

Week 2: Propositional Logic

Wang Xiyu

Propositional logic is about, well, logics about propositions. A proposition is basically a statement that can be evaluated to true or false. For example "1+1=2" and "1+1=3" are both propositions.

1 Syntax

Definition 1.1. *Syntax decides what are allowed to be written*

1.1 Well-formed Formula (WFF)

A formula ψ is obtained by composing $p \in P$ with $c \in C$, where P is the set of propositions and C is the set of connectors. here we use any set of complete logic, like $C = \{\neg, \vee\}$. This is not a precise definition as counter examples that do not form a valid formula can be formulated based on this, for example $\neg \vee p$ means nothing. To formalize the definition we may attempt to define a language with the following grammar:

$$\Sigma = P \cup C \cup \{ (,) \}$$

$$FORM : \{ \psi = p \in P | \neg \psi | \psi \vee \psi | (\psi) \}$$

However this is a recursive definition as the definition of the a formula ψ involves itself. We need to find a declarative way to define this recursive/inductive construction.

Definition 1.2. *FORM is the smallest set of strings over Σ where*

1. $P \subseteq FORM$
2. $\forall \psi \in FORM, \neg \psi \in FORM$
3. $\forall \psi_1, \psi_2 \in FORM, \psi_1 \vee \psi_2 \in FORM$

We can see as a construction rule, we are admitting strings into the set of formula, but we need to somewhat impose restriction on what is not acceptable.

1.2 Closed Set

Let's remove the attributive **the smallest** first and define another set that contains all the set that satisfy these rules listed above:

Definition 1.3. *Closed is a set of strings over Σ where*

1. $P \subseteq Closed$
2. $\forall \psi \in Closed, \neg \psi \in Closed$
3. $\forall \psi_1, \psi_2 \in Closed, \psi_1 \vee \psi_2 \in Closed$

1.2.1 All Closed Set

And we define a set ACS (All-closed sets). We can define FORM via the following constriction:

$$FORM = \bigcap ACS$$

Suppose $BCS = \emptyset$ and from definition

$$U_0 = \bigcap BCS = \{w \in \Sigma^* | \forall S \in BCS, w \in S\}$$

However, $BCS = \emptyset$, The statement $\forall S \in BCS, w \in S$ is vacuously true $\forall w \in \Sigma^*$, which means that

$$U_0 = \Sigma^* = \bigcap BCS$$

A contradiction. Therefore we have proved that ACS cannot be empty, thus $\forall P, \exists S$ such that S is a closed set.

1.3 FORM is the minimum of the set of Closed

First let's prove that FORM is closed. It seems to be trivial that the intersection of closed sets is closed but formal proof is still needed. We denote a set from the intersection construction as T_0 .

Lemma 1.4. T_0 is closed

We are proving that the set T_0 we get from the intersection construction still satisfy the rules that define what is closed.

Proof. From rule 1,

$$\begin{aligned} & \forall S \in ACS, P \subseteq S \\ \implies & \forall p \in P, \forall S \in ACS, p \in S \\ \implies & \forall p \in P, p \in \bigcap ACS \\ \implies & P \in T_0 \end{aligned}$$

From rule 2,

$$\begin{aligned} & \forall S \in ACS, T_0 \subseteq S \\ \text{let } w \in T_0, & \forall S \in ACS, w \in S, \neg w \in S \\ \implies & \forall w \in T_0, \neg w \in \bigcap ACS \\ \implies & \forall w \in T_0, \neg w \in T_0 \end{aligned}$$

From rule 3,

$$\begin{aligned} & \forall S \in ACS, T_0 \subseteq S \\ \text{let } w_1, w_2 \in T_0, & \forall S \in ACS, w_1 \vee w_2 \in S \\ \implies & \forall w_1, w_2 \in T_0, w_1 \vee w_2 \in \bigcap ACS \\ \implies & \forall w_1, w_2 \in T_0, w_1 \vee w_2 \in T_0 \end{aligned}$$

□

Lemma 1.5. T_0 is a smallest Closed set

Proof. The proof of minimality is trivial. $T_0 = \bigcap ACS \implies \forall S \in ACS, T_0 \subseteq S$

□

Lemma 1.6. T_0 is *the* smallest Closed set (unique). In other word,

$$\forall T'_0 \in \{T \mid \forall S \in ACS, T \subseteq S\}, T'_0 = T_0$$

Proof. Suppose otherwise, $\exists T'_0 \in ACS \neq T_0$.

$$\begin{aligned} T_0, T'_0 \in ACS &\implies \forall S \in ACS, T_0 \subseteq S, T'_0 \subseteq S \\ &\implies T_0 \subseteq T'_0 \wedge T'_0 \subseteq T_0 \\ &\implies T_0 = T'_0 \end{aligned}$$

A contradiction. □

Now we can use the generative rule to define formula, which is called **Backus-Naur form**. Note that a context-free grammar is a type of formal language. Backus Naur form is a specification language for this type of grammar. It is used to describe language syntax.

2 Semantics

Definition 2.1. *Semantics describe meanings.*

2.1 Truth Assignment/Proposition evaluation

A truth assignment or an evaluation of propositions is a function that maps from the set of propositions to true and false.

$$\begin{aligned} v : P &\rightarrow \{T, F\} \\ v &\in [P \rightarrow \{T, F\}] \end{aligned}$$

Example 2.2. Consider $v = \{p_1 \mapsto T, p_2 \mapsto F\}$, we can derive

$$\begin{aligned} v(p_1) &= T \\ v(p_2) &= F \\ v(\neg p_2) &= T \\ v(p_1 \vee p_2) &= T \text{ etc.} \end{aligned}$$

2.2 Entail/Model

We denote entailment of an evaluation function to formula as

$$v \models \psi$$

We also define the following rule

$$\begin{aligned} \forall p \in P, v \models p &\iff v(p) \\ v \models \psi &\iff v \not\models \neg\psi \\ v \models \psi_1 \vee \psi_2 &\iff (v \models \psi_1) \vee (v \models \psi_2) \end{aligned}$$