Lógica Proposicional

Motivação

IA estuda como simular
comportamento inteligente

comportamento inteligente
é resultado de **raciocínio** correto
sobre **conhecimento** disponível

conhecimento e raciocínio correto
podem ser representados em **lógica**

o formalismo lógico mais
simples é a **lógica proposicional**

Lógica proposicional

- **É um formalismo composto por:**
  - **Linguagem formal**: usada para representar conhecimento.
  - **Métodos de inferência**: usados para representar raciocínio.
- **Tem como principal finalidade:**
  - **Representar argumentos**, isto é, seqüências de sentenças em que uma delas é uma conclusão e as demais são premissas.
  - **Validar argumentos**, isto é, verificar se sua conclusão é uma conseqüência lógica de suas premissas.

Exercício 1. Intuitivamente, qual dos dois argumentos a seguir é válido?

- ***Se neva, então faz frio. Está nevando. Logo, está fazendo frio.***
- ***Se chove, então a rua fica molhada. A rua está molhada. Logo, choveu.***

Elementos básicos

Proposição

é uma sentença **declarativa** que pode ser verdadeira ou falsa, mas não as duas coisas ao mesmo tempo.

Exercício 2

- **Quais das sentenças a seguir são proposições?**
  - *Abra a porta.*
  - *Excelente apresentação!*
  - *Esta semana tem oito dias.*
  - *Em que continente fica o Brasil?*
  - *A Lua é um satélite da Terra.*
- **Por que a sentença "*esta frase é falsa*" não é uma proposição?**

Elementos básicos

Conectivo

são partículas (**não, e, ou, então**) que permitem construir sentenças complexas a partir de outras mais simples.

Exemplo:

- A partir das sentenças (**proposições atômicas**):
  - Está chovendo
  - A rua está molhada
- Podemos construir as sentenças (**proposições compostas**):
  - Não está chovendo
  - Se está chovendo, então a rua está molhada

Linguagem Formal

Sintaxe: define a estrutura das sentenças

- **Símbolos**
  - Proposições: a, b, c, ...
  - Conectivos: $\neg$, $\wedge$, $\vee$, $\rightarrow$ (da maior para a menor precedência)
- **Fórmulas**
  - Todas as proposições são fórmulas.
  - Se α e β são fórmulas, então também são fórmulas:
    - $\neg\alpha$ (negação)
    - $\alpha\wedge\beta$ (conjunção)
    - $\alpha\vee\beta$ (disjunção)
    - $\alpha\rightarrow\beta$ (implicação)

Linguagem Formal

Semântica: define o significado das sentenças

- **Interpretação:** associação entre proposições e valores-verdade (V ou F)
  - Uma fórmula contendo *n* proposições admite 2^n interpretações distintas.
- **Tabela-verdade:** avalia uma fórmula em cada interpretação possível.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$
F	F	V	F	F	V
F	V	V	F	V	V
V	F	F	F	V	F
V	V	F	V	V	V

- **Tipos de fórmulas:**
  - **Válida** (tautológica): é verdadeira em **toda** interpretação.
  - **Satisfatível** (contingente): é verdadeira em **alguma** interpretação.
  - **Insatisfatível** (contraditória): é verdadeira em **nenhuma** interpretação.

Representação do conhecimento

Conhecimento pode ser representado de duas formas:

- **explícita**: por meio da formalização de sentenças
- **implícita**: por meio de conseqüência lógica (fatos derivados das sentenças)

Passos para formalização de sentenças

- Identificamos as palavras da sentença que correspondem a conectivos.
- Identificamos as partes da sentença que correspondem a proposições atômicas e associamos a cada uma delas um símbolo proposicional.
- Escrevemos a fórmula correspondente à sentença, substituindo suas proposições atômicas pelos respectivos símbolos proposicionais e seus conectivos lógicos pelos respectivos símbolos conectivos

Representação do conhecimento

Exemplo

- *Está chovendo.*
- **Se** *está chovendo,* **então** *a rua está molhada.*
- **Se** *a rua está molhada,* **então** *a rua está escorregadia.*

- **Vocabulário**
  - c : *"está chovendo"*
  - m : *"a rua está molhada"*
  - e : *"a rua está escorregadia"*
- **Formalização**
  - $\Delta = \{c, c \rightarrow m, m \rightarrow e\}$

base de conhecimento

Formalização de argumentos

Um **argumento** é uma seqüência de premissas seguida de uma conclusão

Exemplo

- *Se neva, então faz frio.*
- *Está nevando.*
- ***Logo**, está fazendo frio.*

- **Vocabulário**
  - **n** : "*neve*"
  - **f** : "*frio*"
- **Formalização**
  - $\{n \rightarrow f, n\} \models f$

consequência lógica

Formalização de argumentos

Um **argumento** é uma seqüência de premissas seguida de uma conclusão

Exemplo

- *Se neva, então faz frio.*
- *Está nevando.*
- ***Logo**, está fazendo frio.*

- **Vocabulário**
  - **n** : "*neve*"
  - **f** : "*frio*"
- **Formalização**
  - $\{n \rightarrow f, n\} \models f$

consequência lógica

Validação de Argumentos

Nem todo argumento é válido!

Exemplo: Intuitivamente, qual dos argumentos a seguir é válido?

- **Argumento 1**
  - *Se eu fosse artista, então eu seria famoso.*
  - *Não sou famoso.*
  - *Logo, não sou artista.*
- **Argumento 2**
  - *Se eu fosse artista, então eu seria famoso.*
  - *Sou famoso.*
  - *Logo, sou artista.*

Validação de argumentos

Um argumento é **válido** se a sua conclusão é uma **conseqüência lógica** de suas premissas, ou seja, a veracidade da conclusão está implícita na veracidade das premissas.

- Vamos mostrar três métodos de validação de argumentos:
  - **Tabela-verdade** (semântico)
  - **Prova por dedução** (sintático)
  - **Prova por refutação** (sintático)
- Métodos semânticos são baseados em interpretações
- Métodos sintáticos são baseados em regras de inferência (raciocínio)

Validação de argumentos usando tabela verdade

Um argumento da forma $\{\alpha_1, \ldots, \alpha_n\} \models \beta$ é válido se e somente se a fórmula correspondente $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ é válida (tautológica).

Exemplo

- **Argumento 1**
  - *Se eu fosse artista, seria famoso.*
  - *Não sou famoso.*
  - *Logo, não sou artista.*
- **Vocabulário**
  - **a** : *"artista"*
  - **f** : *"famoso"*
- **Formalização**
  - $\{a \rightarrow f, \neg f\} \models \neg a$

a	f	(a	→	f)	∧	¬	f	→	¬	a
F	F	F	V	F	V	V	F	V	V	F
F	V	F	V	V	F	F	V	V	V	F
V	F	V	F	F	F	V	F	V	F	V
V	V	V	V	V	F	F	V	V	F	V

O argumento é válido!

Validação de argumentos usando tabela verdade

Exemplo

- **Argumento 2**
  - *Se eu fosse artista, seria famoso.*
  - *Sou famoso.*
  - *Logo, sou artista.*
- **Vocabulário**
  - a : *"artista"*
  - f : *"famoso"*
- **Formalização**
  - $\{a \rightarrow f, f\} \vDash a$

a	f	(a	→	f)	∧	f	→	a
F	F	F	V	F	F	F	V	F
F	V	F	V	V	V	V	F	F
V	F	V	F	F	F	F	V	V
V	V	V	V	V	V	V	V	V

O argumento NÃO é válido!

Validação de argumentos usando tabela verdade

- Conseqüência lógica é o elo entre o que um agente "acredita" e aquilo que é explicitamente representado em sua base de conhecimento.
- A tabela-verdade é um método semântico que permite verificar conseqüências lógicas.
- Este método tem a vantagem de ser conceitualmente simples; mas, como o número de linhas na tabela-verdade cresce exponencialmente em função do número de proposições na fórmula, seu uso nem sempre é viável.
- Assim, apresentaremos o **raciocínio automatizado** como uma alternativa mais eficiente para verificação de conseqüência lógica (isto é, validação de argumentos).

Validação de argumentos usando Dedução

Uma **prova por dedução** de uma fórmula φ, a partir de uma base de conhecimento Δ, é uma seqüência finita de fórmulas $\gamma_1, \ldots, \gamma_k$ tal que:

- $\gamma_k = \varphi$;
- para $1 \leq i \leq k$, ou $\gamma_i \in \Delta$ ou, então, γ_i é **derivada** de fórmulas em $\Delta \cup \{\gamma_1, \ldots, \gamma_{i-1}\}$, pela aplicação de uma **regra de inferência**.

Regra de inferência:

é um padrão de manipulação sintática que define como uma fórmula (*conclusão*) pode ser derivada de outras fórmulas (*premissas*)

Validação de argumentos usando Dedução

Regras de inferência clássicas:

- **Modus ponens (MP):** $\{\alpha \rightarrow \beta,\ \alpha\} \vdash \beta$
- **Modus tollens (MT):** $\{\alpha \rightarrow \beta,\ \neg\beta\} \vdash \neg\alpha$
- **Silogismo hipotético (SH):** $\{\alpha \rightarrow \beta,\ \beta \rightarrow \gamma\} \vdash \alpha \rightarrow \gamma$

As regras de inferência clássicas:

- representam "esquemas de raciocínio" válidos
- podemos validar estes esquemas usando tabela-verdade
- podem ser usadas para derivar conclusões que são conseqüências lógicas de suas premissas

Validação de argumentos usando Dedução

Exemplo: validar o argumento $\{j \to g,\ \neg j \to t,\ g \to c,\ \neg c\} \models t$

---

(1)	$j \to g$	Δ
(2)	$\neg j \to t$	Δ
(3)	$g \to c$	Δ
(4)	$\neg c$	Δ
-------------		
(5)	$j \to c$	SH(1,3)
(6)	$\neg j$	MT(5,4)
(7)	t	MP(2,6)

MP: $\{\alpha \to \beta,\ \alpha\} \vdash \beta$

MT: $\{\alpha \to \beta,\ \neg \beta\} \vdash \neg \alpha$

SH: $\{\alpha \to \beta,\ \beta \to \gamma\} \vdash \alpha \to \gamma$

Conclusão: o argumento é válido, pois a fórmula t pode ser derivada de Δ.

Validação de argumentos usando Refutação

Embora a prova por dedução seja um método mais prático que a tabela-verdade, ainda é muito difícil obter algoritmos eficientes para validação de argumentos com base neste método.

Refutação

- **Refutação** é um processo em que se demonstra que uma determinada hipótese contradiz uma base de conhecimento.
- Uma base de conhecimento $\Delta = \{\alpha_1, \ldots, \alpha_n\}$ é consistente se a fórmula correspondente $\alpha_1 \wedge \ldots \wedge \alpha_n$ é satisfatível.
- Se $\Delta = \{\alpha_1, \ldots, \alpha_n\}$ é consistente, provar $\Delta \vDash \gamma$ equivale a mostrar que o conjunto de fórmulas $\{\alpha_1, \ldots, \alpha_n, \neg\gamma\}$ é inconsistente.

Validação de argumentos usando Refutação

- **Argumento**
  (1) Se o time joga bem, então ganha o campeonato.
  (2) Se o time não joga bem, então o técnico é culpado.
  (3) Se o time ganha o campeonato, então os torcedores ficam contentes.
  (4) Os torcedores não estão contentes.
  (5) Logo, o técnico é culpado.

- **Refutação**

(a) O técnico **não** é culpado	**hipótese**
(b) O time joga bem	**MT(a,2)**
(c) O time ganha o campeonato	**MP(b,1)**
(d) Os torcedores ficam contentes	**MP(c,3)**
(e) Contradição!	**Confrontando (d) e (4)**

- **Conclusão:** a hipótese contradiz as premissas, logo o argumento é válido!

Validação de argumentos usando Refutação

Exemplo: validar o argumento $\{j \to g,\ \neg j \to t,\ g \to c,\ \neg c\} \vDash t$

---

(1) $j \to g$ Δ
(2) $\neg j \to t$ Δ
(3) $g \to c$ Δ
(4) $\neg c$ Δ
\-------------
(5) $\neg t$ **Hipótese**
(6) j MT(5,2)
(7) g MP(6,1)
(8) c MP(7,3)
(9) $\square$ **Contradição!**

Conclusão: como $\Delta \cup \{\neg t\}$ é inconsistente, segue que $\Delta \vDash t$.

Forma Normal Conjuntiva

Para simplificar a automatização do processo de refutação, vamos usar **fórmulas normais** (Forma Normal Conjuntiva - FNC).

Passos para conversão para FNC

- **Elimine a implicação:**

$$\alpha \rightarrow \beta \equiv \neg \alpha \vee \beta$$

- **Reduza o escopo da negação:**

$$\neg(\alpha \wedge \beta) \equiv \neg \alpha \vee \neg \beta$$

$$\neg(\alpha \vee \beta) \equiv \neg \alpha \wedge \neg \beta$$

- **Reduza o escopo da disjunção:**

$$\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$

Forma Normal Conjuntiva

Exemplo de conversão para FNC

$p \vee q \rightarrow r \wedge s$

$\equiv \neg (p \vee q) \vee (r \wedge s)$

$\equiv (\neg p \wedge \neg q) \vee (r \wedge s)$

$\equiv ((\neg p \wedge \neg q) \vee r) \wedge ((\neg p \wedge \neg q) \vee s)$

$\equiv (\neg p \vee r) \wedge (\neg q \vee r) \wedge (\neg p \vee s) \wedge (\neg q \vee s)$

FNC

Fórmulas normais: $\{\neg p \vee r, \neg q \vee r, \neg p \vee s, \neg q \vee s\}$

Inferência por resolução

- FNC permite usar inferência por resolução
- A idéia da resolução é:
  - RES($\alpha \vee \beta, \neg \beta \vee \gamma$) = $\alpha \vee \gamma$
  - RES($\alpha, \neg \alpha$) = $\square$

Equivalência entre resolução e regras de inferência clássicas

MP($\alpha \rightarrow \beta, \alpha$) = β	RES($\neg \alpha \vee \beta, \alpha$) = β
MT($\alpha \rightarrow \beta, \neg \beta$) = $\neg \alpha$	RES($\neg \alpha \vee \beta, \neg \beta$) = $\neg \alpha$
SH($\alpha \rightarrow \beta, \beta \rightarrow \gamma$) = $\alpha \rightarrow \gamma$	RES($\neg \alpha \vee \beta, \neg \beta \vee \gamma$) = $\neg \alpha \vee \gamma$

Inferência por resolução

Exemplo: validar o argumento $\{j \rightarrow g,\ \neg j \rightarrow t,\ g \rightarrow c,\ \neg c\} \models t$

(1) $\neg j \vee g$ Δ
(2) $j \vee t$ Δ
(3) $\neg g \vee c$ Δ
(4) $\neg c$ Δ
\------------
(5) $\neg t$ **Hipótese**
(6) j RES(5,2)
(7) g RES(6,1)
(8) c RES(7,3)
(9) $\square$ RES(8,4)

Este é o mecanismo de raciocínio implementado pelo Prolog!

Conclusão: como $\Delta \cup \{\neg t\}$ é inconsistente, segue que $\Delta \models t$.

FIM