CS 440
Introduction to Artificial Intelligence

Lecture 11:

Logic and Planning

20 February 2020

Logistics

Meetings after class

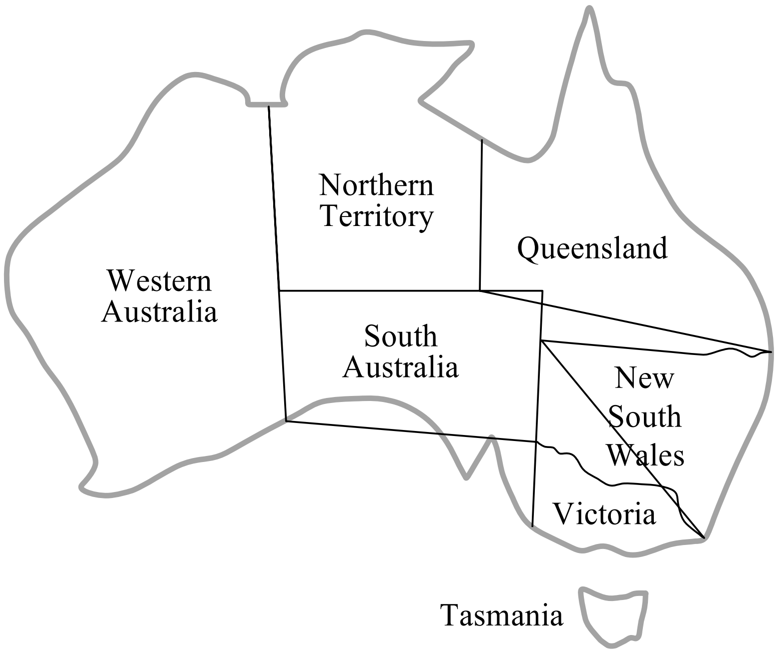
- Out of consideration for next class we should move discussion into lobby
- I will always stay around until all questions are answered

Moved due date of project back 1 day

- Sunday 23rd

Backtracking: Intelligent Backjumping

- Iteratively set each variable to valid value
  - Valid given constraints and values of other variables
- Set variables that have only one valid value
  - All variables set $\Rightarrow$ solution found
- If any variable has no valid values
  - Current setting cannot produce solution
  - Backtrack
    - Roll back variable settings



Backtracking Algorithm

Backtracking(X, C, S)

Input: A set of variables **X**, a set of constraints **C** and a partial setting **S**

While ∃ **x** ∈ **X** that has only one valid value, v

Set **x** to **v** in **S**

If all variables have been set

Return **S**

If ∃ **x** ∈ **X** with no valid values

return **FAIL**

Let x = an unset value in **X**

For all valid settings of **x** as **v**

S' = **S**

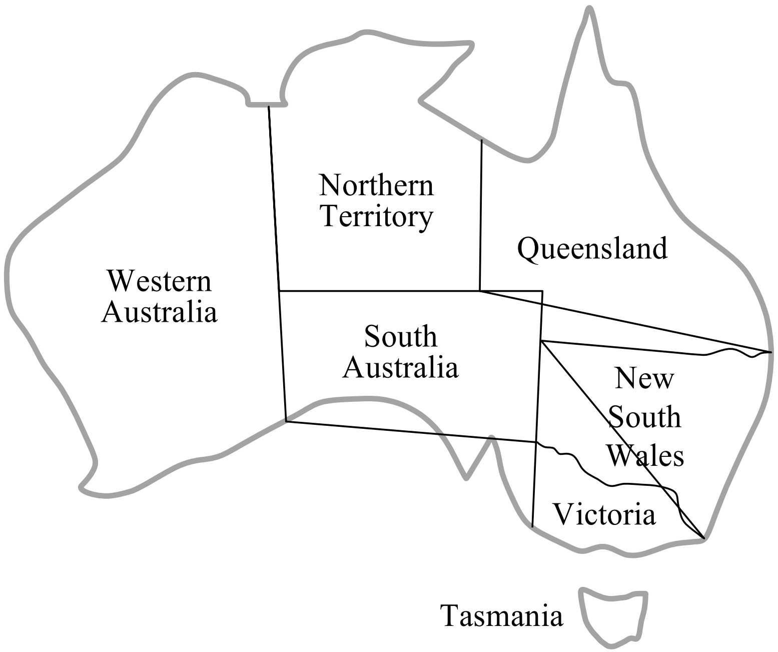
Set x to v in S'

result = Backtracking(**X, C, S'**)

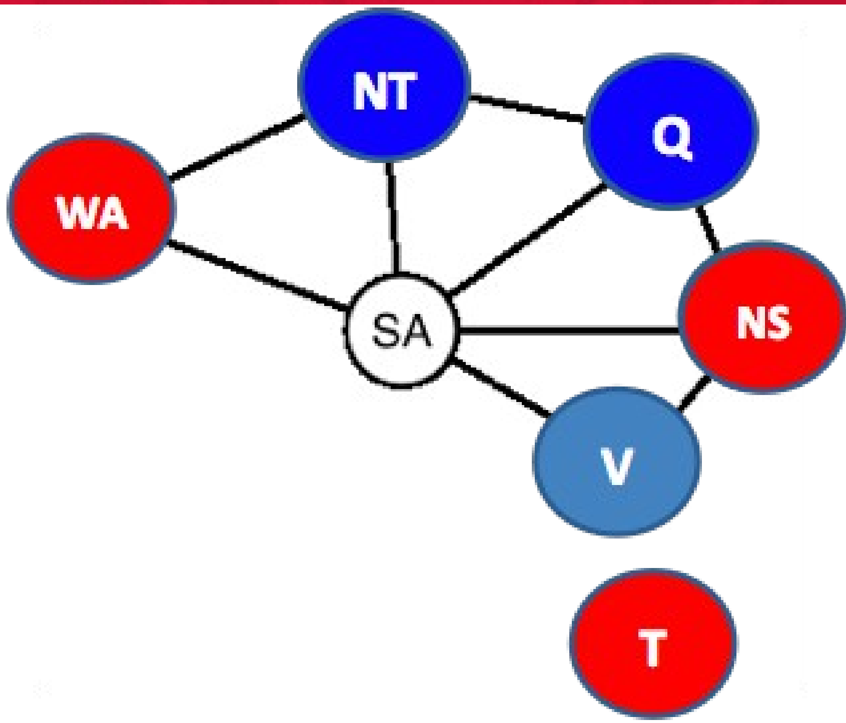
if(result ≠ **FAIL**)

Return **result**

Return **FAIL**



Backtracking:Example



- Order: Q, NS, V, T, SA
- Failure when trying to assign SA
- SA's conflict set {Q, NS, V}
- *Backjump* to the latest node in the conflict set: V
- Skip Tasmania

Assume WA=red and NSW =red, then assign T, NT, Q, SA

SA will cause a conflict, whatever we do...

- Where should the algorithm backjump?

Constraint satisfaction problem

Variables can be set to **true** or **false**

Denote variables as $\mathbf{X} = \{x_1, x_2, \ldots\}$

Constraints introduced by set of logical statements

Examples:

- $f(\mathbf{X}) = \neg x_1$
- $g(\mathbf{X}) = x_1 \vee x_2$
- $h(\mathbf{X}) = f(\mathbf{X}) \wedge g(\mathbf{X})$
  - $h(\mathbf{X}) = \neg x_1 \wedge (x_1 \vee x_2)$
- $j(X) = x_1 \rightarrow x_2$
- $k(X) = l(X) \leftrightarrow m(X)$

Hint: Think about what sets of variable settings satisfy a statement

Axioms

Axioms

- Statements that are always true
- True for all possible settings of variables

$x_1 \vee \neg x_1$

Commutativity:

- $f(X) \vee g(X) \Leftrightarrow g(X) \vee f(X)$
- $f(X) \wedge g(X) \Leftrightarrow g(X) \wedge g(X)$

Transitivity:

- $(f(X) \rightarrow g(X) \wedge g(X) \rightarrow h(X)) \rightarrow (f(X) \rightarrow h(X))$

Distributive:

- $f(X) \vee (g(X) \wedge h(X)) \Leftrightarrow (f(X) \vee g(X)) \wedge (f(X) \vee h(X))$
- $f(X) \wedge (g(X) \vee h(X)) \Leftrightarrow (f(X) \wedge g(X)) \vee (f(X) \wedge h(X))$

How would you prove something is an axiom?

How would you prove something isn't an axiom?

More Axioms

- $f(X) \lor \neg f(X) \Leftrightarrow$ true
- $f(X) \land \neg f(X) \Leftrightarrow$ false
- $f(X) \lor$ true $\Leftrightarrow$ true
- $f(X) \land$ false $\Leftrightarrow$ false
- $f(X) \lor$ false $\Leftrightarrow f(X)$
- $f(X) \land$ true $\Leftrightarrow f(X)$
- $\neg(f(X) \land g(X)) \Leftrightarrow \neg f(X) \lor \neg g(X)$
- $f(X) \land g(X) \rightarrow g(X)$
- $g(X) \rightarrow f(X) \lor g(X)$

Axioms

How would you prove something is an axiom?

- Reduction
  - Reduce to a statement that is trivially true
- Contradiction
  - Assume axiom if false and show contradiction
    - Variable that can neither be true of false

How would you prove something isn't an axiom?

- Setting of variables for which statement is false

How would an AI agent prove something is an axiom?

Example: Reductions

- $\neg\alpha(X) \wedge (\alpha(X) \vee \beta(X)) \rightarrow \beta(X)$
  - $f(X) \wedge (g(X) \vee h(X)) \Leftrightarrow (f(X) \wedge g(X)) \vee (f(X) \wedge h(X))$
- $(\neg\alpha(X) \wedge \alpha(X)) \vee (\neg\alpha(X) \wedge \beta(X)) \rightarrow \beta(X)$
  - $f(X) \wedge \neg f(X) \Leftrightarrow \text{false}$
- $\text{false} \vee (\neg\alpha(X) \wedge \beta(X)) \rightarrow \beta(X)$
  - $f(X) \vee g(X) \Leftrightarrow g(X) \vee f(X)$
- $(\neg\alpha(X) \wedge \beta(X)) \vee \text{false} \rightarrow \beta(X)$
  - $f(X) \vee \text{false} \Leftrightarrow f(X)$
- $\neg\alpha(X) \wedge \beta(X) \rightarrow \beta(X)$
  - $f(X) \wedge g(X) \rightarrow g(X)$
- $\beta(X) \rightarrow \beta(X)$

Given list of rules could an AI agent make such a reduction?

Example: Contradiction

- Negate statement
- Show contradiction
  - Negation unconditionally false
    - No setting of variables will satisfy it
    - Could apply backtracking algorithm and show that returns false
      - Complexity may make it inflatable
    - Could reduce to statement that is unconditionally false

Entailment

f(X) entails g(X)

- $f(X) \models g(X)$
- For all variable settings where f(X) is true, g(X) is also true
  - $g(X) \vee \neg f(X)$
- f(X) is sufficient to show g(X)
- f(X) implies g(X)

Examples:

- $\neg x_1 \wedge (x_1 \vee x_2) \models x_2$

Entailment vs implies

- $\models$ **vs** $\rightarrow$

Modus Ponens

- **If x implies y and x is true then you can entail that y must also be true**
  - $x \rightarrow y,\ y \models y$

- **Example**
  - **If you don't study then you will fail the exam**
  - **You did not study**
  - **Therefore, you will fail the exam**

Binary CSP example: Wumpus World

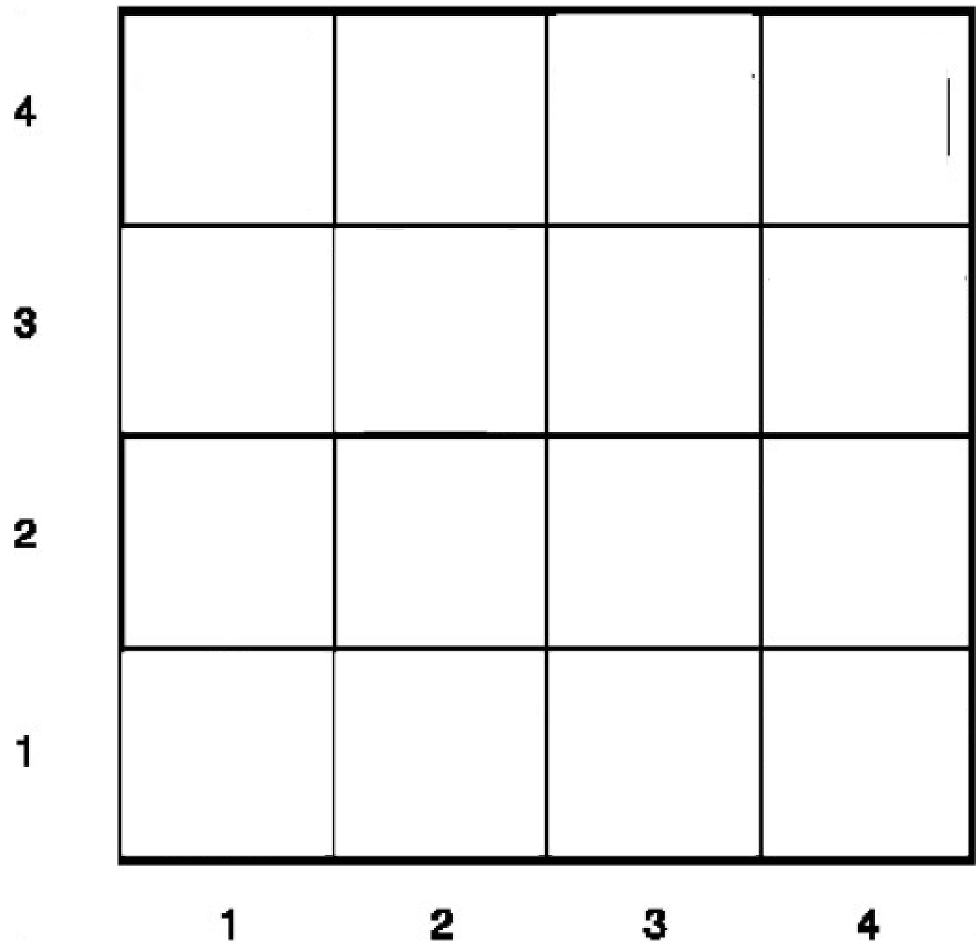
Wumpus

Agent

Pit

Gold

+1000 for gold
-1000 falling into pit
-1 for each action
-10 for using arrow

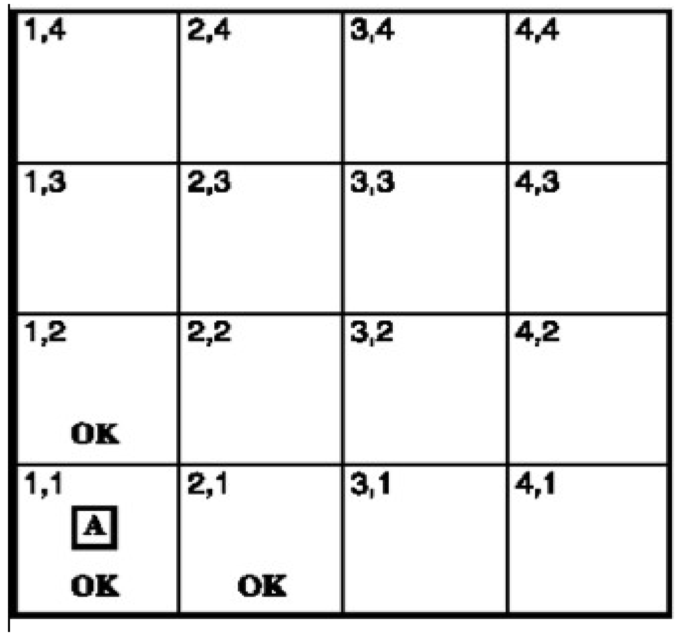


Formulation

- Variables
  - Defined for each cell
  - $S_{i,j}$, $B_{i,j}$, $P_{i,j}$, $W_{i,j}$, $G_{i,j}$
- Define problem in terms of logical statements
  - $S_{i,j} \rightarrow (W_{i-1,j} \lor W_{i+1,j} \lor W_{i,j-1} \lor W_{i,j+1})$
  - $B_{i,j} \rightarrow (P_{i-1,j} \lor P_{i+1,j} \lor P_{i,j-1} \lor P_{i,j+1})$
- Gold/Pit/Wumpus cannot be in same cell
  - $W_{i-,j} \rightarrow (\neg G_{i-,j} \land \neg P_{i-,j})$
  - $G_{i-,j} \rightarrow (\neg W_{i-,j} \land \neg P_{i-,j})$
  - $P_{i-,j} \rightarrow (\neg W_{i-,j} \land \neg G_{i-,j})$

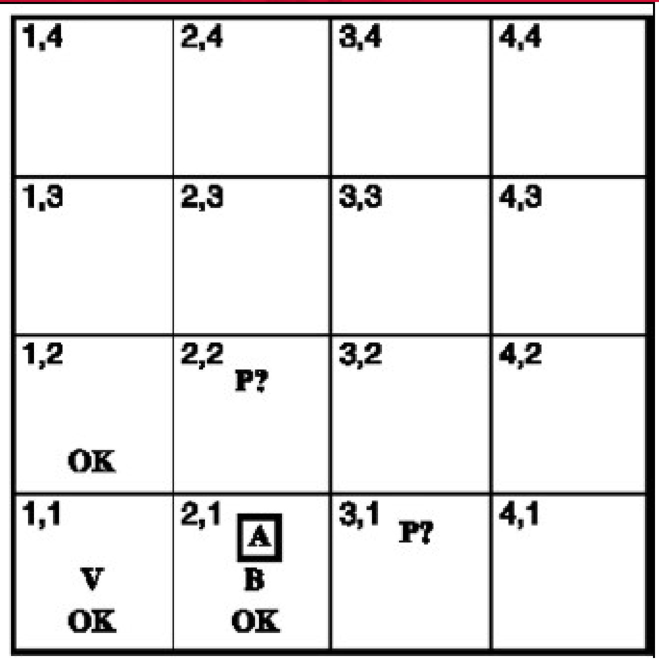
RUTGERS

Wumpus World



- **Agent in 1,1**
  - **No breeze and no smell**
- **Use Entailment**
  - $B_{i,j} \rightarrow (P_{i-1,j} \vee P_{i+1,j} \vee P_{i,j-1} \vee P_{i,j+1}), \neg B_{1,1} \models \neg P_{1,2}, \neg P_{2,1}$
  - $S_{i,j} \rightarrow (W_{i-1,j} \vee W_{i+1,j} \vee W_{i,j-1} \vee W_{i,j+1}), \neg S_{1,1} \models \neg W_{1,2}, \neg W_{2,1}$
- **Cells 1,2 and 2,1 are safe**

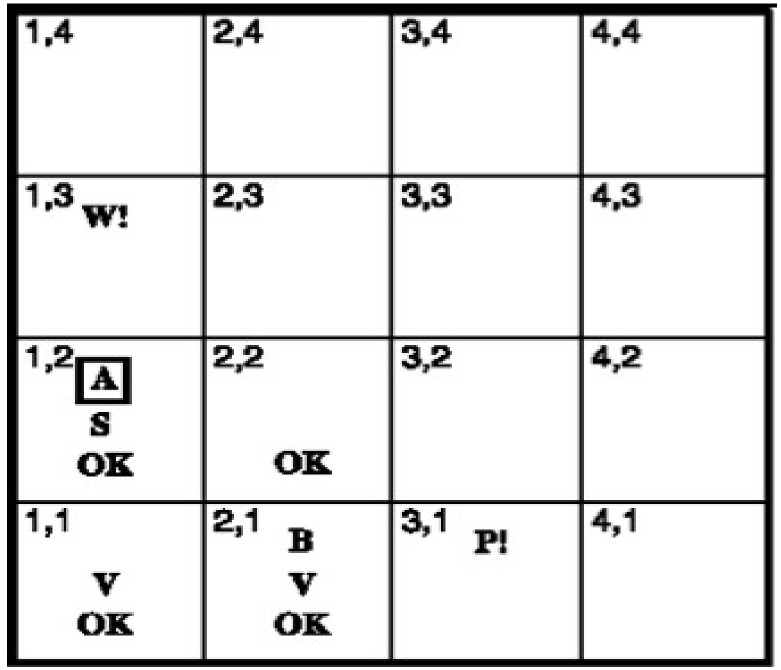
Wumpus World



- Agent in 2,1
  - Breeze and no smell
- Use Entailment
  - $B_{i,j} \rightarrow (P_{i-1,j} \vee P_{i+1,j} \vee P_{i,j-1} \vee P_{i,j+1}),\ B_{2,1} \models P_{2,2} \vee P_{3,1}$
  - $S_{i,j} \rightarrow (W_{i-1,j} \vee W_{i+1,j} \vee W_{i,j-1} \vee W_{i,j+1}),\ \neg S_{2,1} \models \neg W_{2,2},\ \neg W_{3,1}$
- Don't move to 2, 2 or 3, 1
  - Go back and explore 1,2 instead

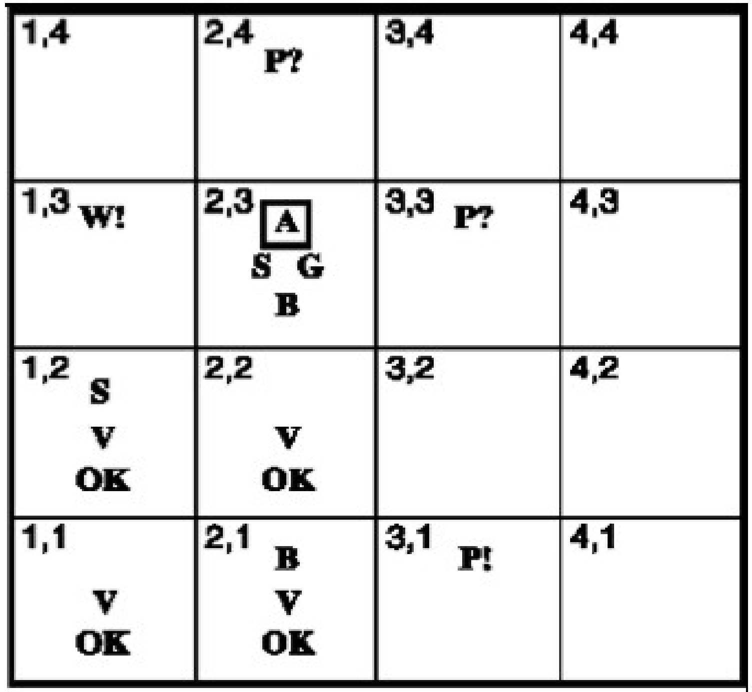
RUTGERS

Wumpus World



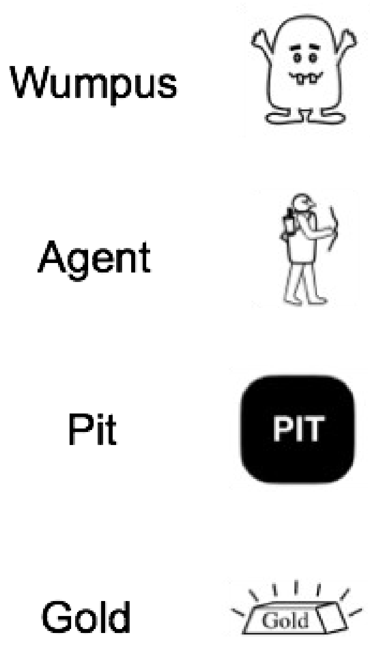
- **Agent in 2,1**
  - **Smell and no breeze**
- $B_{i,j} \rightarrow (P_{i-1,j} \vee P_{i+1,j} \vee P_{i,j-1} \vee P_{i,j+1}),\ \neg B_{2,1} \models \neg P_{1,3},\ \neg P_{2,2}$
- $P_{2,2} \vee P_{3,1},\ \neg P_{2,2} \models P_{3,1}$
- $S_{i,j} \rightarrow (W_{i-1,j} \vee W_{i+1,j} \vee W_{i,j-1} \vee W_{i,j+1}),\ S_{1,2},\ \neg W_{2,2},\ \neg W_{1,1} \models W_{1,3}$
- **Cell 2,2 is safe, move there**

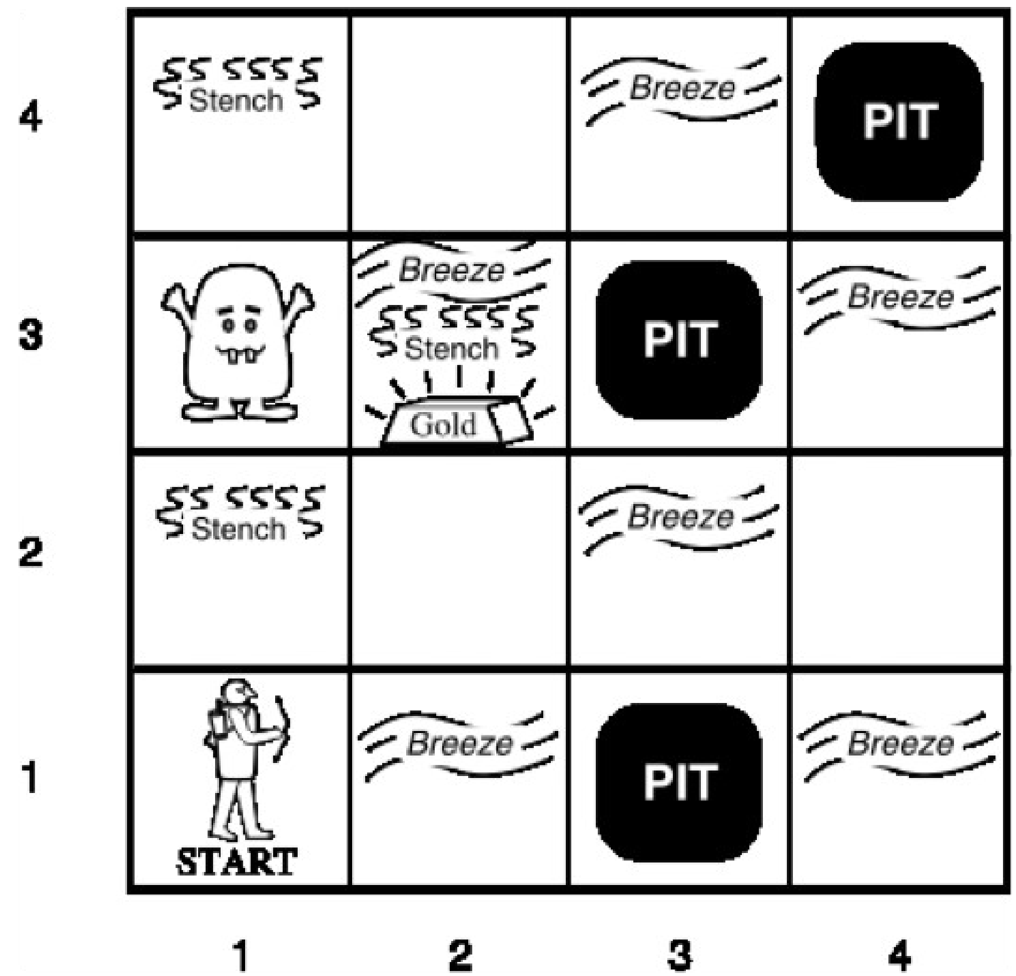
Wumpus World



- **Agent visits 2,2**
  - **No smell and no breeze**
- $B_{i,j} \rightarrow (P_{i-1,j} \vee P_{i+1,j} \vee P_{i,j-1} \vee P_{i,j+1}),\ \neg B_{2,2} \models \neg P_{2,3},\ \neg P_{3,2}$
- $S_{i,j} \rightarrow (W_{i-1,j} \vee W_{i+1,j} \vee W_{i,j-1} \vee W_{i,j+1}),\ \neg S_{2,2} \models \neg W_{2,3},\ \neg W_{3,2}$
- **Cell 3,2 and 2,3 are safe**
  - Move to 2,3
  - Find the gold there!!!!!

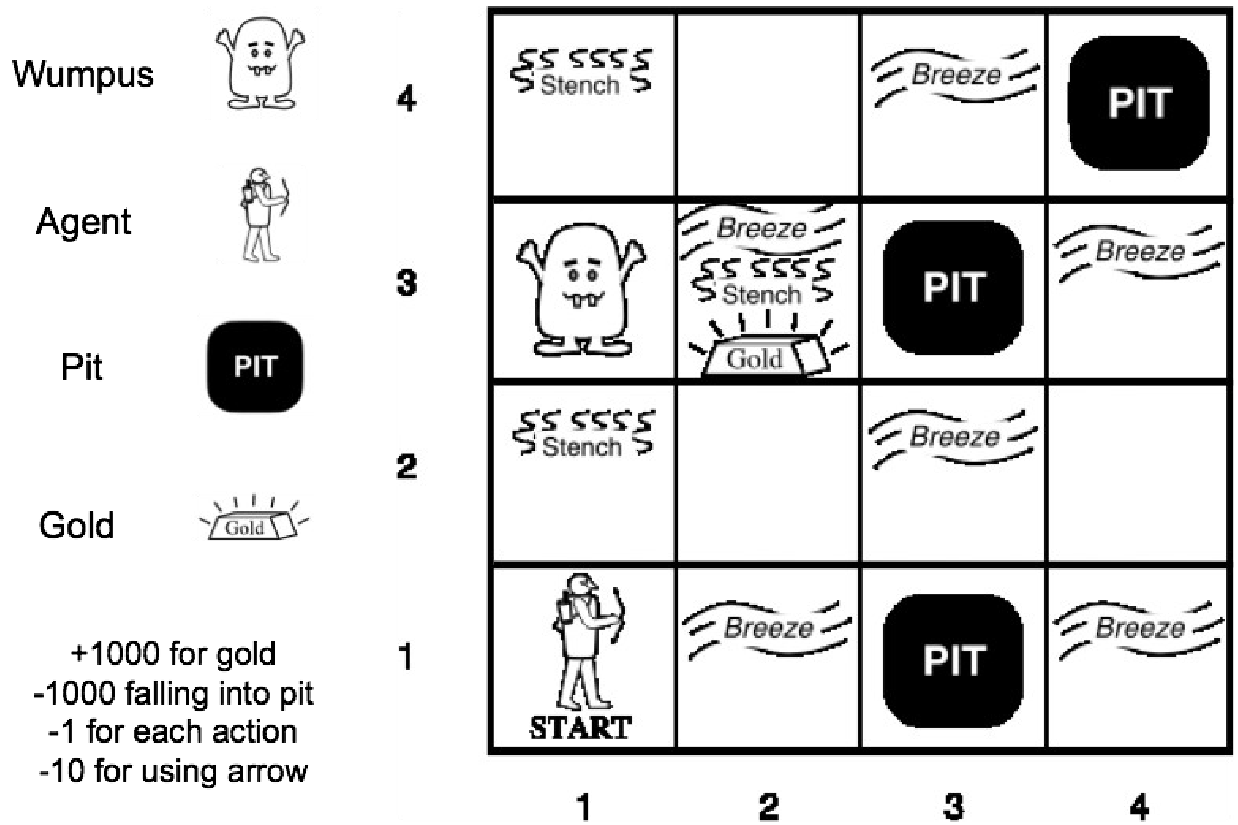
Wumpus World





+1000 for gold
-1000 falling into pit
-1 for each action
-10 for using arrow

Discussion



- How would you automate planning in environment like Wumpus World?
- What is the major difficulty?

Summary of First Third of Class

- **End of section on deterministic reasoning**
  - **Problem Formalization**
    - **State/Action/Transition/Observation/ect.**
  - **Local Search**
    - **Hill Climbing**
    - **Gradient descent**
  - **Search**
    - **BFS/DFS/SPF**
    - **Heuristics**
    - **A* Search**
  - **Adversarial Search**
    - **Minimax Search**
    - **Alpha-Beta Pruning**
  - **Decision Trees**
    - **Constructing Decision Trees**
  - **Constraint Satisfaction**
    - **Backtracking Algorithm**
  - **Logic**