Computing Fundamentals 1
Lecture 3

Lecturer: Patrick Browne
http://www.comp.dit.ie/pbrowne/
Room K308

Based on Chapter 3 Propositional Calculus.
A Logical approach to Discrete Math
By David Gries and Fred B. Schneider
Need for formalism

• It is difficult to reason in natural language:
• “The sun is shinning and I feel happy”\(^1\).
• Are the two facts related, do I feel happy because the sun shining?
• “Roses are red and violets are blue”
• Are roses red because violets are blue?
• Both sentences have the same form, two facts connect with ‘and’.

\(^1\)Note: These examples are taken from the text for demonstration purposes.
Need for formalism

- There is no known way to formally handle all the ambiguities of English. So, we need a separate ‘logical language’ where many of the ambiguities are filtered out.
- We start with propositional logic.
- There are several mathematical systems (calculi) for reasoning about propositions:
  - Truth tables (semantic), Equational Logic (syntactic/semantic) which we use on this course, Boolean Algebra (axiomatic), Natural Deduction (can be used as an alternative to equational logic)
Propositional Calculus

• Boolean expressions can be defined:
  – By how they are evaluated or
  – By how they can be manipulated.
• To help use manipulate Boolean expressions we need proofs, heuristics, and principles.
• A calculus is a method or process of reasoning by calculation with symbols.
• The propositional calculus uses Boolean expressions and propositional variables.
Propositional Calculus

• A calculus is a method of reasoning by calculating with symbols. Logic is the study of correct reasoning. Equational reasoning is a style of logic. The propositional calculus used on this course is called *equational logic* $\mathbf{E}$.

• $\mathbf{E}$ has two parts:
  – A set of axioms,
  – Four inference rules
Propositional Calculus

- The axioms are Boolean expressions that define the basic manipulative properties of the Boolean operators. For example,
  - \( p \lor q \equiv q \lor p \)
  - Indicates that \( \lor \) is symmetric.
  - An inference rule asserts the if the premises are assumed to be truths, then the conclusion is also a truth. On the next slide are the four inference rules for \( E \).
Four Inference Rules for $E$

- **Leibniz:**
  
  $P = Q$
  
  $E[r := P] = E[r := Q]$

- **Transitivity:**
  
  $P = Q, Q = R$
  
  $P = R$

- **Substitution**
  
  $P$
  
  $P[r := Q]$

- **Equanimity**
  
  $E$, $E \equiv F$
  
  $F$

**Example**

- $X + 1 > 0$

  (Example)

- $(X + 1 > 0)[x := 3]$

  $4 > 0$
CafeOBJ's version of Equational Logic

5.1 Equational logic proof calculus (CafeOBJ)

We present the equational logic proof calculus with respect to a fixed equational specification $SP$. This consists of the following inference rules:

[reflexivity] \[
\frac{}{(\forall X) \, t = t}
\]

[symmetry] \[
\frac{(\forall X) \, t = t'}{(\forall X) \, t' = t}
\]

[transitivity] \[
\frac{(\forall X) \, t = t' \quad (\forall X) \, t' = t''}{(\forall X) \, t = t''}
\]

[congruence] \[
\frac{(\forall X) \, t_i = t'_i \text{ for } i \in [n]}{(\forall X) \, \sigma(t_1, \ldots, t_n) = \sigma(t'_1, \ldots, t'_n)}
\]

for all operations $\sigma \in sign(SP)_{s_1, \ldots, s_n}$, and $t_i$ of sort $s_i$ for $i \in [n]$.

[substitutivity] \[
\frac{(\forall X) \, \theta(C) = \text{true}}{(\forall X) \, \theta(t) = \theta(t')}
\]

where $(\forall Y) \, t = t'$ if $C$ is any equation in $SP$ and $\theta: Y \rightarrow T_{sign(SP)}(X)$ is any substitution.
Reminder on observation & interpretation

Transitivity

Observable Reality
Reminder on Interpretation

• Our **scientific theories** about the world can be expressed using mathematics and logic. The **models** of these theories often match the actual world. But our scientific theories can be wrong.
Reminder on Interpretation

• Before 1674 it was generally though that that the main source of water in rivers was from underground springs. In 1674 Pierre Perrault measured and compared rainfall and river flow. He found that the annual rainfall of part of a river catchment exceeded the annual river flow and that no underground water supply was needed to sustain the river. Other reasonable theories had been published previously (Bernard Plassy, 1580), but in general it was thought that rivers were maintained from underground sources rather than rainfall.

• In this case does measurement validate Perrault theory?
Reminder on Interpretation

• We use the following definitions:
  Volume of underground source (Underground)
  Volume rainfall (Rainfall)
  Volume flowing in river (Riverflow)

• Assume that there are only two possible sources. Assume only Rainfall and RiverFlow are measured. A working assumption could be that the volume of the sum of sources is greater or equal to river flow:
  (Underground + Rainfall) ≥ RiverFlow

• Perrault’s experiment tells us: Rainfall > RiverFlow

• Would the above allow us to conclude anything about the value of Underground in this experiment?
Reminder on Interpretation

• Can Perrault’s experiment be used to **prove** that rivers are fed by rainfall only?
• Can Perrault’s experiment be used to **prove** that the flow of a river does not depend on any underground source?
• Are any of the following true?

\[(\text{Rainfall} > \text{RiverFlow}) \Rightarrow (\text{Underground} < \text{RiverFlow})\]

\[(\text{Rainfall} > \text{RiverFlow}) \Rightarrow (\text{Underground} > \text{RiverFlow})\]
Theorems in E

• A theorem of the prepositional calculus E is either:
  – An axiom
  – The conclusion of an inference rule whose premises are theorems
  – A Boolean expression, that using the rules of inference is proved to be equal to an axiom or a previously proved theorem.
Proof Method in E

• To prove that $P \equiv Q$ is a theorem (where $P$ and $Q$ are expressions e.g. $P$ is say $a \lor b$ and $Q$ is $b \lor a$), transform $P$ to $Q$ or $Q$ to $P$ using Leibniz.

• Proofs in E have the form:

\[
\begin{align*}
P \\
< \text{Hint for proof step 1} > \\
R \\
. \\
. \\
< \text{Hint for proof step n} > \\
Q
\end{align*}
\]
Helpful Hints

• Focus on ‘manipulation’ rather than ‘evaluation’. These are syntactic operations that do not consider meaning.

• There are many axioms and theorems, you do not have to memorise them. Details of precedence rules, theorems, axioms, etc. will be supplied for examination questions.
The Axioms of equational propositional logic.

• The next slide present the main axioms of equational propositional logic. The full list is at:

• You are not expected to learn these off by heart! But you should be able to see how they are used in a proof.
### Table 2: Axioms of Logic E

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Associativity of</strong> (\equiv): ((p \equiv q) \equiv r) (\equiv) ((p \equiv (q \equiv r)))</td>
</tr>
<tr>
<td>2</td>
<td><strong>Symmetry of</strong> (\equiv): (p \equiv q \equiv q \equiv p)</td>
</tr>
<tr>
<td>3</td>
<td><strong>Identity of</strong> (\equiv): (\text{true} \equiv q \equiv q)</td>
</tr>
<tr>
<td>4</td>
<td><strong>Definition of</strong> (\text{false}): (\text{false} \equiv \neg \text{true})</td>
</tr>
<tr>
<td>5</td>
<td><strong>Distributivity of</strong> (\neg) <strong>over</strong> (\equiv): (\neg(p \equiv q) \equiv \neg p \equiv q)</td>
</tr>
<tr>
<td>6</td>
<td><strong>Definition of</strong> (\neq): ((p \neq q) \equiv \neg(p \equiv q))</td>
</tr>
<tr>
<td>7</td>
<td><strong>Associativity of</strong> (\lor): ((p \lor q) \lor r \equiv p \lor (q \lor r))</td>
</tr>
<tr>
<td>8</td>
<td><strong>Symmetry of</strong> (\lor): (p \lor q \equiv q \lor p)</td>
</tr>
<tr>
<td>9</td>
<td><strong>Idempotency of</strong> (\lor): (p \lor p \equiv p)</td>
</tr>
<tr>
<td>10</td>
<td><strong>Distributivity of</strong> (\lor) <strong>over</strong> (\equiv): (p \lor (q \equiv r) \equiv p \lor q \equiv p \lor r)</td>
</tr>
<tr>
<td>11</td>
<td><strong>Excluded Middle</strong>: (p \lor \neg p)</td>
</tr>
<tr>
<td>12</td>
<td><strong>Golden rule</strong>: (p \land q \equiv p \equiv q \equiv p \lor q)</td>
</tr>
<tr>
<td>13</td>
<td><strong>Implication</strong>: (p \Rightarrow q \equiv p \lor q \equiv q)</td>
</tr>
<tr>
<td>14</td>
<td><strong>Consequence</strong>: (p \Leftarrow q \equiv q \Rightarrow p)</td>
</tr>
<tr>
<td>15</td>
<td><strong>Anti-implication</strong>: (p \nRightarrow q \equiv \neg(p \Rightarrow q))</td>
</tr>
<tr>
<td>16</td>
<td><strong>Anti-consequence</strong>: (p \nLeftarrow q \equiv \neg(p \Leftarrow q))</td>
</tr>
</tbody>
</table>

Note these **axiom numbers** differ from those in the book.
Heuristics

- **Heuristic**: (3.21) Identify an applicable theorem from table 2 by matching its structure to the given expressions or sub-expressions to be proved. The operators that appear in a Boolean expression and the shape of its sub-expressions can focus the choice of theorems, from table 2, that can be used to manipulate it.

- **Principle**: Structure proofs to avoid repeating the same sub-expression many times.
A Proof

• Prove theorem: \( \neg p \equiv q \equiv p \equiv \neg q \)

\[
\begin{align*}
\neg p & \equiv q \equiv p \equiv \neg q \\
\neg (p \equiv q) & \equiv p \equiv \neg q \\
\neg (p \equiv q) & \equiv \neg (p \equiv q) \\
\text{true}
\end{align*}
\]

= <axiom(3.9,5), \neg (p \equiv q) \equiv \neg \neg p \equiv \neg q>

= <axiom(3.9,3.17), with p,q:=q,p \neg (q \equiv p) \equiv \neg q \equiv p>

= <(3.5,3.8) Reflexivity of \equiv >
(3.33) **Heuristic:** To prove $P \equiv Q$, transform the expression with the most structure (either $P$ or $Q$) into the other. To illustrate this heuristic we prove: $p \lor \text{true} \equiv \text{true}$. Start with LHS.

\[
p \lor \text{true} = \langle (3.3) \text{ Identity of } \equiv \rangle \\
p \lor (p \equiv p)
\]

\[
= \langle (3.27, 3.32) \text{ Distributivity of } \lor \text{ over } \equiv \rangle \\
p \lor p \equiv p \lor p
\]

\[
= \langle (3.3, 3.7) \text{ Identity of } \equiv \rangle \\
\text{true}
\]

Which is the same as the RHS.
Heuristic

(3.23) **Heuristic of definition of elimination**: To prove a theorem concerning an operator $\circ$ that is defined in terms of another operator $\bullet$, expand the definition of $\circ$ to arrive at a formula that contains $\bullet$: exploit the properties of $\bullet$ to manipulate the formula and then (possibly) reintroduce $\circ$ using its definition.
Using a Heuristic

To illustrate the heuristic (3.23) we prove

\[(p \not\equiv q) \equiv (q \not\equiv p),\] where \(\not\equiv\) is \(\not\equiv\) and \(\equiv\) is \(\equiv\)

LHS \((p \not\equiv q)\)

= \(< (3.10)\) Def of \(\not\equiv >\)

\(\neg (p \equiv q)\)

= \(<(3.2)\) Symmetry of \(\equiv >\)

\(\neg (q \equiv p)\)

= \(< (3.10)\) Def of \(\not\equiv\) with \(p, q:=q, p>\)

\((q \not\equiv p)\) RHS
Equational Proof

- Prove that \(-X\) is a right inverse of +. Assume true, then proof produces the identity 0:

1. \(X + (-X)\)
   <reverse of EQ1>

2. \(0 + X + (-X)\)
   <reverse of EQ2, substituted by \((-X)\)>

3. \((-(-X)) + (-X) + X + (-X)\)
   <EQ2>

4. \((-(-X)) + 0 + (-X)\)
   <EQ1>

5. \((-(-X)) + (-X)\)
   <EQ2>

6. 0
The Golden rule\(^1\)

- \(p \land q \equiv p \equiv q \equiv p \lor q\)

- On one hand the Golden Rule\(^2\) can be seen as a **definition** of conjunction (\(\land\)) in terms of equivalence (\(\equiv\)) and disjunction (\(\lor\)):
  
  \[
  (p \land q) \equiv (p \equiv q \equiv p \lor q)
  \]

- But it can also be read in other ways, e.g. assertion of **equality**:
  
  \[
  (p \land q \equiv p) = (q \equiv p \lor q)
  \]

- This can be used to define logical implication.

---

Check that GR is an in CafeOBJ module PROPC:

```cafeobj
set trace on whole
red p \land q <\rightarrow p <\rightarrow q <\rightarrow p \lor q .
```
Golden Rule definition of $\land$

\[ p \land q \equiv p \iff q \iff p \lor q \]

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>$p \land q \equiv p \iff q \iff p \lor q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

\[ P \quad Q \quad (P \land Q) \quad (P \equiv Q) \quad (P \lor Q) \]

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>$(P \land Q)$</th>
<th>$(P \equiv Q)$</th>
<th>$(P \lor Q)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

26
Golden Rule

\[ p \land q \equiv p \equiv q \equiv p \lor q . \]

The golden rule can be seen as a definition of conjunction in terms of equivalence and disjunction if we read it as

\[ (p \land q) = (p \equiv q \equiv p \lor q) . \]

But it can also be read in other ways. For example, the golden rule asserts the equality

\[ (p \land q \equiv p) = (q \equiv p \lor q) . \]

This reading can be used when we define logical implication. It can also be read as a definition of disjunction in terms of conjunction:

\[ (p \land q \equiv p \equiv q) = (p \lor q) . \]

This reading is sometimes useful when in a calculation it is expedient to replace disjunctions by conjunctions.
The Golden rule\(^1\)

- The GR can also be read as a definition of disjunction (\(\lor\)) in terms of conjunction (\(\land\))
  \[(p \land q \equiv p \equiv q) \equiv (p \lor q)\]
- This form can be useful in calculations replacing disjunctions by conjunctions.
- GR can say \(p\) and \(q\) are equal iff their conjunction & disjunction are equal.
  \[(p \equiv q) \equiv (p \land q \equiv p \lor q)\]
Using Golden Rule

• Since the GR is the definition of $\wedge$, it can be used with the heuristic of **Definition Elimination** (3.23). The GR can be used to eliminate $\wedge$. The next slide shows a proof of:

$$p \wedge (\neg p \lor q) \equiv p \wedge q$$
Using Golden Rule(*)

- Prove: \( p \land (\neg p \lor q) \equiv p \land q \)
- LHS

\[
\begin{align*}
\text{LHS} & \quad \text{Using Golden Rule(\(\ast\))} \\
\ & \quad \text{Prove: } p \land (\neg p \lor q) \equiv p \land q \\
\ & \quad \text{LHS} \\
\ & \quad p \land (\neg p \lor q) \\
\ & \quad = < (3.35) \text{ GR, with } q:= \neg p \lor q > \\
\ & \quad p \equiv \neg p \lor q \equiv p \lor \neg p \lor q \\
\ & \quad = < (3.28) \text{ Excluded Middle } > \\
\ & \quad p \equiv \neg p \lor q \equiv \text{true} \lor q \\
\ & \quad = < (3.29) \text{ Zero for } \lor > \\
\ & \quad p \equiv \neg p \lor q \equiv \text{true} \\
\ & \quad = < (3.3) \text{Identity of } \equiv > \\
\ & \quad p \equiv \neg p \lor q \\
\ & \quad = < (3.32) p \lor q \equiv p \lor \neg q \equiv p, \text{ with } p,q:=q,p > \\
\ & \quad p \equiv p \lor q \equiv q \\
\ & \quad = < (3.35) \text{ GR} > \\
\ & \quad p \land q \\
\end{align*}
\]

Which is RHS.

Check that substitution is justified LHS=RHS:

\[
\text{Using GR as definition of ‘and’, the first inference substitutes the } q:= \neg p \lor q \text{ in GR} \\
\text{LHS } = \text{ RHS} \\
p \land q \equiv p \equiv q \equiv p \lor q \text{ (GR)} \\
p \equiv (\neg p \lor q) \equiv p \lor (\neg p \lor q) \\
\text{Hence the } \land \text{ gone after step 1.}
\]

Axiom, Golden rule: \( P \land Q \equiv P \equiv Q \equiv P \lor Q \)

Golden Rule \( (p \land q) = (p \equiv q \equiv p \lor q) \)
DeMorgan and Monotonicity

• 3.47(a)(b) DeMorgan:

\[ \neg (p \land q) \equiv \neg p \lor \neg q \]
\[ \neg (p \lor q) \equiv \neg p \land \neg q \]

• 4.2 Monotonicity of \( \lor \):

\[ (p \Rightarrow q) \Rightarrow (p \lor r \Rightarrow q \lor r) \]
Implication and Consequence

- Axiom, Definition of Implication:
  \[ p \implies q \equiv p \lor q \equiv q \]

- Axiom, Definition of Consequence:
  \[ p \iff q \equiv q \implies p \]

- Rewriting Implication
  \[ p \implies q \equiv \neg p \lor q \]
  \[ p \implies q \equiv p \land q \equiv p \]

- Contrapositive:
  \[ p \implies q \equiv \neg q \implies \neg p \]

- Negation
  \[ \neg (p \implies q) \equiv p \land \neg q \]

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>\neg(p \implies q) \leftrightarrow (p \land \neg q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
Implication, Converse, Inverse

- Original statement: \( p \Rightarrow q \)
- Converse: \( q \Rightarrow p \)
- Inverse: \( \neg p \Rightarrow \neg q \)
- Contrapositive: \( \neg q \Rightarrow \neg p \)
- If a statement is true, the contrapositive is also logically true.
- If the converse is true, the inverse is also logically true.
Negating Implication

- Original statement: $p \Rightarrow q$
- Implication as disjunction: $\neg p \lor q$
- Contrapositive: $\neg (\neg p \lor q)$
- DeMorgan: $p \land \neg q$
Implication, Converse, Inverse

- From a given implication we get three derived implications (inverse, converse, inverse).
- \( P \rightarrow Q \) (original implication)
- \(-Q \rightarrow -P\) (contrapositive of original)
- \( Q \rightarrow P \) (converse of original)
- \(-P \rightarrow -Q\) (inverse of original)
- They all obey the truth table for implication. However, the derived implications contain negations and/or are in a different order than the original implication.
Implication (⇒), or (∨)

• The statement (P ⇒ Q) is equivalent to (not P ∨ Q).
• “If I finish this chapter then I will learn”
• This is equivalent to saying,
• “I do not finish this chapter or I will learn.”

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>(p → q)</th>
<th>(¬p ∨ q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>(p → q)</th>
<th>(¬p ∨ q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Implication ($\Rightarrow$), or ($\lor$)

- The statement \((\text{not } P) \Rightarrow Q\) is propositionally equivalent to \(P \lor Q\).
- “If I don't finish this chapter, I'm in trouble"
- This is equivalent to saying,
- “I (must) finish this chapter or I'm in trouble.”
Negation of $\Rightarrow$ in terms of $\wedge$

- The negation of an implication is an and statement:
  - $\neg(p \text{ implies } q) = (p \text{ and } \neg q)$

$$\neg(p \rightarrow q) \iff (p \land \neg q)$$

- Example: “if I hit my thumb with a hammer, then my thumb will hurt” The negation is: “I hit my thumb with a hammer and my thumb does not hurt.”
Truth Table for Contrapositive

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>q</td>
<td>p → q</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>p</td>
<td>q → p</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

In General
statement: if p then q
converse: if q then p
inverse: if not p then not q
contrapositive: if not q then not p

expression is a **tautology**
Truth Table for Implication

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P \implies Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
</tr>
</tbody>
</table>

P is stronger than Q

In General
statement: if p then q
converse: if q then p
inverse: if not p then not q
contrapositive: if not q then not p

Q is weaker than P.

When \( \implies \) is true then Q is true in more than cases P.
Weakening & Strengthening

• Suppose $P \Rightarrow Q$. Then we say that $P$ is stronger than $Q$ and $Q$ is weaker than $P$. This because $Q$ is true in more (or at least the same) states than $P$. $P$ imposes more restrictions on a state. The strongest formula is false (true in fewer case) and the weakest is true (true in more case).

• The following theorems are called weakening or strengthen depending on whether it is used to transform the antecedent into the consequent, thus weakening it, or transform the consequent into the antecedent, thus strengthening it.
Weakening & Strengthening

• $p \Rightarrow p$ (starting with)
• $p \Rightarrow p \lor q$ (weaken RHS)
• $p \land q \Rightarrow p$ (strengthen LHS)
• $p \land q \Rightarrow p \lor q$ (strg/weak)
• $p \lor (q \land r) \Rightarrow (p \lor q)$
• $(p \land q) \Rightarrow p \lor (q \land r)$
Modus ponens

• Modus ponens.
  \[ p \land (p \implies q) \implies q \]

• In many propositional calculi MP is a major inference rule. It is somewhat less important in E where the emphasis is on equational reasoning.
(3.33) **Heuristic:** To prove $P \equiv Q$, transform the expression with the most structure (either $P$ or $Q$) into the other.

(3.34) **Principle:** Structure proofs to minimize the number of rabbits pulled out of the hat – make each step seem obvious, based on the structure of the expression and the goal of the manipulation.

(3.54) **Principle:** Lemmas (small proofs) can provide structure, bring to light interesting facts, and should shorten proofs.
Implication

• Is the following valid? i.e. true in all states.
• $x > 2 \implies x > 0$
• What about when $x$ is 0?
• What about when $x$ is 1?
• Can we use a truth table?
Definition of implication 3.60

• $3.60 \ p \Rightarrow q \equiv p \land q \equiv p$

• Use TTG

• $(p \Rightarrow q) \iff (p \lor q) \iff p$

  $p \quad q$  Result is a tautology
  \[
  \begin{array}{cccc}
  F & F & T \\
  T & F & T \\
  F & T & T \\
  F & T & T \\
  T & T & T \\
  \end{array}
  \]
### Definition of implication 3.60

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p \implies q</th>
<th>p \land q</th>
<th>p \implies q \equiv p \land q \equiv p</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
</tbody>
</table>

Equivalence is associative and has the lowest precedence. If we evaluate the first line left to right we get 

\( (T \equiv F) \equiv F \), which gives true

If we evaluate the first line right to left we get 

\( T \equiv (F \equiv F) \), which gives true

The two evaluations are the same which demonstrated the associative property.
Proof Implication Theorem

• Prove: \( p \Rightarrow (q \equiv r) \equiv p \land q \equiv p \land r \)

• Start with LHS

\[
p \Rightarrow (q \equiv r) \\
= < \text{Def of Implication (3.60)} > \\
p \land (q \equiv r) \equiv p \\
= < \text{(3.49) relates and equivalence } > \\
p \land q \equiv p \land r
\]

• Which gives us RHS.

• Prove this by truth table
<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>r</th>
<th>p \land q</th>
<th>p \land r</th>
<th>LHS</th>
<th>(q \equiv r)</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Abbreviation for proving implication

- This abbreviation allows us to use \( \Rightarrow \) as well as \( = \) in proof steps.
- Given \( (p \equiv q) \text{ and } (q \Rightarrow r) \)
- We show \( (p \Rightarrow r) \) holds.

We want to prove \( (p \equiv q) \land (q \Rightarrow r) \Rightarrow (p \Rightarrow r) \)

\[
\begin{align*}
&= \text{<Why? } p \equiv q \equiv \text{true}> \\
&\quad \text{true} \land (q \Rightarrow r) \Rightarrow (p \Rightarrow r) \\
&= \text{<Why? } q \Rightarrow r \equiv \text{true}> \\
&\quad \text{true} \land \text{true} \Rightarrow (p \Rightarrow r) \\
&= \text{< (3.38 and 3.73) >} \\
&\quad p \Rightarrow r \text{ (LHS true gives RHS)}
\end{align*}
\]

3.38 idempotency of \( \land : p \land p \equiv p \)
3.73 left identity of \( \Rightarrow : \text{true} \Rightarrow p \equiv p \)
Abbreviation for proving implication

• Previously, we only used $=$ in the proof steps, now we can use $\Rightarrow$.
• A shorter proof of previous slide, given:
• $(p \equiv q)$ and $(q \Rightarrow r)$
• We prove that $(p \Rightarrow r)$ holds.

\[
\begin{align*}
  p &<\text{Why? } p \equiv q> \\
  q &<\text{Why? } q \Rightarrow r > \\
  r &
\end{align*}
\]
• Note we use $\Rightarrow$ in the second proof step.
How does CafeOBJ evaluate expressions?

• To find out, type in the following:
  open BOOL
  set trace on
  red true xor false xor true and false .

• To further clarify the order of evaluation look at the precedence of 'and' and 'xor' by using the following command:
  show module BOOL .

• Note: 1) the lower the number the higher the precedence and 2) the associativity of 'and' and 'xor'.
How does CafeOBJ evaluate expressions?

- CafeOBJ uses left to right re-write rules to evaluate terms. A term can be a variable or a function name applied to zero or more arguments e.g., `add(X,1)`. Zero argument functions are called constants. More complex terms can be built from a vocabulary of function and variable symbols. Terms can be considered as simple strings. A term is well-formed if it has exactly one least parse tree. A ground term has no variables e.g. `a`. 

![Diagram showing an example term]

```
term = f(a + x, h(f(a, b))):
```

53
How does CafeOBJ evaluate expressions?

- CafeOBJ uses left to right re-write rules to evaluate terms. A term can be a variable or a function name applied to zero or more arguments e.g., `add(X,1)`. Zero argument functions are called constants. More complex terms can be built from a vocabulary of function and variable symbols. Terms can be considered as simple strings. A term is well-formed if it has exactly one least parse tree. A ground term has no variables e.g. `a`.

```
term = f(a + x, h(f(a, b))): a
```

How does CafeOBJ evaluate expressions?

- Substitution is a mapping from variables to terms. Substitution are specified by equations. Example: let \( \sigma = \{(x \mapsto f(a, z))\}\) be a substitution function. If we apply \( \sigma \) on the term
- \( t = g(a, g(x, x)) \) we get term
- \( (t) = g(a, g(f(a, z), f(a, z))) \)

\[\text{Substitution.} \quad \text{A substitution is a set } S \text{ of pairs } (v,t) \text{ of a variable } v \text{ and a (ground) term } t, \text{ such that no two pairs contain the same variable. If } (v,t) \text{ is in } S, t \text{ is said to be the value of } v \text{ under } A.\]

\[\text{Instantiation.} \quad \text{Given a substitution } S \text{ and a term } t, \text{ replacing each (occurrence of each) variable in } t \text{ with its value under } S \text{ is called the instantiation of } t \text{ by } S.\]

\[\text{Matching.} \quad \text{Given a ground term } t \text{ and a term } p, \text{ if there is a substitution } S \text{ such that the instantiation of } p \text{ by } S \text{ equals } t, t \text{ is said to be matchable to } p. \text{ To compute such a substitution is called matching.}\]
How does CafeOBJ evaluate expressions?

• Left to right rewriting does not exactly match equational logic. In equational logic the inference rules used for deducing “behaviour” are those familiar for reasoning about equalities, namely, for all $x, y, \text{ and } z$

  • $x = x,$
  • if $x = y,$ then $y = x,$
  • if $x = y$ and $y = z,$ then $x = z,$ and
  • we may “substitute equals for equals”, e.g., if $x = y,$ then $f(x) = f(y),$ etc.
How does CafeOBJ evaluate expressions?

• CafeOBJ uses left to right re-write rules to evaluate terms.
• A term can be a constant, a variable or a function name applied to zero or more arguments e.g., add(X,Y). More complex terms can be built from a vocabulary of function symbols and variable symbols. Terms can be considered as simple strings.
• Term rewriting is a computational method that is based on the repeated application of simplification rules.
• TRS provides methods of replacing sub-terms of a formula with other terms.
• The CafeOBJ keyword for evaluation is reduce, which can be abbreviated to red at the CafeOBJ command line.
• See Section 6.1.1 of the CafeOBJ Manual.
When computing an equivalence relation over the terms we determine which terms give the same result, and which terms give different results.
Initial algebra for a stack and its specification
In the previous slide the proof that $1 + 1 == 2$, which is written as $p\ s\ s\ s\ 0 == s\ 0 + s\ 0$, where $p$ is the predecessor, and $s$ is the successor of a natural number ($p\ 0$ is not defined). Note the equality predicate (denoted as $==$) defines an equivalence relation and is distinct from definitional equality (denoted as $=$) used in equational definitions (e.g. eq $\text{N} + 0 = 0$).
An algebra with two sorts

mod! TWOSORTS {
  [ T V ]
  op f : T -> T
  op g : T -> V

  -- Specific elements of T
  ops t1 t2 t3 t4 t5 : -> T

  -- Specific elements of V
  ops v1 v2 v3 v4 : -> V

  -- equations for f
  eq f(t1) = t3 .
  eq f(t3) = t4 .
  eq f(t4) = t5 .

  -- equations for g
  eq g(t1) = v1 .
  eq g(t2) = v2 .
  eq g(t4) = v4 .
  eq g(t5) = v4 .
}

open TWOSORTS
**> Some reductions
red f(t1) .
red f(f(f(t1))) .
red g(t4) == g(t5) .
Superman Example

• If Superman were able and willing to prevent evil, he would do so. If Superman were unable to prevent evil, he would be ineffective; if he were unwilling to prevent evil, he would be malevolent. Superman does not prevent evil. If Superman exists, he is neither ineffective nor malevolent. Therefore Superman does not exist.
Superman Example

\[ a: \text{Superman is able to prevent evil.} \]
\[ w: \text{Superman is willing to prevent evil.} \]
\[ i: \text{Superman is ineffective.} \]
\[ m: \text{Superman is malevolent.} \]
\[ p: \text{Superman prevents evil.} \]
\[ e: \text{Superman exists.} \]
Superman Example

The first four sentences:

F0 : \( a \land w \implies p \)
F1 : \((\neg a \implies i) \land (\neg w \implies m)\)
F2 : \(\neg p \)
F3 : \(e \implies \neg i \land \neg m\)

Superman argument is equivalent to:

\[ \text{ALL=} \quad F0 \land F1 \land F2 \land F3 \implies \neg e \]

Note that \( F3 \) is the only place \( e \) appears.
Superman Proof Example

Six steps. Assume ALL = F0 ∧ F1 ∧ F2 ∧ F3.

Start at conclusion ¬e.

¬e

⇔ <Contrapositive (3.61) ¬(¬i∧¬m) ⇒ ¬e of F3>
¬(¬i∧¬m)
= < DeMorgan (3.47a)>

i ∨ m
⇔ <First conjunct of F1, monotonicity 4.2>
¬a ∨ m

⇔ <Second conjunct of F1, monotonicity 4.2>

¬a ∨ ¬w

= <DeMorgan (3.47a)>

¬(a ∧ w)

⇔ <Contrapositive (3.61) ¬p ⇒ ¬(a∧w)) of F0>
¬p (this is F2)

(We conclude that ALL is a theorem so the argument of the superman paragraph is sound)
Make Truth Table for Superman

- Enter the following expression into truth table generator at
  http://turner.faculty.swau.edu/mathematics/materialslibrar y/truth/

(((a & w) > p) & (~a > i) & (~w > m) & (e > (~i & ~m)) & ~p) > (~e)

Describe the result.
red e implies (not(i) and not(m))
  iff (not(not(i) and not(m))
       implies (not(e)))

• English (version 1)
• SM existence implies that he is not ineffective and not malevolent if and only if SM is ineffective or SM is malevolent implies that he does not exist.
CafeOBJ Superman in English Example

red (e implies (not(i) and not(m)))
implies (not(not(i) and not(m)))
implies (not(e))

• English (version 2)

• If Superman exists then he is not ineffective and not malevolent, which using the contrapositive rule, means that if he is ineffective or is malevolent then he does not exist.
Superman Proof in CafeOBJ

- Note in the following proof we do not need the truth values of the equations from the SUMPERMAN module in the previous slide. We only need the constants representing the sentences. The proof uses the given equations and general theorems of equational logic. The proof does not depend on the value of the equations in the SUPERMAN module, hence their truth values are not used directly. The following reductions (or evaluations) validate the 6 steps
Superman Proof in CafeOBJ

open SUPERMAN.

-- Step 1 : 3.61 Contrapositive of F3
red e implies (not(i) and not(m)) implies
  (not(not(i) and (not(m)) implies not(e))).

-- Step 2 : 3.47a DeMorgan
red not(not(i) and not(m)) iff (i or m).

-- Step 3 : First conjunct of F1 and
  Monotonicity 4.2 ('or m' added)
red (not(a) implies i) implies ((not(a) or
  m) implies (i or m)).
Superman Proof in CafeOBJ

-- Step 4 : Second conjunct of F1 and Monotonicity 4.2 ('or not(a)' added)
red (not(w) implies m) implies ((not(w) or not(a)) implies (m or not(a))) .

-- Step 5 : 3.47a DeMorgan
red (not(a) or not(w)) iff (not(a and w)) .

-- Step 6 : 3.61 Contrapositive of F0
red ((a and w) implies p) implies ((not(p)) implies (not(a and w))) .
mod! SUPERMAN {
ops a w i m p e : -> Bool
ops aw ai nw pe ee : -> Bool
eq [F0] : aw = (a and w) implies p .
eq [F1a] : ai = not a implies i  .
eq [F1b] : nw = not w implies  m .
eq [F3] : ee = e implies (not i) and (not m) . }  
open SUPERMAN .
  -- Assumptions:  LHS of equations                   GOAL
red (aw and ai and nw and pe and ee) implies (not e) .
Alternative Proof by Contradiction

• This argument is valid. We can also argue by contradiction. Temporarily assume that Superman does exist. Then he is not ineffective, and he is not malevolent (this follows from F3). Therefore by (the contrapositives of) the two parts of F1, we conclude that he is able to prevent evil, and he is willing to prevent evil. By the implication of F0, we therefore ‘know’ that Superman does prevent evil. But this contradicts F2. Since we have arrived at a contradiction, our original assumption must have been false, so we conclude finally that Superman does not exist.
Alternative Proof in CafeOBJ

- CafeOBJ has a built theorem prover which can use proof by contradiction. The next slide shows how CafeOBJ can prove the non-existence of superman by applying equational logic and substitution in a mechanical way.
Alternative Proof using FOPL in CafeOBJ

mod! SUPERMAN {
-- the basic facts
    ops a w i m p e : -> Bool
ax [F0] : (a & w) -> p .
ax [F1] : (~(a) -> i) & (~(w) -> m) .
ax [F2] : ~(p) .
ax [F3] : (e -> (~(i) & ~(m))) .
}
Alternative Proof in CafeOBJ

- The expected output is
- ** PROOF _____________________________________________

- 1:[] ~(a) | ~(w) | p
- 2:[] a | i
- 3:[] m | w
- 4:[] ~(p)
- 5:[] ~(e) | ~(i)
- 6:[] ~(e) | ~(m)
- 7:[] e
- 15:[hyper:7,5,2] a
- 16:[hyper:3,1,15] m | p
- 17:[hyper:16,4] m
- 18:[hyper:17,6,7]
- ** ___________________________________________________
module! BADTEST{
  pr(BOOL)
  pred a : .   pred b : .

  eq a and b = true .
  eq not(a) = true .
  eq not(b) = true . }

open BADTEST
red (not(a) and not(b)) and (a and b) . -- inconsistent
  expression gives true!

open BOOL
**> same expression gives correct result (false) in BOOL
  pred a : . pred b : .
red (not(a) and not(b)) and (a and b) . -- false
Terms

• **syntax**: formal structure of sentences
• **semantics**: *truth* of sentences wrt *models*
• **entailment**: necessary truth of one sentence given another
• **inference**: deriving sentences from other sentences, provides a syntactic mechanism for deriving ‘truth’
• **soundness**: derivations produce only entailed sentences
• **completeness**: derivations can produce all entailed sentences
Formal Logic

• A logic consists of:
  – A set of symbols
  – A set of formulas constructed from the symbols
  – A set of distinguished formulas called axioms
  – A set of inference rules.

• The set of formulas is called the language of the logic, pure syntax.

• Inference rules (premise and conclusion) allow formulas to be derived from other formula.
Formal Logic

• A formula is a *theorem* of the logic if it is an axiom or can be generated from the axioms and already proven theorems.

• A *proof* that a formula is a theorem is an argument that shows how the inference rules are used to generate the formula.
Equational Logic E

• Symbols (, ), ⇒, ⇔, ≡, ≠, ¬, ∧, ∨

• Constants true, false.

• Boolean variables p, q, etc.

• Formulas are expressions using symbols, constants, and variables.

• Axioms e.g. Associativity of ≡.

• Inference rules e.g. Leibniz

• Theorems are formulas that can be shown to be equal to axioms using inference rules.
Model Theory

• The formulas of a logic are intended to be statements about some *domain of discourse* (DOD). Formulas may be *interpreted* as having a meaning with respect to this DOD by defining which formulas are true statements and which statements are false about DOD.

• An *interpretation* assigns meaning to the operators, constants, and variables of a logic.
Model Theory

Smaller theories have bigger models they are less constraining

Bigger theories have smaller models they are more constraining

Recall the theory of Mary & John’s apples:

\[ m = 2 \times j \]

Has many models \( m=2, j=1, \ m=4, j=2, \ m=6, j=, \ldots \)
Model Theory

Smaller theories have bigger models they are less constraining

Bigger theories have smaller models they are more constraining

Recall Mary & John’s apples:
mary = 2 \times john \text{  and  } mary/2 = 2 \times (john - 1)

Has only one model m=4, j=2.
Reminder on Interpretation

Transitivity

Observable Reality
Model Theory

• We distinguish between:
  – Syntax : a formal logic
  – Semantics : the interpretation of a formal logic

• In Lecture 2 we used truth tables to define operations and to evaluate expressions.

• Truth tables provide an interpretation (or meaning) for the Boolean expressions in Lecture 2.

• An interpretation can have two parts:
  – A fixed meaning, based on operators and constants
  – A state based meaning, based on values of the variables.
Interpretation or evaluation of $E$

• An interpretation of an expression in $E$ gives its value in some state. Here is the ‘standard’ interpretation of expressions in $E$.

• $\text{eval}$ (evaluation) gives ‘meaning’ to operators and constants of $E$.

• For expression $P$ without variables, let $\text{eval}.'P'$ be the value of $P$.

• Let $Q$ be any expression with variables and let $s$ be a state that gives values to all the variables of $Q$.

• Define $s.'Q'$ to be a copy of $Q$ in which all of the variables are replaced by their corresponding values in state $s$.

• Then function $f$ given by $f.'Q' = \text{eval}(s.'Q')$ is an interpretation for $Q$.

• The eval operation is called red in CafeOBJ, meaning reduction.
Satisfiability and validity

- Satisfiability and validity for Boolean expressions were defined in Lecture 2. Here satisfiability and validity are defined for any logic.

- Let $S$ be a set interpretations for a logic and $F$ be formulas for the logic. $F$ is satisfiable under $S$ iff at least one interpretation of $S$ maps $F$ to true. $F$ is valid iff every interpretation in $S$ maps $F$ to true.
Sound and Complete

• A logic is sound $\iff$ every theorem is valid. A logic is complete iff every valid formula is a theorem.

• Soundness means that the theorems are true statements about DOD. Completeness means that every valid formula can be proved.

• $E$ is sound and complete wrt standard interpretation.
Sound and Complete

- The soundness and completeness of a logic can be expressed in symbols of a ‘meta-logic’.
- $P_1, P_2, \ldots P_n \vdash Q$ means that there is a proof which infers the conclusion $Q$ from the assumptions $P_1, P_2, \ldots P_n$ using formal inference rules (syntactic turnstile).
- $P_1, P_2, \ldots P_n \models Q$ means that $Q$ must be true if $P_1, P_2, \ldots P_n$ are true but, says nothing about whether we have a proof, or even whether a proof exists (semantic turnstile).
Sound and Complete

• A formal system is sound (or consistent)
  \[
  \text{if } a \vdash b \text{ then } a \models b
  \]

• A system is sound if each proposition is provable using the inference rules is actually true. You can only prove true things.

• A formal system is complete
  \[
  \text{if } a \models b \text{ then } a \vdash b
  \]

• A system is complete if the inferences are powerful enough to prove every proposition is true. You can prove all true things.
Necessary & Sufficient

- Let $r$ be the proposition “I revised”, and
- Let $q$ be the proposition “I passed”.
- Then $r \rightarrow p$ may be expressed as:
  - “if I revised, I passed”;
  - “I passed if I revised”;
  - “I revised only if I passed”.
- See necessary and sufficient in Lecture 2.
Necessary & Sufficient Conditions for classifying concepts

• **Necessary Conditions**: If something is a member of a given concept, then it is necessary to fulfill some conditions.
  – If something is concept member, then it meets condition.

• **Necessary and Sufficient Conditions**: If something fulfills some conditions, then it must be a member of a given class.
  – If something is concept member, then it meets conditions.
  – If something meets condition, then concept member.
Proof by Contradiction

• A **direct proof** of a theorem shows that there cannot be a counterexample of the theorem.
• If a counterexample exists then two properties will hold:
  – P(1) It must make the hypothesis of the implication true
  – P(2) It must make the conclusion of the implication false.
• If no counterexample exists then no states the variables in the theorem will satisfy both P(1) and P(2).
Proof by Contradiction

- Using proof by contradiction we temporarily assume a counterexample exists and then show that this cannot be because it would result in a contradiction. A contradiction is a statement that we know to be false (e.g. $A \land \neg A$). In finding such a counterexample we are showing that $P(1)$ and $P(2)$ are logically incompatible.
Proof by Contradiction

• Hence assuming a counterexample we arrive at a contradiction. So, the assumed counterexample could not hold.
Proof by Contradiction

(4.9) Proof by Contradiction PbC:

• \( \neg p \Rightarrow \text{false} \equiv p \)

• PbC uses (3.74): \( p \Rightarrow \text{false} \equiv \neg p \)

• A theorem \( P \) is assumed \text{false} and we derive a \text{contradiction}.

• Once we have shown \( \neg P \Rightarrow \text{false} \) is a theorem we can conclude that \( P \) is also a theorem.
PbC in CafeOBJ

- CafeOBJ’s theorem prover (similar to OTTER/PROVER9), negates the current goal, and reduces the given equations to the form:
  \[
  \sim \text{goal} \land (a \lor b) \land (c \lor d)
  \]
- \(a, b, c, d\) can be positive or negative.
- The CafeOBJ prover tries to resolve terms to find the empty set (no values can satisfy the equations). By looking for the pattern \((a \land \sim a)\).
PbC in CafeOBJ

• Here is an example of proving Modus Ponens: P, P\(\rightarrow\)Q

1. P : premise
2. \(\neg P \lor Q\) : premise
3. \(\neg Q\) : Negation of conclusion
4. Q : Resolvent of 1,2
5. False : Resolvent of 3,4
Portia’s Suitors Dilemma (simplified)

- Portia has a gold casket and a silver casket and has place a picture of herself in one of them. On the casket’s are written the following inscriptions:
  - Gold: The portrait is not in here.
  - Silver: Exactly one of these inscriptions is true.
Portia’s Suitors Dilemma (simplified)

- Portia explains to her suitor that each inscription may be true or false, but she has placed her portrait in one of the caskets consistent with the truth or falsity of the inscriptions. If the suitor chooses the casket with her portrait, she will marry him. The suitor must determine which casket contains the portrait using only the inscriptions which may be true or false.
Portia’s Suitors Dilemma (simplified)

- The two possible situations are represented as:
  - $gc$: the portrait is in the gold casket.
  - $sc$: the portrait is in the silver casket.

- The two inscriptions on the caskets are represented as:
  - $g$: the portrait is not in the gold casket
  - $s$: exactly one of $g$ and $s$ is true.
Portia’s Suitors Dilemma (simplified)

• The fact that the portrait is exactly one place is written as:
  • $$F_0 : gc \equiv \neg sc$$

• The inscription $g$ on the gold casket is the negation of $gc$ written as:
  • $$F_1 : g \equiv \neg gc$$
Portia’s Suitors Dilemma (simplified)

• We do not know whether the inscription $s$ on the silver casket is true or false, but we do know that inscription $s$ is equivalent to the fact that only one of the inscriptions is true. This is written as:

• $F_2 : s \equiv (s \equiv \neg g)$
Recall Basic Proof

• The basic rule of reasoning is Leibniz ‘substitute equals for equals’. See slide 21 of lecture 1. Where two formats of proof are described: Notation 1

\[
E[z := X] \\
= \langle X = Y \rangle \\
E[z := Y]
\]

• The premise for a proof step is the reason why the step is valid, written as a hint between brackets \(<\>). The conclusion of a proof step consists of the line immediately above and immediately below the premise line related by equality (or iff) using notation 2. The 2\textsuperscript{nd} line is usually the result of a substitution. If whenever the premise true then the conclusion is true, then we can conclude that this is a valid proof step.
Portia’s Suitors Dilemma

• Having formalized the situation as a set logical axioms, we now try to derive either $gc$ or $sc$ from the axioms. We start with $F_2$, because it is the axiom with most structure (heuristic 3.33).

\[ s \equiv s \equiv \neg g \]

\[ = \langle 3.2 \text{ Symmetry of } \equiv, (\neg g \equiv s) \equiv (s \equiv \neg g) \rangle, \]

replace $s \equiv s \equiv \neg g$ with $\neg g >$

\[ \neg g \]

\[ = \langle F_1: g \equiv \neg gc, \text{ negation of both sides and 3.12 double negation, replace } \neg g \text{ with } \neg \neg gc \rangle \]

\[ gc \]
Portia’s Suitors Dilemma (simplified)

• From $F_1$ and $F_2$ we can conclude that the portrait is in the gold casket ($gC$). $F_0$ was not needed to solve this puzzle.

• We must make sure that $F_0$, $F_1$, and $F_2$ do not form a contradiction which can never be satisfied. In other words, we must check that there is a least one assignment of values that makes all three true.

• If $F_0 \land F_1 \land F_2$ were false in every state then they would be inconsistent (see note below) and anything could be proved. Were that to be the case, the assumptions could not be satisfied, so we would conclude that the problem had no solution.
Portia’s Suitors Dilemma (simplified)

• Here is the checking:

• With $gc=\text{true}$, the additional assignments $sc=false$, $g=false$, and $s=false$, satisfy $F0$, $F1$, and $F2$. Note that it does not matter whether the inscription on the silver casket is true or false. That is $s=false$ or $s=true$ still satisfy the three assumptions. This is illustrated in the truth table on next slide
Portia’s Suitors Dilemma Truth Table

<table>
<thead>
<tr>
<th>gc</th>
<th>sc</th>
<th>g</th>
<th>s</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

Truth table for \((gc \leftrightarrow \neg sc) \& (g \leftrightarrow \neg gc) \& (s \leftrightarrow (s \leftrightarrow \neg g))\)
Portia’s Suitors Dilemma (inconsistent)

• So far there is no inconsistency in this problem. Here we add an inconsistency by adding a different inscription on the silver casket:
  • $s'$: This inscription is false.
  • A formalization of this inscription is:
    • $F_2': s' \equiv \neg s'$
  • Then $F_2'$ is true in no state which is equivalent to false.
  • Adding $F_2'$ as an axiom to propositional logic, then would make false an axiom. Recall true is axiom 3.4, so false cannot be an axiom.
Portia’s Suitors Dilemma (inconsistent)

- From false anything can be proved.
- The addition of $F_2'$ makes the logic inconsistent and cannot be part of any mathematical or model of reality. This would lead to a puzzle that has no solution.
- An inconsistency can also arise from an interplay between axioms. Suppose $F_0: gc \equiv \neg sc$ is already an axiom, if we now add axiom $F_5: gc = sc$, then $F_0 \land F_5$ is false, so the system is inconsistent.
CafeOBJ code

mod! PORTIA1{
  protecting(BOOL)
  -- the basic facts are (in this case misleading) called predicates.
  pred gc : pred sc : pred g : pred s :
  -- Different syntax to CafeOBJ equations
  -- 'and' is translated to &
  -- 'implies' is translated to ->
  -- 'not' is translated to ~
  ax [F0] : gc <-> ~(sc) .
  ax [F1] : g <-> ~(gc) .
  ax [F2] : s <-> (s <-> ~(g)) .
  -- there is an implicit '&' between the axioms.}
CafeOBJ set up for proof

- open PORTIA1
- -- The goal is to prove that the portrait is in the gold casket.
- option reset .
- goal(gc) . -- We are only interested goal line
- flag(auto, on) .
- flag(very-verbose,on) .
- flag (universal-symmetry,on)
- param(max-proofs, 1) .
- resolve .
- close
CafeOBJ output

There is a lot of output. We are only concerned that CafeOBJ found the empty clause

** PROOF
4: [unit-del:7] g
5: [] ~s | ~(g)
6: [] s | ~(g)
7: [] ~(gc)
18: [hyper:4,6] s
19: [hyper:18,5,4]

Try to follow this proof by comparing it to the output from SupermanMadeEasy.mod

- CafeOBJ’s theorem prover OTTER will try to use proof by contradiction.
- It will create not(gc) and try to find a contradiction.
CafeOBJ using equational logic

mod! PORTIA2{
pr(BOOL)
  -- the basic facts.
pred gc : pred sc : pred g : pred s :
  -- We are using '==' instead of 'iff'
eq [F0] : gc = not(sc) .
eq [F1] : g = not(gc) .
eq [F2] : s = (s iff not(g)) .
}
open PORTIA2
  vars gc1 s1 g1 sc1 : Bool .
  red (gc1 iff not(sc1)) and (g1 iff not(gc1)) and (s1 iff (s1 iff not(g1))) implies gc1 .
CafeOBJ using equational logic

• There is an implicit and between the axioms. Stating this explicitly makes no difference to the proof of \( gc \).

• **See** Portia2.mod **for further details.**
Propositional proof Example

• Given \((p \equiv q)\) and \((q \implies r)\) prove that \((p \implies r)\) holds.
• You may use the following theorems:
  • (1) Idempotency of \(\land\): \(p \land p \equiv p\) (3.38 in course text)
  • (2) Left identity of \(\implies\): true \(\implies p \equiv p\) (3.73 in course text)

• Solution: Given \((p \equiv q)\) and \((q \implies r)\). We show \((p \implies r)\) holds.
• The numbers (e.g. 3.38) refer to either theorems or axioms in the course text.
• \((p \equiv q) \land (q \implies r) \implies (p \implies r)\)
  • = \(<\text{Why? } p \equiv q \equiv \text{true, given}>\)
  • true \(\land (q \implies r) \implies (p \implies r)\)
  • = \(<\text{Why? } q \implies r \equiv \text{true, given}>\)
  • true \(\land \text{true} \implies (p \implies r)\)
  • = \(<\text{(3.38 and 3.73)}>\)
  • \(p \implies r\) (LHS true gives RHS)
Using Golden Rule

- **Prove:** \( p \land (\neg p \lor q) \equiv p \land q \)
- **LHS**

\[
p \land (\neg p \lor q) = \langle (3.35) \text{ GR, with } q := \neg p \lor q \rangle \\
p \equiv \neg p \lor q = p \lor \neg p \lor q \equiv \langle (3.28) \text{ Excluded Middle } \rangle \\
p \equiv \neg p \lor q = \text{true} \lor q \equiv \langle (3.29) \text{ Zero for } \lor \rangle \\
p \equiv \neg p \lor q = \text{true} \equiv \langle (3.3) \text{ Identity of } \equiv \rangle \\
p \equiv \neg p \lor q \equiv \langle (3.32) \text{ p } \lor q \equiv p \lor \neg q \equiv p \text{ with } p, q = q, p \rangle \\
p \equiv p \lor q = q \equiv \langle (3.35) \text{ GR} \rangle \\
p \land q
\]

Which is RHS.

Using GR as definition of \(\land\), the first inference substitutes the \(q\) in GR

\[
LHS = RHS \\
p \land q \equiv p \equiv q \equiv p \lor q \\
p \equiv (\neg p \lor q) \equiv p \lor (\neg p \lor q)
\]

Hence the \(\land\) from line one is gone in line two.
Write CafeOBJ functions

• A supplier gives the following discounts:
  – 5 Euro if the total bill is 100 Euro or more and
  – 50 Euro if the total is 500 Euro or more.

• Write two CafeOBJ functions called `discount` and `bill` that given the price and quantity that calculates the correct amount of the bill.
module! BILL {
pr (INT)
op bill : Int Int -> Int
op discount : Int -> Int
vars p q r : Int
ceq discount(p) = 0 if p < 100 .
ceq discount(p) = 5 if (p >= 100) and ( p < 500) .
ceq discount(p) = 50 if p >= 500 .
eq bill(p,q) = p * q - discount(p * q) .
}
Functions

• Tests could select p and q to check the boundaries, negative number for either argument: error
• 99, discount 0
• 100, discount -50
• 101, discount -50
• 499, discount -50
• 500, discount -10
• 501, discount -10
Another example of propositional reasoning.

- Write the following statements in propositional logic:
  - I worked hard or I played the piano.
  - If I worked hard, then I will get a bonus.
  - I did not get a bonus.
- \((W \lor P) \land (W \Rightarrow B) \land \neg (B)\)
Another example of propositional reasoning.

- Construct a TT using:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>(wVp)∧(w→b)∧(¬b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

- This Boolean expression is only true in one case.
Another example of propositional reasoning.

- Truth tables for each premise.
  
  \[(W \lor P) \land (W \Rightarrow B) \land \neg(P)\]

- Premise 1

\[
\begin{array}{ccc}
<table>
<thead>
<tr>
<th>w</th>
<th>p</th>
<th>(w \lor p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
\end{array}
\]

- Premise 1

\[
\begin{array}{ccc}
<table>
<thead>
<tr>
<th>w</th>
<th>b</th>
<th>(w \Rightarrow b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
\end{array}
\]

- Premise 3

\[
\begin{array}{ccc}
<table>
<thead>
<tr>
<th>b</th>
<th>\neg b</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
\end{array}
\]

An argument form is valid iff the truth of the conclusion is guaranteed by the truth of the premises.
Another example of propositional reasoning.

- The intermediate truth tables for \((\neg w \lor p) \land (w \implies b) \land \neg (b)\)

- Double negation: \(\neg (\neg (w)) = w\).

Expressing or as \(\implies\) 

<table>
<thead>
<tr>
<th>w</th>
<th>p</th>
<th>(w\lor P)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>w</th>
<th>b</th>
<th>(w\implies b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>b</th>
<th>\neg b</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Expressing \(\lor\) as \(\implies\)

Two ways of expressing or as \(\implies\) ways are equivalent
translation between or and implies

• Two ways to rewrite \( \lor \) as \( \Rightarrow \)

<table>
<thead>
<tr>
<th>p</th>
<th>w</th>
<th>((p \lor w) \iff (\neg w \Rightarrow p))</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>w</th>
<th>((\neg p \Rightarrow w) \iff (\neg w \Rightarrow p))</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

eq \ [e1] : b = false .

ceq \ [e2] : p = true if (not w) .

ceq \ [e3] : w = false if not b
Another example of propositional reasoning.

- Given
- I worked hard or I played the piano.
- If I worked hard, then I will get a bonus.
- I did not get a bonus.
- Can we conclude: I played the piano.
- In logical notation is the following a valid argument?
- \((\neg W \lor P) \land (\neg W \Rightarrow B) \land \neg (B)\)
- \(\therefore P\)
Another example of propositional reasoning.

• **Premises:**
  – I worked hard or I played the piano.
  – If I worked hard, then I will get a bonus.
  – I did not get a bonus.

• **Conclusion:** Therefore I played the piano.

• Can we use \(\text{not(b) implies p}\) to prove that the sentences form a valid argument?
Conditional Equations

• The Inference rule Modus Ponens can be written as a CafeOBJ conditional equation
  • A, A \rightarrow B

• In CafeOBJ this would be:
  • \texttt{ceq B = true if A .}
Valid Argument

- The following is a valid argument
  - [P1] If my program is less than 10 lines long then it is correct.
  - [P2] My program is not correct.
  - [C1] Therefore my program is more than 10 lines long.

- The first two lines are the premisses and the last the conclusion. Let \( P_{10} \) be the proposition that my programme is less than 10 lines long and \( C \) be the proposition that it is correct. Recall the contrapositive rule \( (P_{10} \Rightarrow C) \equiv (\neg C \Rightarrow \neg P_{10}) \).
Invalid argument

• The following is an invalid argument
  – [P1] If I am wealthy then I give away lots of money
  – [P2] I give away lots of money
  – [C1] Therefore I am wealthy

• The reasoning is not valid because from the premisses you cannot derive the conclusion. \((W \Rightarrow M) \neq (M \Rightarrow W)\)
Valid argument: Modus Ponens

• An example a very general valid argument is Modus Ponens (from Latin mode that affirms)

\[
\begin{align*}
1. & \quad P \rightarrow Q \\
2. & \quad P \\
\therefore & \quad Q
\end{align*}
\]

This can be represented in CafeOBJ as a conditional equation:

\[
\text{ceq } Q = \text{true} \text{ if } P .
\]
\[
\text{eq } P = \text{true} .
\]

• Also applies to predicates

\[
\text{ceq } Q(x) = \text{true} \text{ if } P(x) .
\]
Turnstile $\models$

- In propositional logic the double turnstile is a symbol that is placed between propositions and the conclusion.
- Sequent expressions assert that an entailment relation holds between the set of propositions on the left and the conclusion on right-hand sides of the double turnstile.
- $A \models B$ can be read “$A$ entails $B$”
- There is no interpretation which makes $A$ true and $B$ false. As a specification we could say $A$ satisfies $B$.
Single Turnstile $\vdash$

- The intuitive meaning of $A \vdash B$ is that under the assumption of $A$ the conclusion of $B$ is provable. It denotes syntactic consequence.

- **Soundness**: $A \vdash B \Rightarrow A \models B$
- **Completeness**: $A \models B \Rightarrow A \vdash B$

- Soundness prevents proving things that aren't true when we interpret them.
- Completeness means that everything we know to be true on interpretation, we must be able to prove.
Double turnstile $\models$ in relation to $\Rightarrow$

$P_1, \ldots, P_n \models Z$ \iff $
\models (P_1 \land \ldots \land P_n) \Rightarrow Z$

- Recall the work, piano, bonus example.

$(w \lor p), (w \Rightarrow b), (~b) \models p$ \iff $
\models (w \lor p) \land (w \Rightarrow b) \land (~b) \Rightarrow p$

<table>
<thead>
<tr>
<th>$w$</th>
<th>$p$</th>
<th>$b$</th>
<th>$((w \lor p) \land (w \Rightarrow b) \land (~b)) \Rightarrow p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

expression is a tautology
Difference between valid arguments and implies.

• An argument is valid iff it is necessary that under all interpretations (valuations in propositional logic), in which the premises are true the conclusion is true as well: \( P_1, \ldots, P_n \models Z \)

• \( P_1, \ldots, P_n \models Z \) if and only if the statement of the form \( P_1 \) and ... and \( P_n \) implies \( Z \) is necessarily true (a tautology): \( \models (P_1 \& \ldots \& P_n) \Rightarrow Z \)
A formula is **satisfiable** if it is possible to find an interpretation (model) that makes the formula true.

A |= B read “A satisfies B”.

A valid argument is one in which, if the premises are true, then the conclusion must be true.
Building And-Or in CafeOBJ

-- No variables
-- We turn off implicit import of built-in BOOL.
set include BOOL off
-- This allows us to develop our own Boolean operations
mod* BOOLEAN1 {
  signature {
    [ Boolean ]
    op false : -> Boolean
    op true : -> Boolean
    op _\_ : Boolean -> Boolean
    op _\lor\_ : Boolean Boolean -> Boolean
    op _\land\_ : Boolean Boolean -> Boolean
  }

  axioms {
    eq - false = true .
    eq - true = false .
    eq false \lor false = false .
    eq false \lor true = true .
    eq true \lor false = true .
    eq true \lor true = true .}
}

eof
-- Check these reductions?
open BOOLEAN1
var A : Boolean
red true \lor false .
red true \lor A .
red A \lor true .
Alternative Terminology

• G&S do not use the terminology found in many other textbooks. For example, they use the term *expression* for what is often called a *clause* in other texts.

• The next few slides cover this alternative terminology.

A boolean expression is in *conjunctive normal form* if it has the form

\[ E_0 \land E_1 \land \ldots \land E_{n-1} \]

where each \( E_i \) is a disjunction of variables and negations of variables. For example, the following expression is in conjunctive normal form.

\[ (a \lor \neg b) \land (a \lor b \lor c) \land (\neg a) \]
Alternative Terminology

- In propositional calculus:
  - A **literal** is a propositional variable or its negation.
  - A **clause** is a finite disjunction of literals (a OR b OR c ...)
  - A **unit-clause** is composed of a single literal.
  - **Unit deletion** is used to simplify a set of clauses.
Alternative Terminology

• Conjunctive normal form (CNF) is a conjunction of clauses, where a clause is a disjunction of literals. It is an ANDing of ORs.

\[ \neg A \land (B \lor C) \]
\[ (A \lor B) \land (\neg B \lor C \lor \neg D) \land (D \lor \neg E) \]
\[ A \lor B \]
\[ A \land B \]
Alternative Terminology

• If a set of clauses contains the unit clause L, the other clauses are simplified by the application of the two following rules:
  1. every clause (other than the unit clause itself) containing L is removed;
  2. \( \text{not}(L) \) is deleted from other clauses.

• Example \( L = a: \{ a \lor b, \neg a \lor c, \neg c \lor d, a \} \)
  
  (removed) (\( \neg a \) deleted) (unchanged) (unchanged)

  \{ c, \neg c \lor d, a \}
Alternative Terminology

• Unit deletion *propagates* the fact that $a$ is true into the other clauses.

$$\{ \ a \lor b, \ \neg a \lor c, \ \neg c \lor d \ \ a \ \}$$

(removed) $(\neg a$ deleted) (unchanged) (unchanged)

$$\{ \ c, \ \neg c \lor d, \ \ a \ \}$$
Alternative Terminology

- In general, subsumption can be used when deciding whether one description, D1, is more general than another one, D2. That is whether D2 \textit{logically implies} D1. For example, IRISH-PERSON is subsumed by PERSON (i.e. an IRISH-PERSON is-a PERSON).
Alternative Terminology

• Given a clause in a CNF formula, if a subset of its literals constitutes another clause in the formula, then the first clause is said to be *subsumed* by the second clause.

• Clause C subsumes clause D if the variables of C can be instantiated in such a way that it becomes a subclause of D. If C subsumes D, then D is redundant and can be discarded, because it is weaker than or equivalent to C.