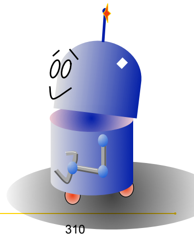
Computer Science

CPSC 433 - Artificial Intelligence

Final Review



Rob Kremer
ICT 748
kremer@cpsc.ucalgary.ca
http://pages.cpsc.ucalgary.ca/~kremer/

- What is AI?
- What are the problems AI addresses?
- Search vs computation

Muddiness

CATEGORY	FACTOR	CLEAN	MUDDY
External Env. (external to brain)	Awareness	Known	Unknown
	Complexity	Simple	Complex
	Conrolledness	Controlled	Uncontrolled
	Variation	Fixed	Changing
	Foreseeability	Foreseeable	Nonforeseeable
Input	Rawness	Symbolic	Real sensor
	Size	Small	Large
	Background	None	Complex
	Variation	Simple	Complex
	Occlusion	None	Severe
	Activeness	Passive	Active
	Modality	Simple	Complex
	Multi-modality	Single	Multiple
Internal Env. (internal to brain)	Size	Small	Large
	Representation	Given	Not given
	Observability	Observable	Unobservable
	Impossibility	Impossible	Nonimpossible
	Time coverage	Simple	Complex
Output	Terminalness	Low	High
	Size	Small	Large
	Modality	Simple	Complex
	Multimodality	Single	multiple

Search Models

A:(S,T)	**Set-based**	**And-tree-based**	**Or-tree-based**
S:possible states	S *F:set of facts*	$S_\wedge$ Atree: (pr, sol:{yes,?}, b n ≥ 0	$S_\vee$ *Otree:* (pr, sol:{yes,?,no}, b n≥0
T:SxS = possible Transitions	T: {(s,s') | ∃ A→B ∈ *Ext* • A ⊆ s ∧ s'=(s-A)∪B}; *Ext* ⊆ {A→B | A,B ⊆ *F*}	$T_\wedge$ *Erw* (pr,?,(pr (pr,?,b (pr,?,b	$T_\vee$ *Erw* (pr,?,b

---

Search Processes

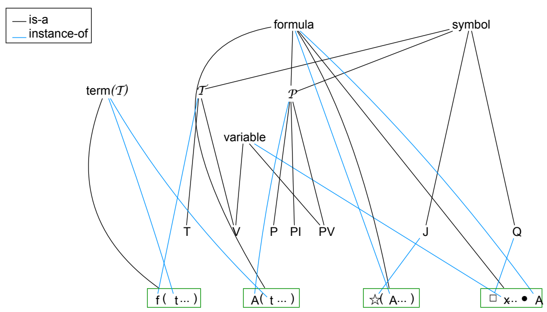
P:(A,Env,K)	**Set-based**	**And-tree-based**	**Or-tree-based**
K:SxEnv→S	$\mathcal{K}$(s,e) = (s-A)∪B		
Utility: SxSxEnv → Nat	∀A'→B' ∈ *Ext* | A'⊆ s • f_{Wert} f_{Wert}	f	f
Select: 2	A ∀ A"→B" ∈ *Ext* | A"⊆ s • f_{Wert} f_{Wert} f_{select}	f	f

---

Instances and Goals

	Set-based	**And-tree-based**	**Or-tree-based**
Ins=(s			
G: s → {yes,no}	s no more expansions possible	s = (pr', yes) ∨ ((pr', ?, b G the sols are compatable) ∨ no more solutions	s = (pr', yes) ∨ ((pr', ?, b G) ∨ can't process any leaves

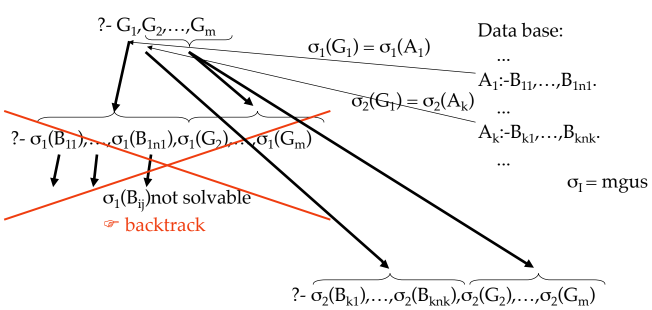
Logics



Logics

- General logics: T, P, J, Q, I
- Propositional logic
- Predicate logic
- Logic of EMycin
- Tautologies, Satisfiable, Unsatisfiable

Prolog



EMycin

Let h be an object-attribute-value triple and e a set of production rules. If $P_1 \wedge \ldots \wedge P_n \rightarrow_i h$ is the only rule in e, then we get

$MB(h,e) = I(P_1 \wedge \ldots \wedge P_n \rightarrow_i h) * max(0,min(I(P_1),\ldots,I(P_n))$

If e = $\{e_1,e_2\}$, then we get

$MB(h,\{e_1,e_2\}) = 0$, if $MD(h,\{e_1,e_2\}) = 1$

$MB(h,\{e_1,e_2\}) = MB(h,\{e_1\}) + MB(h,\{e_2\}) * (1-MB(h,\{e_1\}))$

For more elements just iterate this.

MD is computed similarly, except that e contains all production rules of the form $P_1 \wedge \ldots \wedge P_n \rightarrow_i \neg h$

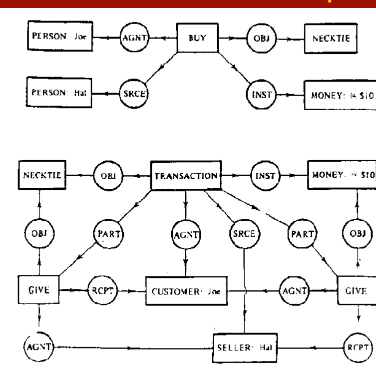
☞ application of Bayes formula for conditional probabilities

Frames: XML/DTD

```
<!DOCTYPE CATALOG [

<!ENTITY AUTHOR "John Doe">
<!ENTITY COMPANY "JD Power Tools, Inc.">
<!ENTITY EMAIL "jd@jd-tools.com">

<!ELEMENT CATALOG (PRODUCT+)>

<!ELEMENT PRODUCT
 (SPECIFICATIONS+,OPTIONS?,PRICE+,NOTES?)>
<!ATTLIST PRODUCT
 NAME CDATA #IMPLIED
 CATEGORY (HandTool|Table|Shop-Professional) "HandTool"
 PARTNUM CDATA #IMPLIED
 PLANT (Pittsburgh|Milwaukee|Chicago) "Chicago"
 INVENTORY (InStock|Backordered|Discontinued) "InStock">

<!ELEMENT SPECIFICATIONS (#PCDATA)>
<!ATTLIST SPECIFICATIONS
 WEIGHT CDATA #IMPLIED
 POWER CDATA #IMPLIED>

<!ELEMENT OPTIONS (#PCDATA)>
<!ATTLIST OPTIONS
 FINISH (Metal|Polished|Matte) "Matte"
 ADAPTER (Included|Optional|NotApplicable) "Included"
 CASE (HardShell|Soft|NotApplicable) "HardShell">

<!ELEMENT PRICE (#PCDATA)>
<!ATTLIST PRICE
 MSRP CDATA #IMPLIED
 WHOLESALE CDATA #IMPLIED
 STREET CDATA #IMPLIED
 SHIPPING CDATA #IMPLIED>

<!ELEMENT NOTES (#PCDATA)>

]>
```

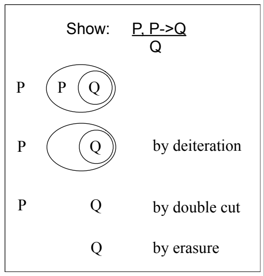
```
<product name="widget" partnum="T123" inventory="Backordered">
<specifications weight="120kg" power="240v">
<price MSRP="$50.00">
</product>
```


Semantic Nets: Conceptual Graphs



Existential Graphs

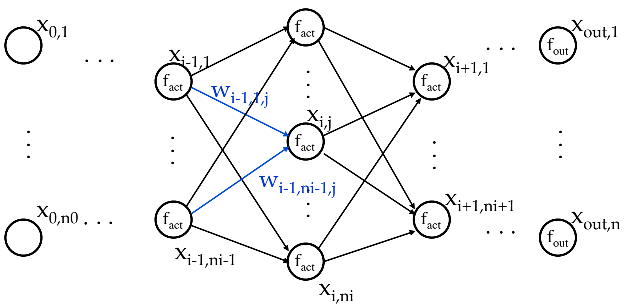
1. Insertion
Any graph can be inserted in to an oddly cut region
2. Erasure
Any graph in an even number of cuts can be removed
3. Iteration
Copy any graph into a region that occurs in all cuts the original occurs in
4. Deiteration
An graph obtained from iteration can be erased
5. Double Cut
2 cuts around a graph can be erased, and 2 cuts can be added around a grap



Semantic Nets: Conceptual Graphs

- Build a conceptual dependency representation for the following sentences:
  - John eats a steak
  - John ate pizza yesterday
- Build the graph for the following question and match it against the knowledge base from above:
  - Who had pizza yesterday?

Neural Nets



$x_{ij} = f_{act}(x_{i-1,1} {}^* w_{i-1,1,j}, \ldots, x_{i-1,ni-1} {}^* w_{i-1,ni-1,j})$

Neural Nets: Perceptron

- No hidden layers
- $f_{act}(a_1,\ldots,a_m) = sig(a_1+\ldots+a_m)$
- $f_{learn}(w_{0j1,old},d,x_{01},\ldots,x_{0k},c) =$
  $w_{0j1,old} + c^*(d - sig(\sum x_{0i} {}^* w_{0i1}))^* x_{0i}$
  where d is the expected output of the web
- Usually there are more inputs to a perceptron than what is suggested by the function to learn: bias-nodes allow for more learning accuracy

Neural Nets: Perceptron

Perform the learning for a perceptron with two input nodes and one bias node with constant value of 1 and a learning rate c = 0.3

Let the initial weights be

$w_{011} = 0.3$; $w_{021} = 0.4$; $w_{031} = -0.2$

Training data:

(1,1) -> 1; (9.4,6.4) -> -1; (8,7.7) -> -1; (0.5,2.2) -> 1

Constraints: Rules of thumb

- Theoretical investigations: what is possible?
  - ☞ Logics
- Knowledge already in very similar format
  - ☞ take format
- Hierarchical structures / inheritance
  - ☞ semantic nets, frames
- Represent certain input-output behavior
  - ☞ neural nets
- Laws, rules
  - ☞ rule sets

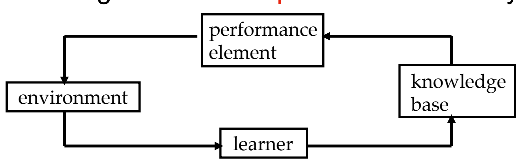
Learning

In general:

Structuring (or restructuring) of knowledge (due to experiences)

In AI systems:

Restructuring in order to improve behavior of system



Cooperation: TECH

- TEams for Cooperative Heterogeneous Search
- Basic idea:
  different agents employ different search processes working on the same search instance;
  both different controls and different search models
- From time to time an agent selects pieces of different kinds of knowledge to communicate them to the other agents
- An agent receiving pieces of knowledge filters them according to its needs and then integrates them into its search

Cooperation: TECH

- Positive information to be integrated into the search state:
  (partial) solutions to problem or subproblems
- Negative information to be integrated into the search state:
  unsolvable subproblems, partial solutions not leading to a solution
- Positive control information:
  control parameters, focus pieces of solutions
- Negative control information:
  transitions to avoid

Agent Communication: CASA