CS 486: Lecture 2, Thursday, Jan 22, 2009

Mark Bickford

January 22, 2009

Outline

- Propositional formulas
- Interpretations and Valuations
- Validity and Satisfiability
- Truth tables and Disjunctive Normal Form
- Truth functional implication and equivalence
- Algebraic Laws (DeMorgan and Distributive)
- Example Application

Propositional Formulas

The set of *propositional formulas*, or formulas of propositional logic, is the smallest set formed from

- variables: "officially" a fixed infinite set of symbols $\{p_0, p_1, p_2, \ldots\}$, informally, any string like "v1", "p", "23" of letters or digits with no spaces.
- connectives $\neg \vee \wedge$: if ψ_1 and ψ_2 are formulas then $(\neg\psi_1)$, $(\psi_1 \vee \psi_2)$, and $(\psi_1 \wedge \psi_2)$ are formulas.

Induction on formulas: If for every variable p, and all formulas ψ_1 and ψ_2,

- $P(p)$ is true
- $P(\psi_1)$ and $P(\psi_2)$ implies $P((\psi_1 \wedge \psi_2))$ and $P((\psi_1 \vee \psi_2))$ and $P((\neg\psi_1))$
- then $P(\psi)$ holds for all formulas ψ.

Interpretations and Valuations

An *interpretation* (also called an *assignment* or a *state*) is a function σ from variables to *truth values* $\{t, f\}$

A *valuation* is a function *val* from formulas to truth values such that:

- $val((\psi_1 \land \psi_2)) = t$ if and only if $val(\psi_1) = t$ and $val(\psi_2) = t$
- $val((\psi_1 \lor \psi_2)) = t$ if and only if $val(\psi_1) = t$ or $val(\psi_2) = t$
- $val((\neg\psi_1)) = t$ if and only if $val(\psi_1) \neq t$ (i.e. $val(\psi_1) = f$)

Definition A valuation *val extends* an interpretation σ if $val(p) = \sigma(p)$ for every variable p.

Lemma For each interpretation there is exactly one valuation that extends it.

proof: By induction on formulas.

Example

If $\sigma(p_1) = t$, $\sigma(p_2) = t$, $\sigma(p_3) = f$, $\sigma(p_4) = f$, and val_σ extends σ, then

$$
\begin{aligned}
val_\sigma(((p_1 \vee (\neg p_2)) \wedge ((\neg p_3) \vee p_4))) &= \\
val_\sigma((p_1 \vee (\neg p_2))) \mathbin{\&} val_\sigma(((\neg p_3) \vee p_4)) &= \\
(val_\sigma(p_1) \ or \ val_\sigma((\neg p_2))) \mathbin{\&} (val_\sigma((\neg p_3)) \ or \ val_\sigma(p_4)) &= \\
(\sigma(p_1) \ or \ not(val_\sigma(p_2))) \mathbin{\&} (not(val_\sigma(p_3)) \ or \ \sigma(p_4)) &= \\
(t \ or \ not(\sigma(p_2))) \mathbin{\&} (not(\sigma(p_3)) \ or \ f) &= \\
(t \ or \ not(t)) \mathbin{\&} (not(f) \ or \ f) &= \\
(t \ or \ f) \mathbin{\&} (t \ or \ f) &= \\
t \mathbin{\&} t &= t
\end{aligned}
$$

Validity and Satisfiability

Definition A formula ψ is *valid* if, for *every* valuation, val, $val(\psi) = t$.

For example: $(p \vee (\neg p))$ is valid

Definition A formula ψ is *satisfiable* if, for *some* valuation, val, $val(\psi) = t$.

For example: $(p \wedge q)$ is satisfiable but not valid, $(p \wedge (\neg p))$ is not satisfiable.

Lemma: If two valuations val_1 and val_2 agree on every variable in formula ψ, then $val_1(\psi) = val_2(\psi)$.

Corollary: If ψ has N variables, then to decide whether ψ is valid or satisfiable, we need check only 2^N valuations.

Truth tables and Disjunctive Normal Form

The *truth table* for formula ψ with N variables is a list of the 2^N assignments, σ, to the variables and the corresponding value $val_\sigma(\psi)$. Each row corresponds to a conjunction of *literals*, a variable or negation of a variable.

x	y	$((x \vee (\neg y)) \wedge ((\neg x) \vee y))$	$((x \wedge y) \vee ((\neg x) \wedge (\neg y)))$
t	t	t	$(x \wedge y)$
t	f	f	$(x \wedge (\neg y))$
f	t	f	$((\neg x) \wedge y)$
f	f	t	$((\neg x) \wedge (\neg y))$

The validity and satisfiability of a formula is immediately evident from its truth table. We can also compute a *disjunctive normal form* (DNF) for the formula ($\vee$ the rows that have val = t).

Truth functional implication and equivalence

Definition Formula ψ_1 *truth functionally implies* formula ψ_2 if, every valuation that makes ψ_1 true also makes ψ_2 true.

Definition Formulas ψ_1 and ψ_2 are *truth functionally equivalent* ($\psi_1 \equiv \psi_2$) if, every valuation that makes one of then true also makes the other true.

Lemma: A formula is truth functionally equivalent to its DNF.

Algebraic Laws (Commutative and Associative)

$$
\begin{aligned}
(\psi_1 \wedge \psi_2) &\equiv (\psi_2 \wedge \psi_1) \\
(\psi_1 \wedge (\psi_2 \wedge \psi_3)) &\equiv ((\psi_1 \wedge \psi_2) \wedge \psi_3) \\
(\psi_1 \vee \psi_2) &\equiv (\psi_2 \vee \psi_1) \\
(\psi_1 \vee (\psi_2 \vee \psi_3)) &\equiv ((\psi_1 \vee \psi_2) \vee \psi_3)
\end{aligned}
$$

These laws imply that for any finite set of formulas $\{\psi_i | i \in I\}$, $\bigvee\{\psi_i | i \in I\}$ and $\bigwedge\{\psi_i | i \in I\}$ are well defined.

Algebraic Laws (DeMorgan and Distributive)

$$
\begin{aligned}
(\neg \bigvee \{\psi_i | i \in I\}) &\equiv \bigwedge \{(\neg \psi_i) | i \in I\} \\
(\neg \bigwedge \{\psi_i | i \in I\}) &\equiv \bigvee \{(\neg \psi_i) | i \in I\} \\
(\bigvee \{\psi_i | i \in I\} \wedge \bigvee \{\phi_j | j \in J\}) &\equiv \bigvee \{(\psi_i \wedge \phi_j) | i \in I, j \in J\} \\
(\bigwedge \{\psi_i | i \in I\} \vee \bigwedge \{\phi_j | j \in J\}) &\equiv \bigwedge \{(\psi_i \vee \phi_j) | i \in I, j \in J\}
\end{aligned}
$$

Example Application