# Intelligent Agents
## Web-Mining Agents


Prof. Dr. Ralf Möller

Dr. Tanya Braun

Universität zu Lübeck

Institut für Informationssysteme

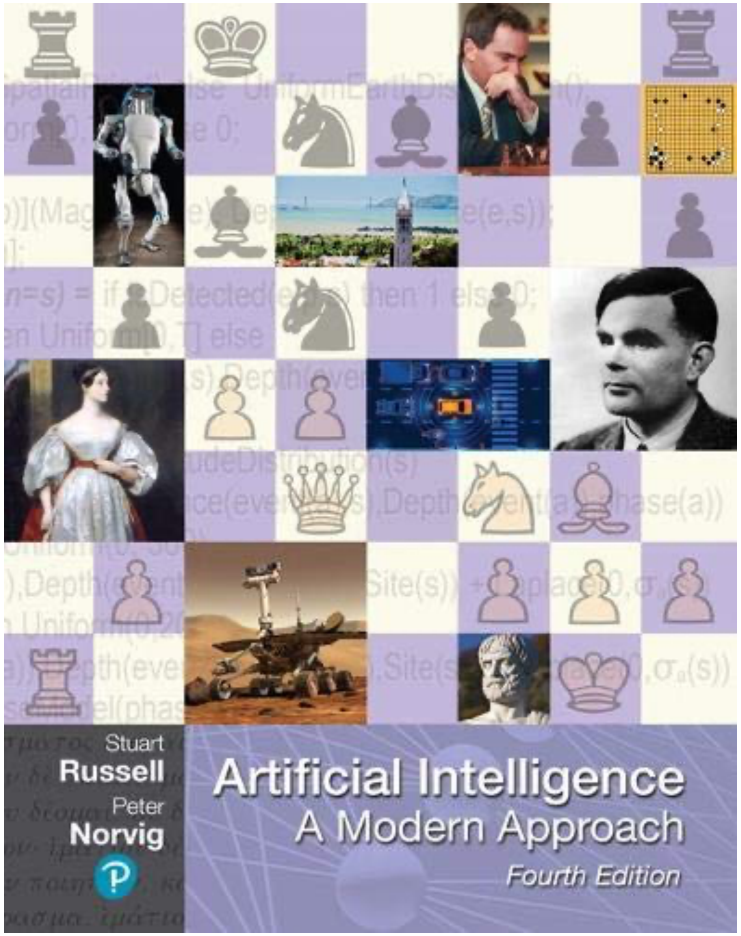
# Acknowledgements

- Some slides have been designed by PD Dr. Özgür Özcep. Since Özgür works in my group, those slides are not explicitly marked. Thank you nevertheless, Özgür.
- Other slides have been take from lecture material provided by researchers on the web. I hope this material is indicated appropriately. Thank you all.

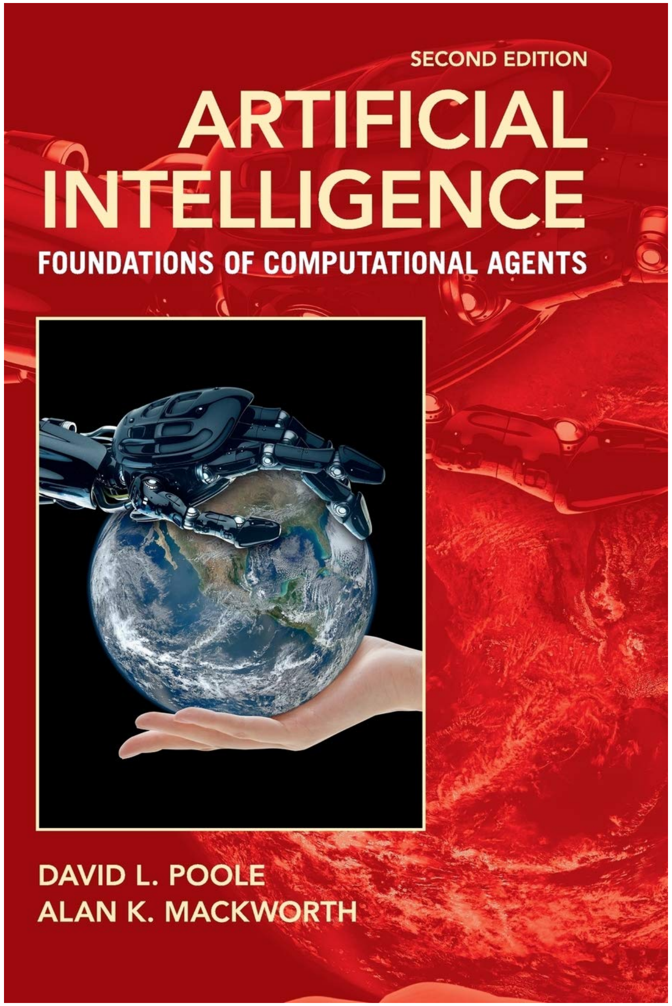
# Artificial Intelligence and Intelligent Agents

- Artificial intelligence (AI) is the science of systematic synthesis and analysis of computational agents that act intelligently
  - Agents are central to AI (and vice versa)
  - Intelligent agent = computational agent that acts intelligently
  - Talking about AI w/o talking about agents misses the point (and vice versa)
- Need to technically define the notion of "acting intelligently"
- AI = Science of Intelligent Systems
  - Systems are called computational agents in AI, or agents for short

# Literature

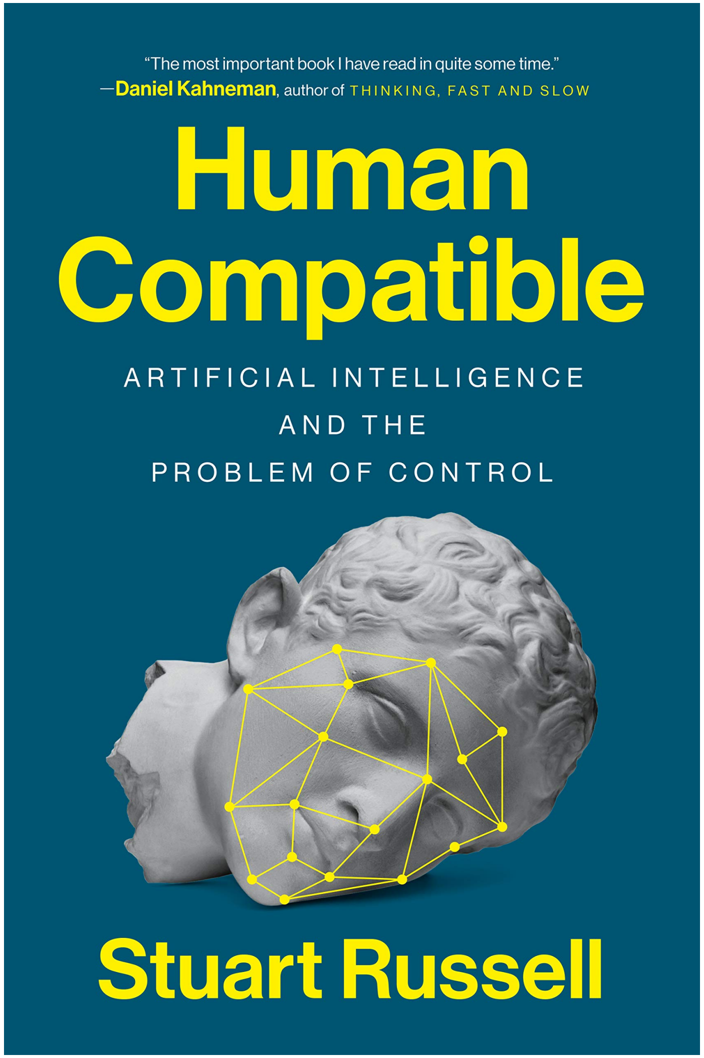


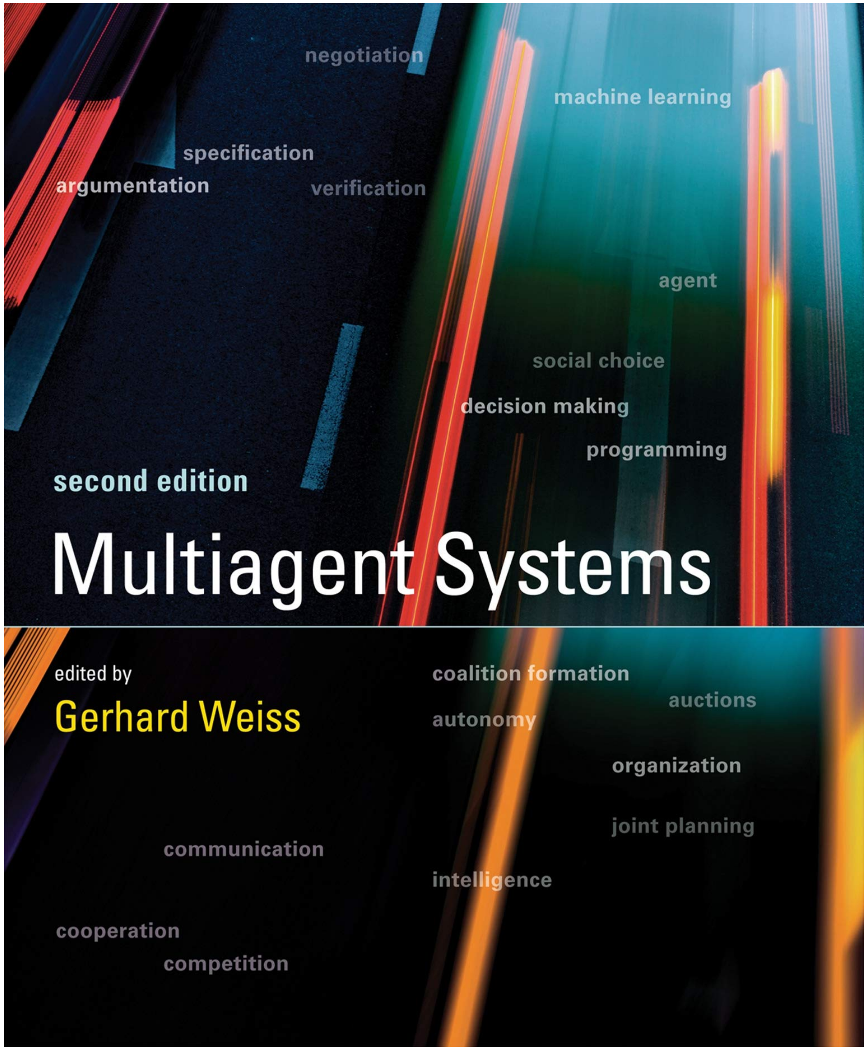

http://aima.cs.berkeley.edu
(AIMA, 1st edition 1995)



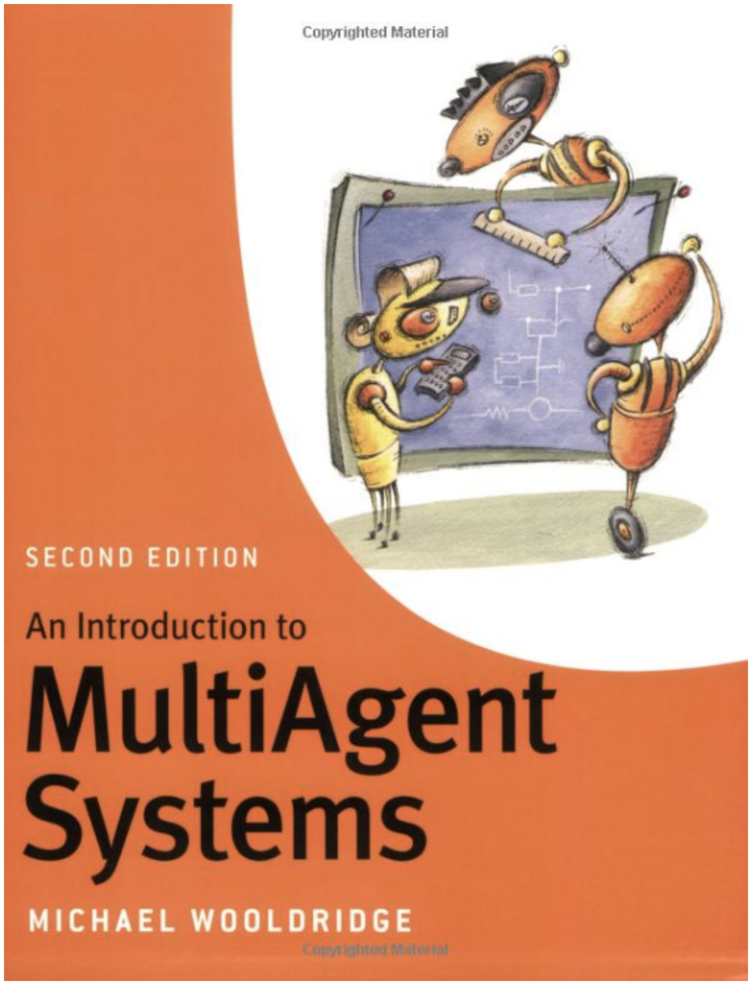

http://artint.info
(AIFCA, 1st edition 2010)

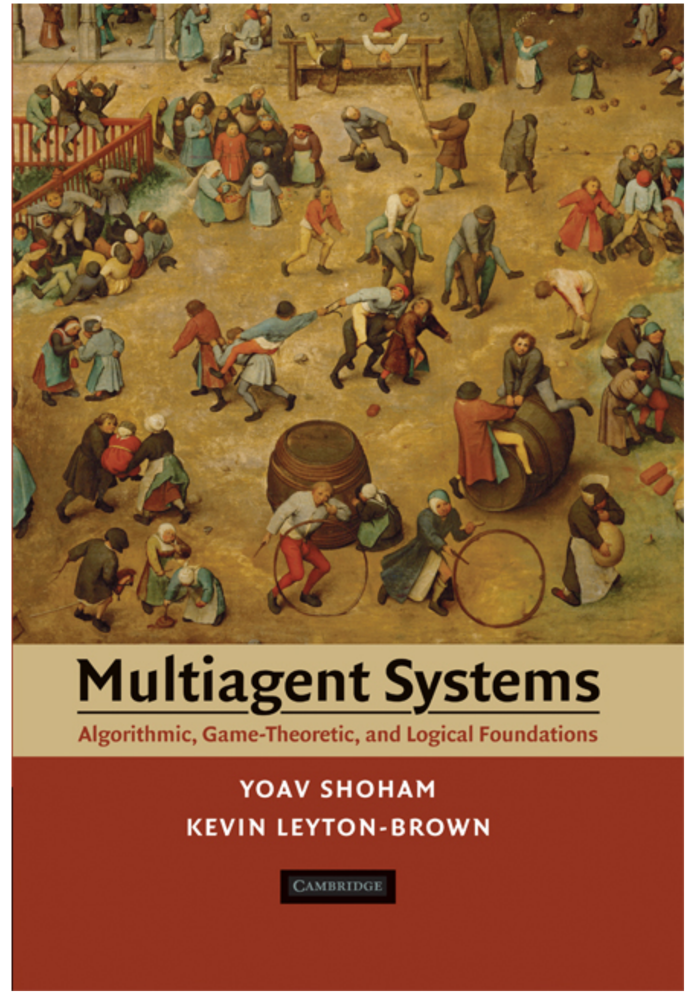
# Literature


"The most important book I have read in quite some time."
—Daniel Kahneman, author of THINKING, FAST AND SLOW
Human Compatible
ARTIFICIAL INTELLIGENCE AND THE PROBLEM OF CONTROL
Stuart Russell


negotiation
machine learning
specification
argumentation
verification
agent
social choice
decision making
programming
second edition
Multiagent Systems
edited by
Gerhard Weiss
coalition formation
auctions
autonomy
organization
joint planning
communication
intelligence
cooperation
competition

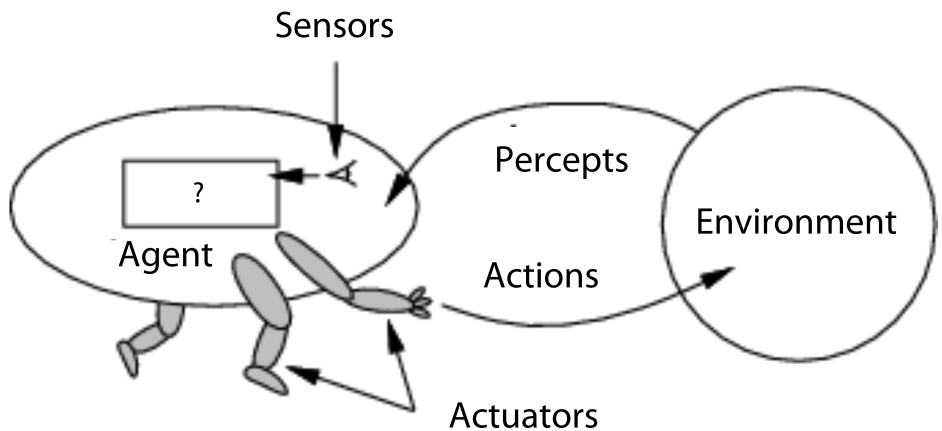
# Literature


Copyrighted Material
SECOND EDITION
An Introduction to
MultiAgent
Systems
MICHAEL WOOLDRIDGE
Copyrighted Material


Multiagent Systems
Algorithmic, Game-Theoretic, and Logical Foundations
YOAV SHOHAM
KEVIN LEYTON-BROWN
CAMBRIDGE

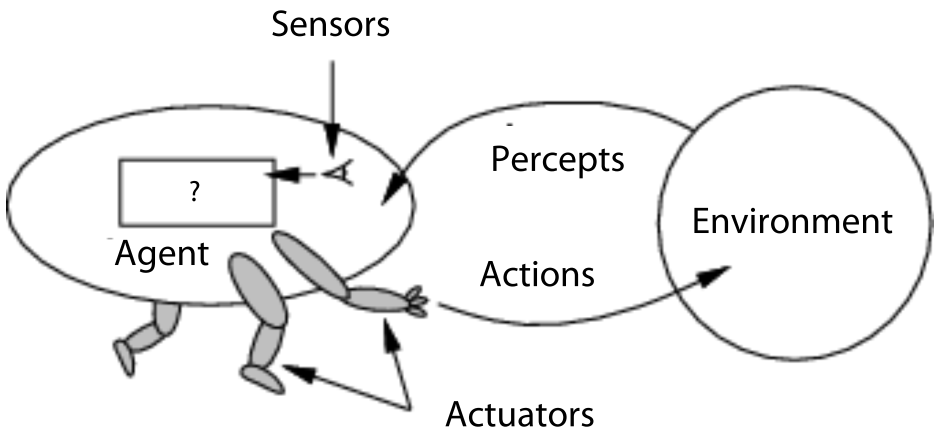
# What is an Agent?

- Anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators
  [AIMA-Def]




  - Human agent
    eyes, ears, and other
    organs for sensors; hands, legs, mouth, and other body parts for actuators
  - Robotic agent
    cameras and infrared range finders for sensors; various motors for actuators
  - Software agent
    interfaces, data integration, interpretation, data manipulation/output

# Abstractions: Agents and Environments



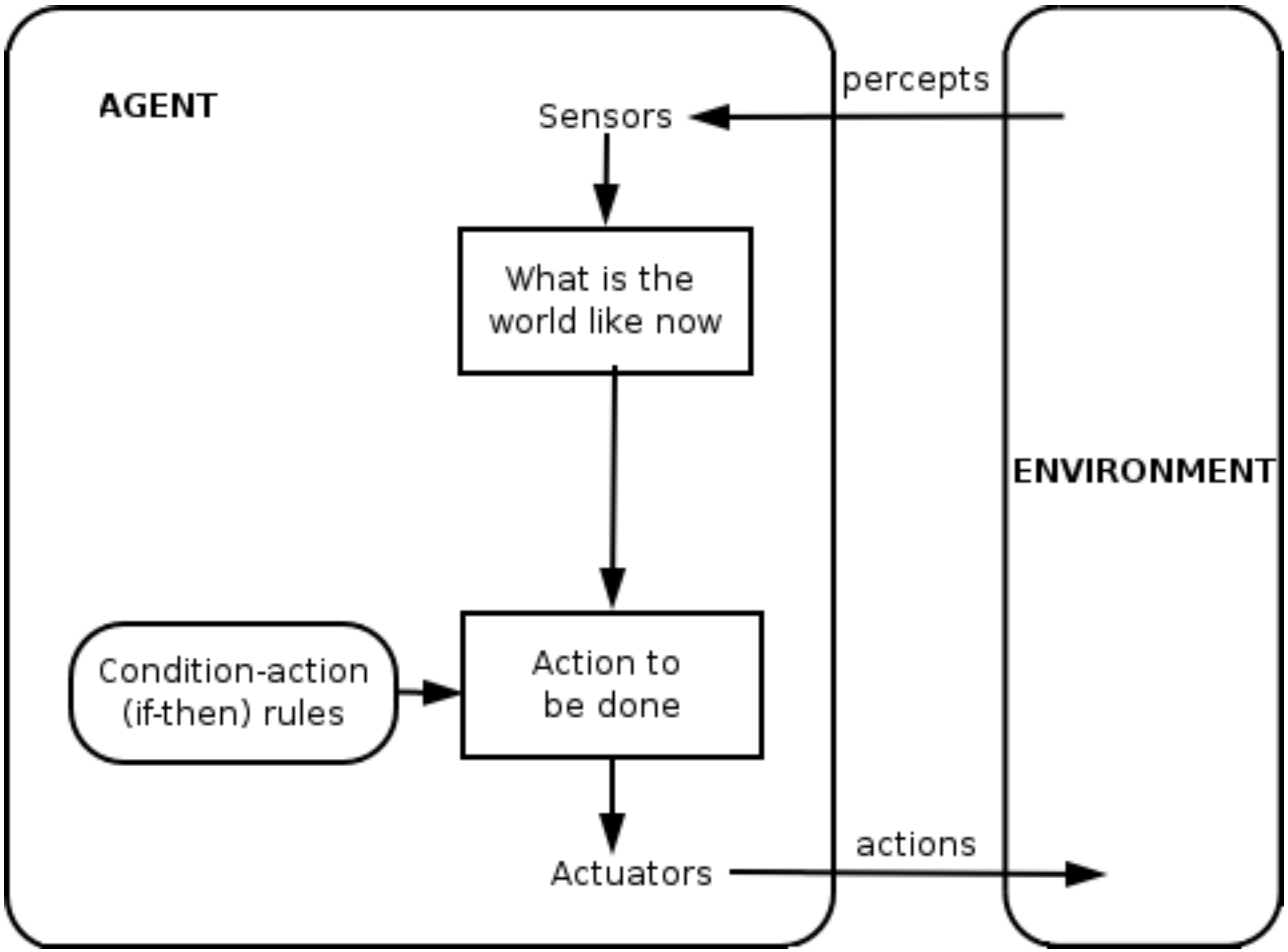

- The agent function maps from percept histories to actions:

$$[f: P^* \rightarrow A]$$

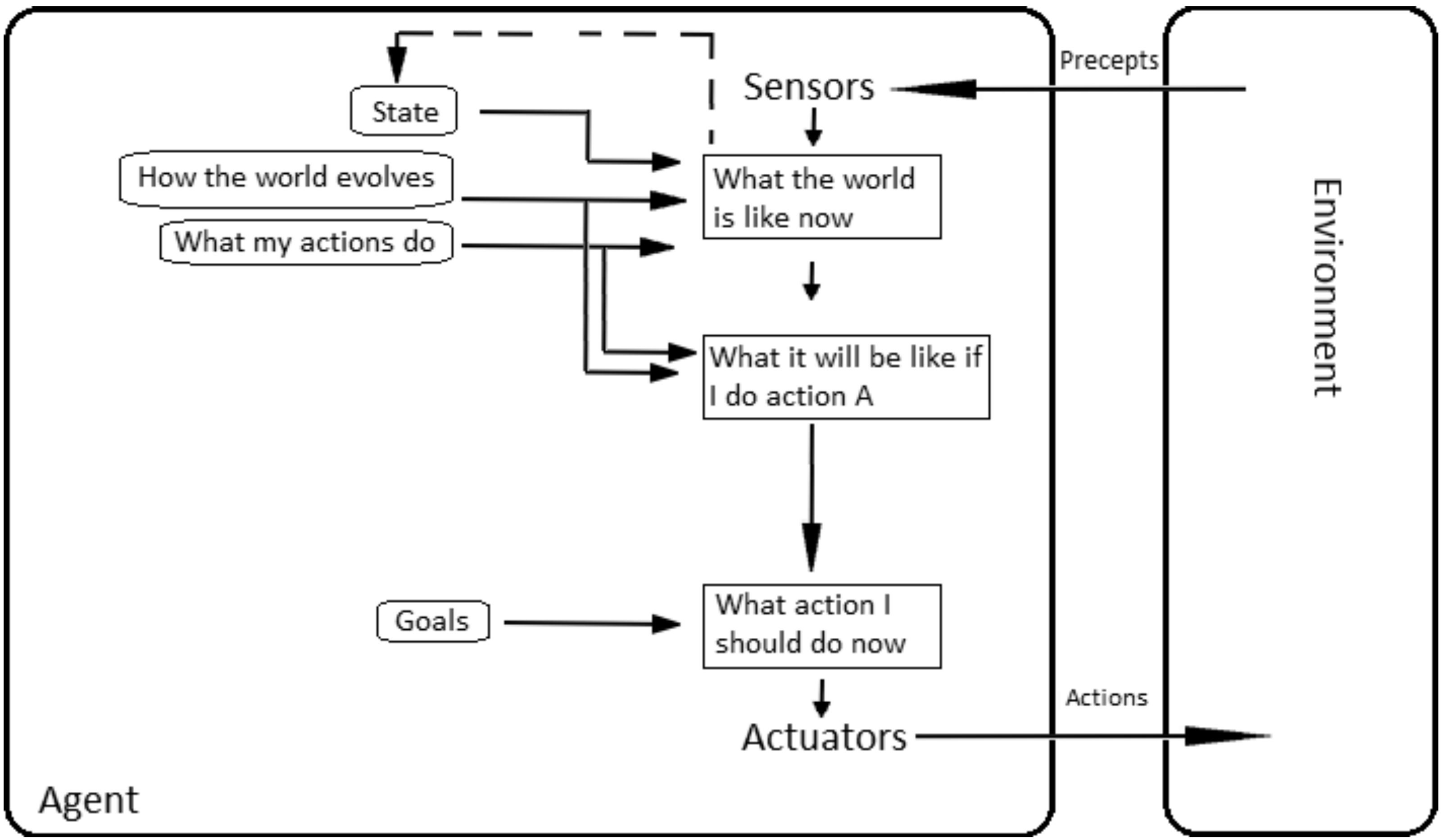
- The agent program runs on a physical architecture to produce $f$
- Agent = architecture + program

Really insist on functional behavior?

# *Reactive* vs. Goal-based Agents



# Reactive vs. *Goal-based* Agents



# Balancing Reactive and Goal-Oriented Behavior

- We want our agents to be reactive, responding to changing conditions in an appropriate fashion (e.g., timely)
- We want our agents to systematically work towards long-term goals
- These two considerations can be at odds with one another
  - Designing an agent that can balance the two remains an open research problem
  - Achieve maximum freedom of action if there is no specific short-term goal (e.g., keep batteries charged)

# Social Ability

- The real world is a multi-agent environment: we cannot go around attempting to achieve goals without taking others into account
- Some goals can only be achieved with the cooperation of others
- Social ability in agents is the ability to interact with other agents (and possibly humans) via some kind of agent-communication language …
- ... with the goal to let other agents to make commitments (of others) or reinforcements (about its own behavior)
- Need to represent and reason about beliefs about other agents

# Rational Agents

- Rational Agent: For each possible percept sequence, a rational agent
  - should select an action
  - that is expected to maximize its local performance measure,
  - given the evidence provided by the percept sequence and
  - whatever built-in knowledge the agent has
- Rational = Intelligent ?
  - There is more to intelligence than meets rationality

# Autonomous Agents

- Rationality is distinct from omniscience (all-knowing with infinite knowledge)
- Computing the best action usually intractable
- Rationality is bounded
- Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration)
- An agent is autonomous if its behavior is determined by its own "experience" (with ability to learn and adapt)
  - What matters for the "experience" is the
    - percept sequence (which the agents can determine), the
    - state representation, and the
    - "computational success" of computing the best action as well as learning and adapting for the future

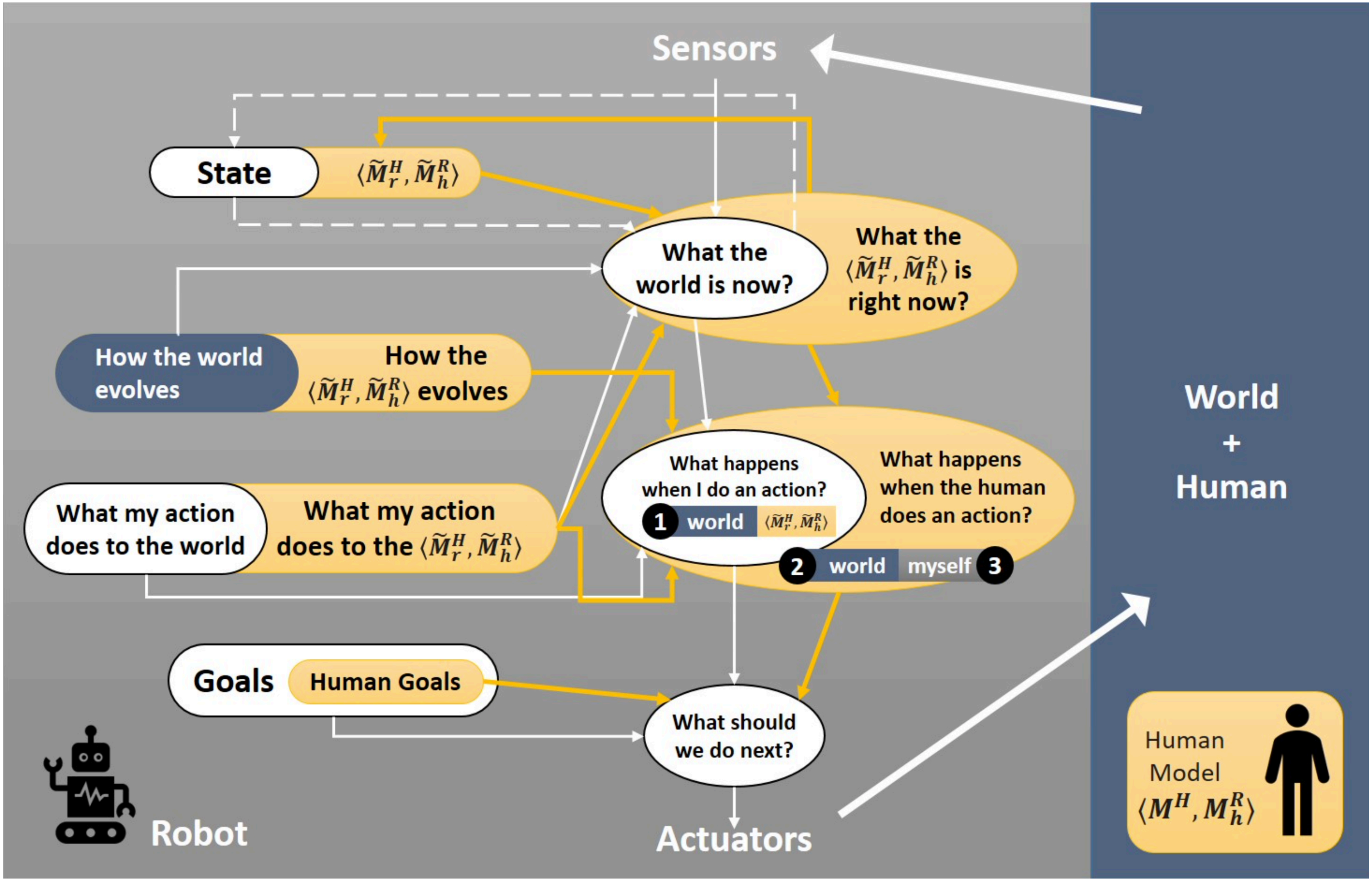
# Human-compatible Behavior

- Agents act on behalf of humans, who specify the goals
- Agent should consider its initial goals to be uncertain (and to be challenged due to underspecification)
- Agent should be able to prove their behavior is beneficial to humans
- Artificial intelligence, agents, and ethics
  - Agents must act in an ethical way (and those, who develop agents, possibly too)
  - Developers should be able to prove …
    … that agents are able to prove
    … that they (the agents) act in an ethical way
  - Simple technology assessment is not enough

# Human-aware Behavior

- Agents interact with humans
- Selected actions must match human expectations
  - Maybe the presumably expected action might not be the best (for the human or the agent, or both)
- Selected actions that are assumed to not match human expectations must be explained

# Agent Model vs. Human Model

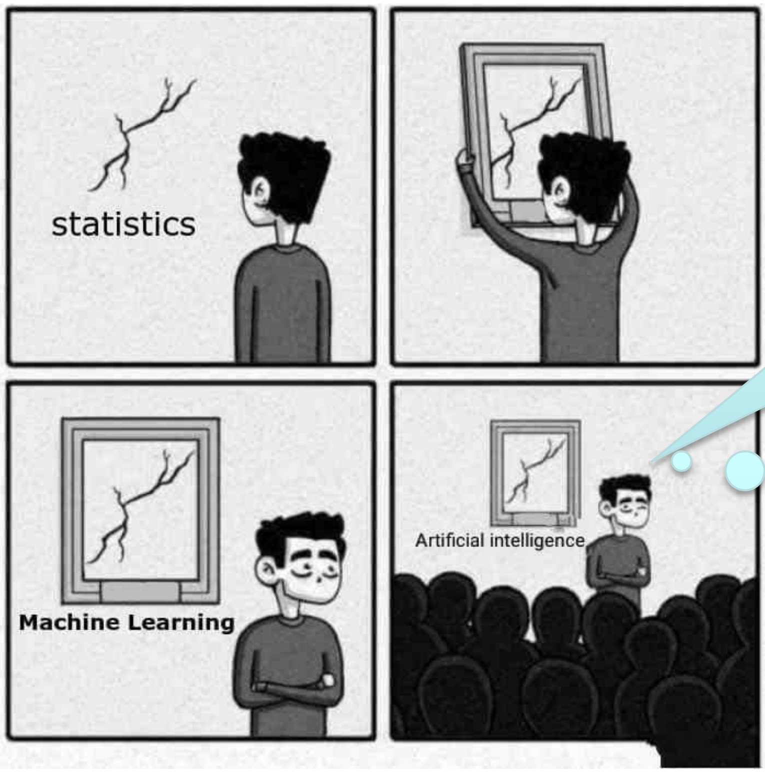


# Learning Agents (Online)

- Ever extented percept sequence (incl. more or less explicitly encoded reinforcement feedback or rewards) is …
  - … sparse (no big data), but gives rise to model updates
  - … with the aim to better (faster) achieve goals
- We say: Agents learn (and we mean: while acting, or online)
  - Optimize a performance measure
- Setting up agents' online learning engines
  - Dedicated knowledge about online learning required
- Setting up an agent's initial model by exploiting data:
  - Dedicated knowledge of machine learning required
  - Also basically optimizing a performance measure

# Machine Learning (ML): Offline

- Statistics vs. Data Science vs. Machine Learning

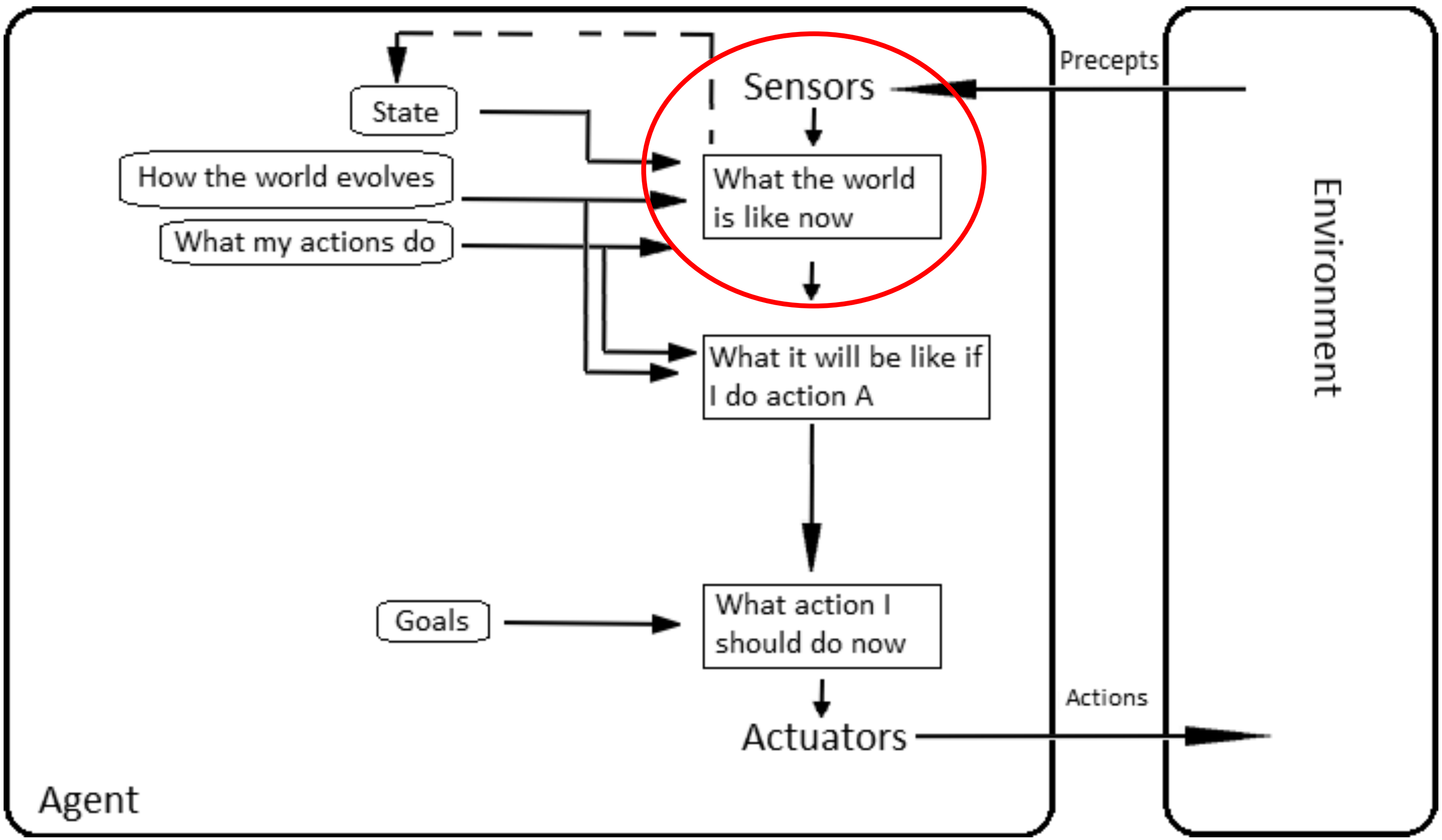



*"When you're fundraising, it's AI. When you're hiring, it's ML. When you're implementing, it's logistic regression."*

It is clear that claiming AI ***is*** machine learning or "contains" machine learning does not make much sense!

- Machine learning scales, but can only do so much
- All fields have evolved and still do evolve

# Reactive vs. *Goal-based* Agents



# Misunderstandings

- Applying ML to implement a function f some people say:
  “I have used ML technique X to create an AI”
- Unconsciously, AI is used as a synonym for agent, but …
  … mostly a very simple one
  - $f: P \rightarrow A$
- Claiming that f is “an AI” is an indication of lack of understanding …
- … even if the last n percepts are considered
  - $f: P \times \ldots \times P \rightarrow A$
- One is lost w/o an understanding of intelligent agents
  - $f: P^* \rightarrow A$

# Frame Agents

- Assume that machine learning techniques are used to build models at agent setup time
- Runtime behavior of agent always depends on last n elements of percept sequence only $f: P \times \ldots \times P \rightarrow A$
- No interaction w/ environment, no feedback
- Agent is fake (simply a frame around standard SW/HW)
  - Also holds when setup training data is camouflaged as initial percepts (but no actions towards goals are computed until training completed)
- Maybe even enlightening for practical applications, but agent idea …
- … does not show its full potential

# Learning-based Software Development

- There is no need to deliberately conflate machine learning with agents and AI !
- No need to invent frame agents !
- Can build extremely cool SW/HW w/ machine learning techniques (e.g., for industrial image processing applications)

- → Probabilistic Differential Programming (CS5071-KP04)
- → Deep Learning Lab (CS5071-KP04)

- There are caveats, however:

Artificial intelligence / Machine learning

# Training a single AI model can emit as much carbon as five cars in their lifetimes

Deep learning has a terrible carbon footprint.

by **Karen Hao**

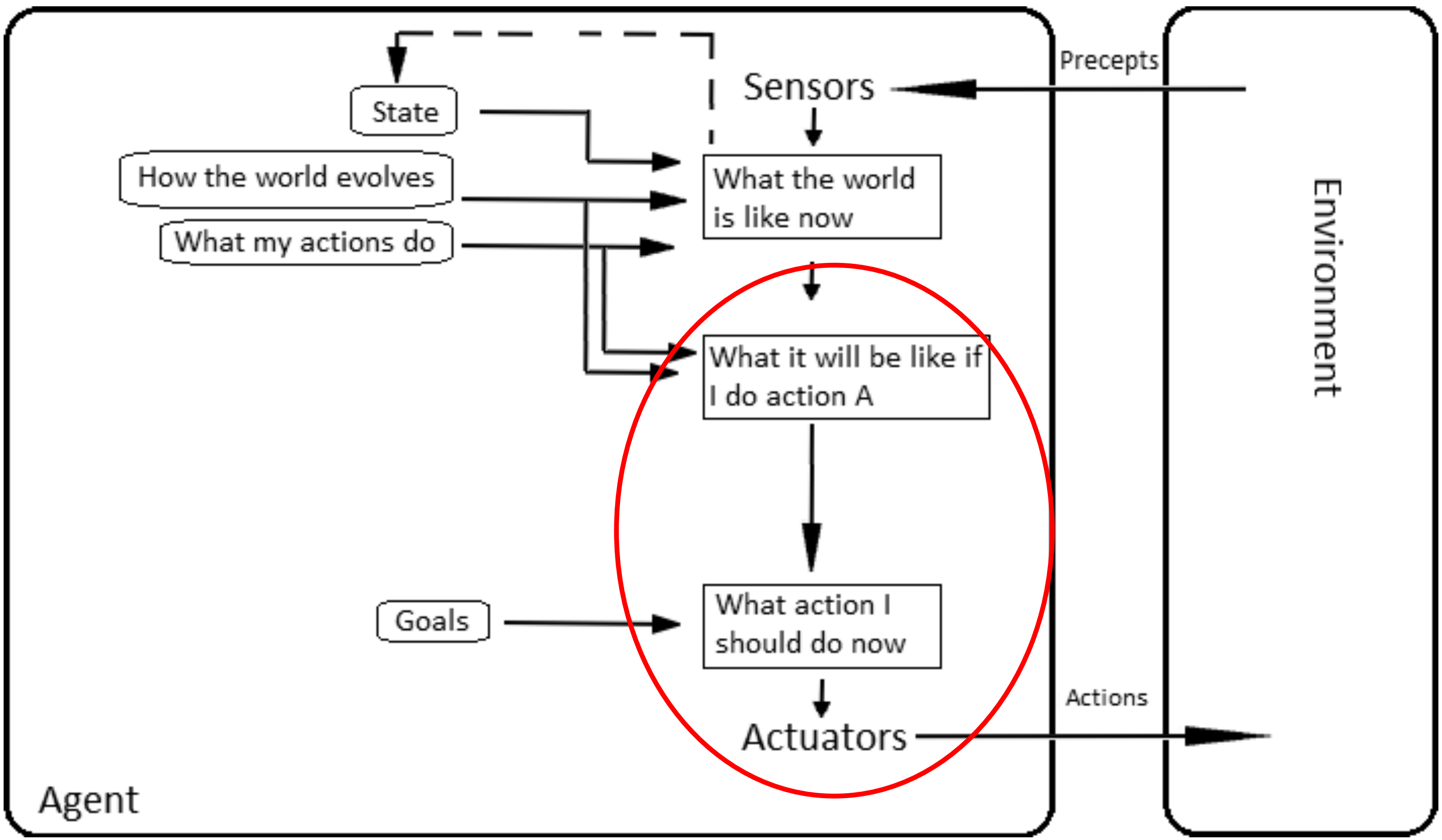
June 6, 2019

# Back to the Future: Human-guided Learning

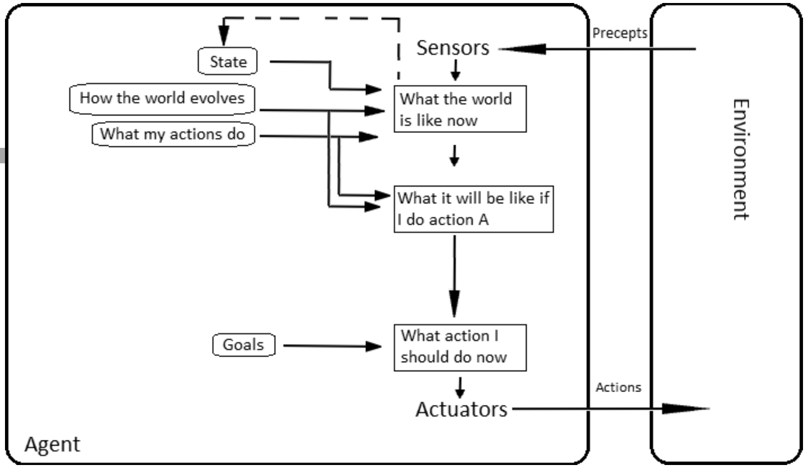
- Develop machine learning techniques that achieve good performace w/o too much training material
- Exploit human capabilities
- Artificial agents and human agents cooperate
- Machine learning becomes agent online learning
  - Motivation for studying agents!
  - Machine learning cannot go w/o agents in the future

- Agents allow for more or less learning (incl. no learning)
- Next: Proper agent with no learning

# Proper Agent: An Example



# Proper Agent: An Example




Given:

- Current state of the environment
- Description of goal state
- Set of action descriptions

→ Find sequence of actions (a plan) for transforming current state into goal state

→ Select first action, and hope that plan can be completed

# STRIPS Formalism

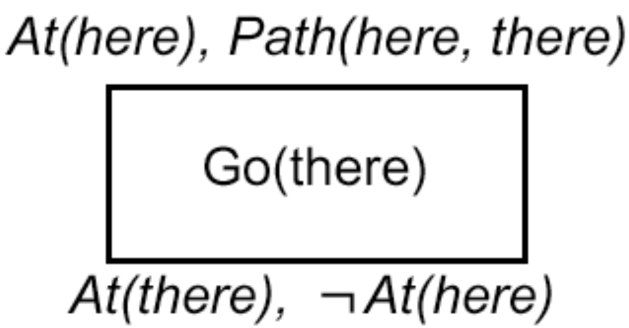
- States modeled as set of ground atoms (database)
  - Current state as well as goal state
  - Example: Blocks World
    - On_Table(A), On_Table(B), On_Table(C)
    - On_Block(C, B), On_Block(B, A)



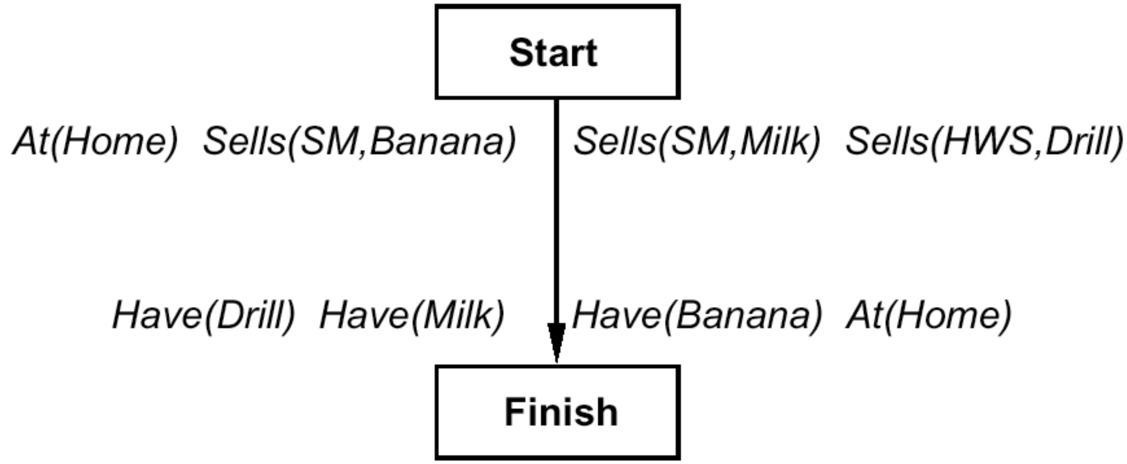
STRIPS = Stanford Research Institute Problem Solver (1971)

# STRIPS Planning Operators

$$
\begin{aligned}
&Op(\text{Action: } Go(there),\\
&\quad \text{Precond: } At(here) \land Path(here, there),\\
&\quad \text{Effect: } At(there) \land \neg At(here))
\end{aligned}
$$



# Complete Plan




**Aktionen:**

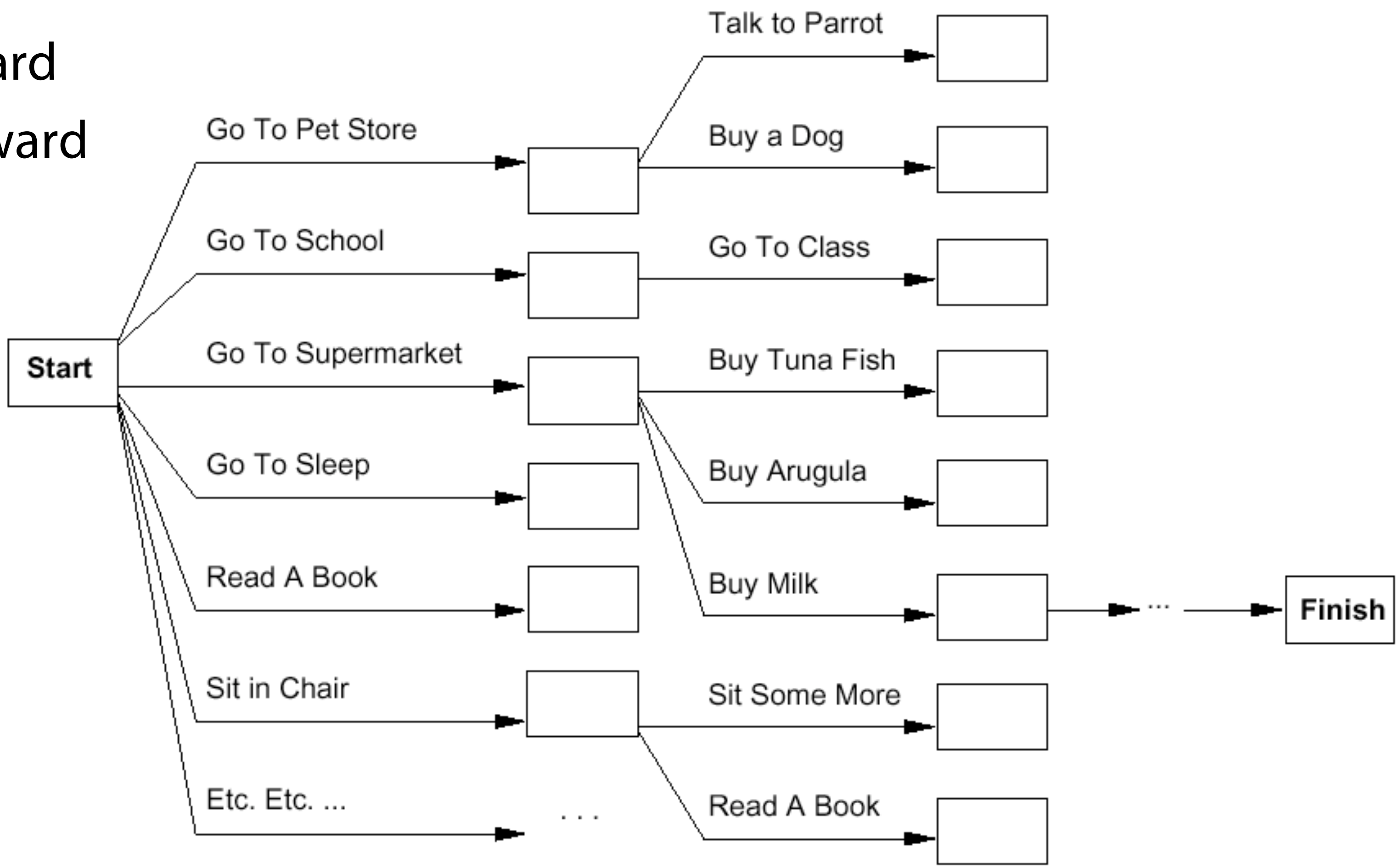
*Op* (Action: *go(there)*,
Precond: *At(here)*,
Effect: $At(there) \wedge \neg At(here)$)

*Op* (Action: *buy(x)*,
Precond: $At(store) \wedge Sells(store,x)$,
Effect: *Have(x)*)

there, here, x, store are variables

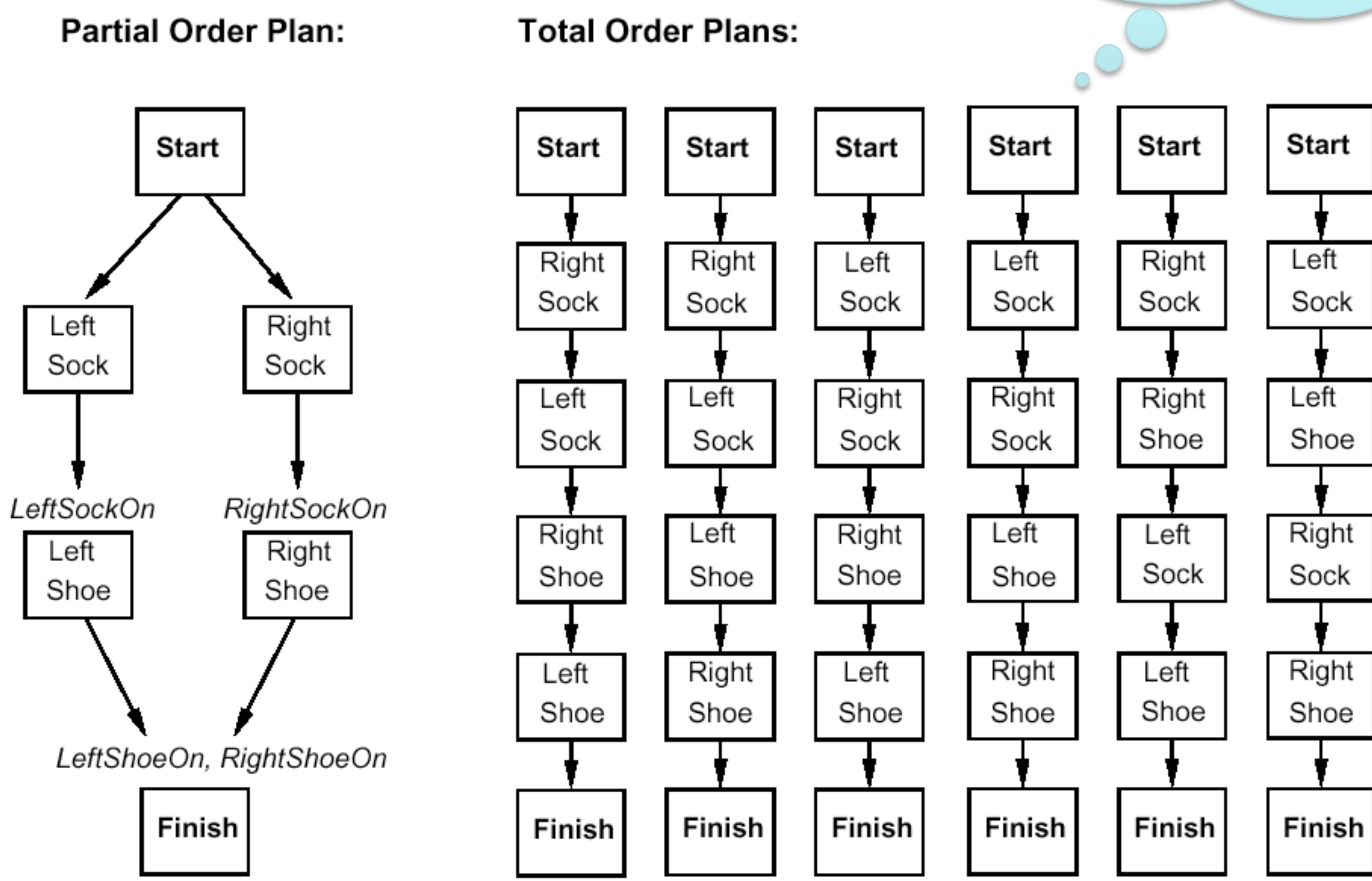
# Planning as Search

- Forward
- Backward



# Plan = Sequence of Actions?

Apply principle of
Least Commitment



# Representation of Partial-Order Plans

- Plan step = STRIPS Operator
- Plan consists of
  - Plan step with partial order (<), where $S_i < S_j$ iff $S_i$ is to be executed before $S_j$
  - Set of variable assignments $x = t$ , where $x$ is a variable and $t$ is a constant or variable
  - Set of causal relations: $S_i \rightarrow^c S_j$ means $S_i$ creates the precondition $c$ of $S_j$ (implies $S_i < S_j$)

- Solutions to planning problems …
  … must satisfy certain conditions



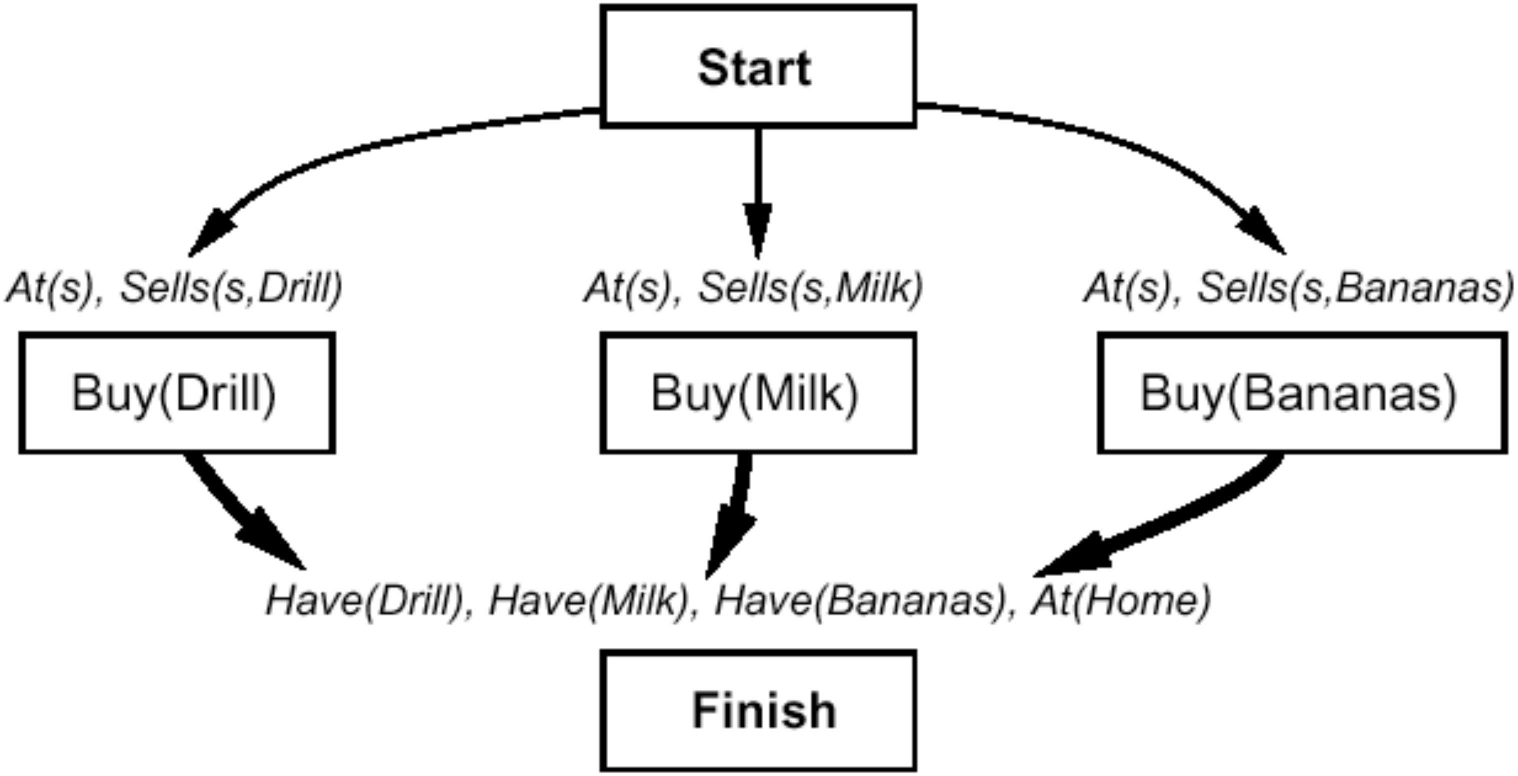
STRIPS = Stanford Research Institute Problem Solver (1971)

# Completeness and Consistency

- Complete plan
  - Every precondition of a step is fulfilled
  - $\forall S_j$ with $c \in Precond(S_j)$,
    - $\exists S_i$ s.t. $S_i < S_j$ and $c \in Effects(S_i)$, and
    - for every linearization it holds that:
      - $\forall S_k$ with $S_i < S_k < S_j$, $\neg c \notin Effects(S_k)$
- Consistent plan
  - If $S_i < S_j$, then $S_j \not< S_i$ and
  - If $x = A$, then $x \neq B$ for different $A$ and $B$ for variable $x$ (Unique Names Assumption)
- Solution of the planning problem: complete and consistent plan

# Plan Refinement (1)

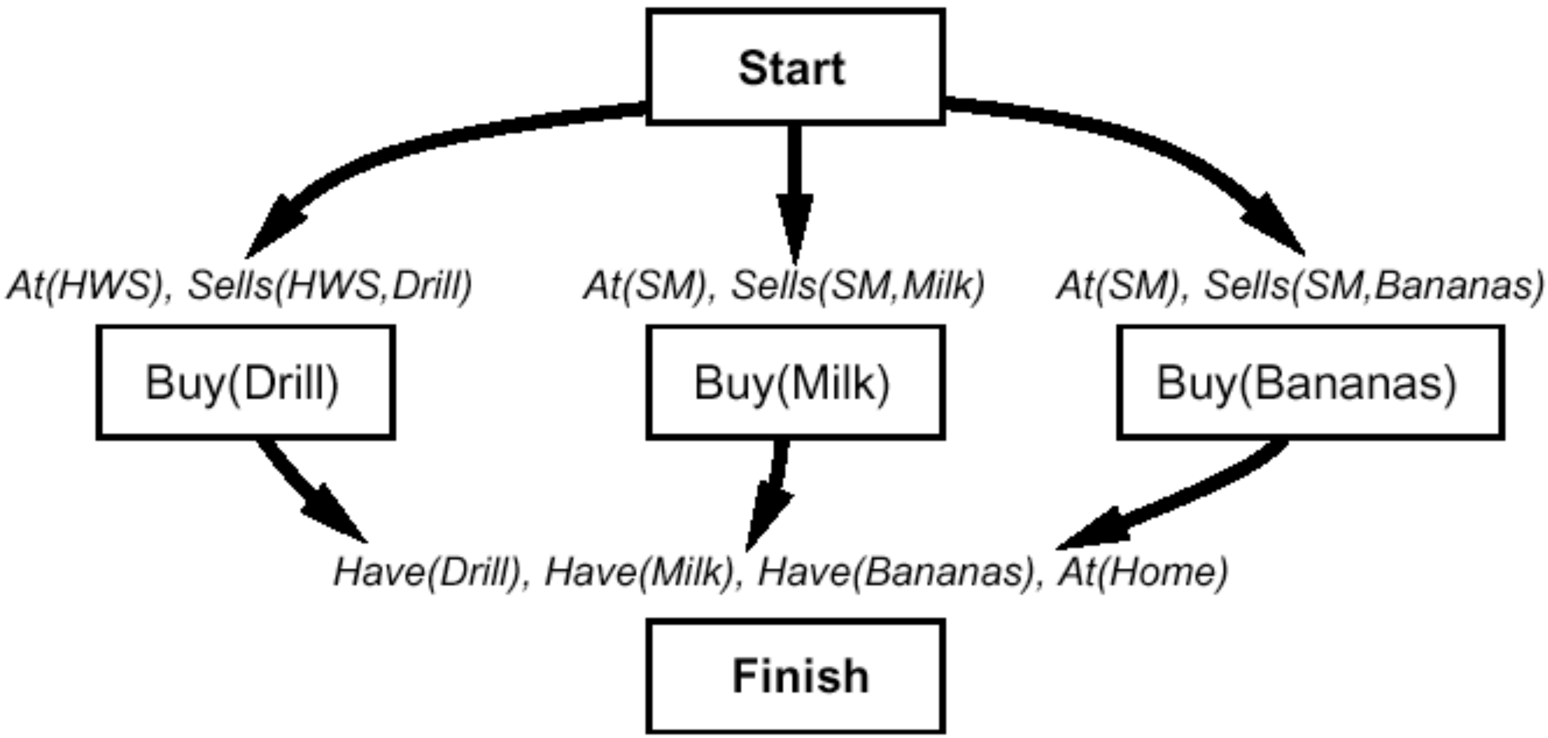
- Backward planning




Thin arrows = <

Fat arrows = causal relation + <

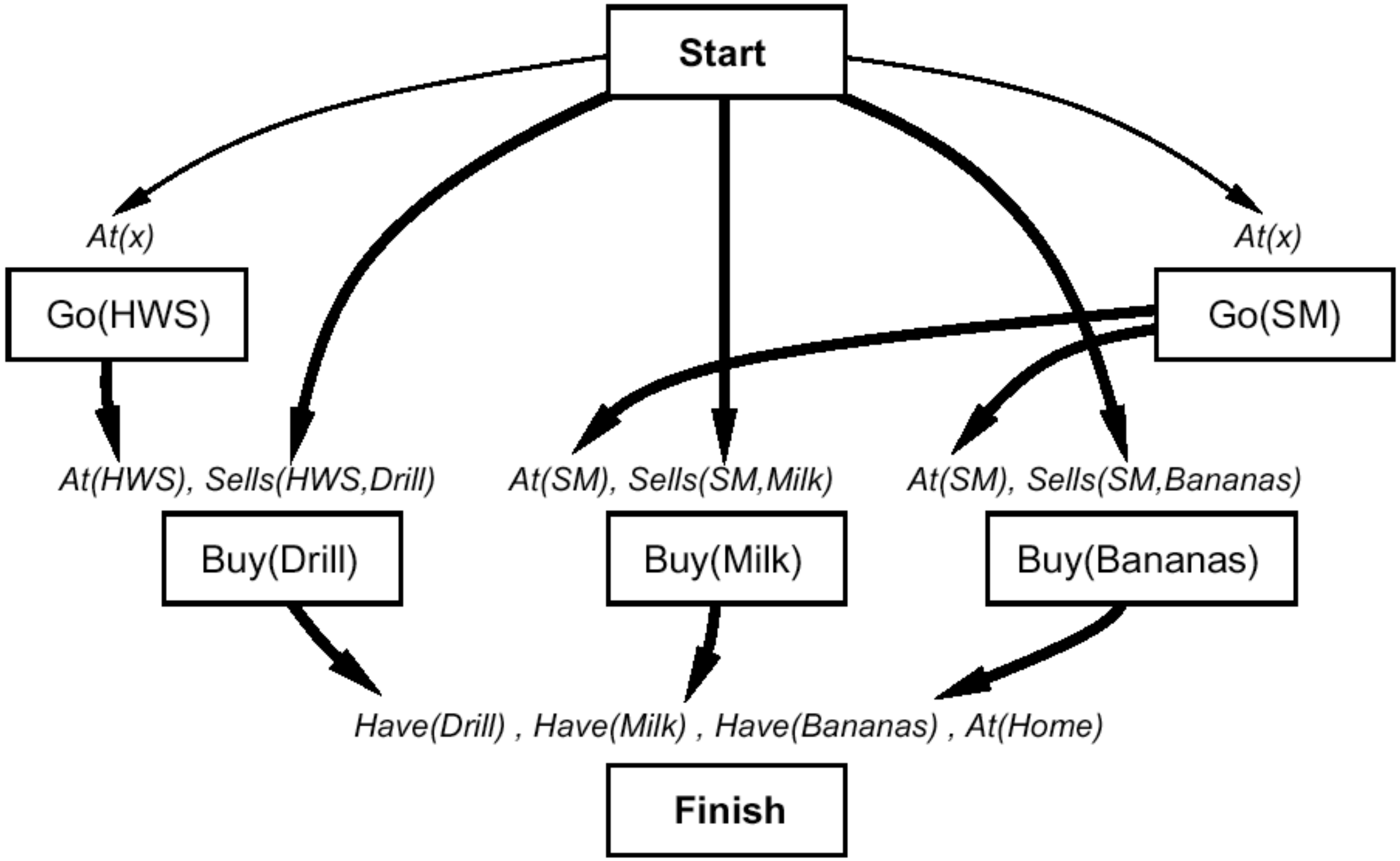
# Plan Refinement (1)




. . . after variable instantiation

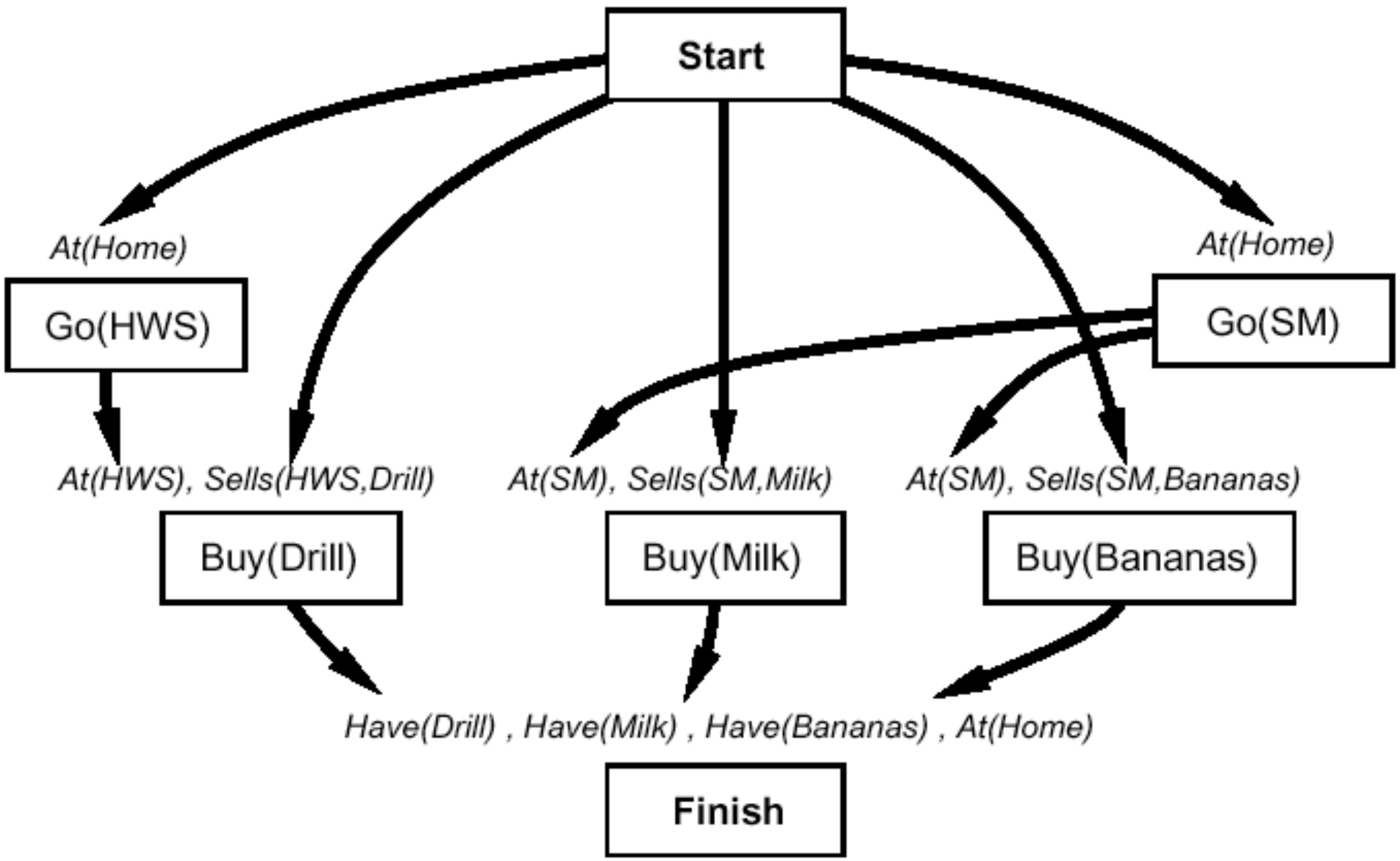
# Plan Refinement (2)

- ... buy at the right store



# Plan Refinement (3)

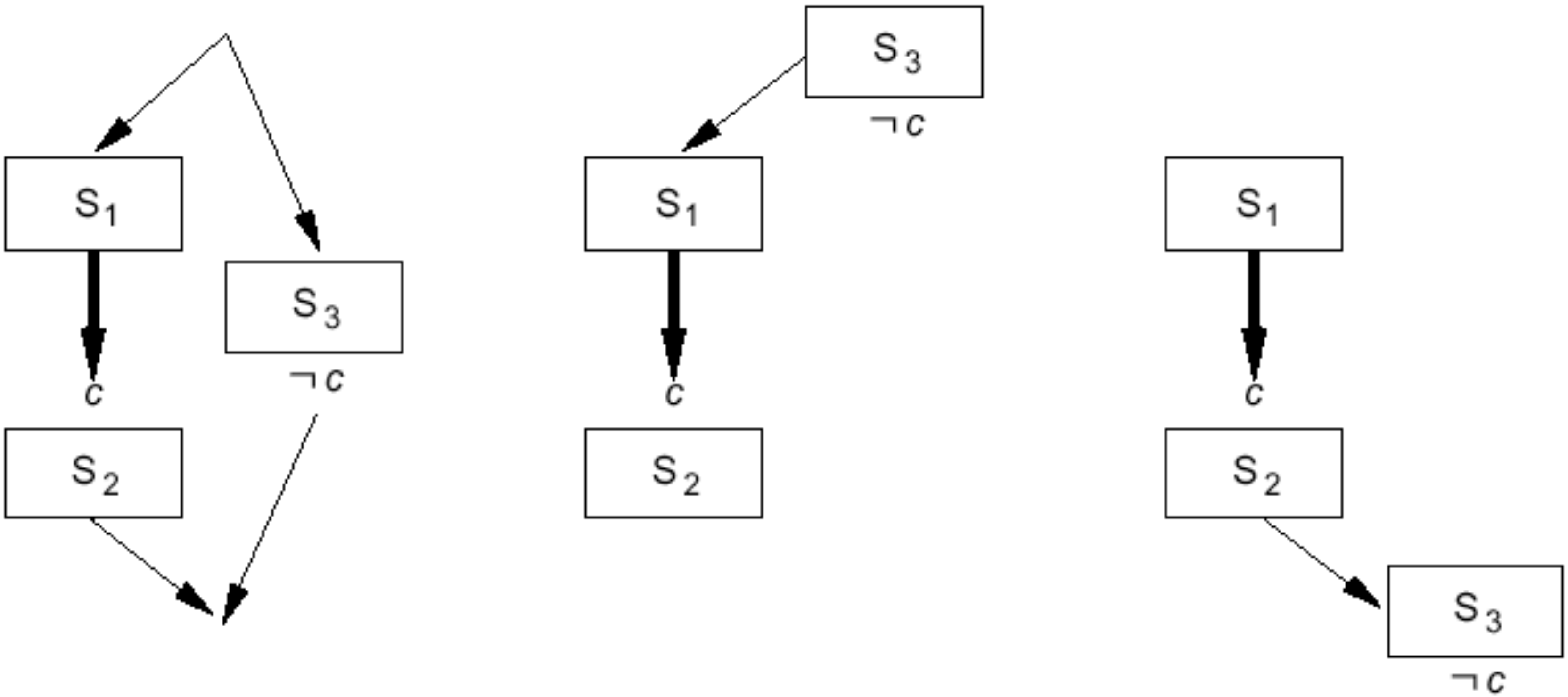
- ... but you must get there



# Plan Refinement (3)

- Note:
  Up to now no search, but simple „backward chaining“
- Now:
  Conflict! After go(HWS) is executed, At(Home) no longer holds (similarly for go(SM))

# Protection of Causal Relations




(a) (b) (c)

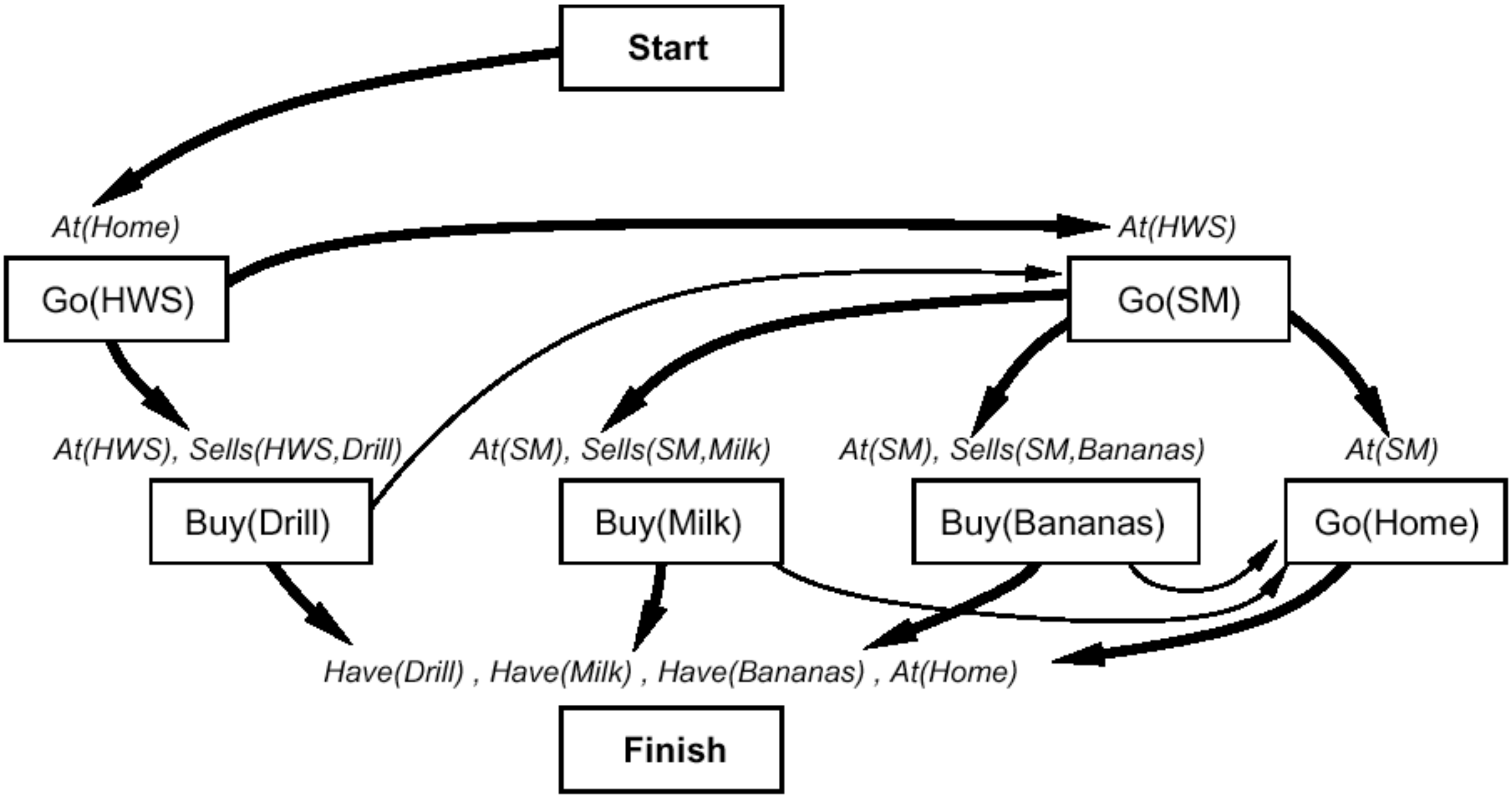
# Protection of Causal Relations

- Conflict:
  - S3 “threatens“ causal relation between S1 and S2
- Conflict resolution:
  - Promotion: Put threat before causal relation
  - Demotion: Put threat after causal relation

# Another Plan Refinement …

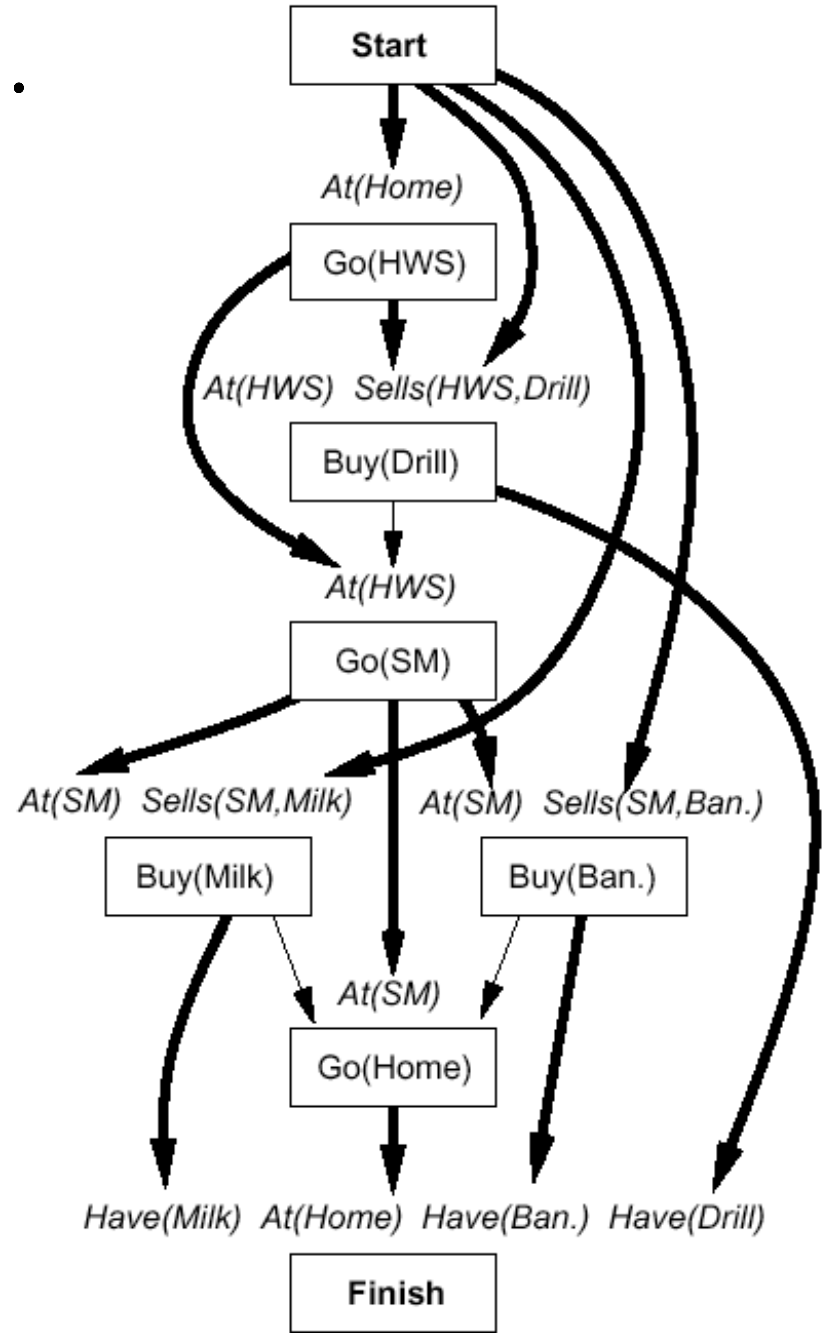
- Assumption: Cannot resolve conflict by protection
- Made a wrong step during plan refinement
- Alternative
  - Select x = HWS (with causal relation) while instantiating At(x) in go(SM)

# Another Plan Refinement …



# The Complete Solution …

- … with all links
- Computation by POP Algorithm
  - Complete
  - … and correct
- Addtionally, not considered here, correct treatment of variables



**function** POP(*initial, goal, operators*) **returns** *plan*

```
plan ← MAKE-MINIMAL-PLAN(initial, goal)
loop do
    if SOLUTION?(plan) then return plan
    S_need, c ← SELECT-SUBGOAL(plan)
    CHOOSE-OPERATOR(plan, operators, S_need, c)
    RESOLVE-THREATS(plan)
end
```

---

**function** SELECT-SUBGOAL(*plan*) **returns** $S_{need}$, $c$

pick a plan step $S_{need}$ from STEPS(*plan*)
    with a precondition $c$ that has not been achieved
**return** $S_{need}$, $c$

**procedure** CHOOSE-OPERATOR(*plan, operators*, $S_{need}$, *c*)

**choose** a step $S_{add}$ from *operators* or STEPS(*plan*) that has *c* as an effect
**if** there is no such step **then fail**
add the causal link $S_{add} \xrightarrow{c} S_{need}$ to LINKS(*plan*)
add the ordering constraint $S_{add} \prec S_{need}$ to ORDERINGS(*plan*)
**if** $S_{add}$ is a newly added step from *operators* **then**
    add $S_{add}$ to STEPS(*plan*)
    add $Start \prec S_{add} \prec Finish$ to ORDERINGS(*plan*)

---

**procedure** RESOLVE-THREATS(*plan*)

**for each** $S_{threat}$ that threatens a link $S_i \xrightarrow{c} S_j$ in LINKS(*plan*) **do**
    **choose** either
        *Promotion:* Add $S_{threat} \prec S_i$ to ORDERINGS(*plan*)
        *Demotion:* Add $S_j \prec S_{threat}$ to ORDERINGS(*plan*)
    **if not** CONSISTENT(*plan*) **then fail**
**end**

# Last Century Planning Systems (Last Decade!)

- UCPOP (Weld, UW)
  (http://www.cs.washington.edu/ai/ucpop.html )
- Sensory Graphplan (Weld, Blum, and Furst: UW)
  (http://aiweb.cs.washington.edu/ai/sgp.html )
- IPP (Köhler and Nebel: Univ. Freiburg)
  (https://idw-online.de/de/news5468)
- Prodigy: Planning and Learning (Veloso: CMU)
  (http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/prodigy/Web/prodigy-home.html )

- All systems have found interesting applications

# Planning is an Active Field of Research

- More powerful successors
  - Systems learn how to plan fast for specific problem instances
  - Can deal with uncertainty
    - About state estimation
    - About effects of actions
- Very powerful problem solvers can be set up …
  - w/ less effort/knowledge than with mathematical optimization theory and respective tools

→Automated Planning and Acting (CS5072-KP04)

( https://www.ifis.uni-luebeck.de/index.php?id=dski-aktuell-ss20&L=2 )

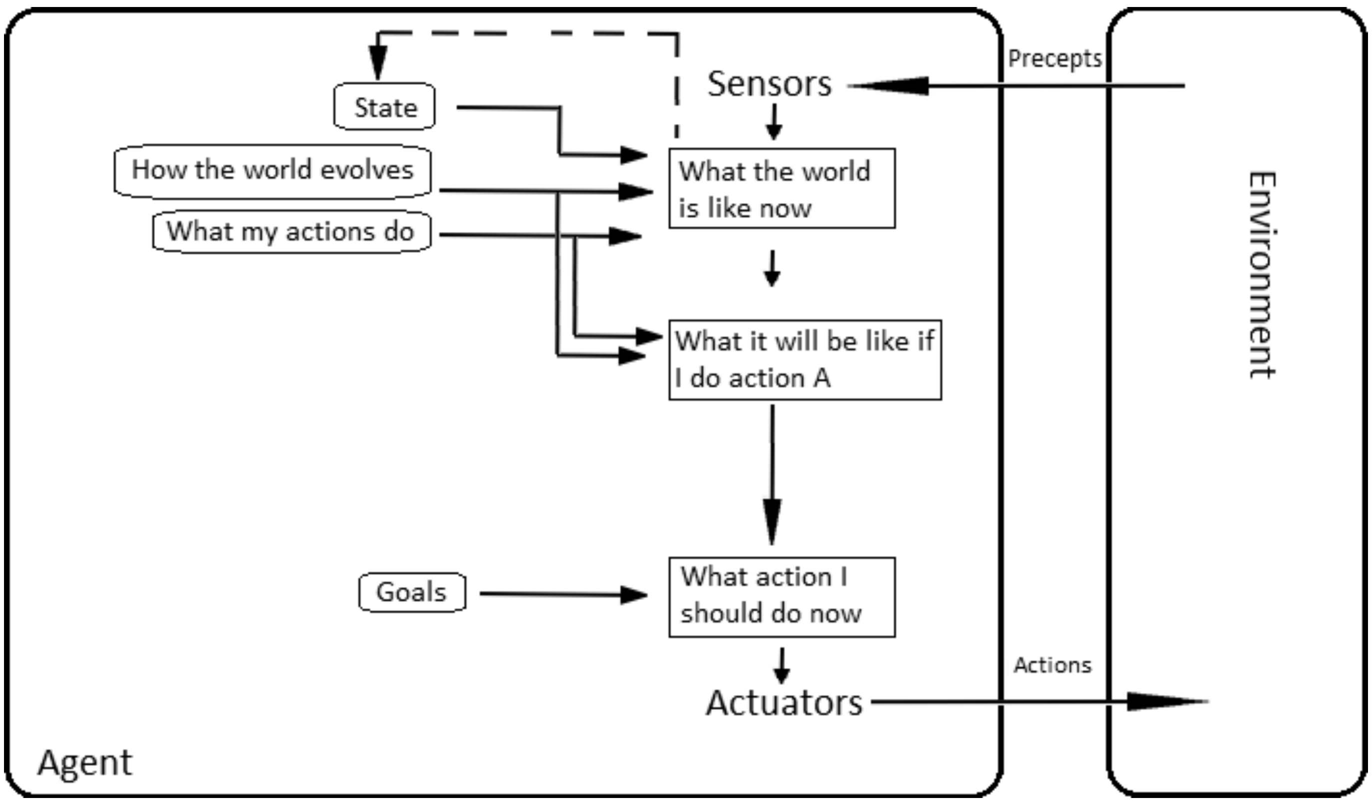
# Back to Intelligent Agents / Acting Intelligently

- Rational agents that:
  - Act autonomously and are persistent
  - Achieve goals surprisingly fast (despite bounded rationality)
  - Learn how to behave in a clever way (even learn computational strategies)
- Can adapt their goals to anticipate humans needs and expectations
  - Human compatibility, human awareness
- Can learn new models online to
  - Keep high performance over time
  - Support human-guided machine learning

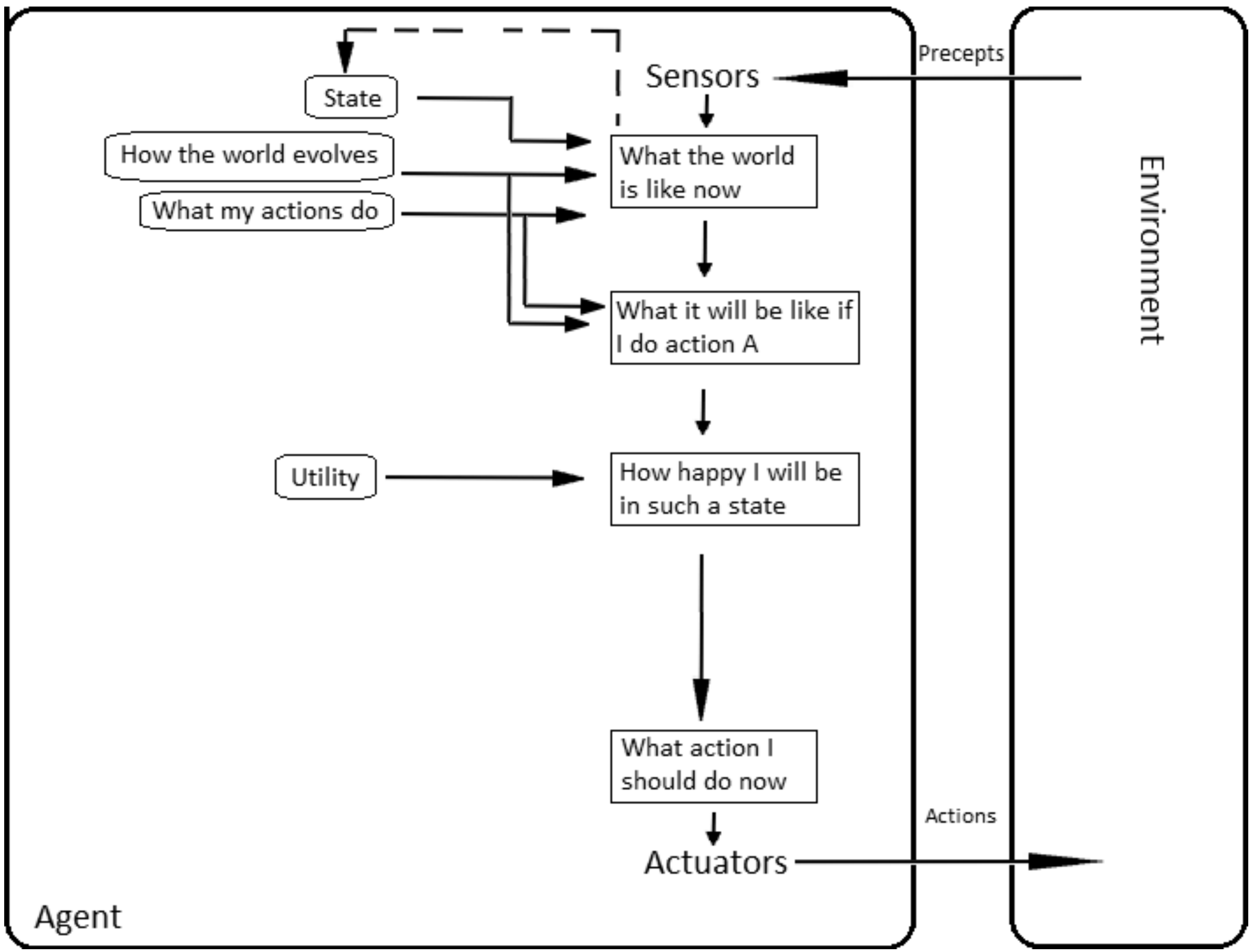
# Perspectives

- There are no specific "AI methods" applied by agents
  - Agents solve application problems (e.g., information retrieval)
  - Constructing agents humans solve AI problems
- Intelligence attributed to agents by humans from the outside
  - It is agents that can be intelligent, not functions (no framing desired)
- If different agents can achieve a certain goal easily,
  - Win 1000 of 1000 chess games
- … i.e., solve a certain application problem perfectly
- … then, intelligence attribution becomes less and less likely
  - NB: Different agents!
  - Cheating, i.e., sometimes showing stupidity to just not loose intelligence label, is not a dominant strategy for agents

# Goals…


State
How the world evolves
What my actions do
Sensors
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Actuators
Agent
Precepts
Actions
Environment

# … or Utilities




Goals can change, and so can utility measures