Knowledge Representation
Chapter 2. Propositional Representation and Reasoning

Pedro Cabalar

Dept. Computer Science
University of Corunna, SPAIN

February 14, 2021

Propositional Logic: Syntax

- Def. Propositional Signature Σ: set of propositions or atoms. E.g. $\Sigma = \{happy, rain, weekend\}$.
- Def. Propositional language $\mathcal{L}_\Sigma$, set of well formed formulas (wff).

$$\boxed{\begin{array}{ccccc} p & \top & \bot & \neg\alpha & \\ \alpha \vee \beta & \alpha \wedge \beta & \alpha \rightarrow \beta & \alpha \leftrightarrow \beta & (\alpha) \end{array}}$$

where $p \in \Sigma$ and $\alpha, \beta \in \mathcal{L}_\Sigma$.

- Alternative notations: implication $\rightarrow$, $\supset$, $\Rightarrow$; equivalence $\equiv$, $=$, $\leftrightarrow$, $\Leftrightarrow$
- Precedence: $\equiv, \rightarrow, \vee, \wedge, \neg$. Binary ops. left associative.
- Def. literal = an atom p or its negation $\neg p$.
- Def. theory = set of formulas $\Gamma \subseteq \mathcal{L}_\Sigma$.

Propositional Logic: Semantics

- Def. interpretation is a function $\mathcal{I} : \Sigma \longrightarrow \{1, 0\}$
  Example: $\mathcal{I}(\mathit{happy}) = 1, \ \mathcal{I}(\mathit{rain}) = 0, \ \mathcal{I}(\mathit{weekend}) = 1$
- Alternative representation: set $\mathcal{I} \subseteq \Sigma$ of (true) atoms.
  Example: $I = \{\mathit{happy}, \mathit{weekend}\}$
- We extend its use to formulas $\mathcal{I} : \mathcal{L}_\Sigma \longrightarrow \{1, 0\}$.
  $\mathcal{I}(\alpha) =$ replace each $p \in \Sigma$ in α by $\mathcal{I}(p)$ and apply:

$$\begin{aligned} \mathcal{I}(\top) &= 1 \\ \mathcal{I}(\bot) &= 0 \end{aligned}$$

	$\neg$
1	0
0	1

		$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

- Example: $\mathcal{I}(\neg \mathit{rain} \rightarrow \neg \mathit{weekend})\ \mathcal{I}(\neg 0 \rightarrow \neg 1)\ \mathcal{I}(1 \rightarrow 0) = 0$

Propositional Logic: Semantics

- Def. $\mathcal{I}$ satisfies α, written $\mathcal{I} \models \alpha$, iff $\mathcal{I}(\alpha) = 1$.
- Satisfaction can also be defined inductively as follows:

i)	$\mathcal{I} \models \top$	and $\mathcal{I} \not\models \bot$.
ii)	$\mathcal{I} \models p$	iff $\mathcal{I}(p) = 1$.
iii)	$\mathcal{I} \models \neg\alpha$	iff $\mathcal{I} \not\models \alpha$.
iv)	$\mathcal{I} \models \alpha \wedge \beta$	iff $\mathcal{I} \models \alpha$ and $\mathcal{I} \models \beta$.
v)	$\mathcal{I} \models \alpha \vee \beta$	iff $\mathcal{I} \models \alpha$ or $\mathcal{I} \models \beta$ (or both).
vi)	$\mathcal{I} \models \alpha \rightarrow \beta$	iff $\mathcal{I} \not\models \alpha$ or $\mathcal{I} \models \beta$ (or both).
vii)	$\mathcal{I} \models \alpha \equiv \beta$	iff ($\mathcal{I} \models \alpha$ iff $\mathcal{I} \models \beta$).

- $\mathcal{I}$ is a *model* of Γ, written $\mathcal{I} \models \Gamma$, iff it satisfies all formulas in Γ.

Propositional Logic: Semantics

- We can define $M(\Gamma)$ = the set of models of a theory (or formula) Γ. Example: $M(a \vee b) = \{\{a, b\}, \{a\}, \{b\}\}$
- The models of a formula can be inspected by structural induction:

$$
\begin{aligned}
M(\alpha \vee \beta) &= M(\alpha) \cup M(\beta) \\
M(\alpha \wedge \beta) &= M(\alpha) \cap M(\beta) \\
M(\neg \alpha) &= 2^{\Sigma} \setminus M(\alpha)
\end{aligned}
$$

- Two formulas α, β are equivalent if $M(\alpha) = M(\beta)$ (same models)

Propositional Logic: Semantics

- From a set S of interpretations: do you know a method to get a formula α s.t. $M(\alpha) = S$?
- Example: find α to cover $M(\alpha) = \{\{a, c\}, \{b, c\}, \{a, b, c\}\}$
- Does this formula α always exist?

Propositional Logic: Semantics

- Def. relation $\Gamma \models \alpha$ is called logical consequence or entailment and defined as $M(\Gamma) \subseteq M(\alpha)$.
  Example $\{happy, (rain \rightarrow \neg happy)\} \models \neg rain$
- If $M(\alpha) = \emptyset$ (no models!), α is inconsistent or unsatisfiable
  Examples: $rain \wedge \neg rain$, $\bot$, $\ldots$
- If $M(\alpha) = 2^{\Sigma}$ (all interpretations are models), α, is valid or a tautology. Examples: $rain \vee \neg rain$, $\top$, $b \wedge c \wedge d \rightarrow (d \rightarrow b)$, $\ldots$
- We write $\models \alpha$ to mean that α is a tautology
  Note: this is $\emptyset \models \alpha$, so we require $M(\emptyset) = 2^{\Sigma} \subseteq M(\alpha)$

Propositional Logic: Semantics

Theorem

$\models \alpha \to \beta$ is equivalent to $\alpha \models \beta$.

Definition (Weaker/stronger formula)

When $\models \alpha \to \beta$, or just $M(\alpha) \subseteq M(\beta)$, we say that α is stronger than β (or β is weaker α).

- Which are the strongest and weakest possible formulae?
- Examples: for each pair, which is the strongest?

$$\begin{array}{ccc} p & \leftarrow & p \wedge q \\ p & \to & p \vee \neg q \\ p \vee q & \leftarrow & p \wedge q \\ p & \to & (q \to p) \\ p \wedge \neg q & & \neg p \wedge q \end{array}$$

Propositional Reasoning

Reasoning: $\{P_1, \ldots, P_n\} \models C$

does conclusion C follow from premises $\{P_1, \ldots, P_n\} = KB$ (the Knowledge Base)?

Example: $KB =$ but we need formulas, not sentences?

P_1: On *w*eekends, I don't watch *tv*

P_2: I'm *h*appy when it *r*ains, except in the *w*eekend

P_3: I'm watching *tv* but I'm not *h*appy

Can I conclude this?

C: it is not *r*aining

From human to formal language . . .

$A \rightarrow B$	A *implies* B
	A is a *sufficient condition* for B
	B is a *necessary condition* for A
	if A then B
	B if A
	A only if B
	B given that A
	B provided that A
$A \leftrightarrow B$	A is *equivalent* to B
	A if and only if (iff) B
$A \vee B$	A or B (inclusive or)
	A unless B, A except B
$\neg(A \leftrightarrow B)$	A or B (exclusive or)

Del lenguaje humano al formal ...

$A \rightarrow B$	A *implica* B
	A es *suficiente* para B
	B es *necesario* para A
	si A entonces B
	B si A
	A sólo si B
	B siempre que A
$A \leftrightarrow B$	A *equivale* a B
	A si y sólo si B
$A \vee B$	A ó B (inclusivo)
	A a no ser que (a menos que) B
	A excepto si B
$\neg(A \leftrightarrow B)$	A ó B (exclusivo)

Propositional Reasoning

Reasoning: $\{P_1, \ldots, P_n\} \models C$

does conclusion C follow from premises $\{P_1, \ldots, P_n\} = KB$ (the Knowledge Base)?

Example: $KB =$

P_1: On *w*eekends, I don't watch *tv* $(w \to \neg tv)$

P_2: I'm *h*appy when it *r*ains, except in the *w*eekend $(r \wedge \neg w \to h)$

P_3: I'm watching *tv* but I'm not *h*appy $(tv \wedge \neg h)$

Can I conclude this?

C: it is not *r*aining $(\neg r)$

Propositional Reasoning

Definition (Entailment)

A theory KB entails conclusion C, written $KB \models C$, when all models of KB are models of C. If so, C is called a semantic consequence of KB.

- In propositional logic, $\{P_1, P_2, P_3\} \models C$ is the same as checking that the formula $P_1 \wedge P_2 \wedge P_3 \rightarrow C$ is a tautology or, equivalently, that its negation $P_1 \wedge P_2 \wedge P_3 \wedge \neg C$ is inconsistent

Definition (SAT decision problem)

Decision problem $SAT(\alpha) \in \{yes, no\}$ checks whether a formula α has some model. (Time) complexity: **NP**-complete problem.

- In other words:
  $\{P_1, P_2, P_3\} \models C$ iff $SAT(P1 \wedge P2 \wedge P3 \wedge \neg C) = no$.

Propositional Reasoning

- First naive method: check all interpretations ($2^4 = 16$) one by one (truth table) to obtain a 0 in all cases.
- $\mathcal{I}(P_1 \wedge P_2 \wedge P_3 \wedge \neg C) = 0$ when some conjunct is 0.

h	tv	w	r	P_1 $(w \rightarrow \neg tv)$	P_2 $(r \wedge \neg w \rightarrow h)$	P_3 $tv \wedge \neg h$	$\neg C$ r
0	0	0	0	1	1	0	0
		$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$
0	1	0	0	1	1	1	0
0	1	0	1	1	0	1	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
		$\vdots$					

Propositional Reasoning

- Computational cost is exponential = 2^n with $n = |\Sigma|$ number of atoms. Can we perform better?
- Not much hope for the worst case: **NP**-complete!
- However, enumeration of interpretations always forces worst case. We can do better in particular cases.
- In our example: $tv \wedge \neg h$ and r fix the truth of 3 atoms: $\mathcal{I}(h) = 0$, $\mathcal{I}(tv) = 1$ and $\mathcal{I}(r) = 1$. Only w needs to be checked

$$
\begin{array}{crcl}
 & (w \to \neg tv) & \wedge & (r \wedge \neg w \to h) \\
\equiv & (\neg w \vee \neg tv)(\neg w \vee \neg tv)(\neg w \vee \neg \top) & \wedge & (\neg r \vee w \vee h)(\neg r \vee w \vee h \\
\equiv & (\neg w \vee \bot) & \wedge & (\bot \vee w \vee \bot) \\
\equiv & \neg w & \wedge & w \quad \text{inconsistent!}
\end{array}
$$

SAT solvers

- SAT solvers: nowadays, SAT is an outstanding state-of-the-art research area for search algorithms. There exist many efficient tools and commercial applications. See `www.satlive.com`
- SAT keypoint: instead of designing an *ad hoc* search algorithm, encode the problem into propositional logic and use SAT as a backend.
- SAT solvers represent the input (*KB* and conclusions) as a set (conjunction) of “clauses”, where clause = disjunction of literals. This is called Conjunctive Normal Form (CNF).

Conjunctive Normal Form (CNF)

Getting the CNF. Example:

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

1. replace $\alpha \rightarrow \beta$ by $\neg\alpha \vee \beta$ and $\alpha \leftrightarrow \beta$ by $(\alpha \wedge \beta) \vee (\neg\alpha \wedge \neg\beta)$
2. Negation Normal Form (NNF):
   apply De Morgan laws until $\neg$ only applied to atoms
3. apply distributivity $\wedge, \vee$ and associativity to get conjunction of disjunctions

- Warning: distributivity may have an exponential cost. Example $(a \wedge b) \vee (c \wedge d) \vee (e \wedge f) \vee (h \wedge i)$
- Some techniques [Tseitin68] allow generating a CNF in polynomial time but introducing new auxiliary atoms.

Conjunctive Normal Form (CNF)

- If *KB* is a set of facts and implications involving literals, it is (almost) in CNF!
- Example: just change the sign of left literals in $\rightarrow$

$$\begin{array}{ccc} P_1 & \wedge & P_2 \\ (w \rightarrow \neg tv)(w \rightarrow \neg tv)(w \rightarrow \neg tv) & \wedge & (r \wedge \neg w \rightarrow h)(r \wedge \neg w \rightarrow h)(r \wedge \\ (\neg w \vee \neg tv)\underbrace{(\neg w \vee \neg tv)}_{C_1} & \wedge & (\neg r \vee w \vee h)\underbrace{(\neg r \vee w \vee}_{C_2} \end{array}$$

  we get five clauses: C_3, C_4, C_5 are unit clauses.
- We will call constraint to the negation of a CNF clause

$$\underbrace{(w \wedge tv)}_{\neg C_1} \quad \underbrace{(r \wedge \neg w \wedge \neg h)}_{\neg C_2} \quad \underbrace{\neg tv}_{\neg C_3} \quad \underbrace{h}_{\neg C_4} \quad \underbrace{\neg r}_{\neg C_5}$$

- Constraints can be easily obtained from implications of literals: change the sign of the right literals in $\rightarrow$.