Universidade do Sul de Santa Catarina - UNISUL
Ciência da Computação

Técnicas de Inteligência Artificial

Aula 05
Lógica Proposicional
Lógica de Predicados

Prof. Max Pereira

logica

Proposicional

$$\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$$

A lógica está relacionada com **raciocínio** e com a **validade** de argumentos. Normalmente estamos preocupados com a validade das sentenças, não com sua **veracidade**.

Isto quer dizer que, apesar do seguinte argumento ser claramente lógico, ele não é algo que consideraríamos como verdadeiro:

Todos os limões são azuis.
Maria é um limão
Então, Maria é azul.

Esse conjunto de sentenças é considerado como **válido**, pois a conclusão (Maria é azul) segue **logicamente** as outras duas sentenças, que chamamos de **premissas**.

A razão pela qual validade e veracidade podem ser separadas desse modo é simples: um raciocínio é considerado como válido se sua conclusão for verdadeira, nos casos em que suas premissas também sejam verdadeiras.

Podemos dizer que: *um raciocínio é válido se ele conduzir a uma conclusão verdadeira em todas as situações nas quais as premissas sejam verdadeiras*.

Lógica está envolvida com **valores-verdade**. Os possíveis valores-verdade são <u>verdadeiro</u> ou <u>falso</u>. Estes podem ser considerados as **unidades fundamentais** de lógica e quase toda lógica está, em última análise, envolvida com esses valores-verdade.

Lógica e Inteligência Artificial

Lógica é amplamente utilizada em computação e, em especial, em Inteligência Artificial. Lógica é utilizada como um **método representacional**, permitindo raciocinar facilmente sobre <u>negativas</u> (como "este livro não é vermelho") e <u>disjunções</u> ("*ou*" – como "ele é engenheiro ou cientista").

Sentenças lógicas devem ser expressas em termos de verdade ou falsidade – não é possível raciocinar, em lógica clássica, sobre **possibilidades**.

Operadores Lógicos

Ao raciocinar sobre valores-verdade, precisamos utilizar diversos operadores, que podem ser aplicados a valores-verdade. Estamos familiarizados com vários desses operadores.

Gosto de maçãs ***e*** laranjas.
Você pode comer um bolo ***ou*** sorvete.
Se você é brasileiro, ***então*** fala português,
Não sou militar.

Nas sentenças anteriores estão os operadores mais básicos utilizados no nosso dia-a-dia. Os operadores são:

- e
- ou
- não
- se ... então ... (geralmente chamado de **implicação**)

Os operadores são escritos utilizando símbolos:

e	$\wedge$
ou	$\vee$
não	$\neg$
implicação (se...então...)	$\rightarrow$
equivalência (se e somente se)	$\leftrightarrow$

"*se e somente se*" é uma forma mais forte de implicação que é verdadeira se uma coisa implica a outra e também se a segunda coisa implica a primeira (equivalência).

Tradução entre Português e Notação Lógica

Para usar lógica, é necessário primeiro converter **fatos** e **regras** sobre o mundo real em **expressões lógicas**, utilizando os operadores lógicos.

Vamos considerar os operadores simples $\wedge$, $\vee$ e $\neg$.

Sentenças que usam a palavra "*e*" em Português, para expressar mais de um conceito, todos sendo simultaneamente verdadeiros, podem ser facilmente traduzidas para lógica utilizando o operador "E" ($\wedge$).

"Está chovendo e é sábado".

Poderia ser expressa como: $C \wedge S$

Em que *C* quer dizer "está chovendo" e *S* quer dizer "é sábado"

Tradução entre Português e Notação Lógica

Se não for necessário expressar onde está chovendo, a variável *C* será suficiente. Se precisarmos escrever sentenças como "*está chovendo em Orleans*" ou "*está chovendo muito*" ou mesmo "*choveu por 30 minutos no sábado*", então a variável *C* não será suficiente.

Podemos expressar conceitos mais complexos, geralmente utilizando **predicados**.

Podemos traduzir por exemplo:

"Está chovendo em Orleans".

$O(C)$ ou $C(O)$

Utilizar uma ou outra forma, depende de considerarmos a chuva como sendo uma **propriedade** de Orleans ou vice-versa.

Tradução entre Português e Notação Lógica

Em outras palavras, quando escrevemos O(C), estamos dizendo que uma propriedade da chuva é que ela ocorre em Orleans, enquanto com C(O) estamos dizendo que uma propriedade de Orleans é que está chovendo.

Qual será utilizada **dependerá do problema** que estamos solucionando. Se estivéssemos solucionando um problema sobre Orleans, utilizaríamos C(O), enquanto se estivéssemos solucionando um problema sobre a localização de vários tipos de clima, deveríamos utilizar O(C).

Tradução entre Português e Notação Lógica

Voltando aos operadores lógicos...

A expressão "*está chovendo em Orleans e estou ficando doente ou estou cansado*" pode ser expressa como:

$C(O) \wedge (D(E) \vee C(E))$

Aqui utilizamos os operadores $\wedge$ e $\vee$, para expressar um conjunto de sentenças. A sentença pode ser dividida em duas seções, o que é indicado pelo uso de parênteses. A seção entre parênteses é $D(E) \vee C(E)$, que quer dizer "estou ficando doente **OU** estou cansado". Esta expressão é conectada com a parte de fora dos parênteses, que é C(O).

É importante mencionar que as variáveis (letras) utilizadas podem ser substituídas por palavras.

Chovendo(**O**rleans) $\wedge$ (**D**oente(**E**u) $\vee$ **C**ansado(**E**u))

Tradução entre Português e Notação Lógica

O operador $\neg$ é aplicado para expressar **negação**. Por exemplo,

"*Não está chovendo em Orleans*", poderia ser expresso como:

$\neg C(O)$ ou $\neg Chovendo(Orleans)$

É importante colocar o símbolo "$\neg$" no lugar correto. Por exemplo: "*não estou bem ou estou cansado*" pode ser traduzida como

$\neg B(E) \vee C(E)$

Aqui o símbolo "$\neg$" indica que ele está ligado a B(E) e não tem qualquer influência sobre C(E).

Tradução entre Português e Notação Lógica

Vamos verificar como o operador "$\rightarrow$" é utilizado. Frequentemente quando lidamos com lógica, estamos discutindo **regras**, que expressam conceitos com "*se estiver chovendo então ficarei molhado*".

Essa sentença seria traduzida para lógica como:

$C \rightarrow M(E)$

Isto é lido "C *implica* M(E)" ou "*Se* C *Então* M(E)".

É essencial compreender que a escolha de variáveis e de predicados é importante, mas que podemos escolher quaisquer variáveis e predicados que se ajustem melhor ao problema e que ajudem a solucioná-lo.

Tradução entre Português e Notação Lógica

"*Não está chovendo em Florianópolis*"

$\neg$C(F) ou $\neg$chuva(Florianópolis)

"*Não estou bem ou estou muito cansado*"

$\neg B(E) \vee C(E)$

"*Se o relógio está parado e hoje é segunda-feira, então estou atrasado*".

$P(R) \wedge S(H) \rightarrow A(E)$

Tabelas-Verdade

É comum representar o **comportamento** dos operadores lógicos utilizando tabelas-verdade. Uma tabela-verdade mostra os possíveis valores que podem ser gerados pela aplicação de um operador aos valores-verdade.

TABELA-VERDADE DAS PROPOSIÇÕES COMPOSTAS BÁSICAS

p	q	$p \wedge q$	$p \vee q$	$p \veebar q$	$p \rightarrow q$	$p \leftrightarrow q$
V	V	V	V	F	V	V
V	F	F	V	V	F	F
F	V	F	V	V	V	F
F	F	F	F	F	V	V

O símbolo $\veebar$ identifica o operador lógico *ou-exclusivo* (XOR).

Tabelas-Verdade

As tabelas-verdade não estão limitadas a mostrar valores para operadores únicos. Por exemplo, uma tabela-verdade pode ser usada para apresentar os possíveis valores para $A \wedge (B \vee C)$.

A	B	C	$A \wedge (B \vee C)$
V	V	V	V
V	V	F	V
V	F	V	V
V	F	F	F
F	V	V	F
F	V	F	F
F	F	V	F
F	F	F	F

O uso de parênteses nessa expressão é importante. $A \wedge (B \vee C)$ não é o mesmo que $(A \wedge B) \vee C$.

Tabelas-Verdade

Para evitar **ambiguidade**, os operadores lógicos respeitam uma **ordem de precedência**, assim como os operadores matemáticos. A ordem de precedência utilizada é:

$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

Portanto, em uma afirmação como

$\neg A \vee \neg B \wedge C$

O operador $\neg$ tem a maior precedência, ou seja, a sentença pode ser expressa como:

$(\neg A) \vee ((\neg B) \wedge C)$

Da mesma forma, quando escrevemos
$\neg A \vee B$
Isto é o mesmo que $(\neg A) \vee B$
em vez de $\neg (A \vee B)$

Tautologia

Considere a seguinte tabela-verdade:

A	$A \lor \neg A$
F	V
V	V

O valor da expressão $A \lor \neg A$ é *verdadeiro* independente do valor de A. Uma expressão como essa, que é sempre verdadeira, é chamada de **tautologia**.

Se A for uma tautologia, escrevemos:

$\vDash A$

Uma expressão lógica que seja uma tautologia é frequentemente descrita como **válida**. Uma expressão válida é definida como sendo aquela que é verdadeira sob qualquer **interpretação**.

Tautologia

Se a expressão for *falsa* em qualquer interpretação, ela é descrita como sendo **contraditória**. As seguintes expressões são contraditórias:

$A \wedge \neg A$
$(A \vee \neg A) \rightarrow (A \wedge \neg A)$

Não importa o que A signifique nestas expressões, o resultado não pode ser *verdadeiro*.

Tautologia

Algumas expressões são **satisfatórias**, porém não são <u>válidas</u>. Isso significa que elas são *verdadeiras* **sob alguma interpretação**, mas não sob todas as interpretações. As seguintes expressões são satisfatórias:

$A \vee B$

$(A \wedge B \vee \neg C) \rightarrow (D \wedge E)$

Uma expressão contraditória é claramente não satisfatória, e, então, é descrita como sendo **não satisfatória**.

Equivalência

Considere as duas expressões:

$A \wedge B$
$B \wedge A$

É bastante claro que estas duas expressões terão sempre o mesmo valor para um dado par de valores para A e B. Em outras palavras, dizemos que a primeira expressão é **logicamente equivalente** à segunda expressão. Escrevemos isso como:

$A \wedge B \equiv B \wedge A$ (isso significa que o operador $\wedge$ é **comutativo**).

Equivalências lógicas

$A \vee A \equiv A$

$A \wedge A \equiv A$

$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$

$A \vee (A \wedge B) \equiv A$

$A \rightarrow B \equiv \neg A \vee B$

$A \vee B \equiv \neg(\neg A \wedge \neg B)$

$A \wedge B \equiv \neg(\neg A \vee \neg B)$

Todas essas equivalências podem ser provadas elaborando-se as tabelas-verdade para cada lado da equivalência e verificando se as duas tabelas são idênticas.

Equivalência

A seguinte é uma equivalência muito importante:

$A \to B \equiv \neg A \vee B$

Pode-se verificar isso examinando as tabelas-verdade. O motivo disto ser útil é por significar que não precisamos usar o símbolo $\to$, já que podemos substituí-lo por uma combinação de $\neg$ e $\vee$. Da mesma forma, as seguintes equivalências mostram que não precisamos usar $\wedge$ ou $\leftrightarrow$.

$A \wedge B \equiv \neg(\neg A \vee \neg B)$
$A \leftrightarrow B \equiv \neg(\neg(\neg A \vee B) \vee \neg(\neg B \vee A))$

De fato, qualquer operador lógico binário (que atua sobre duas variáveis) pode ser expresso utilizando $\neg$ e $\vee$.

Equivalência

Finalmente, as seguintes equivalências são conhecidas como **Leis de DeMorgan**.

$A \wedge B \equiv \neg(\neg A \vee \neg B)$
$A \vee B \equiv \neg(\neg A \wedge \neg B)$

Utilizando estas e outras equivalências, expressões lógicas podem ser **simplificadas**. Por exemplo,

$(C \wedge D) \vee ((C \wedge D) \wedge E)$

Pode ser simplificada utilizando a seguinte regra:

$A \vee (A \wedge B) \equiv A$

Portanto,
$(C \wedge D) \vee ((C \wedge D) \wedge E) \equiv C \wedge D$

Dessa forma, é possível eliminar expressões que não contribuam para o valor total da expressão.

Qual das proposições representa a negação da sentença:

“Júlia gosta de manteiga mas detesta creme”?

a. Júlia detesta manteiga e creme.

b. Júlia não gosta de manteiga nem de creme.

c. Júlia não gosta de manteiga mas adora creme.

d. Júlia odeia manteiga ou gosta de creme.

Vamos escrever a sentença usando a notação lógica, para isso, faremos M = “gosta manteiga” e C = “detesta creme”. Assim podemos escrever que Júlia $M \wedge \neg C$. Para negar a sentença temos $\neg(M \wedge \neg C)$.

“Júlia gosta de manteiga mas detesta creme”?

a. Júlia detesta manteiga e creme.

b. Júlia não gosta de manteiga nem de creme.

c. Júlia não gosta de manteiga mas adora creme.

d. Júlia odeia manteiga ou gosta de creme.

Agora podemos aplicar a Lei de **DeMorgan**:

$\neg(M \wedge \neg C) \equiv \neg M \vee C$

Ou seja, algo como “*não gosta de manteiga ou gosta de creme*”. A opção “**d**” é a mais adequada.

Lógica Proposicional

Existem vários sistemas de lógica. O que estamos analisando é chamado de **lógica proposicional**. Um sistema lógico pode ser definido em termos de sua **sintaxe** (o alfabeto de símbolos e como eles podem ser combinados), sua **semântica** (o que o símbolo significa) e um conjunto de **regras de dedução** que nos permite derivar uma expressão a partir de um conjunto de outras expressões e, desse modo, fazer <u>argumentações e provas</u>.

Dedução

Se tivermos um conjunto de premissas $\{A_1, A_2, ..., A_n\}$ e, a partir dessas premissas, formos capazes de derivar uma conclusão, C, então dizemos que **deduzimos** C a partir das premissas, o que é escrito como:

$\{A_1, A_2, ..., A_n\} \vdash C$

Se C pode se concluído sem qualquer premissa, então escrevemos

$\vdash C$

Para derivar uma conclusão a partir de um conjunto de premissas, aplicamos um conjunto de **regras de inferência**. Para distinguir uma regra de inferências de uma sentença, frequentemente escrevemos $A \vdash B$ como se segue:

$$\frac{A}{B}$$

Regras de Inferência

$\wedge$-Introdução

$$\frac{A \qquad B}{A \wedge B}$$

Esta regra é bem direta. Ela diz: dados A e B podemos deduzir A $\wedge$ B.

$\wedge$-Eliminação

$$\frac{A \wedge B}{A}$$

$$\frac{A \wedge B}{B}$$

Essas regras dizem que dado A $\wedge$ B podemos deduzir A e também podemos deduzir B separadamente.

Regras de Inferência

Ou-Introdução

$$\frac{A}{A \wedge B}$$

$$\frac{B}{A \wedge B}$$

Essas regras dizem que a partir de A podemos deduzir a **disjunção** (operador "*ou*") de A com qualquer expressão. Por exemplo, a partir da sentença "*gosto de lógica*" podemos deduzir expressões como:

"*gosto de lógica ou gosto de literatura*", "*gosto de lógica ou não gosto de lógica*", "*gosto de lógica ou peixes podem cantar*", "*gosto de lógica ou 2 + 2 = 123*" e assim por diante. Isto acontece porque "*verdadeiro* $\vee$ B" é **verdadeiro** para qualquer valor de B.

Regras de Inferência

$\rightarrow$Eliminação

Essa regra é conhecida como ***modus ponens*** e é uma das regras mais comumente utilizadas em dedução lógica. Ela é expressa como

$$\frac{A \quad A \rightarrow B}{B}$$

Em outras palavras, se A for verdadeiro e A implica B for verdadeiro, então sabemos que B será verdadeiro. Por exemplo, se substituirmos A por "*está chovendo*" e B por "*preciso de um guarda-chuva*", então produziremos o seguinte:

Está chovendo. Se estiver chovendo, precisarei de um guarda-chuva. Logo, preciso de um guarda-chuva.

Este tipo de raciocínio é claramente válido.

Regras de Inferência

Reductio Ad Absurdum

Precisamos de uma nova notação para esta regra:

$$\frac{\begin{array}{c}\neg A \\ \vdots \\ \bot\end{array}}{A}$$

O símbolo $\bot$ é chamado de ***falsum*** e é utilizado para indicar um **absurdo** ou uma **contradição**. Por exemplo, $\bot$ pode ser deduzido a partir de $A \wedge \neg A$.

A regra *reductio ad absurdum* simplesmente diz que, se assumirmos que A seja falso ($\neg$ A) e isto levar a uma contradição ($\bot$), então poderemos deduzir que A é *verdadeiro*. Isto é conhecido como prova por contradição.

Regras de Inferência

→Introdução

$$\frac{\begin{array}{c} A \\ . \\ : \\ C \end{array}}{A \rightarrow C}$$

Esta regra mostra que, ao conduzir uma prova, se começarmos a partir de uma premissa A e derivarmos uma conclusão C, então poderemos concluir que A → C.

<u>Regras de Inferência</u>

¬ ¬Eliminação

$$\frac{\neg\neg A}{A}$$

Esta regra estabelece que se, tivermos uma sentença negada duas vezes, poderemos concluir uma sentença propriamente dita, sem a negação.

Lógica dos Predicados

A lógica dos predicados nos permite **raciocinar sobre propriedades** de objetos e **relacionamento** entre objetos.

"Maria gosta de laranjas"
gosta(Maria, laranjas)

"Lucas mora em Curitiba"
mora(Lucas, Curitiba)

"Gabriela é irmã de Arthur"
irmã(Gabriela, Arthur)

Relação(x, y)

Quantificadores

$\forall$ - para todo ou para todos
$\exists$ - existe um ou existe pelo menos um

Considere a seguinte sentença:

Para todo *x*, $x > 0$
Notação lógica com a utilização do quantificador: $(\forall x)(x > 0)$

$(x > 0)$ é a **propriedade** de *x*, aquilo que se afirma sobre o objeto *x*, ou seja, $P(x)$.

$(\forall x)P(x)$
$P(x) = x > 0$

Atribuindo valores-verdade as sentenças com quantificadores

$(\forall x)P(x)$ = V ou F?

$P(x) = x > 0$

Não podemos definir o valor-verdade da sentença sem conhecermos o seu **domínio** (conjunto universo).

$(\forall x)P(x)$ = V

$P(x) = x > 0$

Domínio: inteiros positivos

Exemplos:

Qual o valor lógico da expressão $(\forall x)P(x)$?

a. $P(x)$ é a propriedade que x é amarelo e o conjunto universo é o conjunto de todas as flores.

b. $P(x)$ é a propriedade que x é uma planta e o conjunto universo é o conjunto de todas as flores.

E para $(\exists x)P(x)$?

É possível encontrar uma interpretação na qual, ao mesmo tempo, $(\forall x)P(x)$ seja verdadeiro e $(\exists x)P(x)$ seja falso?

É possível encontrar uma interpretação na qual, ao mesmo tempo, $(\forall x)P(x)$ seja falso e $(\exists x)P(x)$ seja verdadeiro?

Os predicados podem ser binários, envolvendo propriedades de duas variáveis.

$(\forall x)(\exists y)P(x, y) = ?$

$P(x, y) = x < y$

Domínio = inteiros

$(\exists y)(\forall x)P(x, y) = ?$

Exemplos:

Domínio = inteiros

$(\exists y)(\forall x)(x + y = x)$?

$(\forall x)(\forall y)(x < y \vee y < x)$?

$(\exists x)(\exists y)(x^2 = y)$?

Representando conhecimento:

Calabar foi enforcado = enforcado(Calabar)

Getúlio foi presidente = presidente(Getúlio)

Todo traidor é enforcado = $(\forall x)\text{traidor}(x) \rightarrow \text{enforcado}(x)$

Todos os índios eram selvagens = $(\forall x)\text{índio}(x) \rightarrow \text{selvagem}(x)$

Tiradentes não era índio = $\neg\text{índio}(\text{Tiradentes})$

Tiradentes foi considerado traidor = traidor(Tiradentes)

Exemplos:

Expressar a idéia de que todos gostam de cerveja.

$(\forall x)Pessoa(x) \rightarrow Gosta(x,cerveja)$

Expressar a idéia de que nem todos gostam de cerveja.

$\neg(\forall x)Pessoa(x) \rightarrow Gosta(x,cerveja)$

Expressar a idéia de que existe uma pessoa que não gosta de cerveja

$(\exists x)Pessoa(x) \rightarrow \neg Gosta(x,cerveja)$

Considerar os seguintes predicados:

G(x) = x é um gato

R(x) = x é um rato

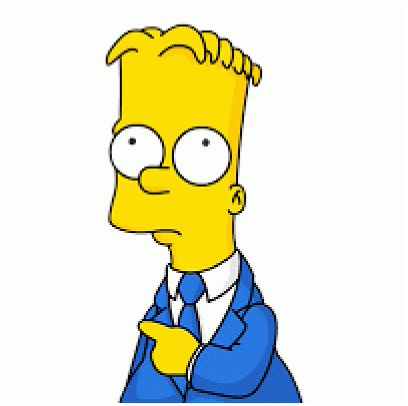
P(x,y) = x caça y

a. Todos os gatos caçam todos os ratos.

$(\forall x)G(x) \land (\forall y)R(y) \rightarrow P(x,y)$

b. Alguns gatos caçam todos os ratos.

$(\exists x)G(x) \land (\forall y)R(y) \rightarrow P(x,y)$



b. Apenas gatos caçam ratos.

$(\forall x)(\forall y)R(y) \land P(x,y) \rightarrow G(x)$

Qual o valor-verdade da sentença?

$$(\forall x)(\forall y)(\forall z)[pai(x,y) \wedge pai(y,z) \rightarrow neto(z,x)]$$

Praticando....

Dados os valores de A=V, B=F e C=V, qual o valor-verdade de cada uma das sentenças?

a. $A \wedge (B \wedge C)$

b. $(A \wedge B) \vee C)$

c. $\neg(A \wedge B) \vee C$

d. $\neg A \vee \neg(\neg B \wedge C)$

Praticando....

Construa as tabelas-verdade:

a. $(A \rightarrow B) \leftrightarrow \neg A \vee B$

b. $A \wedge B \rightarrow \neg A$

c. $A \rightarrow (B \rightarrow A)$

Praticando....

D(x) = x é dia, S(x) = x está fazendo sol, C(x) = x está chovendo, M(x) = é segunda-feira, T(x) = é terça-feira.

a. Todos os dias faz sol.

b. Alguns dias não está chovendo.

c. Todo dia que não está fazendo sol, está chovendo.

d. Alguns dias faz sol e chove.

e. Segunda-feira fez sol.

f. É um dia de sol apenas se não estiver chovendo.

g. Choveu na segunda e na terça-feira.

h. Nenhum dia fez sol.