781 - 2013-02-19

Note Title 2013-02-19

In computer:

$$light1\_broken \leftarrow sw\_up \wedge power \wedge unlit\_light1.$$
$$sw\_up.$$
$$power \leftarrow lit\_light2.$$
$$unlit\_light1.$$
$$lit\_light2.$$

In user's mind:

- *light1_broken*: light #1 is broken
- *sw_up*: switch is up
- *power*: there is power in the building
- *unlit_light1*: light #1 isn't lit
- *lit_light2*: light #2 is lit

Conclusion: *light1_broken*

- The computer doesn't know the meaning of the symbols
- The user can interpret the symbol using their meaning

This will be used next class.

CNF and DNF

How to convert a prop. calculus formula to CNF & DNF.

Two main techniques.

One involves using the substitution theorem and a number of equivalences, including:

1. The "definition" of implication $P \to Q \equiv \sim P \vee Q$
2. The "definition" of equivalence $P \leftrightarrow Q \equiv (P \to Q) \wedge (Q \to P)$

3. Idempotency $(F \wedge F) \equiv F$, $(F \vee F) \equiv F$

4. Commutativity $(F \wedge G) \equiv (G \wedge F)$, $(F \vee G) \equiv (G \vee F)$

5. Associativity $((F \wedge G) \wedge H) \equiv (F \wedge (G \wedge H)) \; (\equiv (F \wedge G \wedge H))$

$((F \vee G) \vee H) \equiv (F \vee (G \vee H)) \; (\equiv (F \vee G \vee H))$

6. Absorption $(F \wedge (F \vee G)) \equiv F$, $(F \vee (F \wedge G)) \equiv F$

7. Distributivity $(F \wedge (G \vee H)) \equiv ((F \wedge G) \vee (F \wedge H))$

$(F \vee (G \wedge H)) \equiv ((F \vee G) \wedge (F \vee H))$

8. Double Negation $\neg\neg F \equiv F$

9. De Morgan's Laws $\neg(F \wedge G) \equiv (\neg F \vee \neg G)$

$\neg(F \vee G) \equiv (\neg F \wedge \neg G)$

10. Tautology Laws $(F \vee G) \equiv F$ if F is a tautology

$(F \wedge G) \equiv G$ if F is a tautology

11. Unsatisfiability Laws $(F \vee G) \equiv G$ if F is unsatisfiable

$(F \wedge G) \equiv F$ if F is unsatisfiable

To convert F to CNF,

1. Push negation inwards, using

$$\neg\neg G \Rightarrow G$$

$$\neg(G \wedge H) \Rightarrow (\neg G \vee \neg H)$$

$$\neg(G \vee H) \Rightarrow (\neg G \wedge \neg H)$$

until no subformulas of the form on the LHS of these rules occur.

2. Substitute in F each occurrence of a subformula of the form

$$(F \vee (G \wedge H)) \Rightarrow ((F \vee G) \wedge (F \vee H))$$

$$((F \wedge G) \vee H)) \Rightarrow ((F \vee H) \wedge (G \vee H))$$

until no such subformulas occur.

The resulting formula is in CNF (except for possible occurrences of tautologies).

(End of method)

Another technique is based on truth tables.
For formula F, to convert to DNF, take each row of its truth-table and build a conjunction with a positive literal of the corresponding variable is assigned 1 (T), and a negative literal of the corresponding variable is assigned 0.

To convert F to CNF, interchange the roles of 0 and 1, and construct a disjunction for each row of the truth table.

Example. Convert to DNF and CNF the following formula

$$F = ((\neg A \to B) \wedge ((A \wedge \neg C) \leftrightarrow B))$$

A	B	C	$\underset{}{\overset{A \vee B}{(\neg A \to B)}}$	$(A \wedge \neg C)$	$((A \wedge \neg C) \leftrightarrow B)$	F
0	0	0	0	0	1	0
0	0	1	0	0	1	0
0	1	0	1	0	0	0
0	1	1	1	0	0	0
1	0	0	1	1	0	0
1	0	1	1	0	1	1
1	1	0	1	1	1	1
1	1	1	1	0	0	0

The equivalent DNF formula is

$$(A \wedge \neg B \wedge C) \vee (A \wedge B \wedge \neg C)$$

The equivalent CNF formula is

$$(A \vee B \vee C) \wedge (A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C) \wedge (\neg A \vee B \vee C) \wedge (\neg A \vee \neg B \vee \neg C)$$

, which is very opaque,

b/c this is also an equivalent CNF formula.

$$A \wedge (B \text{ XOR } C) \equiv A \wedge (B \vee C) \wedge (\neg B \vee \neg C)$$

Let's try the other method (for CNF):

$$F = ((\neg A \to B) \wedge ((A \wedge \neg C) \leftrightarrow B)) \equiv$$

$$\equiv ((\neg\neg A \vee B) \wedge (((A \wedge \neg C) \to B) \wedge (B \to (A \wedge \neg C)))) \equiv$$

$$\equiv ((A \vee B) \wedge ((\neg (A \wedge \neg C) \vee B) \wedge (\neg B \vee (A \wedge \neg C)))) \equiv$$

$$\equiv (A \vee B) \wedge (((\neg A \vee C) \vee B) \wedge (\neg B \vee (A \wedge \neg C))) \equiv$$

$$\equiv (A \vee B) \wedge ((\neg A \vee C \vee B) \wedge ((\neg B \vee A) \wedge (\neg B \vee \neg C))) \equiv$$

$$\equiv \underline{(A \vee B)} \wedge (\neg A \vee B \vee C) \wedge \underline{(A \vee \neg B)} \wedge (\neg B \vee \neg C) \equiv$$

$\equiv A \wedge (\neg A \vee B \vee C) \wedge (\neg B \vee \neg C) \equiv$

$\equiv A \wedge (B \vee C) \wedge (\neg B \vee \neg C)$

Resolution

The (propositional) resolution rule:

Assume a formula in (conjunctive) clausal form;

start with F in CNF, so

$F := (L_{1,1} \vee \dots \vee L_{1,n_1}) \wedge \dots \wedge (L_{k,1} \vee \dots \vee L_{k,n_k})$

(F is a conjunction of k clauses, where the ith clause contains n_i literals)

Write each clause as a set, so F is expressed as a set of clauses:

$$F = \{\{L_{1,1}, \ldots, L_{1,n_1}\}, \ldots\ldots, \{L_{k,1}, \ldots, L_{k,n_k}\}\}$$

<u>Definition</u> <u>Resolvent</u>.

C_1 and C_2 are clauses. Then R is a <u>resolvent</u> of C_1 and C_2 if there is a literal $L \in C_1$ such that $\bar{L} \in C_2$ and $R = (C_1 - \{L\}) \cup (C_2 - \{\bar{L}\})$.

$$\bar{L} = \begin{cases} \neg A_i & \text{if } L = A_i \\ A_i & \text{if } L = \neg A_i \end{cases}$$

Graphical notation

$$\begin{tikzcd}[column sep=small] \{A, \neg C\} \arrow[dr, dash] & & \{A, B, C\} \arrow[dl, dash] \\ & \{A, B\} & \end{tikzcd}$$

The resolvent $\longrightarrow$ $\{A, B\}$
of $\{A, \neg C\}$ and $\{A, B, C\}$

Theorem. Let F be a CNF formula, represented as a set of clauses. Let R be a resolvent of two clauses C_1 and C_2 in F. Then, F and $F \cup \{R\}$ are equivalent.

<u>Definition</u> Let F be a set of clauses. Then $Res(F)$ is defined as:

$$Res(F) = F \cup \{R \mid R \text{ is a resolvent of two clauses in } F\}$$

Furthermore, define

$Res^0(F) = F$

$Res^{n+1}(F) = Res(Res^n(F))$ for $n \geq 0$, and

finally let

$$Res^*(F) = \bigcup_{n \geq 0} Res^n(F).$$

It can be proved that for every finite F, there is a k s.t. $Res^k(F) = Res^*(F)$.

<u>Defn.</u> The empty resolvent is the resolvent of $C_1 = \{L\}$ and $C_2 = \{\bar{L}\}$.

This is also called the <u>empty clause</u>, and is denoted by $\square$.

$$
\begin{tikzcd}
\{A_3\} \arrow[dr, dash] & & \{\neg A_3\} \arrow[dl, dash] \\
& \square &
\end{tikzcd}
$$

The <u>Resolution Theorem</u> (of propositional logic)

[J. A. Robinson proved this for FOL around 1960.]

A clause set F is unsatisfiable iff $\square \in Res^*(F)$.

<u>Example</u> (Ex 33 Schöning) Using resolution, show that $(A \wedge B \wedge C) = G$ is a consequence of

The clause set $F = \{\{\neg A, B\}, \{\neg B, C\}, \{A, \neg C\}, \{A, B, C\}\}$

Recall: $F \models G$ iff $\vdash F \to G$ iff $F \wedge \neg G$ is unsatisfiable

So. Negate G and show that $F \wedge \neg G$ is unsat.

$\neg G$: $\{\neg A, \neg B, \neg C\}$. Show that

$F \cup \neg G$: $\{\{\neg A, B\}, \{\neg B, C\}, \{A, \neg C\}, \{A, B, C\}, \{\neg A, \neg B, \neg C\}\}$

is unsat., i.e., by the res thm, show that

$\mathrm{Res}^*(F \cup \neg G) = \square$.

$\{A, \sim C\}$ $\{A, B, C\}$ $\{\sim A, \sim B, \sim C\}$ $\{A, B\}$

$\{A, B\}$ $\{\sim B, \sim C\}$