Lógica proposicional

O que é a lógica? Adotaremos a definição de Irving Copi: *Estudo dos métodos e princípios para dinstinguir o raciocínio correto do incorreto.* Isso é bem diferente de uma "ciência das leis do pensamento". O pensamento é um processo mental que não inclui somente a distinção correto/incorreto. Tampouco, a lógica não é a "ciência do raciocínio". Existem raciocínios que usam, por exemplo, intuições ou analogias, e isso não é no domínio da lógica.

Além de esclarecer parte do raciocínio humano, a lógica também permite várias aplicações em ciência da computação. No esforço atual para desenvolver programas ditos "inteligentes", capaz de resolver problemas complexos, a lógica ocupa um espaço importante. Ela oferece uma maneira clara e elegante de representar a capacidade de inferência que esses programas precisam.

Nesse curso, tentaremos explicar o que é a lógica simbólica clássica, descrevendo primeiro uma forma simple, a lógica proposicional, e apresentando depois a lógica de primeira ordem, um pouco mais complexa, mas que oferece um poder expressivo bem maior. Teremos sempre a preocupação de ver como essas lógicas podem ser implementadas no computador.

1 Sintaxe

Suponhamos a existência de uma lista infinita, P_1, P_2, ..., de *símbolos proposicionais.* Intuitivamente, esses símbolos, que por conveniência representaremos por letras maiúsculas P, Q, R, ..., representam uma *proposição*, uma coisa dita sobre o mundo, que pode ser verdadeira ou falsa. A única condição sobre a lista de símbolos poposicionais é que os elementos que ela contém sejam distintos.

Uma *fórmula* é construída a partir dos símbolos proposicionais utilizando conectores. Essencialmente, os conectores são os seguintes:

Conector	Tipo	Nome	Exemplo
$\neg$	unário	Negação	$\neg P$
$\vee$	binário	Disjunção	$P \vee Q$
$\wedge$	binário	Conjunção	$P \wedge Q$
$\supset$	binário	Implicação	$P \supset Q$

É possível definir mais conectores, mas todos podem ser definidos em função dos conectores que são enumerados nessa tabela. Alias, é possível ser mais econômico ainda, pois dois conectores serião suficentes.

Além dos símbolos proposicionais e conectores, a lógica proposicional contém duas constantes $\top$ e $\bot$, que representam, respectivamente, os valores **verdade** e **falso**.

Finalmente, precisaremos de parênteses para escrever fórmulas complexas que contêm mais de um conector.

Fórmulas da lógica proposicional:

> Uma *fórmula atómica* é um símbolo proposicional ou uma das duas constantes $\top$ e $\bot$.
>
> O conjunto de *fórmulas proposicionais* é o menor conjunto $\mathbf{P}$ tal que:
>
> 1. Se A é uma fórmula atómica, $A \in \mathbf{P}$;
> 2. Se $X \in \mathbf{P}$, $\neg X \in \mathbf{P}$;
> 3. Seja $\circ$ um conector binário. Se $X, Y \in \mathbf{P}$, $(X \circ Y) \in \mathbf{P}$.

Eis exemplos de fórmulas da lógica proposicional: $\top$, $(P \vee (Q \wedge \neg R))$, $(P \supset \bot)$, $\neg\neg\neg\neg(P \supset \neg R)$. Para simplificar, omitiremos parênteses em casos onde a ausência delas não causa confusão. Eis exemplos de fórmulas que não fazem parte da linguagem da lógica proposicional (por que?): $(P \neg \vee Q)$, $(P \vee Q \supset R)$, $((P \vee Q))$, $(\neg(\neg P))$.
Vamos agora definir as noções de *subfórmula imediata* e *subfórmula*.

Subfórmula imediata:

- Os símbolos proposicionais, assim como os símbolos $\bot$ e $\top$, não têm subfórmula imediata.
- A única subfórmula imediata de $\neg X$ é X.

- Seja $\circ$ um conector binário. A fórmula $(X \circ Y)$ tem exatamente duas subfórmulas: X e Y.

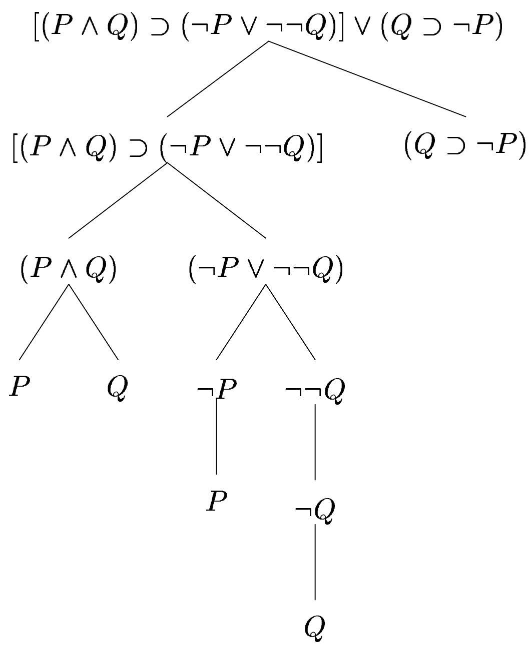
Subfórmula:

- Se X é uma subfórmula imediata de Y, ou se X é idêntico a Y, X é uma subfórmula de Y.
- Se X é uma subfórmula de Y e Y é uma subfórmula de Z, X é uma subfórmula de Z.

Usando esse conceito de subfórmula, podemos representar uma fórmula como uma árvore de formação. Seja por exemplo a seguinte fórmula:

$$[(P \wedge Q) \supset (\neg P \vee \neg\neg Q)] \vee (Q \supset \neg P)$$

Eis a árvore de formação dessa fórmula:



2 Semântica

A semântica da lógica proposicional usa o conjunto **Tr** $= \{$**T**,**F**$\}$, onde **T F** significam *verdadeiro* (true) ou *falso.* Chamaremos *valores de verdade* os elementos do conjunto **Tr**. Para interpretar uma fórmula da lógica proposicional, usamos uma função de valoração booleana. Seja S o conjunto de fórmulas da lógica proposicional. A função de valoração é uma função $v : S \mapsto \mathbf{Tr}$ definida assim:

Seja $X \in S$ uma fórmula da lógica proposicional,

- Se X é $\bot$, $v(X) = \mathbf{F}$; se X é $\top$, $v(X) = \mathbf{T}$
- Se X é um símbolo proposicional, $v(X)$ é um dos valores $\mathbf{T}$ ou $\mathbf{F}$.
- Se X é da forma $\neg\alpha$, o valor $v(X) = \mathbf{T}$ se $v(\alpha) = \mathbf{F}$, e $v(X) = \mathbf{F}$ se $v(\alpha) = \mathbf{T}$.
- Se X é da forma $(\alpha \circ \beta)$, onde $\circ$ é um conector binário, o valor $v(X)$ é determinado segundo as seguintes tabelas de verdade:

$v(\alpha)$	$v(\beta)$	$v(\alpha \wedge \beta)$	$v(\alpha \vee \beta)$	$v(\alpha \supset \beta)$
T	T	T	T	T
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

Note que os conectores $\neg$ e $\wedge$ formam um conjunto completo de conectores. Isso significa que qualquer outro conector da lógica proposicional pode ser definido em função desses dois. Considere por exemplo os conectores $\vee$ e $\supset$:

$(X \vee Y)$ é equivalente a $\neg(\neg X \wedge \neg Y)$
$(X \supset Y)$ é equivalente a $\neg(X \wedge \neg Y)$

É interessante notar que o conector $\neg$ com qualquer um dos outros conectores forma um conjunto completo.

Exercício: Mostre que todo conector binário pode ser definido usando somente $\neg$ e qualquer outro conector.

Com a definição da função de valoração dada em cima, podemos estabelecer o valor de verdade de qualquer fórmula da lógica proposicional. Esse valor vai ser completamente determinado pelos valores retornados pela função para os símbolos proposicionais. Seja f uma função que retorna um valor de verdade para qualquer símbolo proposicional. Chamamos f uma *interpretação*. Podemos verificar que:

a) Para cada função de interpretação f, existe uma função de valoração booleana v que concorda com f para os símbolos proposicionais.

b) Se v_1 e v_2 sã duas funções de valoração que concordam com um conjunto S de símbolos proposicionais, v_1 e v_2 concordam com qualquer fórmula que contém somente símbolos de S.

No que tange à semântica da lógica proposicional, há duas noções importantes:

> Uma fórmula proposicional X é uma *tautologia* (ou é dita *válida*) se $v(X) = \mathbf{T}$ para toda função de valoração v.
>
> Um conjunto S de fórmulas proposicionais é *satisfatível* se existe uma função de valoração booleana v tal que $v(X) = \mathbf{T}$ para todo $X \in S$. Um conjunto de fórmula que não é satisfatível é dito *insatisfatível* ou *inconsistente*.

O problema que consiste em determinar se uma fórmula X é uma tautologia é *decidível*. É só verificar o valor $v(X)$ para todas as funções de valoração possíveis. Considerando as proposições a) e b) dadas em cima, sabemos que a valoração de X depende unicamente dos valores atribuídos aos símbolos proposicionais que X contém. Note também que X é uma tautologia se e somente se $\{\neg X\}$ é insatisfatível.

Exemplo 1 . Eis as tabelas de verdade para mostrar que a seguinte fórmula é uma tautologia:

$$[(P \wedge Q) \supset (\neg P \vee \neg\neg Q)] \vee (Q \supset \neg P)$$

P	Q	$(P \wedge Q)$	$\neg P$	$\neg Q$	$\neg\neg Q$	$(\neg P \vee \neg\neg Q)$	X^*	$(Q \supset \neg P)$	$X^* \vee (Q \supset \neg P)$
T	**T**	**T**	**F**	**F**	**T**	**T**	**T**	**F**	**T**
T	**F**	**F**	**F**	**T**	**F**	**F**	**T**	**T**	**T**
F	**T**	**F**	**T**	**F**	**T**	**T**	**T**	**T**	**T**
F	**F**	**F**	**T**	**T**	**F**	**T**	**T**	**T**	**T**

$X^* = (P \wedge Q) \supset (\neg P \vee \neg\neg Q)$

Exemplo 2 . Vamos utilizar a técnica das tabelas de verdade para mostrar que a seguinte fórmula é satisfatível:

$$(P \supset Q) \supset (P \supset R)$$

P	Q	R	$(P \supset Q)$	$(P \supset R)$	$(P \supset Q) \supset (P \supset R)$
T	**T**	**T**	**T**	**T**	**T**
T	**T**	**F**	**T**	**F**	**F**
T	**F**	**T**	**F**	**T**	**T**
T	**F**	**F**	**F**	**F**	**T**
F	**T**	**T**	**T**	**T**	**T**
F	**T**	**F**	**T**	**T**	**T**
F	**F**	**T**	**T**	**T**	**T**
F	**F**	**F**	**T**	**T**	**T**

Nesse exemplo, vemos que a fórmula não é uma tautologia, pois existe uma interpretação que a torna false, mas ela é satisfatível, pois existe várias interpretações que a tornam verdadeira.

Exemplo 3 Vamos mostrar agora uma fórmula inconsistente:

$$(P \supset \neg Q) \wedge (P \wedge Q)$$

P	Q	$\neg Q$	$(P \supset \neg Q)$	$(P \wedge Q)$	$(P \supset \neg Q) \wedge (P \wedge \neg Q)$
T	**T**	**F**	**F**	**T**	**F**
T	**F**	**T**	**T**	**F**	**F**
F	**T**	**F**	**T**	**F**	**F**
F	**F**	**T**	**T**	**F**	**F**

Existe um conector que não faz parte da lista dada em cima, mas que às vezes é muito útil. É o conector de equivalência $\equiv$. Esse conector pode ser definido assim: $(X \equiv Y) = (X \supset Y) \wedge (Y \supset X)$. É fácil de verificar que $v(X \equiv Y) = \mathbf{T}$ se $v(X) = v(Y)$. Senão, $v(X \equiv Y) = \mathbf{F}$.

Exercício 1 Considere os seguintes enunciados da lógica proposional. Para cada um, indique se ele e satisfatível, inconsistente ou válido.

a) $(P \supset (Q \supset P))$
b) $(P \supset P)$
c) $(P \supset P) \supset Q$
d) $(P \vee Q) \vee (P \supset Q)$
e) $P \wedge (Q \vee \neg Q)$
f) $(P \wedge Q) \vee \neg Q$
g) $((P \wedge R) \wedge (P \supset Q)) \wedge (Q \supset \neg R)$
h) $((P \supset Q) \supset P) \supset P$
i) $((P \supset Q) \wedge (Q \equiv \bot)) \supset \neg P$
j) $(P \wedge \neg Q) \equiv \neg(\neg P \vee Q)$

---

3 Notação uniforme

O conjunto de conectores binários da lógica proposicional pode ser dividido em duas classes: os conectores *conjuntivos* e os conectores *disjuntivos*. Chamamos *literal* uma fórmula proposicional que é uma fórmula atómica ou a negação de uma fórmula atómica. As fórmulas que usam os conectores conjuntivos podem ser rescritas em uma conjunção de dois literais. Fórmulas que usam os conectores disjuntivos podem ser rescritas em uma disjunção de dois literais. Isso é ilustrado na seguinte tabela:

Conjuntivos			Disjuntivos		
α	α_1	α_2	β	β_1	β_2
$(X \wedge Y)$	X	Y	$\neg(X \wedge Y)$	$\neg X$	$\neg Y$
$\neg(X \vee Y)$	$\neg X$	$\neg Y$	$(X \vee Y)$	X	Y
$\neg(X \supset Y)$	X	$\neg Y$	$(X \supset Y)$	$\neg X$	Y

4 Forma normal

O objetivo das formas normais é a obtenção de fórmulas padronizadas. A principal vantagem das fórmas normais é que elas facilitam a automatização da demonstração de teorema.

Podemos definir dois tipos de fórmulas estandardizadas. Seja $X_1, X_2, \ldots X_n$ literais.

Cláusula: Escrita da forma $[X_1, X_2, ..., X_n]$, ela representa a disjunção $(X_1 \vee X_2 \vee \ldots X_n)$.

Cláusula dual Escrita da forma $\langle X_1, X_2, ..., X_n \rangle$, ela representa a conjunção $(X_1 \wedge X_2 \wedge \ldots X_n)$.

A cada uma dessas forma corresponde uma forma normal. A *forma conjuntiva normal* é uma conjunção de cláusulas. A *forma disjuntiva normal* é uma disjunção de cláusulas duais. Aqui será apresentada somente a forma conjuntiva normal.

4.1 Forma conjuntiva normal

Eis o algoritmo para transformar uma fórmula em formal conjuntiva normal:

Transformação em forma conjuntiva normal: Seja X a fórmula a transformar. Seja F_n a forma normal que queremos obter. Começamos com a conjunção $F_n = \langle [X] \rangle$. Enquanto F_n não é uma forma conjuntiva normal, efetuamos as seguintes operações. Seja $F_n = \langle D_1, D_2, \ldots, D_k \rangle$. Selecionamos um D_i que contém elementos que não são literais. Seja N um desses elementos que não é um literal.

- Se N é $\neg\top$, substituímos N por $\bot$.
- Se N é $\neg\bot$, substituímos N por $\top$.
- Se N é $\neg\neg Z$, substituímos N por Z.
- Se N é uma fórmula β, substituímos N pelas duas fórmulas β_1 e β_2.
- Se N é uma fórmula α, substituímos a disjunção D_i por duas disjunções, uma como D_i exceto que α é substituído por α_1 e a outra como D_i mas com α substituído por α_2.

Exemplo. Vamos utilizar esse algoritmo para obter a forma conjuntiva normal da fórmula $(P \supset (Q \supset R)) \supset ((P \wedge S) \supset R)$:

$$
\begin{array}{ll}
1) & \langle [(P \supset (Q \supset R)) \supset ((P \wedge S) \supset R)] \rangle \\
2) & \langle [\neg(P \supset (Q \supset R)), ((P \wedge S) \supset R)] \rangle \\
3)) & \langle [P, ((P \wedge S) \supset R)], [\neg(Q \supset R)), ((P \wedge S) \supset R)] \rangle \\
4) & \langle [P, \neg(P \wedge S), R], [\neg(Q \supset R)), ((P \wedge S) \supset R)] \rangle \\
5) & \langle [P, \neg P \neg S, R], [\neg(Q \supset R)), ((P \wedge S) \supset R)] \rangle \\
6) & \langle [P, \neg P \neg S, R], [\neg(Q \supset R)), \neg(P \wedge S), R)] \rangle \\
7) & \langle [P, \neg P \neg S, R], [\neg(Q \supset R)), \neg P, \neg S, R)] \rangle \\
8) & \langle [P, \neg P \neg S, R], [Q, \neg P, \neg S, R)], [\neg R, \neg P, \neg S, R)] \rangle
\end{array}
$$

Note que na forma normal obtida, tem duas cláusulas que contém no mesmo um símbolo proposicional e a negação dele. Então essas cláusulas representam tautologias e podem ser eliminadas.

Exercício 2 Transforme em forma conjuntiva normal as seguintes fórmulas:

a) $P \vee \neg P$
b) $P \wedge \neg P$
c) $P \supset (Q \supset P)$
d) $\neg(\neg(P \vee Q) \wedge (P \vee Q))$
e) $\neg((P \supset (Q \supset R)) \supset ((P \supset Q) \supset (P \supset R)))$
f) $(P \vee \neg Q) \supset (R \vee W)$

Exercício 3 Mostre que os seguintes enunciados são tautologias (n e k podem ser 0):

a) $[A_1, \ldots, A_n, (U \vee V), B_1, \ldots, B_k] \equiv [A_1, \ldots, A_n, U, V, B_1, \ldots, B_k]$
b) $[A_1, \ldots, A_n, (U \wedge V), B_1, \ldots, B_k] \equiv$
$[A_1, \ldots, A_n, U, B_1, \ldots, B_k] \wedge [A_1, \ldots, A_n, V, B_1, \ldots, B_k]$

4.2 Cláusula de Horn

Existe um caso particular de forma normal: a que contém somente **cláusulas de Horn**, isto é, cláusulas que contêm no máximo um literal positivo. Por exemplos, as cláusulas $[\neg P, Q, \neg R]$, $[P]$ e $[\neg P, \neg R]$ são cláusulas de Horn. A cláusula $[\neg P, Q, \neg R, S]$ não é uma cláusula de Horn.

Nem todos os conjuntos de fórmulas podem ser traduzidos em forma normal que contém somente cláusulas de Horn. Considere por exemplo a fórmula $P \supset (Q \vee R)$. A forma normal é $[\neg P, Q, R]$ que não é uma cláusula de Horn.

A importância das cláusulas de Horn vai aparecer mais tarde quando será apresentada a técnica de prova por resolução. A perda em poder expressivo será compensada por ganho em eficiência.

5 Consequência lógica

O que nos interessa no uso da lógica não é somente saber se uma fórmula é uma tautologia. Mais interessante é saber se uma certa fórmula pode ser deduzida a partir de um conjunto de fórmulas. Em outras palavras, queremos saber o que podemos concluir a partir de um certo conhecimento. Em lógica, isso representado pelo conceito de *consequência lógica*, definido assim:

> Uma fórmula X é uma consequência lógica de um conjunto de fórmula S se toda valuação que torna S verdadeiro também torna X verdadeiro. Utilizaremos a notação $S \models X$ para expressar o fato que X é consequência lógica de S.

Assim, se sempre que um conjunto de fatos é verdadeiro, um fato X é verdadeiro, diremos que X é uma consequência lógica desse conjunto. Considere, por exemplo, a seguinte situação:

> Se o presidente não conseguir diminuir o déficit, a situação econômica do país não vai melhorar, ou o FMI não vai emprestar o dinheiro e a situação vai piorar. O presidente não consegue diminuir o déficit e a situação não está piorando.

Suponhamos a seguintes proposições:

P: O presidente diminua o déficit.
Q: A situação melhora.
R: O FMI empresta o dinheiro.
S: A situação piora.

Os fatos podem ser representados com a seguinte base de conhecimento:

BC: (1) $\neg P \supset \neg Q \vee (\neg R \wedge S)$
(2) $\neg P$
(3) $\neg S$

Com essa base de conhecimento, não podemos deduzir nada sobre a proposição R, isto é, não é possível saber se o FMI vai ou não vai emprestar o dinheiro. Em outros termos, não é possível mostrar $BC \models R$, nem $BC \models \neg R$. Mas é possível fazer uma dedução sobre Q. Podemos mostrar $BC \models \neg Q$, isto é, a situação não vai melhorar. Tem ao menos duas maneiras para fazer essa demonstração. Podemos usar a semântica, usando as tabelas de verdade, para mostrar a conseqüências lógica:

P	Q	R	S	BC	$\neg Q$
V	V	V	V	F	F
V	V	V	F	F	F
V	V	F	V	F	F
V	V	F	F	F	F
V	F	V	V	F	V
V	F	V	F	F	V
V	F	F	V	F	V
V	F	F	F	F	V
F	V	V	V	F	F
F	V	V	F	F	F
F	V	F	V	F	F
F	V	F	F	F	F
F	F	V	V	F	V
F	F	V	F	V	V
F	F	F	V	F	V
F	F	F	F	V	V

Segunda a tabela de verdade, só tem duas interpretações que tornam verdadeira a base de conhecimento. E nessas duas interpretações, a proposição Q é falsa, isto é, $\neg Q$ é verdadeira. Então, $BC \models \neg Q$. Podemos afirmar que a situação não vai melhorar.

6 Inferência em lógica proposicional

Para efetuar inferências apresentaremos mais para frente um mtodo que pode ser automatizado. Por enquanto, veremos comos realizar inferncia usando um conjunto de regras de inferência e regras algébricas.

6.1 Regras de inferência

Modus Ponens $$\frac{\alpha \supset \beta \quad \alpha}{\beta}$$

Exemplo: Sabendo que um incêndio implica a presência de fumaça e que há um inceêndio, então podemos concluir que há tambem fumaça:

P: Tem incêndio
Q: Tem fumaça
$P \supset Q$
P
$$\frac{P \supset Q \quad P}{Q}$$

Modus Tolens $$\frac{\alpha \supset \beta \quad \neg\beta}{\neg\alpha}$$

Exemplo: Suponhamos que sabemos que um incêndio implica a presência de fumaça. Se também sabemos que não tem fumaça, podemos concluir que não tem incêncio:

P: Tem incêndio
Q: Tem fumaça
$P \supset Q$
$\neg Q$
$$\frac{P \supset Q \quad \neg Q}{\neg P}$$

Eliminação de conjunção $$\frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}{\alpha_i}$$

O sentido dessa regra é que se uma conjunção de fatos é verdadeira, então necessariamente cada fato considerado isoladamente tem que ser verdadeiro.

Exemplo: Sabendo que Paulo é irmão de Maria e de Roberto, podemos fazer essas afirmações separadamente: Paulo e o irmão de Maria e Paulo e o irmão de Roberto:

P: Paulo é o irmão de Maria
Q: Paulo é o irmão de Roberto
$P \wedge Q$
$$\frac{P \wedge Q}{P} \quad \frac{P \wedge Q}{Q}$$

Introdução da conjunção $$\frac{\alpha_1 \quad \alpha_2 \quad \ldots \quad \alpha_n}{\alpha_i \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}$$

O sentido dessa regra é que se fatos considerados isoladamente são verdadeiro, então a conjunção desses fatos é também verdadeira.

Exemplo:

P: Paulo é o irmão de Maria
Q: Paulo é o irmão de Roberto
P
Q
$$\frac{P \quad Q}{P \wedge Q}$$

Eliminação da dupla negação $$\frac{\neg\neg\alpha}{\alpha}$$

Resolução unitária $$\frac{\alpha \vee \beta \qquad \neg\beta}{\alpha}$$

O sentido dessa regra é o seguinte. Sabendo que, entre dois fatos, um ou o outro é verdadeiro, se também temos conhecimento de que um desses fatos é falso, então necessariamente o outro tem que ser verdadeiro.

Resolução $$\frac{\alpha \vee \beta \qquad \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

O principe de resolução é mais complicado. Visto que β não pode ser verdadeiro e falso no mesmo tempo, um dos fatos α e γ tem que ser verdadeiro.

A seguinte tabela de verda mostra a validade da regra de resolução (cada vez que $\alpha \vee \beta$ e $\neg\beta \vee \gamma$ são verdadeiro, $\alpha \vee \gamma$ também é verdadeiro.

α	β	γ	$\alpha \vee \beta$	$\neg\beta \vee \gamma$	$\alpha \vee \gamma$
F	F	F	F	V	F
F	F	V	F	V	V
F	V	F	V	F	F
F	**V**	**V**	**V**	**V**	**V**
V	**F**	**F**	**V**	**V**	**V**
V	**F**	**V**	**V**	**V**	**V**
V	V	F	V	F	V
V	**V**	**V**	**V**	**V**	**V**

6.2 Regras algébricas

Além das regras de inferências, existem regras algébricas que permitem reescrever um enunciado:

Leis de De Morgan:
a) $\neg(P \vee Q) \equiv (\neg P \wedge \neg Q)$
b) $\neg(P \wedge Q) \equiv (\neg P \vee \neg Q)$
Associatividade:
c) $P \wedge (Q \wedge R) \equiv (P \wedge Q) \wedge R$
d) $P \vee (Q \vee R) \equiv (P \vee Q) \vee R$
Comutatividade:
e) $P \wedge Q \equiv Q \wedge P$
f) $P \vee Q \equiv Q \vee P$
Distributividade:
g) $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
h) $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$
Outras:
i) $(P \equiv Q) \equiv ((P \supset Q) \wedge (Q \supset P))$
j) $(P \supset Q) \equiv (\neg P \vee Q)$

6.3 Monotonicidade

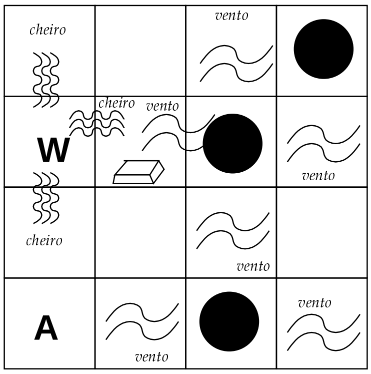
A monotonicidade é uma propriedade importante. Uma lógica é **monotónica** se quando se acresenta novos enunciados, todos os enunciados que eram logicamente implicados pela base de conhecimento continuam a ser logicamente implicados pela nova base aumentada:

Se $KB_1 \models \alpha$ então $(KB_1 \cup KB_2) \models \alpha$.

Uma conseqüência importante da propriedade de monotonicidade e que, uma vez estabelecida a verdade de um fato, essa informação não mudará no futuro. Se uma lógica não é monotónica, precisamos de um mecanismo para manter a coerência da base de conhecimento quando acrescentamos novos fatos.

6.4 Exemplo

Vamos ver agora um exemplo que mostra as vantágem da lógica para realizar um sistema que raciocina. Considere o mundo do Wumpus, ilustrado na figura abaixo.



No inicio, o agente está na posição (1,1) e o objetivo é de achar o ouro, sem entrar em um lugar onde ele pode ser matado pelo Wumpus (que se encontra na posição (1,3) na ilustração) e sem cair numa fossa (no exemplo, tem 3 fossas).

Suponhamos que o agente não sabe onde se encontram as fossas e o Wumpus. Ele pode suspeitar a presência desses perigos pelo cheiro e o vento perceptíveis nos quadrados adjacentes à localisação do Wumpus e das fossas, respectivamente.

Para se deslocar até o ouro evitando os perigos, o agente vai ter que fazer deduções a partir das informações sobre o cheiro e o vento que contem o seu percept em cada localização.

Eis uma maneira de representar esse problema em lógica proposicional:

Conhecimento sobre o mundo:

$C_{i,j}$ significa que tem cheiro no quadrado i,j
$W_{i,j}$ significa que o Wumpus está no quadrado i,j
$V_{i,j}$ significa que tem um vento no quadrado i,j
$F_{i,j}$ significa que tem uma fossa no quadrado i,j

O conhecimento que o agente tem inicialmente é o seguinte. Primeiro, temos um conjunto de fatos (um para cada posição na caverna) para exprimir que se tem cheiro em uma posição, o Wumpus necessariamente está localizado nesse lugar ou em uma posição adjacente:

$C_{1,1} \supset W_{1,1} \vee W_{1,2} \vee W_{2,1}$
$C_{1,2} \supset W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$
...

Outros fatos exprimem, para cada posição, que se o Wumpus está nessa posição, tem cheiro nas posições adjacentes:

$W_{1,1} \supset C_{1,1}$
$W_{1,1} \supset C_{1,2}$
$W_{1,1} \supset C_{2,1}$
...
$W_{1,3} \supset C_{1,3}$
$W_{1,3} \supset C_{1,2}$
$W_{1,3} \supset C_{1,4}$
$W_{1,3} \supset C_{2,3}$
...

Finalmente, temos fatos semelhantes pelas fossas:

$V_{1,1} \supset F_{1,2} \vee F_{2,1}$
$V_{2,1} \supset F_{1,1} \vee F_{2,2} \vee F_{3,1}$
...
$F_{3,1} \supset V_{2,1}$
$F_{3,1} \supset V_{3,2}$
$F_{3,1} \supset V_{4,1}$

A isso, vamos acrescentar mais um conjunto de fatos, para indicar que um lugar é seguro, isto é, temos certeza que nesse lugar não tem nem o Wumpus nem uma fossa:

$SEGURO_{i,j}$ significa que não tem perigo na posição i,j
$\neg W_{1,1} \wedge \neg F_{1,1} \supset SEGURO_{1,1}$
$\neg W_{1,2} \wedge \neg F_{1,2} \supset SEGURO_{1,2}$
...
$\neg W_{4,4} \wedge \neg F_{4,4} \supset SEGURO_{4,4}$

No início, o agente está no lugar (1,1). O seu percept indica que ele não está morto e que nesse lugar não se encontra o Wumpus e não tem fossa. O percept indica também que não tem cheiro nem vento nesse lugar. Então, podemos acrescentar os seguintes fatos na base de conhecimento:

$\neg C_{1,1}$
$\neg V_{1,1}$
$SEGURO_{1,1}$
$\neg W_{1,1}$
$\neg F_{1,1}$

Para escolher o próximo lugar a visitar, o agente tem que fazer deduções para identificar, entre os lugares possíveis, os que são seguros.

As deduções que o agente pode fazer nesse momento são as seguintes:

$$\frac{W_{1,1} \supset C_{1,1} \qquad \neg C_{1,1}}{\neg W_{1,1}} \quad \text{Modus Tolens}$$

$$\frac{W_{1,2} \supset C_{1,1} \qquad \neg C_{1,1}}{\neg W_{1,2}} \quad \text{Modus Tolens}$$

$$\frac{W_{2,1} \supset C_{1,1} \qquad \neg C_{1,1}}{\neg W_{2,1}} \quad \text{Modus Tolens}$$

$$\frac{F_{2,1} \supset V_{1,1} \qquad \neg V_{1,1}}{\neg F_{2,1}} \quad \text{Modus Tolens}$$

$$\frac{F_{1,2} \supset V_{1,1} \qquad \neg V_{1,1}}{\neg F_{1,2}} \quad \text{Modus Tolens}$$

Então, podemos acrescentar na base de conhecimento os seguintes fatos: $\neg W_{1,1}$, $\neg W_{1,2}$, $\neg W_{2,1}$, $\neg F_{2,1}$, $\neg F_{1,2}$. Com essa informação, podemos fazer mais inferências que são importantes para identificar o próximo lugar a visitar:

$$\frac{\neg W_{2,1} \qquad \neg F_{2,1}}{\neg W_{2,1} \wedge \neg F_{2,1}} \quad \text{Introd. conj.}$$

$$\frac{\neg W_{1,2} \qquad \neg F_{1,2}}{\neg W_{1,2} \wedge \neg F_{1,2}} \quad \text{Introd. conj.}$$

$$\frac{\neg W_{2,1} \wedge \neg F_{2,1} \supset SEGURO_{2,1} \qquad \neg W_{2,1} \wedge \neg F_{2,1}}{SEGURO_{2,1}} \quad \text{Modus Ponens}$$

$$\frac{\neg W_{1,2} \wedge \neg F_{1,2} \supset SEGURO_{1,2} \qquad \neg W_{1,2} \wedge \neg F_{1,2}}{SEGURO_{1,2}} \quad \text{Modus Ponens}$$

Agora, sabendo que as duas posições (1,2) e (2,1) são seguras, o agente pode escolher qualquer uma. Vamos supor que ele vai escolher a posição (2,1). As informações que ele pode extrair do percept são as seguintes: $\neg C_{2,1}$ e $V_{2,1}$ (não tem cheiro mas tem vento).

Eis as inferências que ele pode fazer:

$$\frac{W_{2,2} \supset C_{2,1} \quad \neg C_{2,1}}{\neg W_{2,2}} \quad \text{Modus Tolens}$$

$$\frac{W_{3,1} \supset C_{2,1} \quad \neg C_{2,1}}{\neg W_{3,1}} \quad \text{Modus Tolens}$$

$$\frac{V_{2,1} \supset F_{1,1} \vee F_{2,2} \vee F_{3,1} \quad V_{2,1}}{F_{1,1} \vee F_{2,2} \vee F_{3,1}} \quad \text{Modus Ponens}$$

$$\frac{F_{1,1} \vee F_{2,2} \vee F_{3,1} \quad \neg F_{1,1}}{F_{2,2} \vee F_{3,1}} \quad \text{Resolução}$$

Nesse ponto, não tem jeito de provar que as posições (3,1) e (2,2) são seguras. Então, o mais prudente para o agente será voltar no lugar (1,1) e considerar a outra possibilidade que ele deixou de lado. Chegando no lugar (1,2), as informações extraídas do percept são as seguintes: $C_{1,2}$ e $\neg V_{1,2}$ (tem cheiro mas não tem vento). O agente pode fazer as seguintes deduções:

$$\frac{C_{1,2} \supset W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1} \quad C_{1,2}}{W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}} \quad \text{Modus Ponens}$$

$$\frac{W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1} \quad \neg W_{1,1}}{W_{1,3} \vee W_{1,2} \vee W_{2,2}} \quad \text{Resolução}$$

$$\frac{W_{1,3} \vee W_{1,2} \vee W_{2,2} \quad \neg W_{2,2}}{W_{1,3} \vee W_{1,2}} \quad \text{Resolução}$$

$$\frac{W_{1,3} \vee W_{1,2} \quad \neg W_{1,2}}{W_{1,3}} \quad \text{Resolução}$$

$$\frac{F_{1,3} \supset V_{1,2} \quad \neg V_{1,2}}{\neg F_{1,3}} \quad \text{Modus Tolens}$$

$$\frac{F_{2,2} \supset V_{1,2} \qquad \neg V_{1,2}}{\neg F_{2,2}} \qquad \text{Modus Tolens}$$

$$\frac{F_{2,2} \lor F_{3,1} \qquad \neg F_{2,2}}{F_{3,1}} \qquad \text{Resolução}$$

O agente já consiguiu deduzir que o Wumpus está na posição (1,3) e que tem uma fossa na posição (3,1). Também, tendo deduzido o fato $\neg F_{2,2}$, o agente pode combinar essa informação com o outro fato $\neg W_{2,2}$ para deduzir o fato $SEGURO_{2,2}$, usando o Modus Ponens:

$$\frac{\neg W_{2,2} \land \neg F_{2,2} \supset SEGURO_{2,2} \qquad \neg W_{2,2} \land \neg F_{2,2}}{SEGURO_{2,2}} \qquad \text{Modus Ponens}$$

Então, o agente pode ir sem problema na posição (2,2).

Exercício 4 Seja Δ a seguinte base de conhecimento:

1) $P \lor \neg Q \supset R \lor W$
2) $Q \supset (\neg P \supset W)$
3) $(\neg W \supset P) \supset R$

Prove $\Delta \models R$:

a) Utilizando as tabelas de verdade.
b) Utilizando as regras de inferência.
