

For Friday

- Read chapter 18, sections 1-3
- No homework (work on program 3?)

Program 3

- Any questions?

Partial Order Plan

- Plan which does not specify unnecessary ordering.
- Consider the problem of putting on your socks and shoes.

Plans

- A **plan** is a three tuple $\langle A, O, L \rangle$
 - A: A set of **actions** in the plan, $\{A_1, A_2, \dots, A_n\}$
 - O: A set of **ordering constraints** on actions $\{A_i \prec A_j, A_k \prec A_l, \dots, A_m \prec A_n\}$. These must be consistent, i.e. there must be at least one total ordering of actions in A that satisfy all the constraints.
 - L: a set of **causal links** showing how actions support each other

Causal Links and Threats

- A **causal link**, $A_p \rightarrow^Q A_c$, indicates that action A_p has an effect Q that achieves precondition Q for action A_c .
- A threat, is an action A_t that can render a causal link $A_p \rightarrow^Q A_c$ ineffective because:
 - $O \cup \{A_p < A_t < A_c\}$ is consistent
 - A_t has $\neg Q$ as an effect

Threat Removal

- Threats must be removed to prevent a plan from failing
- **Demotion** adds the constraint $A_t < A_p$ to prevent clobbering, i.e. push the clobberer before the producer
- **Promotion** adds the constraint $A_c < A_t$ to prevent clobbering, i.e. push the clobberer after the consumer

Initial (Null) Plan

- Initial plan has
 - $A = \{ A_0, A_\infty \}$
 - $O = \{ A_0 < A_\infty \}$
 - $L = \{ \}$
- A_0 (Start) has no preconditions but all facts in the initial state as effects.
- A_∞ (Finish) has the goal conditions as preconditions and no effects.

Example

Op(Action: Go(there); Precond: At(here);

Effects: At(there), \neg At(here))

Op(Action: Buy(x), Precond: At(store), Sells(store,x);

Effects: Have(x))

- A_0 :
 - At(Home) Sells(SM,Banana) Sells(SM,Milk)
Sells(HWS,Drill)
- A_∞
 - Have(Drill) Have(Milk) Have(Banana)
At(Home)

POP Algorithm

- Stated as a **nondeterministic** algorithm where choices must be made. Various search methods can be used to explore the space of possible choices.
- Maintains an **agenda** of goals that need to be **supported** by links, where an agenda element is a pair $\langle Q, A_i \rangle$ where Q is a precondition of A_i that needs supporting.
- Initialize plan to null plan and agenda to conjunction of goals (preconditions of Finish).
- Done when all preconditions of every action in plan are supported by causal links which are not threatened.

POP($\langle A, O, L \rangle$, agenda)

- 1) **Termination**: If agenda is empty, return $\langle A, O, L \rangle$.
Use topological sort to determine a totally ordered plan.
- 2) **Goal Selection**: Let $\langle Q, A_{\text{need}} \rangle$ be a pair on the agenda
- 3) **Action Selection**: Let A_{add} be a nondeterministically chosen action that adds Q . It can be an existing action in A or a new action. If there is no such action return failure.

$$L' = L \cup \{A_{\text{add}} \rightarrow Q A_{\text{need}}\}$$

$$O' = O \cup \{A_{\text{add}} < A_{\text{need}}\}$$

if A_{add} is new then

$$A' = A \cup \{A_{\text{add}}\} \text{ and } O' = O' \hat{=} \{A_0 < A_{\text{add}} < A_{\infty}\}$$

else $A' = A$

4) **Update goal set:**

Let $\text{agenda}' = \text{agenda} - \{ \langle Q, A_{\text{need}} \rangle \}$

If A_{add} is new then for each conjunct Q_i of its precondition,
add $\langle Q_i, A_{\text{add}} \rangle$ to agenda'

5) **Causal link protection:** For every action A_t that threatens a causal link $A_p \rightarrow^Q A_c$ add an ordering constraint by choosing nondeterministically either

(a) **Demotion:** Add $A_t < A_p$ to O'

(b) **Promotion:** Add $A_c < A_t$ to O'

If neither constraint is consistent then return failure.

6) **Recurse:** $\text{POP}(\langle A', O', L' \rangle, \text{agenda}')$

Example

Op(Action: Go(there); Precond: At(here);

Effects: At(there), \neg At(here))

Op(Action: Buy(x), Precond: At(store), Sells(store,x);

Effects: Have(x))

- A_0 :
 - At(Home) Sells(SM,Banana) Sells(SM,Milk)
Sells(HWS,Drill)
- A_∞
 - Have(Drill) Have(Milk) Have(Banana)
At(Home)

Example Steps

- Add three buy actions to achieve the goals
- Use initial state to achieve the Sells preconditions
- Then add Go actions to achieve new preconditions

Handling Threat

- Cannot resolve threat to $At(\text{Home})$ preconditions of both $Go(\text{HWS})$ and $Go(\text{SM})$.
- Must backtrack to supporting $At(x)$ precondition of $Go(\text{SM})$ from initial state $At(\text{Home})$ and support it instead from the $At(\text{HWS})$ effect of $Go(\text{HWS})$.
- Since $Go(\text{SM})$ still threatens $At(\text{HWS})$ of $Buy(\text{Drill})$ must promote $Go(\text{SM})$ to come after $Buy(\text{Drill})$. Demotion is not possible due to causal link supporting $At(\text{HWS})$ precondition of $Go(\text{SM})$

Example Continued

- Add Go(Home) action to achieve At(Home)
- Use At(SM) to achieve its precondition
- Order it after Buy(Milk) and Buy(Banana) to resolve threats to At(SM)

Machine Learning

- What do you think it is?

Machine Learning

- Definition by Herb Simon: “Any process by which a system improves performance.”

Tasks

- Classification:
 - medical diagnosis, credit-card applications or transactions, investments, DNA sequences, spoken words, handwritten letters, astronomical images
- Problem solving, planning, and acting
 - solving calculus problems, playing checkers, chess, or backgamon, balancing a pole, driving a car

Performance

- How can we measure performance?
- That is, what kinds of things do we want to **get** out of the learning process, and how do we tell whether we're getting them?

Performance Measures

- Classification accuracy
- Solution correctness and quality
- Speed of performance

Why Study Learning?

- (Other than your professor's interest in it)

Study Learning Because ...

- We want computer systems with new capabilities
 - Develop systems that are too difficult or impossible to construct manually because they require specific detailed knowledge or skills tuned to a particular complex task (**knowledge acquisition bottleneck**).
 - Develop systems that can automatically adapt and customize themselves to the needs of individual users through experience, e.g. a personalized news or mail filter, personalized tutoring.
 - Discover knowledge and patterns in databases, **data mining**, e.g. discovering purchasing patterns for marketing purposes.

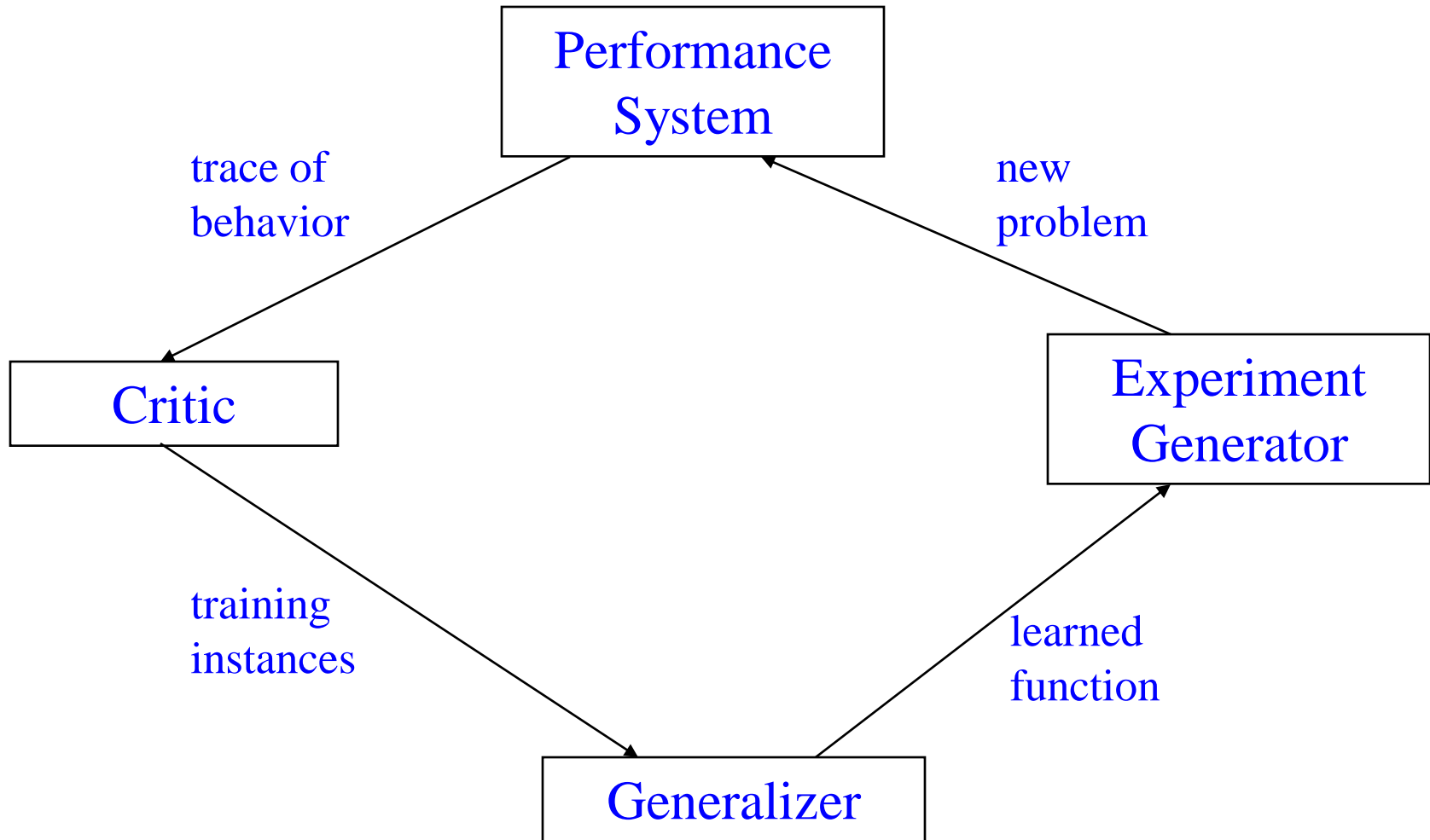
Study Learning Because ...

- Understand human and biological learning and teaching better.
 - Power law of practice.
 - Relative difficulty of learning disjunctive concepts.
- Time is right:
 - Initial algorithms and theory in place.
 - Growing amounts of on-line data.
 - Computational power available.

Designing a Learning System

- Choose the **training experience**.
- Choose what exactly is to be learned, i.e. the **target function**.
- Choose how to **represent** the target function.
- Choose a **learning algorithm** to learn the target function from the experience.
- Must distinguish between the **learner** and the **performance element**.

Architecture of a Learner



Training Experience Issues

- Direct or Indirect Experience
 - Direct: Chess boards labeled with correct move extracted from record of expert play.
 - Indirect: Potentially arbitrary sequences of moves and final games results.
- Credit/Blame assignment:
 - How do we assign blame to individual choices or moves when given only indirect feedback?

More on Training Experience

- Source of training data:
 - “Random” examples outside of learner’s control (negative examples available?)
 - Selected examples chosen by a benevolent teacher (near misses available?)
 - Ability to query oracle about correct classifications.
 - Ability to design and run experiments to collect one's own data.
- Distribution of training data:
 - Generally assume training data is representative of the examples to be judged on when tested for final performance.

Supervision of Learning

- Supervised
- Unsupervised
- Reinforcement