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.
  o 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.
  o 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 satisfiability – we want to find an assignment of variables that satisfy some checkable statement.
• The difficulty of a problem can characterized by the underlying satisfiability problem we are attempting to solve:
  o Suppose the sentence being satisfied has $m$ clauses and $n$ variables.
  o 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** ¬ – negation
      - **and** ∧ – conjunction
      - **or** ∨ – disjunction
      - **implies** ⇒ – implication \((α \Rightarrow β) \equiv (¬α ∨ β)\) **Note:** if \(α\) is false, \(α \Rightarrow β\) says nothing about \(β\).
      - **if and only if** ⇔ – biconditional
      - **xor** ⊕ – \((α ⊕ β) \equiv (α ∧ ¬β) ∨ (¬α ∧ β)\)
    - **order of operations** (high->low): ¬, ∧, ∨, ⇒, ⇔
  - 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: \(α \models β\) 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.

\[
KB \models α \Rightarrow KB ∧ β \models α
\]
• **conjunctive normal form** (CNF) – every sentence of propositional logic is *logically equivalent* to a conjunction of disjunctions of literals.

\[
(l_{1,1} \lor \ldots \lor l_{1,n_1}) \land \ldots \land (l_{m,1} \lor \ldots \lor l_{m,n_m})
\]

1. Eliminate biconditionals: \( \alpha \leftrightarrow \beta \equiv (\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha) \)
2. Eliminate implications \( \alpha \Rightarrow \beta \equiv \neg \alpha \lor \beta \)
3. Move \( \neg \) inwards
4. Distribute \( \land \) over \( \lor \).

• 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.

\[
\neg n_1 \lor \ldots \lor \neg n_m \lor \neg p \equiv \underbrace{n_1 \land \ldots \land n_m}_\text{body} \Rightarrow \underbrace{p}_\text{head}
\]

  o head – the positive literal.
  o body – the negative literals; the premises.
  o fact – a definite clause with no negative literals.
  o **Horn clause** – a disjunction of literals at most one of which is positive.
    
    ▪ can be written as an implication whose conclusion is \( True \).
    ▪ called an integrity constraint.

  o Inference with Horn clauses can be done with forward/backward chaining.
  o Deciding entailment with Horn clauses is *linear* in the size of the KB.
Common Patterns

<table>
<thead>
<tr>
<th></th>
<th>Modus Pones</th>
<th>And Eliminate</th>
<th>Bidirectional</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Premises</td>
<td>$\alpha \Rightarrow \beta$, $\alpha$</td>
<td>$\alpha \land \beta$</td>
<td>$\alpha \equiv \beta$</td>
<td>$\ell_1 \lor \ldots \lor \ell_k$, $\neg \ell_i$</td>
</tr>
<tr>
<td>Conclusion</td>
<td>$\beta$</td>
<td>$\alpha$</td>
<td>$(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)$</td>
<td>$\ell_1 \lor \ldots \lor \ell_{i-1} \lor \ell_{i+1} \lor \ldots \lor \ell_k$</td>
</tr>
</tbody>
</table>

Full Resolution Rule

\[
\frac{\ell_1 \lor \ldots \lor \ell_k \lor m_1 \lor \ldots \lor m_n}{\ell_1 \lor \ldots \lor \ell_{i-1} \lor \ell_{i+1} \lor m_1 \lor \ldots \lor m_{j-1} \lor m_{j+1} \lor \ldots \lor m_n}
\]

where $\ell_i$ 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 $KB \land \neg \alpha$ is unsatisfiable.
    - $KB \land \neg \alpha$ 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, $\alpha$ is not entailed.
      - if the empty clause $\{\}$ is derived, $\alpha$ 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](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 $N \sim O(N^{2N})$).*
<table>
<thead>
<tr>
<th>Order (&quot;N&quot;)</th>
<th>Ordinary Queens Total Solutions</th>
<th>Ordinary Queens Unique Solutions</th>
<th>Probability of Total Solutions</th>
<th>Probability of Unique Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0.0833333333</td>
<td>0.041666667</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>2</td>
<td>0.0833333333</td>
<td>0.016666667</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>1</td>
<td>0.005555556</td>
<td>0.001388889</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
<td>6</td>
<td>0.007936508</td>
<td>0.001190476</td>
</tr>
<tr>
<td>8</td>
<td>92</td>
<td>12</td>
<td>0.002281746</td>
<td>0.000297619</td>
</tr>
<tr>
<td>9</td>
<td>352</td>
<td>46</td>
<td>0.000970018</td>
<td>0.000126764</td>
</tr>
<tr>
<td>10</td>
<td>724</td>
<td>92</td>
<td>0.000199515</td>
<td>2.53527E-05</td>
</tr>
<tr>
<td>11</td>
<td>2,680</td>
<td>341</td>
<td>6.71397E-05</td>
<td>8.54277E-06</td>
</tr>
<tr>
<td>12</td>
<td>14,200</td>
<td>1,787</td>
<td>2.9645E-05</td>
<td>3.73068E-06</td>
</tr>
<tr>
<td>13</td>
<td>73,712</td>
<td>9,233</td>
<td>1.18374E-05</td>
<td>1.48273E-06</td>
</tr>
<tr>
<td>14</td>
<td>365,596</td>
<td>45,752</td>
<td>4.19366E-06</td>
<td>5.2481E-07</td>
</tr>
<tr>
<td>15</td>
<td>2,279,184</td>
<td>285,053</td>
<td>1.74293E-06</td>
<td>2.17985E-07</td>
</tr>
<tr>
<td>16</td>
<td>14,772,512</td>
<td>1,846,955</td>
<td>7.06049E-07</td>
<td>8.82748E-08</td>
</tr>
<tr>
<td>17</td>
<td>95,815,104</td>
<td>11,977,939</td>
<td>2.6938E-07</td>
<td>3.36755E-08</td>
</tr>
<tr>
<td>18</td>
<td>666,090,624</td>
<td>83,263,591</td>
<td>1.04038E-07</td>
<td>1.30051E-08</td>
</tr>
<tr>
<td>19</td>
<td>4,968,057,848</td>
<td>621,012,754</td>
<td>4.08406E-08</td>
<td>5.10512E-09</td>
</tr>
<tr>
<td>20</td>
<td>39,029,188,884</td>
<td>4,876,666,808</td>
<td>1.60422E-08</td>
<td>2.00529E-09</td>
</tr>
<tr>
<td>21</td>
<td>314,666,222,712</td>
<td>39,333,324,973</td>
<td>6.15894E-09</td>
<td>7.69869E-10</td>
</tr>
<tr>
<td>22</td>
<td>2,691,008,701,644</td>
<td>336,376,244,042</td>
<td>2.39413E-09</td>
<td>2.99267E-10</td>
</tr>
<tr>
<td>24</td>
<td>227,514,171,973,736</td>
<td>?</td>
<td>3.66693E-10</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>2,207,893,435,808,350</td>
<td>?</td>
<td>1.42342E-10</td>
<td></td>
</tr>
</tbody>
</table>
Probability of a N-queens Configuration being a Soln

![Graph showing the probability of a N-queens configuration being a solution for various values of N. The x-axis represents N, ranging from 1 to 25, while the y-axis represents the probability, ranging from $10^{-10}$ to 1.

The graph displays two sets of data points:
- Total Solns, represented by a blue line.
- Unique Solns, represented by a pink line.

As N increases, the probability decreases, indicating a higher likelihood of configurations being non-solutions.