CSE 579: Knowledge Representation & Reasoning

Module 1: Propositional Logic

Ali Altunkaya
Spring 2023

Outline

1. Propositional Logic (PL)
    1. Introduction
    2. Syntax (alphabet)
    3. Semantics (meaning)
    4. Notions in PL
        1. Satisfiability
        2. Tautology
        3. Equivalence
        4. Entailment
    5. Computing Propositional Logic (PL): DPLL algorithm

Module 1 Highlights

We will study 3 different formal languages during the course:

1-) Propositional Logic

2-) First Order Logic

3-) Answer Set Theory

Ultimate goal: Encode <u>natural language</u> into a <u>formal language</u> so we can do ….(reasoning)

Module 1 Highlights

Introduction to Propositional Logic (PL):

– Goal: How we can use PL to represent (encode) knowledge so we can manipulate it to derive new information.

– PL is a branch of logic, which studies ways of combining or altering statements/propositions to form more complicated statements/propositions.

– Study of propositions (declarative sentences or statements) about the world which can be given a truth value (True/False).

– PL uses components : not, and, or, if then.

– PL is compositional. You can combine propositions (F and G).

Module 1 Highlights

Introduction to Propositional Logic (PL):

An **atom** represents a proposition, which is either true or false.

- The sum of 3 and 5 equals to 8. (an atom)
- ~~Ready, steady, go. (not an atom)~~

Propositional connectives are used to compose the meaning.

- If a number is divisible by 4, then it is divisible by 2.
- $\text{divisibleBy4} \rightarrow \text{divisibleBy2}$

Reasoning example: If the train arrives late and there are no taxis at the station then John is late for his meeting.

- $\text{TrainLate} \land \neg\text{Taxi} \rightarrow \text{JohnLate}$
- $p \land \neg q \rightarrow r$

Module 1 Highlights

What is syntax?

What is semantics?

What is the relation or difference between them?

Module 1 Highlights

What is syntax, semantics, and the relation or difference between them?

<u>Syntax:</u> Grammar of a statement in a language (nouns, verbs etc).

<u>Semantics:</u> Meaning of a statement in the real-world.

Interpretation from syntax to semantics.

<u>Example:</u> “Colorless green ideas sleep furiously.”

Syntax: correct

Semantics: incorrect

Module 1 Highlights

Syntax: Alphabet of Propositional Logic (PL):

A propositional signature is a set of symbols called atoms:

- TrainLate, TaxiLate, JohnLate, p, q, r

Three types of propositional connectives :

- 2-place (binary): $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (implication)
- 1-place (unary): $\neg$ (negation or not)
- 0-place: $\perp$ (bottom, FALSE), T (top, TRUE)

The alphabet of propositional logic consists of:

- The atoms from the signature
- The propositional connectives
- Parentheses

Module 1 Highlights

Syntax: Alphabet of Propositional Logic (PL):

A propositional formula of signature σ is defined recursively:

- Every atom is a formula.
- Both 0-place connectives ($\bot$ T) are formulas.
- If F is a formula then is ¬F a formula too.
- For any binary connective ◌ (∧, ∨, →), if F and G are formulas then (F◌G) is a formula too.

$$((\neg p \wedge q) \rightarrow \left(p \wedge (q \vee \neg r)\right))$$

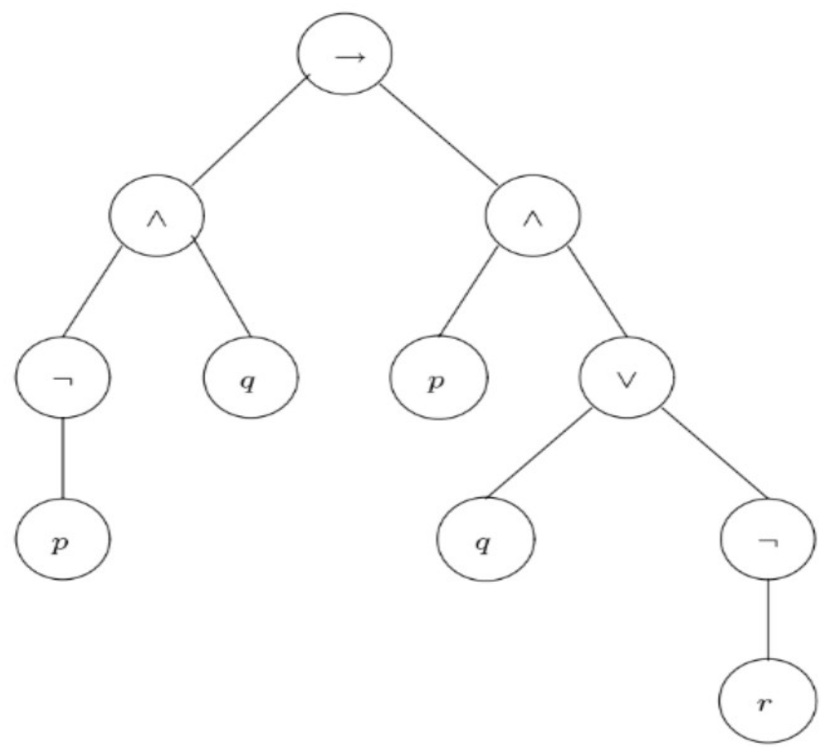
Module 1 Highlights

Syntax: Alphabet of Propositional Logic (PL):

Subformulas of are the formulas corresponding to the subtrees of the parse tree of:

$$((\neg p \wedge q) \rightarrow (p \wedge (q \vee \neg r)))$$

9 subformulas



Module 1 Highlights

Syntax: Alphabet of Propositional Logic (PL):

Binding precedence allows us to avoid many parentheses:

1) ¬
2) ∧,∨
3) →, ↔

$$((\neg p \wedge q) \rightarrow (p \wedge (q \vee \neg r)))$$

$\neg p \wedge q \rightarrow p \wedge (q \vee \neg r)$

Module 1 Highlights

Semantics of Propositional Logic (PL):

– What is interpretation?

– How many interpretations are there for a given signature?

– What are truth values?

– What is truth table?

– Tables associated with propositional connectives

– Evaluation of a Formula

– Satisfaction

- see slides...

Module 1 Highlights

Semantics of Propositional Logic (PL):

Interpretation: is a function which maps signature symbols to truth values.

– How many interpretations are there for a given signature? 2^n

– What are truth values: True/False.

– Truth table

x	y	$\wedge(x,y)$	$\vee(x,y)$	$\rightarrow(x,y)$	$\leftrightarrow(x,y)$
f	f	f	f	t	t
f	t	f	t	t	f
t	f	f	t	f	f
t	t	t	t	t	t

– Tables associated with propositional connectives

– Evaluation of a Formula

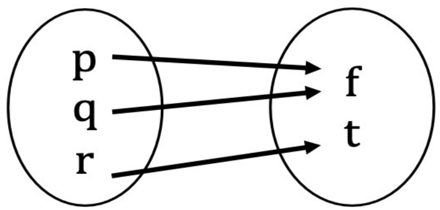
– <u>Interpretations</u> that satisfies a formula are called <u>models</u>.

Module 1 Highlights

Satisfiability: If a formula F is TRUE for some interpretation, then we say that the interpretation I satisfies F. Only one interpretation is sufficient. For a given formula F, is there any interpretation that satisfies it?

<u>NP-complete problem:</u> For n different atoms in the signature, there are 2^n possible interpretations. And we need to check all of them in worst-case scenario.

p	q	r
f	f	t



There are 10^{82} atoms in the universe.

$10^{82} =\sim 2^{300}$.

Formula: $((\neg p \wedge q) \rightarrow (p \wedge (q \vee \neg r))$

Q: The truth value of the formula

1. if I(p)=t, I(q)=t, I(r)=t?

t

2. if I(p)=f, I(q)=t, I(r)=f?

f

Q: How many interpretations for {p, q, r}?

p	q	r
f	f	f
f	f	t
f	t	f
f	t	t
t	f	f
t	f	t
t	t	f
t	t	t

Module 1 Highlights

Notions in Propositional Logic (PL):

– **Satisfiability**: only one interpretation is sufficient

– **Tautology**: all interpretations should satisfy it

– **Equivalence**: F is equivalent to G,

if for every interpretation I, $F^I = G^I$.

– **Entailment**: A set of formulas θ, entails a formula F,

if every interpretation that satisfies all formulas in θ, also satisfies F.

also called logical consequence.

– Some Useful Equivalence formulas (e.g. DeMorgan rules), used for reductions.

– **Reductions between problems**: all these 4 notions are strongly related with each other.

See slides....

Module 1 Highlights

Propositional Logic (PL) & Knowledge Representation (KR):

Syntax:

In Natural Languages, there are words and sentences (or statements).

In Propositional Logic, there are atoms and formulas.

Semantics:

In Natural Languages, meaning can be any object in Real-World.

In Propositional Logic, meaning can be only TRUE or FALSE.

Module 1 Highlights

Computing Propositional Logic (PL):

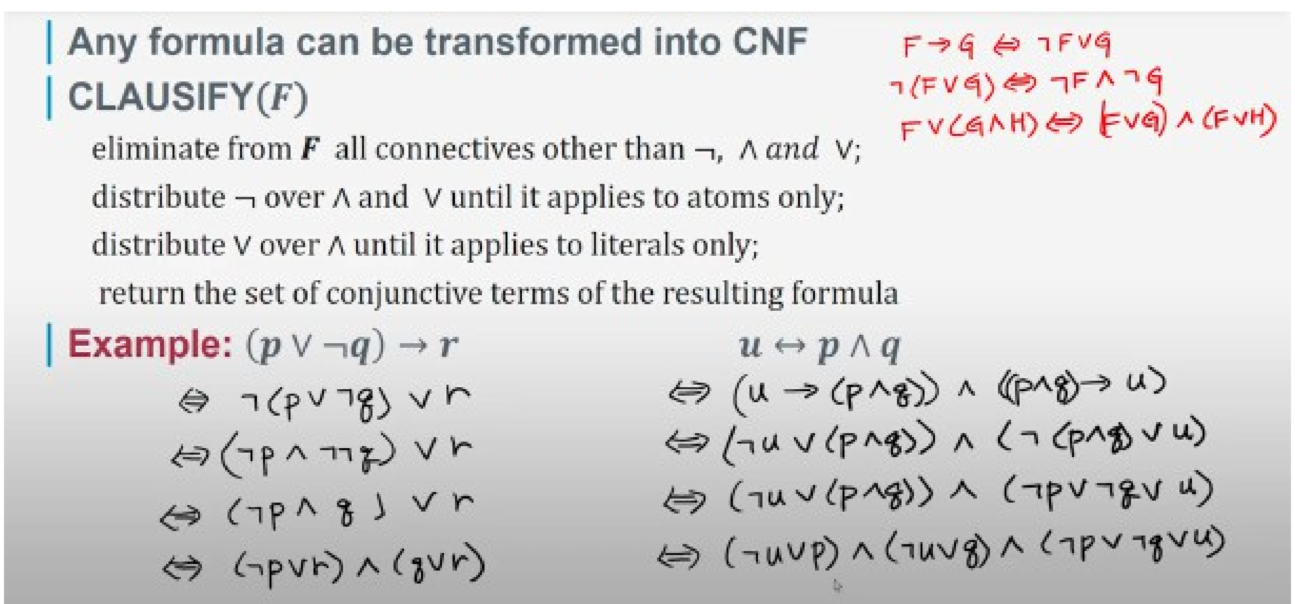
Satisfiability (SAT) problem: We want to make sure, there is no contradiction between our proposition formulas.

– Most SAT solvers accept Conjunctive Normal Form (CNF) as input.

– Any formula can be transformed into CNF, using CLAUSIFY algorithm and equivalence transformations and De Morgan's laws.

– While transforming a formula into CNF, too many clauses can be generated 2^n. We can solve it by using auxiliary atoms.

$$\boxed{\begin{array}{c}\text{F}\\ (\boldsymbol{p} \vee \boldsymbol{q}) \wedge \boldsymbol{r}\end{array}} \xrightarrow{\text{Tr}} \boxed{\begin{array}{c}\Gamma\\ \boldsymbol{u} \leftrightarrow \boldsymbol{p} \vee \boldsymbol{q}\\ \boldsymbol{u} \wedge \boldsymbol{r}\end{array}}$$



Any formula can be transformed into CNF

CLAUSIFY(F)

eliminate from F all connectives other than $\neg$, $\wedge$ *and* $\vee$;

distribute $\neg$ over $\wedge$ and $\vee$ until it applies to atoms only;

distribute $\vee$ over $\wedge$ until it applies to literals only;

return the set of conjunctive terms of the resulting formula

$F \rightarrow G \Leftrightarrow \neg F \vee G$

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

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

Example: $(p \vee \neg q) \rightarrow r$

$\Leftrightarrow \neg(p \vee \neg q) \vee r$

$\Leftrightarrow (\neg p \wedge \neg\neg q) \vee r$

$\Leftrightarrow (\neg p \wedge q) \vee r$

$\Leftrightarrow (\neg p \vee r) \wedge (q \vee r)$

$u \leftrightarrow p \wedge q$

$\Leftrightarrow (u \rightarrow (p \wedge q)) \wedge ((p \wedge q) \rightarrow u)$

$\Leftrightarrow (\neg u \vee (p \wedge q)) \wedge (\neg (p \wedge q) \vee u)$

$\Leftrightarrow (\neg u \vee (p \wedge q)) \wedge (\neg p \vee \neg q \vee u)$

$\Leftrightarrow (\neg u \vee p) \wedge (\neg u \vee q) \wedge (\neg p \vee \neg q \vee u)$

Module 1 Highlights

Computing Propositional Logic (PL):

Unit propagation: Assume formulas are in CNF.

- If there is a unit clause (single literal), we can simplify the formula by assuming these two options: Either it is TRUE or FALSE.
- Then we can easily check the satisfiability.
- If there is no unit clause, just choose one atom and guess (T, F)

DPLL algorithm is actually just application of Unit-propagation in a recursive algorithm.

$$
\begin{array}{l}
\mathrm{DPLL}(F, U) \\
\quad \textsc{Unit-propagate}(F, U); \\
\quad \textbf{if } F \text{ contains the empty clause } \textbf{then} \text{ return}; \\
\quad \textbf{if } F = \top \textbf{ then} \text{ exit with a model of } U; \\
\quad L \leftarrow \text{a literal containing an atom from } F; \\
\quad \mathrm{DPLL}(F|_L, U \cup \{L\}); \\
\quad \mathrm{DPLL}(F|_{\overline{L}}, U \cup \{\overline{L}\})
\end{array}
$$

Module 1 Highlights

Computing Propositional Logic (PL):

Q: Apply DPLL to $(\neg p \vee q) \wedge (\neg p \vee r) \wedge (q \vee r) \wedge (\neg q \vee \neg r)$

$DPLL((\neg p \vee q) \wedge (\neg p \vee r) \wedge (q \vee r) \wedge (\neg q \vee \neg r),\ \emptyset)$

$UP($ " $,\ \emptyset)$

$L := p$

$DPLL(q \wedge r \wedge (q \vee r) \wedge (\neg q \vee \neg r),\ \{p\})$

$UP($ " $,\ \{p\})$

$\{p, q\})$

$F := r \wedge \neg r$ $\quad \{p, q, r\})$

$F := T \wedge \neg T$

$DPLL((q \vee r) \wedge (\neg q \vee \neg r),\ \{\neg p\})$

$L := q$

$DPLL(\neg r$ $\quad \{\neg p, q\})$

$UP(\neg r$ $\quad \{\neg p, q\})$

$F := T$ $\quad \boxed{\{\neg p, q, \neg r\})}$

Thanks

&

Questions