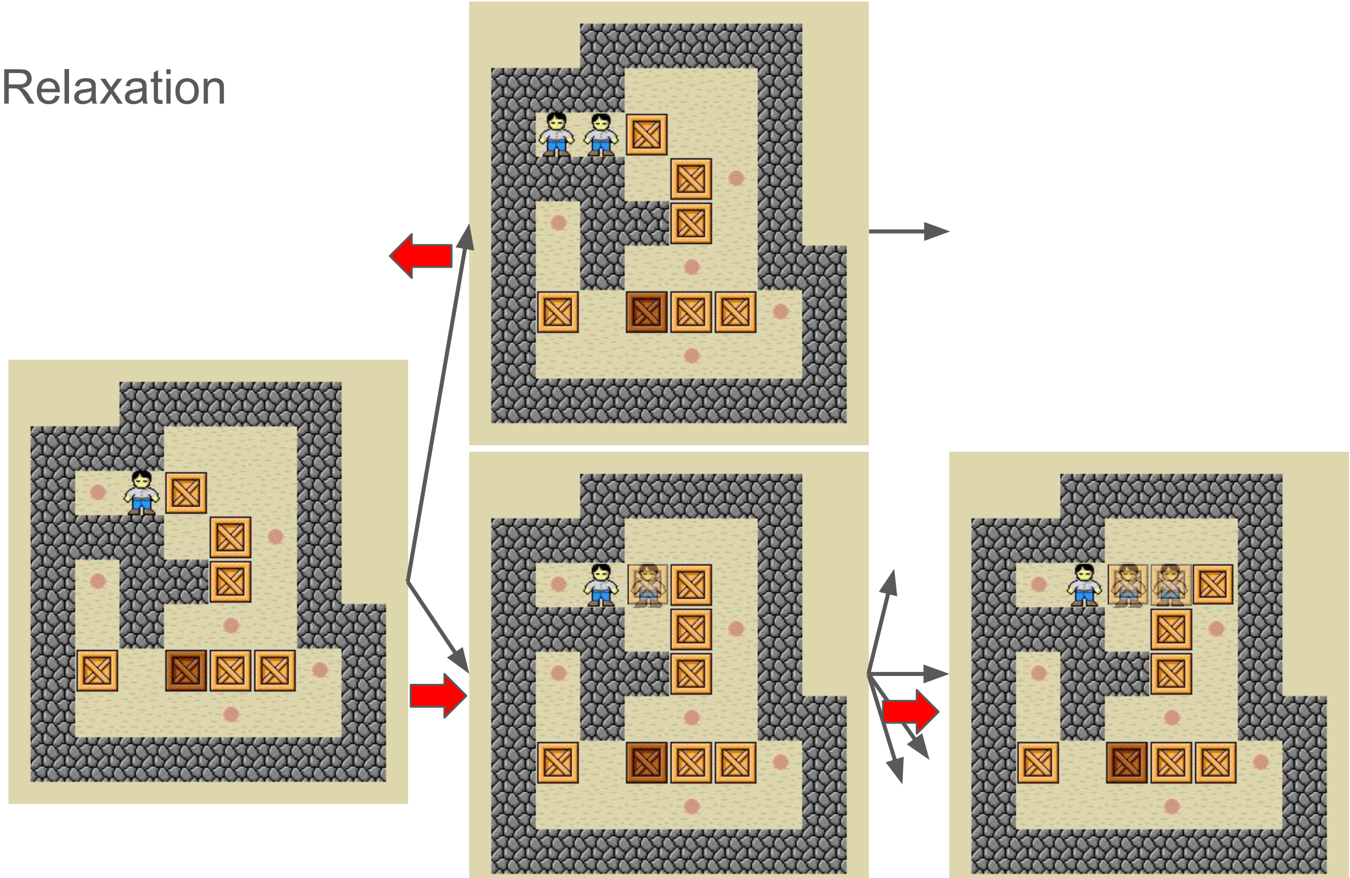
# Representation → Search + **Heuristics**

Relaxation



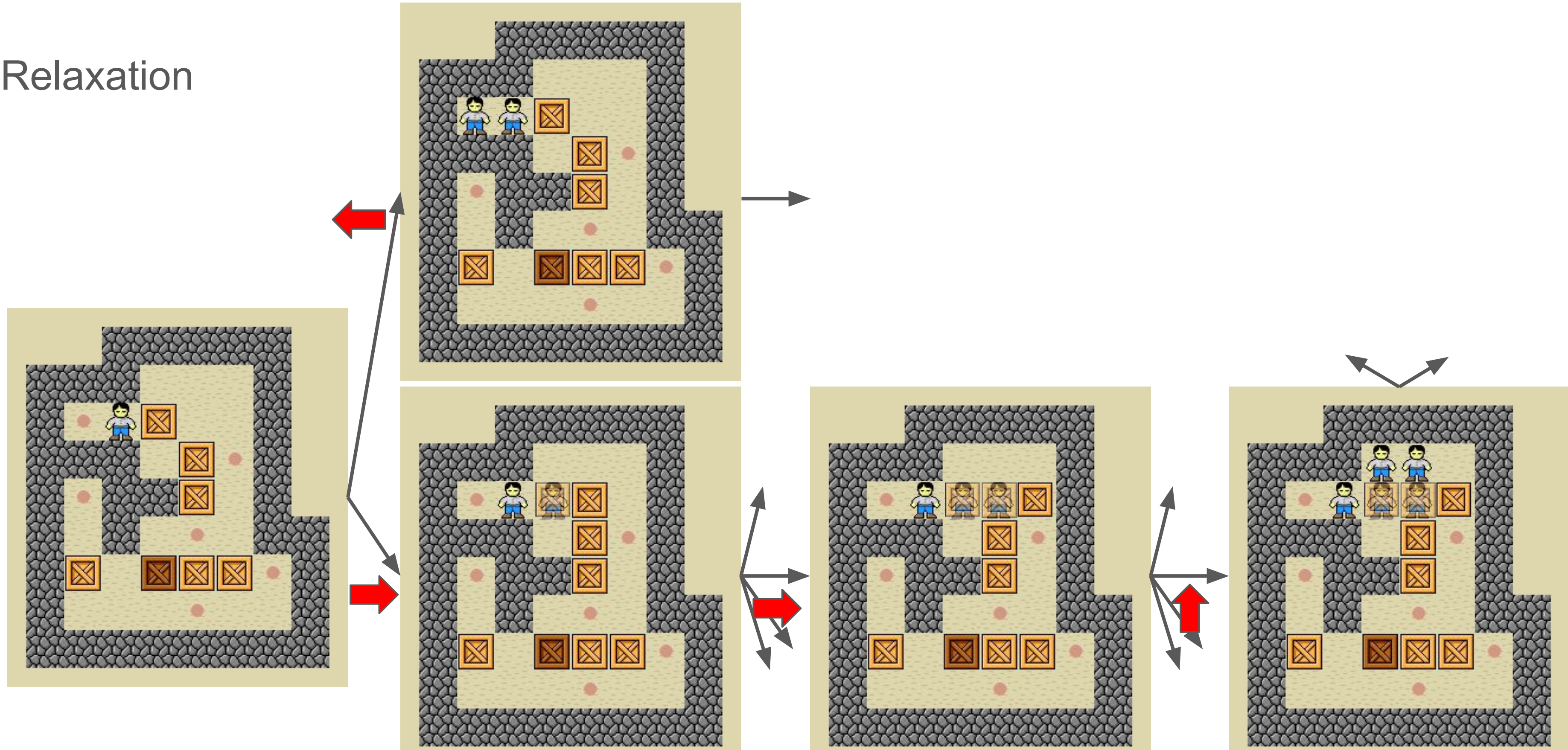
# Representation → Search + **Heuristics**

Relaxation



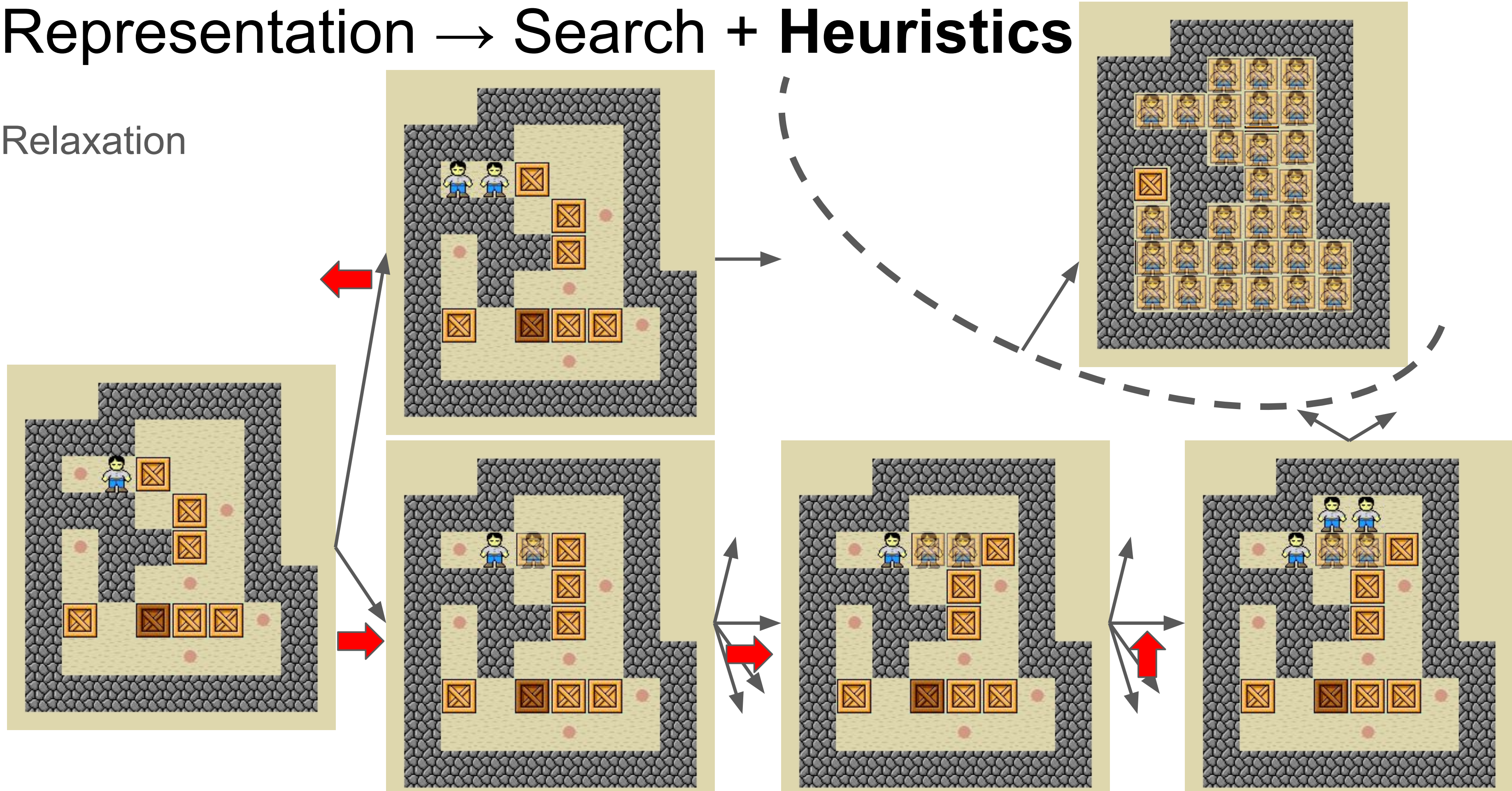
# Representation → Search + **Heuristics**

Relaxation



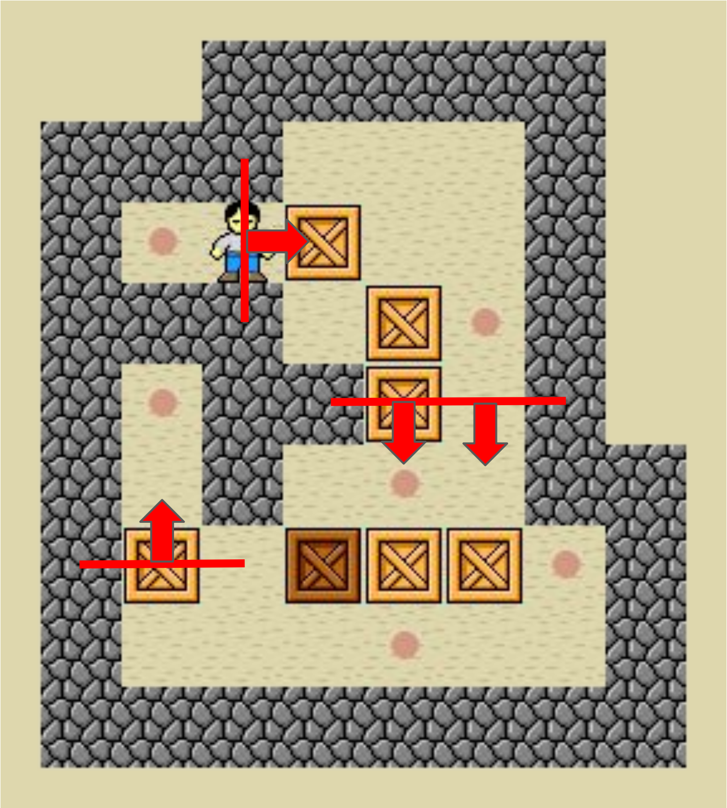
# Representation → Search + **Heuristics**

Relaxation



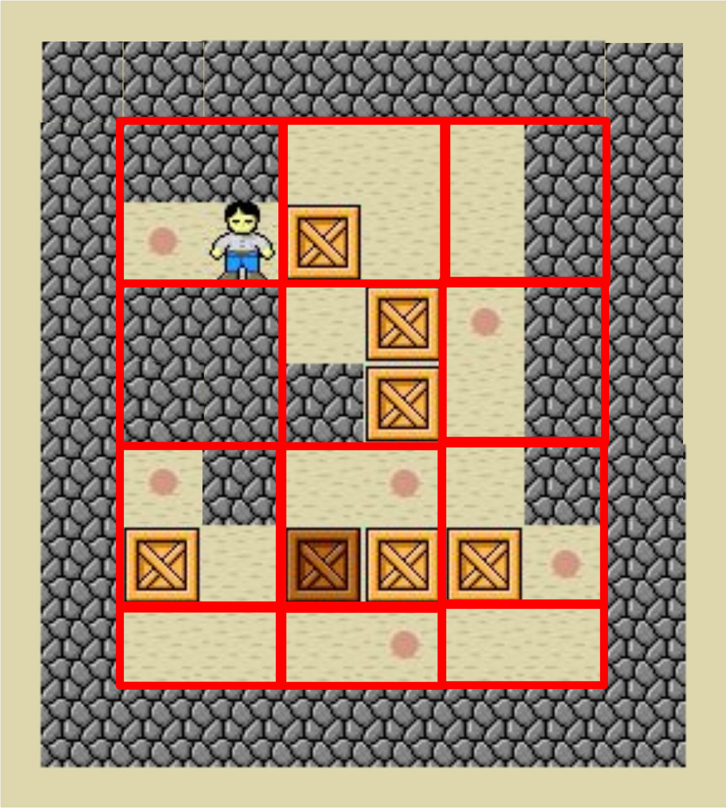
# Representation → Search + **Heuristics**

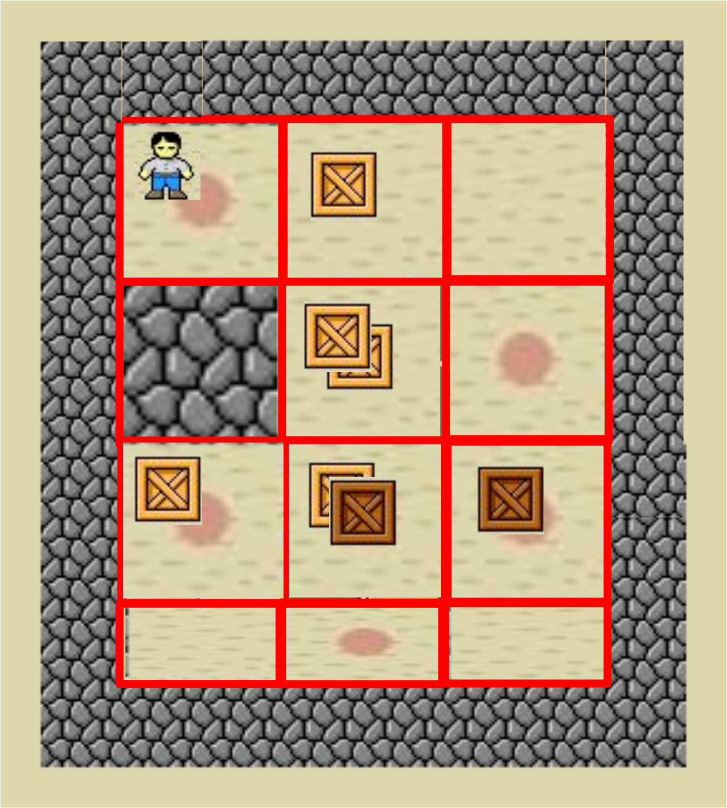
Landmarks



# Representation → Search + **Heuristics**

Abstraction





# Not Enough?

Are you trying to understand fundamentals of solving toy problems and puzzles?

# Not Enough?

Are you trying to understand fundamentals of solving toy problems and puzzles?

That’s not enough for me!

# Not Enough? Good.

- **domain-independent** ↔ domain-specific
- **off-line** ↔ on-line
- **deterministic** ↔ stochastic ↔ non-deterministic
- **fully-observable** ↔ partially-observable ↔ unobservable
- **instantaneous actions** ↔ durative actions
- **discrete** ↔ continuous fluents
- **linear** ↔ partially ordered/temporal
- **hard goals** ↔ soft goals
- **satisficing** (approximative) ↔ optimal
- **single-agent** ↔ multi-agent
- **plan** ↔ policy

# Not Enough? Good.

- domain-independent ↔ **domain-specific**
- off-line ↔ **on-line**
- deterministic ↔ **stochastic** ↔ **non-deterministic**
- fully-observable ↔ **partially-observable** ↔ **unobservable**
- instantaneous actions ↔ **durative actions**
- discrete ↔ **continuous fluents**
- linear ↔ **partially ordered/temporal**
- hard goals ↔ **soft goals**
- satisficing (approximative) ↔ **optimal**
- single-agent ↔ **multi-agent**
- plan ↔ **policy**

# Not Enough? Good. But Be Aware, Because ...

- domain-inde
- off-line ↔ **o**
- deterministic
- fully-observa
- instantaneou
- discrete ↔ **c**
- linear ↔ **pa**
- hard goals ↔
- satisficing (a
- single-agent ↔ **multi-agent**
- plan ↔ **policy**

# Automated Planning Elements (Recall)

- States
- Actions
- Initial state
- Goal states

# Automated Planning Elements (→ Real World)

- Agents
- States
- Observations
- Actions
- Transitions
- Costs
- Stochasticity
- Temporal, deontic, modal logics
- (Unknown) Initial state
- Common (Cumulative) Reward
- ...

# Automated Planning Elements (→ Real World)

- Agents
- States
- Observations
- Actions
- Transitions
- Costs
- Stochasticity
- Temporal, dec
- (Unknown) Ini
- Common (Cumulative) Reward
- ...

Challenge Accepted!