Outline for 4/9

- **Recap**
- **Constraint Satisfaction Techniques**
  - Backjumping (BJ)
  - Conflict-Directed Backjumping (CBJ)
  - Forward checking (FC)
  - Dynamic variable ordering heuristics
  - Preprocessing Strategies
- **Combinatorial Optimization**
- **Knowledge Representation I**
  - Propositional Logic: Syntax
  - Propositional Logic: Semantics
  - Propositional Logic: Inference
  - Compilation to SAT

Unifying View of AI

- Knowledge Representation
  - Expressiveness
  - Reasoning (Tractability)
- Search
  - Space being searched
  - Algorithms & performance

Specifying a search problem?

- What are states (nodes in graph)?
- What are the operators (arcs between nodes)?
- Initial state?
- Goal test?
- [Cost?, Heuristics?, Constraints?]

E.g., Eight Puzzle

1	2	3
4	5	6
7	8	

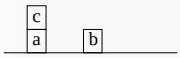
Towers of Hanoi

- What are states (nodes in graph)?
- What are the operators (arcs between nodes)?
- Initial state?
- Goal test?

4

Planning

- What is Search Space?
  - What are states?
  - What are arcs?
- What is Initial State?
- What is Goal?
- Path Cost?
- Heuristic?



PickUp(Block)
PutDown(Block)

5

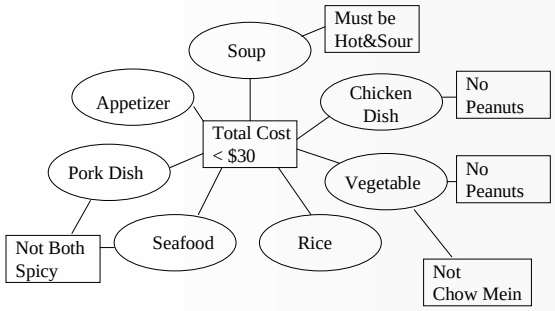
Search Summary

		Time	Space	Complete?	Opt?
Brute force	DFS	b^d	d	N	N
	BFS	b^d	b^d	Y	Y
	Iterative deepening	b^d	bd	Y	Y
	Iterative broadening	b^d			
Heuristic	Best first	b^d	b^d	N	N
	Beam	b^d	b+L	N	N
	Hill climbing	**b^d**	b	N	N
	Simulated annealing	**b^d**	b	N	N
	Limited discrepancy	b^d	bd	Y/N	Y/N
Optimizing	A*	b^d	b^d	Y	Y
	IDA*	b^d	b	Y	Y
	SMA*	b^d	[b-max]	Y	Y

Constraint Satisfaction

- Chronological Backtracking (BT)
- Backjumping (BJ)
- Conflict-Directed Backjumping (CBJ)
- Forward checking (FC)
- Dynamic variable ordering heuristics
- Preprocessing Strategies

Chinese Constraint Network



CSPs in the real world

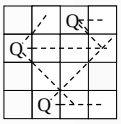
- Scheduling Space Shuttle Repair
- Transportation Planning
- Computer Configuration
- Diagnosis
- Etc...

Binary Constraint Network

- Set of n variables: $x_1 \ldots x_n$
- Value domains for each variable: $D_1 \ldots D_n$
- Set of binary ***constraints*** (also known as relations)
  - $R_{ij} \subseteq D_i \times D_j$
  - Specifies which values of x_i are consistent w/ those of x_j
- Partial ***assignment*** of values with a tuple of pairs
  - {...(x,a)…} means variable x gets value a...
  - ***Consistent*** if all constraints satisfied on all vars in tuple
  - Tuple = full solution if consistent & all vars included
- Tuple $\{(x_i, a_i) \ldots (x_j, a_j)\}$ ***consistent w/ a set of vars*** $\{x_m \ldots x_n\}$ iff $\exists\ a_m \ldots a_n$ such that this tuple is consistent: $\{(x_i, a_i) \ldots (x_j, a_j), (x_m, a_m) \ldots (x_n, a_n)\}$ }

N Queens

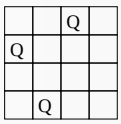
- Variables = board columns
- Domain values = rows
- $R_{ij} = \{(a_i, a_j) : (a_i \neq a_j) \wedge (|i-j| \neq |a_i - a_j|)$
  - e.g. $R_{12} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\}$



- $\{(x_1, 2), (x_2, 4), (x_3, 1)\}$ consistent with (x_4)
- Shorthand: "{2, 4, 1} consistent with x_4"

CSP as a search problem?

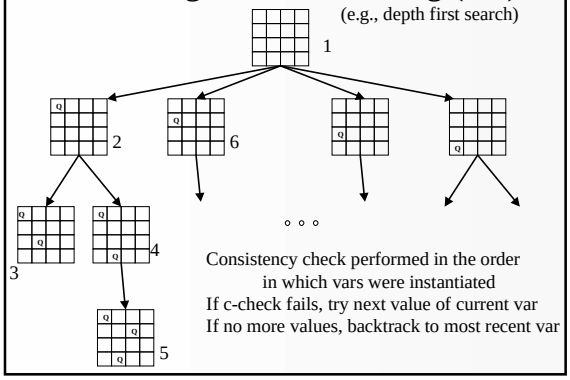
- What are states (nodes in graph)?
- What are the operators (arcs between nodes)?
- Initial state?
- Goal test?



12

Chronological Backtracking (BT)

(e.g., depth first search)



Consistency check performed in the order in which vars were instantiated
If c-check fails, try next value of current var
If no more values, backtrack to most recent var

Backjumping (BJ)

- Similar to BT, but more efficient when no consistent instantiation can be found for the current var
- Instead of backtracking to most recent var…
- BJ reverts to deepest var which was checked against the current var

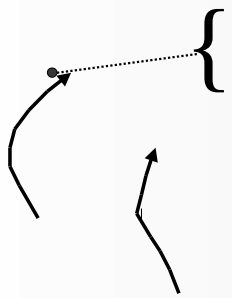
BJ Discovers
(2, 5, 3, 6) inconsistent with x_6
No sense trying other values of x_5

Conflict-Directed Backjumping (CBJ)

- More sophisticated backjumping behavior
- Each variable has ***conflict set CS***
  - Set of vars that failed consistency checks w/ current val
  - Update this set on every failed c-check
- When no more values to try for x_i
  - Backtrack to deepest var, x_h, in $CS(x_i)$
  - ***And*** update $CS(x_h) := CS(x_h) \cup CS(x_i) - \{x_h\}$

CBJ Discovers
(2, 5, 3) inconsistent with $\{x_5, x_6\}$

BT *vs.* BJ *vs.* CBJ



Forward Checking (FC)

- Perform Consistency Check Forward
- Whenever assign var a value
  - Prune inconsistent values from
  - As-yet unvisited variables
  - Backtrack if domain of any var ever collapses

FC only visits consistent nodes
but not *all* such nodes
skips (2, 5, 3, 4) which CBJ visits
But FC can't detect that
(2, 5, 3) inconsistent with $\{x_5, x_6\}$

Number of Nodes Explored

BT=BM *More*

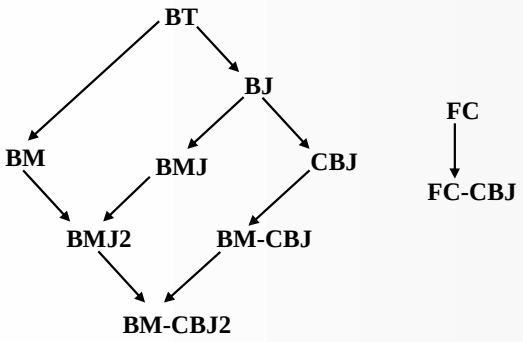
BJ=BMJ=BMJ2

FC

CBJ=BM-CBJ=BM-CBJ2

FC-CBJ *Fewer*

Number of Consistency Checks



Dynamic variable ordering

- In the N-queens examples we assumed
  - First x_1 then x_2 then ...
- But this order not required
  - ***Any*** order ok with respect to completeness
  - A good order leads to huge speedup
- A good heuristic:
  - Choose variable w/ minimum remaining values
    - This is easy if one is doing FC

Add Some DVO Numbers

Preprocessing Strategies

- Even FC-CBJ is O(b^d) time worst case
- Sometimes useful to spend polynomial time preprocessing to achieve local consistency before doing exponential search
- Arc consistency
  - Consider all pairs of vars
  - Can any values be eliminated from a domain ala FC
  - Propagate
  - O(d^2) time where d= number of vars

Combinatorial Optimization

- Nonlinear Programs
- Convex Programs
  - Local optimality ⇒ global optimality
  - Kuhn Tucker conditions for optimality
- Linear Programs
  - Simplex aAlgorithm
- Flow and Matching
- Integer Programming

Today's Outline

- **Recap**
- **Constraint Satisfaction Techniques**
  - Backjumping (BJ)
  - Conflict-Directed Backjumping (CBJ)
  - Forward checking (FC)
  - Dynamic variable ordering heuristics
  - Preprocessing Strategies
- **Combinatorial Optimization**
- **Knowledge Representation I**
  - Propositional Logic: Syntax
  - Propositional Logic: Semantics
  - Propositional Logic: Inference
  - Compilation to SAT

In the Knowledge Lies the Power

Ed Feigenbaum
Stanford University

If the patient has a bacterial skin infection, and
specific organisms are not seen in the patient's blood test,
Then there is evidence that the organism causing the infection is Staphylococcus

Some KR Languages

- Propositional Logic
- Predicate Calculus
- Frame Systems
- Rules with Certainty Factors
- Bayesian Belief Networks
- Influence Diagrams
- Semantic Networks
- Concept Description Languages
- Nonmonotonic Logic

In Fact…

- All popular knowledge representation systems are equivalent to (or a subset of)
  - Logic (Propositional Logic or Predicate Calculus)
  - Probability Theory

AI = Knowledge Representation & Reasoning

- Syntax
- Semantics
- Inference Procedure
  - Algorithm
  - Sound?
  - Complete?
  - Complexity

Knowledge Engineering

What is logic?

- Study of **proof** and **justification** (Aristotle 400BC)
  - All human beings are mortal;
  - All Greeks are human beings;
  - Therefore, all Greeks are mortal.
- **Profound idea**: logic can be formalized (Frege 1879)
- Statements are true or false by virtue of their **form** ("shape") not their content (what they mean)

$\forall x, h(x) \Rightarrow m(x)$
$\forall y, g(y) \Rightarrow h(y)$
Therefore, $\forall z, g(x) \Rightarrow m(x)$

29

Propositional Logic

- Syntax
  - Atomic sentences: P, Q, …
  - Connectives: $\wedge$, $\vee$, $\neg$, $\Rightarrow$
- Semantics
  - Truth Tables
- Inference
  - Modus Ponens
  - Resolution
  - DPLL
  - GSAT
  - Resolution
- Complexity

30

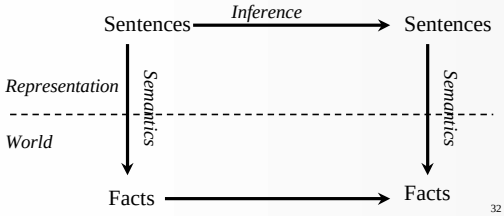
First Order Logic *vs* Prop. Logic

- Ontology
  - Objects, properties, relations *vs*.Facts
- Syntax
  - Sentences have more structure: terms
  - Variables & Quantification ∀, ∃
- Semantics
  - Much more complicated, but who cares
- Inference
  - Much more complicated

31

Semantics

- ***Syntax***: a description of the *legal* arrangements of symbols (Def "sentences")
- ***Semantics***: what the arrangement of symbols *means* in the world



32

Propositional Logic: **SEMANTICS**

- Multiple interpretations
  - Assignment to each variable either T or F
  - Assignment of T or F to each connective via defns

P \ Q	T	F
T	T	F
F	F	F

P ∧ Q

P \ Q	T	F
T	T	T
F	T	F

P ∨ Q

P \ Q	T	F
T	T	F
F	T	T

P ⇒ Q

P	
T	F
F	T

¬ P

Note: (P ⇒ Q) equivalent to ¬ P ∨ Q

Notation

$\Rightarrow$
$\supset$ } **Implication (syntactic symbol)**
$\rightarrow$

$\vdash$ **Inference**

$\models$ **Entailment**

- **Sound** $\vdash$ **implies** $\models$
- **Complete** $\models$ **implies** $\vdash$

34

Definitions

- **valid** = tautology = always true
- **satisfiable** = sometimes true
- **unsatisfiable** = never true

1) smoke $\Rightarrow$ smoke
   $\neg$smoke $\vee$ smoke **valid**
2) smoke $\Rightarrow$ fire
   $\neg$ smoke $\vee$ fire **satisfiable**
3) (smoke $\Rightarrow$ fire) $\Rightarrow$ ($\neg$smoke $\Rightarrow$ $\neg$fire)
   (smoke $\wedge$ $\neg$fire) $\vee$ $\neg$smoke $\vee$ $\neg$fire **valid**
4) smoke $\vee$ fire $\vee$ $\neg$fire
   valid

35

Prop. Logic: **Knowledge Engr**

1) Lisa is not next to Dave in the ranking
2) Jim is immediately ahead of a bio major
3) Dave is immediately ahead of Jim
4) One of the women is a biology major
5) Mary or Lisa is ranked first

- Choose Vocabulary
  - Universe: Lisa, Dave, Jim, Mary
  - LiaD = "Lisa is immediately ahead of Dave"
  - BioD = "Dave is a Bio Major"
- Choose initial sentences (wffs)
  1) $\neg$LiaD $\wedge$ $\neg$DiaL
  2) (JiaD $\wedge$ BioD) $\vee$ (JiaM $\wedge$ BioM) $\vee$ ...
  ...

Propositional Logic: **Inference**

- Backward Chaining (Goal Reduction)
  - Based on rule of ***modus ponens***
  - If know $P_1 \wedge ... \wedge P_n$ and know $(P_1 \wedge ... \wedge P_n) \Rightarrow Q$
  - Then can conclude Q
- Resolution (Proof by Contradiction)
- GSAT
- Davis Putnam

Horn Clauses

- If every sentence in KB is of the form:

$$A \wedge B \wedge C \wedge ... \wedge F \Rightarrow Z$$

equivalently $\neg A \vee \neg B \vee \neg C \vee ... \vee \neg F \vee Z$

- Then Modus Ponens is
  - Polynomial time, and
  - Complete!
- Hence, Prolog Implementation

Clause means a big disjunction

38

Resolution

$$\frac{A \vee B \vee C, \quad \neg C \vee D \vee \neg E}{A \vee C \vee D \vee \neg E}$$

- Refutation Complete
  - Given an unsatisfiable KB in CNF,
  - Resolution will eventually deduce the empty clause
- Proof by Contradiction
  - To show $\Sigma \models Q$
  - Show $\Sigma \cup \{\neg Q\}$ is unsatisfiable!

Normal Forms

- CNF = Conjunctive Normal Form
- Conjunction of disjuncts (each disjunct = "clause")

$(P \wedge Q) \Rightarrow R$ $\quad \neg(P \wedge Q) \vee R$ $\quad \neg P \vee \neg Q \vee R$

$(P \vee Q) \Rightarrow R$ $\quad (\neg P \wedge \neg Q) \vee R$

$(\neg P \vee R) \wedge (\neg Q \vee R)$

Terminology

- Literal u or $\neg$u, where u is a variable
- Clause disjunction of literals
- Formula, Φ, conjunction of clauses
- Φ(u) take Φ and set all instances of u true; simplify
  - e.g. Φ=((P, $\neg$Q)(R, Q)) then Φ(Q)=P
- Pure literal var appearing in a formula either as a negative literal or a positive literal (but not both)
- Unit clause clause with only one literal

Davis Putnam (DPLL)

[1962]

Procedure DPLL (CNF formula: Φ)
  If Φ is empty, return yes.
  If there is an empty clase in Φ return no.
  If there is a pure literal u in Φ return DPLL(Φ(u)).
  If there is a unit clause {u} in Φ return DPLL(Φ(u)).
  Else
    Select a variable v mentioned in Φ.
    If DPLL(Φ(v))=yes,
      then return yes.
    Else
      return DPLL(Φ($\neg$v)).

GSAT

[1992]

```
Procedure GSAT (CNF formula: Φ, max-restarts, max-climbs)
   For I := I o max-restarts do
         A := randomly generated truth assignment
         for j := 1 to max-climbs do
            if A satisfies Φ  then return yes
            A := random choice of one of best successors to A
                  ;; successor means only 1 var val changes from A
                  ;; best means making the most clauses true
```

FOL Definitions

- ***Constants***: a,b, dog33.
  - Name a specific object.
- ***Variables***: X, Y.
  - Refer to an object without naming it.
- ***Functions***: father-of
  - Mapping from objects to objects.
- ***Terms***: father-of(father-of(dog33))
  - Refer to objects
- ***Atomic Sentences***: in(father-of(dog33), food6)
  - Can be true or false
  - Correspond to propositional symbols P, Q

44

Propositional. Logic *vs* **First Order**

Ontology	Facts (P, Q)	Objects, Properties, Relations
Syntax	Atomic sentences Connectives	Variables & quantification Sentences have structure: terms father-of(mother-of(X)))
Semantics	Truth Tables	Interpretations (Much more complicated)
Inference	Efficient SAT algorithms	Unification Forward, Backward chaining Prolog, theorem proving

45