Chapter 2: Propositional Logic

Adnan Darwiche[1]

[1] Lecture slides for *Modeling and Reasoning with Bayesian Networks*, Adnan Darwiche, Cambridge University Press, 2009.

Propositional Logic

One needs a language for expressing events before one can write statements that declare their truth or specify their probabilities.

Propositional logic, which is also known as Boolean logic or Boolean algebra, provides such a language:

- Syntax: rules for forming propositional sentences.
- Semantics: rules for interpreting propositional sentences.

Syntax of Propositional Sentences

Consider an alarm is meant for detecting burglaries and may also be triggered by an earthquake.

Syntax of Propositional Sentences

Consider an alarm is meant for detecting burglaries and may also be triggered by an earthquake.

Example

Burglary $\vee$ Earthquake.

Burglary and Earthquake are called propositional variables, and $\vee$ represents logical disjunction (or).

Syntax of Propositional Sentences

Consider an alarm is meant for detecting burglaries and may also be triggered by an earthquake.

Example

Burglary $\vee$ Earthquake.

Burglary and Earthquake are called propositional variables, and $\vee$ represents logical disjunction (or).

Example

Burglary $\vee$ Earthquake $\implies$ Alarm

$\implies$ represents logical implication.

Syntax of Propositional Sentences

Consider an alarm is meant for detecting burglaries and may also be triggered by an earthquake.

Example

Burglary $\vee$ Earthquake.

Burglary and Earthquake are called propositional variables, and $\vee$ represents logical disjunction (or).

Example

Burglary $\vee$ Earthquake $\implies$ Alarm

$\implies$ represents logical implication.

Example

$\neg$Burglary $\wedge$ $\neg$Earthquake $\implies$ $\neg$Alarm

$\neg$ represents logical negation (not), and $\wedge$ represents logical conjunction (and).

Syntax of Propositional Sentences

Propositional sentences are formed using a set of propositional variables, $P_1, \ldots, P_n$, which take one of two values, typically true and false.

These variables are also called Boolean variables or binary variables.

Syntax of Propositional Sentences

Propositional sentences are formed using a set of propositional variables, $P_1, \ldots, P_n$, which take one of two values, typically true and false.

These variables are also called Boolean variables or binary variables.

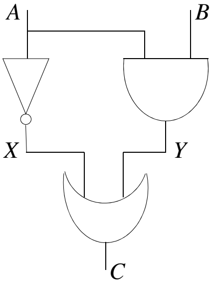
Propositional sentences are formed as follows

- Every propositional variable P_i is a sentence.
- If α and β are sentences, then $\neg\alpha$, $\alpha \wedge \beta$, and $\alpha \vee \beta$ are also sentences.

Note: $\alpha \implies \beta$ is shorthand for $\neg\alpha \vee \beta$.

Propositional Knowledge Bases

Circuit:



Knowledge base:

$$\Delta = \left\{ \begin{array}{lcr} A & \Longrightarrow & \neg X, \\ \neg A & \Longrightarrow & X, \\ A \wedge B & \Longrightarrow & Y, \\ \neg(A \wedge B) & \Longrightarrow & \neg Y, \\ X \vee Y & \Longrightarrow & C, \\ \neg(X \vee Y) & \Longrightarrow & \neg C. \end{array} \right.$$

Semantics of Propositional Sentences

Propositional logic provides a formal framework for defining properties of sentences, such as consistency and validity, and relationships among them, such as implication, equivalence and mutual exclusiveness.

Semantics of Propositional Sentences

Propositional logic provides a formal framework for defining properties of sentences, such as consistency and validity, and relationships among them, such as implication, equivalence and mutual exclusiveness.

- Most people would agree that
  - $A \wedge \neg A$ is inconsistent (will never hold),
  - $A \vee \neg A$ is valid (always holds),
  - A and $(A \implies B)$ imply B, and
  - $A \vee B$ is equivalent to $B \vee A$.

Semantics of Propositional Sentences

Propositional logic provides a formal framework for defining properties of sentences, such as consistency and validity, and relationships among them, such as implication, equivalence and mutual exclusiveness.

- Most people would agree that
  - $A \wedge \neg A$ is inconsistent (will never hold),
  - $A \vee \neg A$ is valid (always holds),
  - A and $(A \implies B)$ imply B, and
  - $A \vee B$ is equivalent to $B \vee A$.
- It may not be as obvious that $A \implies B$ and $\neg B \implies \neg A$ are equivalent, or that $(A \implies B) \wedge (A \implies \neg B)$ implies $\neg A$.

Worlds, Models and Events

A world

is a particular state of affairs in which the value of each propositional variable is known.

Worlds, Models and Events

A world

is a particular state of affairs in which the value of each propositional variable is known.

world	Earthquake	Burglary	Alarm
ω_1	true	true	true
ω_2	true	true	false
ω_3	true	false	true
ω_4	true	false	false
ω_5	false	true	true
ω_6	false	true	false
ω_7	false	false	true
ω_8	false	false	false

It is often called a truth assignment, a variable assignment, or a variable instantiation.

Worlds, Models and Events

A sentence α is either true or false at a particular world ω.

Worlds, Models and Events

A sentence α is either true or false at a particular world ω.

world	Earthquake	Burglary	Alarm
ω_1	true	true	true
ω_2	true	true	false
ω_3	true	false	true
ω_4	true	false	false
ω_5	false	true	true
ω_6	false	true	false
ω_7	false	false	true
ω_8	false	false	false

$\omega \models \alpha$ reads

α is true/hold at ω

α is satisfied/entailed by ω

Example

$\omega_1 \models$ Burglary: world assigns true to Burglary.

$\omega_3 \models \neg$Burglary: world assigns false to Burglary.

$\omega_4 \models$ Burglary $\vee$ Earthquake: world assigns true to Earthquake.

Worlds, Models and Events

Models of sentence α are worlds that satisfy α

$$Mods(\alpha) \stackrel{def}{=} \{\omega : \omega \models \alpha\}.$$

- Every sentence α can be viewed as representing a set of worlds $Mods(\alpha)$, which is called the event denoted by α.
- We will use sentence and event interchangeably.

Worlds, Models and Events

Models of sentence α are worlds that satisfy α

$$Mods(\alpha) \stackrel{def}{=} \{\omega : \ \omega \models \alpha\}.$$

- Every sentence α can be viewed as representing a set of worlds $Mods(\alpha)$, which is called the event denoted by α.
- We will use sentence and event interchangeably.

Using the definition of satisfaction ($\models$)

- $Mods(\alpha \wedge \beta) = Mods(\alpha) \cap Mods(\beta)$.
- $Mods(\alpha \vee \beta) = Mods(\alpha) \cup Mods(\beta)$.
- $Mods(\neg\alpha) = \overline{Mods(\alpha)}$.

Worlds, Models and Events

Following are some example sentences and their truth at worlds:

- Earthquake is true at worlds $\omega_1, \ldots, \omega_4$:

$$\mathit{Mods}(\text{Earthquake}) = \{\omega_1, \ldots, \omega_4\}.$$

- $\neg$Earthquake is true at worlds $\omega_5, \ldots, \omega_8$:

$$\mathit{Mods}(\neg\text{Earthquake}) = \overline{\mathit{Mods}(\text{Earthquake})}.$$

- $\neg$Burglary is true at worlds $\omega_3, \omega_4, \omega_7, \omega_8$.
- Alarm is true at worlds $\omega_1, \omega_3, \omega_5, \omega_7$.
- $\neg$(Earthquake $\vee$ Burglary) is true at worlds ω_7, ω_8:

$$\mathit{Mods}(\neg(\text{Earthquake}\vee\text{Burglary}) = \overline{\mathit{Mods}(\text{Earthquake}) \cup \mathit{Mods}(\text{Burglary})}.$$

- $\neg$(Earthquake $\vee$ Burglary) $\vee$ Alarm is true at worlds $\omega_1, \omega_3, \omega_5, \omega_7, \omega_8$.
- (Earthquake $\vee$ Burglary) $\implies$ Alarm is true at worlds $\omega_1, \omega_3, \omega_5, \omega_7, \omega_8$.
- $\neg$Burglary $\wedge$ Burglary is not true at any world.

Logical Properties: Consistency

Sentence α is consistent iff

Logical Properties: Consistency

Sentence α is consistent iff

there is at least one world ω at which α is true, $Mods(\alpha) \neq \emptyset$.
Otherwise, the sentence α is inconsistent, $Mods(\alpha) = \emptyset$.

Logical Properties: Consistency

Sentence α is consistent iff

there is at least one world ω at which α is true, $Mods(\alpha) \neq \emptyset$.
Otherwise, the sentence α is inconsistent, $Mods(\alpha) = \emptyset$.

- Symbol false used to denote a sentence which is unsatisfiable.
- Terms satisfiable/unsatisfiable interchangeable with consistent/inconsistent.

Logical Properties: Validity

Sentence α is valid iff

Logical Properties: Validity

Sentence α is valid iff

it is true at every world, $Mods(\alpha) = \Omega$, where Ω is the set of all worlds.

Logical Properties: Validity

Sentence α is valid iff

it is true at every world, $Mods(\alpha) = \Omega$, where Ω is the set of all worlds.

- Symbol true used to denote a sentence which is valid.
- Common to write $\models \alpha$ when α is valid.

Logical Relationships

A logical property applies to a single sentence, while a logical relationship applies to two or more sentences.

Logical Relationships

A logical property applies to a single sentence, while a logical relationship applies to two or more sentences.

Sentences α and β are equivalent iff

Logical Relationships

A logical property applies to a single sentence, while a logical relationship applies to two or more sentences.

Sentences α and β are equivalent iff

they are true at the same set of worlds: $Mods(\alpha) = Mods(\beta)$.

Logical Relationships

A logical property applies to a single sentence, while a logical relationship applies to two or more sentences.

Sentences α and β are equivalent iff

they are true at the same set of worlds: $Mods(\alpha) = Mods(\beta)$.

Sentences α and β are mutually exclusive iff

Logical Relationships

A logical property applies to a single sentence, while a logical relationship applies to two or more sentences.

Sentences α and β are equivalent iff

they are true at the same set of worlds: $Mods(\alpha) = Mods(\beta)$.

Sentences α and β are mutually exclusive iff

they are never true at the same world: $Mods(\alpha) \cap Mods(\beta) = \emptyset$.

Logical Relationships

A logical property applies to a single sentence, while a logical relationship applies to two or more sentences.

Sentences α and β are equivalent iff

they are true at the same set of worlds: $Mods(\alpha) = Mods(\beta)$.

Sentences α and β are mutually exclusive iff

they are never true at the same world: $Mods(\alpha) \cap Mods(\beta) = \emptyset$.

Sentences α and β are exhaustive iff

Logical Relationships

A logical property applies to a single sentence, while a logical relationship applies to two or more sentences.

Sentences α and β are equivalent iff

they are true at the same set of worlds: $Mods(\alpha) = Mods(\beta)$.

Sentences α and β are mutually exclusive iff

they are never true at the same world: $Mods(\alpha) \cap Mods(\beta) = \emptyset$.

Sentences α and β are exhaustive iff

each world satisfies at least one of the sentences:
$Mods(\alpha) \cup Mods(\beta) = \Omega$.

Logical Relationships

A logical property applies to a single sentence, while a logical relationship applies to two or more sentences.

Sentences α and β are equivalent iff

they are true at the same set of worlds: $Mods(\alpha) = Mods(\beta)$.

Sentences α and β are mutually exclusive iff

they are never true at the same world: $Mods(\alpha) \cap Mods(\beta) = \emptyset$.

Sentences α and β are exhaustive iff

each world satisfies at least one of the sentences: $Mods(\alpha) \cup Mods(\beta) = \Omega$.

Sentence α implies sentence β iff

Logical Relationships

A logical property applies to a single sentence, while a logical relationship applies to two or more sentences.

Sentences α and β are equivalent iff

they are true at the same set of worlds: $Mods(\alpha) = Mods(\beta)$.

Sentences α and β are mutually exclusive iff

they are never true at the same world: $Mods(\alpha) \cap Mods(\beta) = \emptyset$.

Sentences α and β are exhaustive iff

each world satisfies at least one of the sentences:
$Mods(\alpha) \cup Mods(\beta) = \Omega$.

Sentence α implies sentence β iff

β is true whenever α is true: $Mods(\alpha) \subseteq Mods(\beta)$.

Notation

$\omega \models \alpha$ means that world ω satisfies sentence α

The symbol $\models$ denotes a relationship between a world and a sentence.

Notation

$\omega \models \alpha$ means that world ω satisfies sentence α

The symbol $\models$ denotes a relationship between a world and a sentence.

$\alpha \models \beta$ means that sentence α implies sentence β

This symbol $\models$ denotes a relationship between two sentences.

Equivalences

Schema	*Equivalent Schema*	*Name*
$\neg$true	false	
$\neg$false	true	
false $\wedge\ \beta$	false	
$\alpha\ \wedge$ true	α	
false $\vee\ \beta$	β	
$\alpha\ \vee$ true	true	
$\neg\neg\alpha$	α	double negation
$\neg(\alpha \wedge \beta)$	$\neg\alpha \vee \neg\beta$	de Morgan
$\neg(\alpha \vee \beta)$	$\neg\alpha \wedge \neg\beta$	de Morgan
$\alpha \vee (\beta \wedge \gamma)$	$(\alpha \vee \beta) \wedge (\alpha \vee \gamma)$	distribution
$\alpha \wedge (\beta \vee \gamma)$	$(\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$	distribution
$\alpha \implies \beta$	$\neg\beta \implies \neg\alpha$	contraposition
$\alpha \implies \beta$	$\neg\alpha \vee \beta$	definition of $\implies$
$\alpha \iff \beta$	$(\alpha \implies \beta) \wedge (\beta \implies \alpha)$	definition of $\iff$

Reductions

Implication, equivalence, mutual exclusiveness, and exhaustiveness can all be defined in terms of satisfiability and validity.

Relationship	Property
α implies β	$\alpha \wedge \neg\beta$ is unsatisfiable
α implies β	$\alpha \implies \beta$ is valid
α and β are equivalent	$\alpha \iff \beta$ is valid
α and β are mutually exclusive	$\alpha \wedge \beta$ is unsatisfiable
α and β are exhaustive	$\alpha \vee \beta$ is valid

The Monotonicity of Logical Reasoning

- Someone communicates to us the following sentence

$$\alpha : \quad (\text{Earthquake} \vee \text{Burglary}) \implies \text{Alarm}.$$

- By accepting α, we are considering some worlds as impossible: those that do not satisfy the sentence α are ruled out.
- Our state of belief now characterized by the set of worlds:

$$Mods(\alpha) = \{\omega_1, \omega_3, \omega_5, \omega_7, \omega_8\}.$$

The Monotonicity of Logical Reasoning

world	Earthquake	Burglary	Alarm	*Possible?*
ω_1	true	true	true	yes
ω_2	true	true	false	no
ω_3	true	false	true	yes
ω_4	true	false	false	no
ω_5	false	true	true	yes
ω_6	false	true	false	no
ω_7	false	false	true	yes
ω_8	false	false	false	yes

Possible worlds according to the sentence
(Earthquake $\vee$ Burglary) $\implies$ Alarm.

The Monotonicity of Logical Reasoning

- Suppose now that we also learn

$$\beta: \quad \text{Earthquake} \implies \text{Burglary},$$

for which $Mods(\beta) = \{\omega_1, \omega_2, \omega_5, \omega_6, \omega_7, \omega_8\}$.

- Our state of belief is now characterized by the worlds:

$$Mods(\alpha \wedge \beta) = Mods(\alpha) \cap Mods(\beta) = \{\omega_1, \omega_5, \omega_7, \omega_8\}.$$

- Learning the new information β had the effect of ruling out world ω_3 in addition to those worlds ruled out by α.

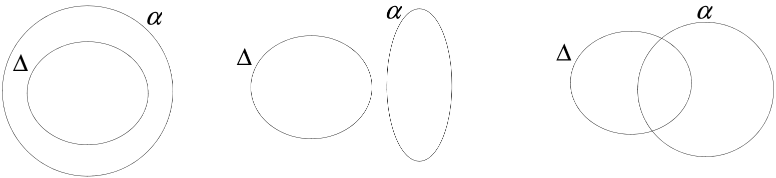
The Monotonicity of Logical Reasoning

- If α implies some sentence γ, then *Mods*$(\alpha) \subseteq$ *Mods*(γ) by definition of implication.
- Since *Mods*$(\alpha \wedge \beta) \subseteq$ *Mods*(α), we must also have *Mods*$(\alpha \wedge \beta) \subseteq$ *Mods*(γ).
- Hence, $\alpha \wedge \beta$ must also imply γ.

This is monotonicity

The belief in γ cannot be given up as a result of learning some new information β.

Logical Beliefs



(a) $\Delta \models \alpha$ (b) $\Delta \models \neg\alpha$ (c) $\Delta \not\models \alpha$ and $\Delta \not\models \neg\alpha$

(a) Δ implies α (α is believed);

(b) Δ implies the negation of α ($\neg\alpha$ is believed); or

(c) Δ neither implies α nor implies its negation.

Multi-Valued Variables: Syntax

One can generalize propositional logic to allow for multi-valued variables.

Consider an alarm that triggers either high or low, leading to three values for variable Alarm: low, high and off.

Example

$\text{Burglary} \implies \text{Alarm}=\text{high}$

Example

$\text{Burglary}=\text{true} \implies \text{Alarm}=\text{high}$

Multi-Valued Variables: Semantics

Worlds can be extended to multi–valued variables.

world	Earthquake	Burglary	Alarm
ω_1	true	true	high
ω_2	true	true	low
ω_3	true	true	off
ω_4	true	false	high
ω_5	true	false	low
ω_6	true	false	off
ω_7	false	true	high
ω_8	false	true	low
ω_9	false	true	off
ω_{10}	false	false	high
ω_{11}	false	false	low
ω_{12}	false	false	off

$\neg$Earthquake $\wedge$ $\neg$Burglary $\implies$ Alarm$=$off
is satisfied by worlds $\omega_1, \ldots, \omega_9, \omega_{12}$.

Variable Instantiations

Variable instantiation

A sentence of the form $(A = a) \wedge (B = b) \wedge (C = c)$, where a, b and c are values of variables A, B, C.

- Will use a, b, c instead of $(A = a) \wedge (B = b) \wedge (C = c)$.
- More generally, the conjoin operator ($\wedge$) may be replaced a comma (,) writing α, β instead of $\alpha \wedge \beta$.
- Trivial instantiation: an instantiation of an empty set of variables. Corresponds to a valid sentence and denoted by $\top$.

More Notations

Variables will be denoted by upper-case letters (A)

their values by lower-case letters (a); and
their cardinalities (number of values) by $|A|$.

More Notations

Variables will be denoted by upper-case letters (A)

their values by lower-case letters (a); and
their cardinalities (number of values) by $|A|$.

Sets of variables will be denoted by bold-face upper-case letters ($\mathbf{A}$)

their instantiations by bold-face lower-case letters ($\mathbf{a}$); and
their number of instantiations by $\mathbf{A}^{\#}$.

More Notations

Variables will be denoted by upper-case letters (A)

their values by lower-case letters (a); and
their cardinalities (number of values) by $|A|$.

Sets of variables will be denoted by bold-face upper-case letters ($\mathbf{A}$)

their instantiations by bold-face lower-case letters ($\mathbf{a}$); and
their number of instantiations by $\mathbf{A}^{\#}$.

$\mathbf{X}$ and $\mathbf{Y}$ are sets of variables; $\mathbf{x}$ and $\mathbf{y}$ are instantiations

Statements such as $\neg\mathbf{x}$, $\mathbf{x} \vee \mathbf{y}$ and $\mathbf{x} \implies \mathbf{y}$ are therefore legitimate sentences in propositional logic.

More Notations

Variables will be denoted by upper-case letters (A)

their values by lower-case letters (a); and
their cardinalities (number of values) by $|A|$.

Sets of variables will be denoted by bold-face upper-case letters ($\mathbf{A}$)

their instantiations by bold-face lower-case letters ($\mathbf{a}$); and
their number of instantiations by $\mathbf{A}^{\#}$.

$\mathbf{X}$ and $\mathbf{Y}$ are sets of variables; $\mathbf{x}$ and $\mathbf{y}$ are instantiations

Statements such as $\neg\mathbf{x}$, $\mathbf{x} \vee \mathbf{y}$ and $\mathbf{x} \implies \mathbf{y}$ are therefore legitimate sentences in propositional logic.

For a propositional variable A with values true and false

We may use a to denote A=true and $\bar{a}$ to denote A=false.

- A, A=true and a are all equivalent sentences.
- $\neg A$, A=false and $\bar{a}$ are all equivalent sentences.

More Notations

$\mathbf{x} \sim \mathbf{y}$

means that instantiations **x** and **y** are compatible; that is, they agree on the values of all their common variables.

Example

Instantiations $a, b, \bar{c}$ and $b, \bar{c}, \bar{d}$ are compatible.

Example

Instantiations $a, b, \bar{c}$ and $b, c, \bar{d}$ are not compatible as they disagree on the value of variable C.