Propositional Formulas

Syntax

Definition 1 *Propositional (Boolean) Formulas* are built from *atoms* $P_1, P_2, P_3, \ldots$, the unary connective $\neg$ (NOT), the binary connectives $\wedge, \vee$ (AND,OR, respectively), and parenthesis (,).

We use $a, b, c, \ldots, x, y, z, \ldots, P, Q, R, \ldots$, as different metavariables that stand for atoms.

We define propositional formulas by structural induction:

1. Any atom P is a formula.
2. If α, β are formulas, then so are $\neg\alpha$,$(\alpha \wedge \beta)$, and $(\alpha \vee \beta)$.

Example 1 $P, (P \vee Q), (\neg(P \wedge Q) \wedge (\neg P \vee \neg Q))$

We will use $\alpha \supset \beta$ (implies) to express $(\neg\alpha \vee \beta)$, and

$\alpha \leftrightarrow \beta$ (is equivalent to) to express $((\alpha \supset \beta) \wedge (\beta \supset \alpha))$

Alternatively, we can define propositional formulas with a context-free grammar:

$G_{\text{prop}} = (\{S\}, \{a - z, (,), \wedge, \vee, \neg\}, \{S \longrightarrow \texttt{a-z} | (S \wedge S) | (S \vee S) | \neg S\}, S)$.

We convert G_{prop} to Chomsky Normal Form:

$X_0 \longrightarrow \texttt{a-z}, X_1 \longrightarrow (, X_2 \longrightarrow), X_3 \longrightarrow \wedge, X_4 \longrightarrow \vee, X_5 \longrightarrow \neg$

$X_0 \longrightarrow X_1 X_6, X_6 \longrightarrow X_0 X_7, X_7 \longrightarrow X_3 X_8, X_8 \longrightarrow X_0 X_2$

$X_0 \longrightarrow X_1 X_9, X_9 \longrightarrow X_0 X_{10}, X_{10} \longrightarrow X_4 X_{11}, X_{11} \longrightarrow X_0 X_2$

$X_0 \longrightarrow X_5 X_0$

Exercise 1 Note that the above grammar only generates propositional formulas with variables in the set $\{a, b, \ldots, z\}$. How would you modify this grammar so that *all* atoms $P_1, P_2, P_3, \ldots$, could be generated ? (Hint: we did this in class; you have to generate subscripts.)

Theorem 1 (Unique Readability Theorem) Suppose $\alpha, \beta, \alpha', \beta'$ are formulas, c, c' are binary connectives, and $(\alpha c \beta) =_{\text{synt}} (\alpha' c' \beta')$. Then $\alpha =_{\text{synt}} \alpha'$ and $\beta =_{\text{synt}} \beta'$ and $c =_{\text{synt}} c'$.

Note that this theorem says that the grammar for generating formulas is unambiguous.

Here $\alpha =_{\text{synt}} \alpha'$ denotes that α and α' are equal as string of symbols (syntactic identity, rather than semantic identity).

PROOF: Assign weights to all symbols as follows:

0 to	$\neg$
1 to	$\wedge, \vee, ($
-1 to	$), P$, for each atom P

Define the weight of a formula to be the sum of the weights of all the symbols in it.

Lemma 1 The weight of any formula α is -1, but the weight of any proper initial segment is ≥ 0. (Hence no proper initial segment of a formula is a formula).

PROOF:(of lemma) By structural induction on the length of α.

Basis Case: $w(P) = -1$, for any atom P.

Induction Step: Three cases: $\neg\alpha$, $(\alpha \wedge \beta)$ and $(\alpha \vee \beta)$. $\square$ $\square$

Semantics

Definition 2 A *truth assignment* is a map $\tau : \{\text{atoms}\} \longrightarrow \{T, F\}$.

Here $\{T, F\}$ represents True and False (sometimes denoted 0,1).

The truth assignment τ can be extended to assign either T of F to every formula, as follows:

1. $(\neg\alpha)^\tau = T$ iff $\alpha^\tau = F$
2. $(\alpha \wedge \beta)^\tau = T$ iff $\alpha^\tau = T$ and $\beta^\tau = T$
3. $(\alpha \vee \beta)^\tau = T$ iff $\alpha^\tau = T$ or $\beta^\tau = T$

Definition 3

1. We say that τ *satisfies* α iff $\alpha^\tau = T$.
2. τ *satisfies* a set of formulas Φ iff τ satisfies all $\alpha \in \Phi$.
3. Φ is *satisfiable* if some τ satisfies it; otherwise,
4. Φ is *unsatisfiable*
5. $\Phi \models \alpha$ (α is a *logical consequence* of Φ) iff τ satisfies α for every τ such that τ satisfies Φ.
6. A formula α is *valid* iff $\models \alpha$ (i.e., $\alpha^\tau = T$ for all τ).
7. A valid propositional formula is called a *tautology*.
8. α and β are *equivalent* formulas (written $\alpha \iff \beta$) iff $\alpha \models \beta$ and $\beta \models \alpha$.
   Note that '$\iff$' and '$\leftrightarrow$' have different meanings (semantic and syntactic).

Example 2 The following are tautologies: $P \vee \neg P, P \supset P, \neg(P \wedge \neg P)$.

Logical consequence: $(P \wedge Q) \models (P \vee Q)$.

Equivalence: $\neg(P \vee Q) \iff (\neg P \wedge \neg Q)$ (De Morgan's Law).

Exercise 2 Show that if $\Phi \models \alpha$ and $\Phi \cup \{\alpha\} \models \beta$, then $\Phi \models \beta$.

Exercise 3 Prove the following **Duality Theorem**: Let α' be the result of interchanging $\vee$ and $\wedge$ in α, and replacing P by $\neg P$ for each atom P. Then $\alpha \iff \alpha'$.

Exercise 4 Prove the **Craig Interpolation Theorem:** Let α and β be any two propositional formulas. Let $\mathrm{Var}(\alpha)$ be the set of variables that occur in α. Let $S = \mathrm{Var}(\alpha) \cap \mathrm{Var}(\beta)$. Assume S is not empty. If $A \supset B$ is valid, then there exists a formula C such that $\mathrm{Var}(C) = S$, called an "interpolant" such that $A \supset C$ and $C \supset B$ are both valid.

Gentzen's system PK

One way to establish that a formula α with n atoms is a tautology is to verify that $A^\tau = T$ for all 2^n truth assignments τ to the atoms of A.

A similar exhaustive method can be used to verify that $\Phi \vDash \alpha$ (if Φ is finite).

Another way, is to use the notion of a formal proof. We present PK.

In the propositional sequent calculus system PK, each line in a proof is a *sequent* of the form:

$$S = \alpha_1, \ldots, \alpha_k \rightarrow \beta_1, \ldots, \beta_l$$

where $\rightarrow$ is a new symbol, and $\alpha_1, \ldots, \alpha_k$ and $\beta_1, \ldots, \beta_l$ are sequences of formulas ($k, l \geq 0$) called *cedents* (*antecedent* and *succedent*, respectively).

A truth assignment τ *satisfies* the sequent S iff τ falsifies some α_i or τ satisfies some β_i, i.e., iff τ satisfies the formula:

$$\alpha_S = (\alpha_1 \wedge \cdots \wedge \alpha_k) \supset (\beta_1 \vee \cdots \vee \beta_l)$$

If antecedent is empty, $\rightarrow \alpha$ is equivalent to α, and if the succedent is empty, $\alpha \rightarrow$ is equivalent to $\neg\alpha$.

If both antecedent and succedent are empty, then $\rightarrow$ is false (unsatisfiable).

We have the analogous definitions of validity and logical consequence for sequents.

Example 3 Valid sequents: $\alpha \rightarrow \alpha$, $\rightarrow \alpha, \neg\alpha$, $\alpha, \alpha \supset \beta \rightarrow \beta$, $\alpha \wedge \neg\alpha \rightarrow$

A formal *proof* in PK is a finite rooted tree in which the nodes are labeled with sequents.

The sequent at the root (bottom) is what is being proved: the *endsequent*.

The sequents at the leaves (top) are *logical axioms*, and must be of the form $\alpha \rightarrow \alpha$, where α is a formula.

Each sequent other than the logical axioms must follow from its parent sequent(s) by one of the following rules of inference (where Γ, Δ denote finite sequences of formulas).

Weak structural rules

exchange-left: $\dfrac{\Gamma_1, \alpha, \beta, \Gamma_2 \to \Delta}{\Gamma_1, \beta, \alpha, \Gamma_2 \to \Delta}$ exchange-right: $\dfrac{\Gamma \to \Delta_1, \alpha, \beta, \Delta_2}{\Gamma \to \Delta_1, \beta, \alpha, \Delta_2}$

contraction-left: $\dfrac{\Gamma, \alpha, \alpha \to \Delta}{\Gamma, \alpha \to \Delta}$ contraction-right: $\dfrac{\Gamma \to \Delta, \alpha, \alpha}{\Gamma \to \Delta, \alpha}$

weakening-left: $\dfrac{\Gamma \to \Delta}{\alpha, \Gamma \to \Delta}$ weakening-right: $\dfrac{\Gamma \to \Delta}{\Gamma \to \Delta, \alpha}$

Cut rule

$$\frac{\Gamma \to \Delta, \alpha \quad \alpha, \Gamma \to \Delta}{\Gamma \to \Delta}$$

Rules for introducing connectives

$\neg$-left: $\dfrac{\Gamma \to \Delta, \alpha}{\neg\alpha, \Gamma \to \Delta}$ $\neg$-right: $\dfrac{\alpha, \Gamma \to \Delta}{\Gamma \to \Delta, \neg\alpha}$

$\wedge$-left: $\dfrac{\alpha, \beta, \Gamma \to \Delta}{(\alpha \wedge \beta), \Gamma \to \Delta}$ $\wedge$-right: $\dfrac{\Gamma \to \Delta, \alpha \quad \Gamma \to \Delta, \beta}{\Gamma \to \Delta, (\alpha \wedge \beta)}$

$\vee$-left: $\dfrac{\alpha, \Gamma \to \Delta \quad \beta, \Gamma \to \Delta}{(\alpha \vee \beta), \Gamma \to \Delta}$ $\vee$-right: $\dfrac{\Gamma \to \Delta, \alpha, \beta}{\Gamma \to \Delta, (\alpha \vee \beta)}$

Table 1: PK rules

Exercise 5 Give PK proofs for each of the following valid sequents: $\neg P \vee \neg Q \to \neg(P \vee Q)$, $\neg(P \vee Q) \to \neg P \wedge \neg Q$, and $\neg P \wedge \neg Q \to \neg(P \vee Q)$.

Exercise 6 Show that the contraction rules can be derived from the cut rule (with exchanges and weakenings).

Exercise 7 Suppose that we allowed $\supset$ as a primitive connective, rather than one introduced by definition. Give the appropriate left and right introduction rules for $\supset$.