Announcements

- **Seminar**: Bjarn Stroustrup, Inventor of C++
  2-3:30pm Friday (Feb 15), Zach 102

- **Programming Contest**: sponsored by ACM (and TACS)
  - project contest (not a problem solving contest)
  - 2 to 4 team members (1 graduate student allowed)
  - send email to programmingcontest@hotmail.com:
    - 1. names and emails of team members
    - 2. project topic

Overview

- Lisp tips: compiling, and measuring execution time
- Resolution in propositional logic
- Resolution examples

Homework 1 Tips

- Test all the base cases and make sure that they work, and try different orders of arguments:
  
  (deriv '+ 1 x)' x)  (deriv '+ x 1)' x)
  (deriv '* 10 x)' x)  (deriv '* x 10)' x)
  (deriv '+ 10 5)' x)  (deriv '- x)' x)
  etc.

- Include results from both (deriv ...) and (deriv-val ...)

- be careful when doing cut-and-paste from another editor into unix, because it can screw up line-breaks.

Lisp Tips

Measuring execution time: `(time (function args))`

>`(time (+ 10 20))`

real time : 0.000 secs
run time : 0.000 secs
30
## Compiling Lisp

**UNIX:**

```
gcl -compile file.lsp \rightarrow this gives you file.o
```

**GCL:**

```
> (load "file.lsp") ; vanilla lisp
> (time (ida-star '( 5 6 7 4 0 8 3 2 1 )))
... real time : 51.000 secs
run time : 42.620 secs
> (load "file.o") ; compiled lisp
Loading eight5.o
start address -T 317000 Finished loading eight5.o
11996
> (time (ida-star '( 5 6 7 4 0 8 3 2 1 )))
... real time : 4.000 secs ; BIG IMPROVEMENT!
run time : 3.650 secs
```

## Resolution: An Overview

Given formulas in conjunctive normal form $F = F_1 \land F_2 \land \ldots \land F_n$, where each $F_i$ is a **clause** *(i.e. disjunctions of literals)*, and the desired conclusion $G$, to show $G$ is a logical consequence of $F$, follow these steps:

1. **negate** $G$ and add it to the list of clauses (make it into CNF if necessary):
   
   $$F_1, F_2, \ldots, F_n, \neg G$$

2. choose two clauses that have **exactly one** pair of literals that are complementary, e.g.:
   
   $$F_n : \neg P \lor Q \lor R$$
   $$F_m : S \lor \neg P$$

3. Produce a new clause by deleting the complimentary pair and producing a new formula, e.g.:
   
   $$Q \lor R \lor S$$

4. repeat until the new clause generated is $\mathbf{F}$

This assumes that the premises are consistent.

## Resolution: An Example

Given the above, we want to prove that $E$ is true. We simply add the negation of the desired conclusion, and try to draw a contradiction:

$\neg A$ \hspace{0.5cm} (1)

$\neg B$ \hspace{0.5cm} (2)

$\neg D$ \hspace{0.5cm} (3)

$\neg A \lor \neg B \lor C$ \hspace{0.5cm} (4)

$\neg C \lor \neg D \lor E$ \hspace{0.5cm} (5)

$\neg E$ \hspace{0.5cm} (6)

## Resolution: Solution

<table>
<thead>
<tr>
<th>Given</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>(1) $\neg B \lor C$ (7)</td>
</tr>
<tr>
<td>$B$</td>
<td>(2) $C$ (8)</td>
</tr>
<tr>
<td>$D$</td>
<td>(3) $\neg C \lor \neg D$ (9)</td>
</tr>
<tr>
<td>$\neg A \lor \neg B \lor C$</td>
<td>(4)</td>
</tr>
<tr>
<td>$\neg C \lor \neg D \lor E$</td>
<td>(5)</td>
</tr>
<tr>
<td>$\neg E$</td>
<td>(6)</td>
</tr>
</tbody>
</table>
Resolution: Why Does It Work

The goal of resolution is to show that $G$ is a logical consequence of $F_1 \land \ldots \land F_n$ is valid. This is equivalent to showing that $F_1 \land \ldots \land F_n \land \neg G$ is inconsistent.

Note that if $H$ is a logical consequence of $F_1 \land \ldots \land F_n$, then $F_1 \land \ldots \land F_n = F_1 \land \ldots \land F_n \land H$:

When $F_1 \land \ldots \land F_n$ is

1. $\text{T}$ : then $H$ must also be true.
2. $\text{F}$ : both sides are false, thus $H$ does not matter.

Thus, we can add any logical consequence of $F_1 \land \ldots \land F_n$ or of any subset of the $F_i$'s without changing the value of the result. Recall that we added newly derived formulas to the list in the previous slide.

Resolution Algorithm

1. Convert premises $F_1 \land \ldots \land F_n$ into CNF, and make a list of resulting clauses.
2. Negate the conclusion, convert to CNF, and add to the clause list.
3. Resolution Step: pick two clauses from the list with exactly one complementary literal; any other literals if they appear on both clauses must have the same sign. Form a new clause by disjunction w/o the complementary literals, and add to the list.

\[
(P \lor C_i), (\neg P \lor C_j) \Rightarrow (C_i \lor C_j)
\]

$F_i$ and $F_j$

Add to list

4. If $\text{F}$ was added to the list of clauses, in step 3, stop; theorem proved. Otherwise, go to step 3.

What Resolution Is Not

If $C_1 \land C_2 \rightarrow H$

- then $C_1 \land C_2 \land H = C_1 \land C_2$
- but not $C_1 \land C_2 = H$

In other words,

\[
(C_1 \land C_2 \rightarrow H) \rightarrow ((C_1 \land C_2 \land H) \leftrightarrow (C_1 \land C_2))
\]

Exercise: Verify the above with $C_1 = (A \lor B), C_2 = (\neg B \lor C)$, and $H = (A \lor C)$.

Examples of Resolution Step

1. $F_1 = P \lor R$
   $F_2 = \neg P \lor Q$
   complimentary literals: $P, \neg P$; resolvent: $R \lor Q$

2. $F_1 = \neg P \lor Q \lor R$
   $F_2 = \neg Q \lor S$
   complimentary literals: $Q, \neg Q$; resolvent: $\neg P \lor R \lor S$

3. $F_1 = \neg P \lor Q$
   $F_2 = \neg P \lor R$
   resolution not possible: no complementary literals

4. $F_1 = \neg P \lor Q \lor S$
   $F_2 = P \lor \neg Q \lor R$
   cannot apply resolution: two complementary literals
Another Example of Resolution

Show that $Q$ is a logical consequence of $P \land (P \rightarrow Q)$:

**Premises:**
1. $P$
2. $\neg P \lor Q$
3. $\neg Q$ (negated conclusion)

**Proof:**

```
   P  \neg P \lor Q  \neg Q  \text{ or }  P  \neg P \lor Q  \neg Q
   \hline
   \begin{array}{c}
   \neg Q \\
   \hline
   F
   \end{array}
   \quad
   \begin{array}{c}
   \neg P \\
   \hline
   F
   \end{array}
```

Exercise: A Problem

**Given:**
1. If it rains, the aquaphobes won’t vote.
2. John will win *only if* the aquaphobes and vegetarians vote.
3. Either John or Peter will win, but not both.

**Prove:** If it rains, Peter will win.

Exercise: English to Propositional Logic

**Given:**
1. If it rains ($R$), the aquaphobes won’t vote ($\neg A$).
   $$R \rightarrow \neg A$$

2. John will win ($J$) *only if* the aquaphobes and vegetarians vote ($A \land V$).
   $$\neg (A \land V) \rightarrow \neg J, \quad \text{i.e.} \quad J \rightarrow (A \land V)$$

3. Either John ($J$) or Peter ($P$) will win, but not both.
   $$(J \rightarrow \neg P) \land (\neg J \rightarrow P), \quad \text{i.e.} \quad J \leftrightarrow P$$

**Prove:** If it rains ($R$), Peter will win ($P$), i.e. $R \rightarrow P$

Negated, this becomes

4. $\neg (R \rightarrow P)$.

Exercise: In CNF, and Actual Proof

**Given:**
1. $\neg R \lor \neg A$
2. (a) $\neg J \lor A$, (b) $\neg J \lor V$
3. (a) $\neg J \lor \neg P$, (b) $J \lor P$
4. (a) $R$, (b) $\neg P$

**Resolution steps**

<table>
<thead>
<tr>
<th>Resolution steps</th>
<th>Expression</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>3b,4b</td>
<td>$J$</td>
<td></td>
</tr>
<tr>
<td>2a,5</td>
<td>$A$</td>
<td></td>
</tr>
<tr>
<td>6,1</td>
<td>$\neg R$</td>
<td></td>
</tr>
<tr>
<td>4a,7</td>
<td>$F$</td>
<td></td>
</tr>
</tbody>
</table>

13

14

15

16
**Exercise: Exercise 6.6, p. 181**

Given:

If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

Prove or disprove:

1. The unicorn is mythical.
2. The unicorn is magical.
3. The unicorn is horned.

Hint: do not use redundant terms.

---

**Representing Resolution Problems in LISP**

Separate positive vs. negative literals

$\langle \text{clause-number} \rangle \langle \text{pos-literal-list} \rangle \langle \text{neg-literal-list} \rangle$

Example:

1. $\neg R \lor \neg A$
   
   (1 NIL (A R ))

2. (a) $\neg J \lor A$, (b) $\neg J \lor V$
   
   (2 (A) (J ))

3. (a) $\neg J \lor \neg P$, (b) $J \lor P$
   
   (3 (V) (J ))

4. (a) $R$, (b) $\neg P$
   
   (4 NIL (J P ))

Comparison to find resolvents become easy. When resolution reaches $\langle \text{n} \rangle \text{NIL} \text{NIL}$, the theorem has been proved.

---

**Key Points**

- understand why resolution works: proof by contradiction
- standard formulas used in resolution (CNF, and negation of conclusion).
- know how to follow resolution steps to prove (or disprove) a theorem.
- given a natural language description, know how to convert to propositional logic formulas.

---

**Next Time**

Chapter 7 (7.1–7.3) and parts of Chapter 9

- Predicate calculus (first-order logic)
Limitation of Propositional Logic

Propositional logic does not allow us to perform any reasoning based on the use of general rules, so its usefulness is limited. Predicate Calculus generalizes Propositional Calculus to allow the expression and use of general rules.

New concepts are introduced:

- terms
- predicates
- quantifiers

Formulas in Predicate Calculus

- Variables: $x, y, z, \ldots$
- Functions: $f(x), g(y), h(z), \text{father}(\text{John}), \ldots$
  - maps \text{constant(s)} to a \text{constant}
- Constants: John, Mary, 3
- Predicates:
  - $P(x, y), \text{GREATER}(x, 3), \text{LOVE}(\text{father}(\text{John}), \text{John})$
  - function whose value is $T$ or $F$
- Quantifiers: $\forall$ (for all), $\exists$ (there exists)

Terms in Predicate Calculus

A \text{T}erm is:

- constant
- variable
- $f(t_1, \ldots, t_n)$, where $f$ is a function symbol and $t_1, t_2, \ldots, t_n$ are terms.

Terms refer to objects in a domain.
In Predicate Calculus

- $P$: All men are mortal $\forall x, \text{MAN}(x) \rightarrow \text{MORTAL}(x)$
- $Q$: Socrates is a man $\text{MAN}(\text{Socrates})$
- $R$: Socrates is mortal $\text{MORTAL}(\text{Socrates})$