CS3243: Introduction to Artificial Intelligence **Fall 2020**

Lecture 12: November 11

Lecturer: Prof. Kuldeep S. Meel *Scribe: Ang Zheng Yong*

12.1 Propositional Logic

In the last lecture, we have discussed the idea of a knowledge base, which can be seen as a database of everything that the agent knows. In this lecture, we will be representing the knowledge base as KB.

We would want every statement $\alpha \in KB$ to also be in $FORM$. In other words, we do not want to deal with sentences which are not in $FORM$, such as $()pq$.

12.1.1 Truth assignments

A **truth assignment** τ is defined as $\tau : PROP \to \{0,1\}$. For each statement, there are $2^{|PROP|}$ possible truth assignments, where $|PROP|$ represents the number of distinct propositions in the statement.

We say τ models ϕ, or $\tau \models \phi$, if:

- ϕ holds at τ, or
- ϕ is true at τ, or
- τ satisfies ϕ.

Hence, if we are given the statement $p \vee q$, we can construct the following truth table:

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

and we see that $\tau \models \phi$ when τ has either of the following assignments: $(p \to 0, q \to 1)$, $(p \to 1, q \to 0)$, or $(p \to 1, q \to 1)$. We notice that $\tau \models \phi$ if and only if $\phi(\tau) = 1$.

Another example given in lecture is the statement $\phi = ((p \vee q) \wedge (\neg r))$. For this statement, if we were given the following two truth assignments:

$$\tau_1 : \begin{cases} p \to 1 \\ q \to 1 \\ r \to 1 \end{cases} \qquad \text{and} \qquad \tau_2 : \begin{cases} p \to 1 \\ q \to 0 \\ r \to 0 \end{cases}$$

we notice that $\phi(\tau_1) = 0$ and $\phi(\tau_2) = 1$, so we conclude that $\tau_1 \models \phi$ and $\tau_2 \not\models \phi$.

12.1.2 Satisfiability

In this section, we began with the following definitions:

1. ϕ is *SAT* (satisfiable) if $\exists\tau$ such that $\tau \models \phi$.
2. ϕ is *VALID* if $\forall\tau,\ \tau \models \phi$.
3. ϕ is *UNSAT* (unsatisfiable) if $\forall\tau,\ \tau \not\models \phi$.
   - In other words, there does not exist τ such that $\tau \models \phi$.

Notice that ϕ is *VALID* implies that ϕ is a tautology, and thus a statement that is *VALID* will also be *SAT* (i.e. *VALID* $\implies$ *SAT*). An example of a statement that is *VALID* is $(p \vee \neg p)$, and an example of a statement that is *SAT* but not *VALID* is $(p \vee q)$. On the other hand, examples of statements that are *UNSAT* include $(p \wedge \neg p)$ and $(p \vee q) \wedge (\neg p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee \neg q)$.

We also define the following:

- $\phi \models \psi$ if and only if $\phi \rightarrow \psi$ is *VALID*.
- $\phi \equiv \psi$ if and only if $\phi \leftrightarrow \psi$ is *VALID*.
- $\phi \leftrightarrow \psi$ is *VALID* if $\forall\tau,\ \phi(\tau) = \psi(\tau)$.
  - This is equivalent to saying that ϕ and ψ are semantically equivalent.

12.1.3 Conjunctive Normal Form (CNF)

Suppose we have $C_1, C_2, C_3, C_4 \in FORM$. Then, we can see that

$$C_1 \wedge C_2 \wedge C_3 \wedge C_4 \Leftrightarrow (C_1 \wedge (C_2 \wedge (C_3 \wedge C_4))) \Leftrightarrow (((C_1 \wedge C_2) \wedge C_3) \wedge C_4)$$

In fact, the order of the bracketing would not matter, if all the operators in between the clauses C_i are $\wedge$ (note that if the operators consist of $\vee$ and $\wedge$, then it may no longer be the case). We say that such a statement has a **Conjunctive Normal Form** (CNF).

More formally, we say that a statement is a CNF if that statement has the form

$$C_1 \wedge C_2 \wedge \ldots \wedge C_n$$

and every **clause** C_i is of the form

$$C_i = (l_1 \vee l_2 \vee \ldots \vee l_k)$$

and each **literal** l_j is of the form

$$l_j \in \{p \mid p \in PROP\} \cup \{\neg p \mid p \in PROP\}$$

An example of a CNF consisting of the literals p, q, r, s is $(p \vee q \vee r) \wedge (q \vee \neg s \vee r) \wedge (p \vee \neg q \vee s)$.

Theorem 12.1. *Every formula ϕ can be converted into a CNF.*

This theorem states that we can assume that $KB \in CNF$. This is essential for our subsequent discussion on resolution.

For example, if we were given the statement $\phi = ((p \wedge q) \vee (r \wedge s))$, we can convert it to its CNF:

$$\begin{aligned}
\phi &= ((p \wedge q) \vee (r \wedge s)) & \\
&= (p \vee (r \wedge s)) \wedge (q \vee (r \wedge s)) & \text{distributive law} \\
&= ((p \vee r) \wedge (p \vee s)) \wedge ((q \vee r) \wedge (q \vee s)) & \text{distributive law} \\
&\equiv (p \vee r) \wedge (p \vee s) \wedge (q \vee r) \wedge (q \vee s) & \because \text{CNF}
\end{aligned}$$

12.1.4 Resolution

We are interested to know if $KB \to \alpha$ is *VALID*, and we say $KB \models \alpha$ (i.e. KB entails α) if and only if $KB \to \alpha$ is *VALID*. Observe that ϕ is *VALID* if and only if $\neg\phi$ is *UNSAT*. Thus, we can determine whether ϕ is *VALID* by checking whether $\neg\phi$ is *UNSAT*. How could we check if ϕ is *SAT* or *UNSAT*?

One way to do so is by checking the value $\phi(\tau)$ for all τ. However, recall that there are many possible truth assignments τ, on the scale of $2^{|PROP|}$.

Another way to check if ϕ is *SAT* or *UNSAT* is by checking whether there exists a contradiction in the statement. In other words, if we find a clause C_i in the CNF such that C_i and $\neg C_i$ coexist, then we can immediately conclude that the CNF is *UNSAT*.

- For example, if we are given a statement $\phi = (p_1) \wedge (\neg p_1) \wedge (p_2 \vee p_3 \vee p_4) \wedge \ldots$ where $PROP = \{p_1, p_2, \ldots, p_{100}\}$, we can immediately conclude that ϕ is *UNSAT* because $p_1 \wedge \neg p_1$ is *UNSAT*.

By generalizing the above idea, suppose now that we are given $\phi = (\alpha \vee p) \wedge (\neg p \vee \beta)$. We can construct the truth table for this statement

α	β	p	ϕ	$\alpha \vee \beta$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	1	1
1	0	0	1	1
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

and from the truth table, we can derive that $\phi \to (\alpha \vee \beta)$. This step is also known as a **resolution**. Another important observation is that if $C_1 \wedge C_2 \to C_3$ is *VALID*, then $(C_1 \wedge C_2) \leftrightarrow (C_1 \wedge C_2 \wedge C_3)$. By riding on this observation and the idea of resolutions, we can solve a CNF by repeatedly applying resolution, and if we encounter $(C_i \wedge \neg C_i)$ at any point in time, we can then conclude that the formula is *UNSAT*.

For example, for the question posed in tutorial, we can solve it using the resolution approach in the following manner:

Given $(p \vee q \vee r) \wedge (\neg p \vee q \vee s) \wedge (\neg s \vee r) \wedge (\neg q \vee r) \wedge (\neg q \vee \neg r) \wedge (q \vee \neg r)$,
Assuming $C_1, C_2, \ldots, C_6$ are assigned to each of the above clauses,
C_7(resolution over $C_1 \wedge C_2$) : $(q \vee r \vee s)$
C_8(resolution over $C_7 \wedge C_3$) : $(q \vee r)$
C_9(resolution over $C_8 \wedge C_4$) : r
C_{10}(resolution over $C_5 \wedge C_6$) : $\neg r$
C_{11}(resolution over $C_9 \wedge C_{10}$) : $UNSAT$

Recall from last week that we have discussed the problem posed by Greek philosophers, and modelled it using propositions:

$$\text{All Greeks are human} : g \to h \equiv (\neg g \vee h)$$
$$\text{All humans are mortal} : h \to m \equiv (\neg h \vee m)$$

How could we determine if $(g \to m)$ is SAT? We could also use the resolution approach to determine whether $(\neg g \vee h) \wedge (\neg h \vee m)$ entails $g \to m \equiv (\neg g \vee m)$.

Previously, we have mentioned that $KB \models \alpha$ if and only if $KB \to \alpha$ is *VALID*. So,

$$(KB \to \alpha) \text{ is } \mathit{VALID} \leftrightarrow (\neg KB \vee \alpha) \text{ is } \mathit{VALID} \leftrightarrow (KB \wedge \neg\alpha) \text{ is } \mathit{UNSAT}$$

and thus we can determine if $(g \to m)$ is SAT by checking if $(\neg g \vee h) \wedge (\neg h \vee m) \wedge g \wedge (\neg m)$ is *UNSAT*. Using the resolution approach,

Let $C_1 = (\neg g \vee h), C_2 = (\neg h \vee m), C_3 = g, C_4 = \neg m$, then
C_5(resolution over $C_1 \wedge C_3$) : h
C_6(resolution over $C_2 \wedge C_4$) : $\neg h$
C_7(resolution over $C_5 \wedge C_6$) : $UNSAT$

Thus, we can conclude that $(g \to m)$ is *VALID*.

12.1.5 Resolution refutation

We define that resolution refutation of ϕ (given in CNF) to be a list of clauses $C_1, C_2, \ldots, C_t$ such that either $C_i \in \phi$ or C_i is derived from C_a, C_b such that $a, b < i$, and $C_i = UNSAT$.

So, for our example from the tutorial, the resolution refutation would be $C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{11}$.

Theorem 12.2. *ϕ is UNSAT if and only if there exists a resolution refutation of ϕ.*

Thus, if we can find a resolution refutation for a statement ϕ, we would have proven that ϕ is *UNSAT*.

Remark: However, note that we still need to find a way of applying resolution between clauses, and this step is hard and may take an exponential amount of time. In fact, the brute force approach may have a better performance than the resolution approach in some circumstances. Given ϕ, checking if ϕ is SAT is *NP*-complete.