Session 5

I. Announcements [5 minutes]

- Assignment 2 has been extended to Sunday. Get it done early and enjoy the weekend.
- The midterm is coming up and *will not* be extended. It is on 10/18. I'm going to be spending parts of section reviewing for the exam.
- **Feedback**: *I'd like to get feedback from everybody in section on how I can improve sections: 3 things you like, 3 things you dislike.*
    - I've sent out a link to a *Remailer* you can send email to me from anonymously. Also the link is posted on my webpage.
    - I know you're busy with Sudoku's, train tracks, and so forth, but take 10 minutes and send me feedback – it'll make the course better for you and me.
- **Handouts**: I'm now handing out a weekly topic sheet. These sheets should be good for *reviewing* for the exam, but will not replace reading the course material. I'll make them available in class and on my website – are people able to get to my website?

II. Questions

II. Exam Question [10 minutes]

- Go over previous exam question

IV. How hard is a problem? [10 minutes]

Earlier in the course, we talked about the N-Queens problem. It turns out the problem can be solved by CSPs and local search algorithms for large N quickly. Earlier we looked in depth at the solution space of N-Queens (see below). As it turns out, the N-Queens problem is not very difficult

- Problems can be formulated in terms of satisfiablity – we want to find an assignment of variables that satisfy some checkable statement.
- The difficulty of a problem can characterized by the underlying satisfiablity problem we are attempting to solve:
    - Suppose the sentence being satisfied has m clauses and n variables.
    - The ratio m/n is a relative indicator of the problem's difficulty.
        - (**Underconstrained**) For m/n *SMALL*, there are many feasible assignments to the variables.
        - (**Overconstrained**) For m/n *LARGE*, there are no feasible assignments to the variables.
        - Both the above cases are easy (n-Queens is overconstrained), but somewhere in between there is a ***critical point*** where there is only a few assignments. Empirically, problems in this region take longer to solve.

V. Propositional Logic [25 minutes]

- Hopefully, everybody has a basic understanding of propositional logic – a language that allows us to state truths about the world.
  - The atoms are simply propositions about the world – they can either be true or false.
  - These propositions can be made into complex sentences via logical connectives:
    - **logical connectives:**
      - **not** $\neg$ **–** negation
      - **and** $\wedge$ **–** conjunction
      - **or** $\vee$ **–** disjunction
      - **implies** $\Rightarrow$ **–** implication ($(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$) *Note: if* α *is false,* $\alpha \Rightarrow \beta$ *says nothing about* β*.*
      - **if and only if** $\Leftrightarrow$ **–** biconditional
      - **xor** $\otimes$ **-** $(\alpha \otimes \beta) \equiv (\alpha \wedge \neg\beta) \vee (\neg\alpha \wedge \beta)$
    - **order of operations (high->low):** $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
  - It is important that you're able to manipulate logical sentences – look in the AIMA book at page 210 for a list of important laws.
    - De Morgan's Law.
- All information is stored in a **knowledge base (KB) –** a set of sentences describing the world.
  - **background knowledge –** initial knowledge in the KB
  - **knowledge level –** we only need to specify what the agent knows and what its goals are in order to specify its behavior
  - **Tell(P) –** function that adds knowledge P to the KB.
  - **Ask(P) –** function that queries the agent about the truth of P.
- **logical entailment –** the concept of 1 sentence following from another sentence:

  $\alpha \models \beta$ if α is true, then β must also be true.

  Note: while similar to the notion of implication, entailment is a meta-statement, not a part of the language itself. That is, statements using entailment are used to describe other logical statements.
  - Monotonicity – a set of entailed sentences can only *increase* in information as information is added to the knowledge base.

$$KB \models \alpha \quad \Rightarrow \quad KB \wedge \beta \models \alpha$$

- **conjunctive normal form (CNF)** – every sentence of propositional logic is *logically equivalent* to a conjunction of disjunctions of literals.

$$\left(l_{1,1} \vee \ldots \vee l_{1,n_1}\right) \wedge \ldots \wedge \left(l_{m,1} \vee \ldots \vee l_{m,n_m}\right)$$

    1. Eliminate biconditionals: $\alpha \Leftrightarrow \beta \equiv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$
    2. Eliminate implications $\alpha \Rightarrow \beta \equiv \neg\alpha \vee \beta$
    3. Move $\neg$ inwards
    4. Distribute $\wedge$ over $\vee$.
- A complex sentence can always be represented in CNF.
    1. literal – an atomic sentence (positive literal) or a negated atomic sentence (negative literal).
    2. clause – a disjunction of literals
    3. sentence – a conjunction of clauses.
- In CNF, the knowledge base is simply a gigantic conjunction of all sentences it has received – all true.
- *Definite Clauses* – disjunction of literals of which exactly one is positive.

$$\neg n_1 \vee \ldots \vee \neg n_m \vee p \quad \equiv \quad \underbrace{n_1 \wedge \ldots \wedge n_m}_{body} \Rightarrow \underbrace{p}_{head}$$

    - **head** – the positive literal.
    - **body** – the negative literals; the premises.
    - **fact** – a definite clause with no negative literals.
    - **Horn clause** – a disjunction of literals at most one of which is positive.
        - can be written as an implication whose conclusion is *False*.
        - called an *integrity constraint*.
    - Inference with Horn clauses can be done with forward/backward chaining.
    - Deciding entailment with Horn clauses is *linear* in the size of the KB.

Common Patterns

	Modus Pones	And Eliminate	Bidirectional	Resolution
Premises	$\alpha \Rightarrow \beta, \quad \alpha$	$\alpha \wedge \beta$	$\alpha \Leftrightarrow \beta$	$\ell_1 \vee \ldots \vee \ell_k, \quad \neg \ell_i$
Conclusion	β	α	$(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$	$\ell_1 \vee \ldots \vee \ell_{i-1} \vee \ell_{i+1} \vee \ldots \vee \ell_k$

Full Resolution Rule

$$\frac{\ell_1 \vee \ldots \vee \ell_k \qquad m_1 \vee \ldots \vee m_n}{\ell_1 \vee \ldots \vee \ell_{i-1} \vee \ell_{l+1} \vee \ldots \vee \ell_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n}$$

where ℓ_i and m_j are complementary literals

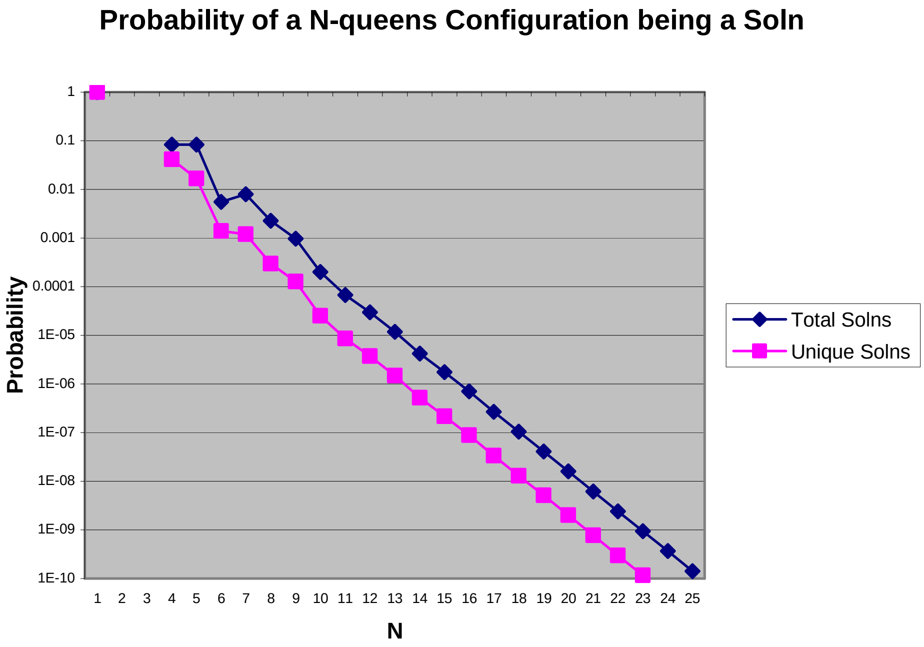
- **resolution** – a *sound* inference algorithm based on the resolution rule.
  - *By applying the only the resolution rule, any complete search algorithm can derive any conclusion entailed by any knowledge base in propositional logic (but possibly in exponential time).*
  - **refutation completeness** – resolution can be used to confirm or refute any sentence, but it cannot enumerate all true sentences.
  - **resolution algorithm**
    - to show $KB \models \alpha$ we will show that $KB \wedge \neg\alpha$ is unsatisfiable.
    - $KB \wedge \neg\alpha$ is converted into CNF… a sequence of clauses
    - The resolution rule is applied to resulting clauses… each pair with complementary literals is resolved into a new clause.
      - **factoring** – removal of redundant literals from a clause.
      - if no new clauses can be added, α is not entailed.
      - if the *empty clause* {} is derived, α is entailed.
    - Note: in applying resolution, only a single pair of literals can be negated per step of resolution

Resolution Worksheet

Solution Density

How Common are the N-queens solutions? The following table came from http://www.durangobill.com/N_Queens.html and shows the number of solutions (and unique solutions) along with their probabilities. *These probabilities are "inflated" in that I assumed the queens each had to be in separate rows or columns (N! such configurations) whereas, there are far more dumb solutions (N^2 choose N ~ $O(N^{2N})$).*

Order ("N")	Ordinary Queens Total Solutions	Ordinary Queens Unique Solutions	Probability of Total Solutions	Probability of Unique Solutions
1	1	1	1	1
2	0	0	0	0
3	0	0	0	0
4	2	1	0.083333333	0.041666667
5	10	2	0.083333333	0.016666667
6	4	1	0.005555556	0.001388889
7	40	6	0.007936508	0.001190476
8	92	12	0.002281746	0.000297619
9	352	46	0.000970018	0.000126764
10	724	92	0.000199515	2.53527E-05
11	2,680	341	6.71397E-05	8.54277E-06
12	14,200	1,787	2.9645E-05	3.73068E-06
13	73,712	9,233	1.18374E-05	1.48273E-06
14	365,596	45,752	4.19366E-06	5.2481E-07
15	2,279,184	285,053	1.74293E-06	2.17985E-07
16	14,772,512	1,846,955	7.06049E-07	8.82748E-08
17	95,815,104	11,977,939	2.6938E-07	3.36755E-08
18	666,090,624	83,263,591	1.04038E-07	1.30051E-08
19	4,968,057,848	621,012,754	4.08406E-08	5.10512E-09
20	39,029,188,884	4,878,666,808	1.60422E-08	2.00529E-09
21	314,666,222,712	39,333,324,973	6.15894E-09	7.69869E-10
22	2,691,008,701,644	336,376,244,042	2.39413E-09	2.99267E-10
23	24,233,937,684,440	3,029,242,658,210	9.3741E-10	1.17176E-10
24	227,514,171,973,736	?	3.66693E-10	
25	2,207,893,435,808,350	?	1.42342E-10	



Probability of a N-queens Configuration being a Soln