Capítulo 1

Classificação

Sentença Uma sentença é uma sequência finita de símbolos que se distingue da fórmula pelo facto de não possuir variáveis livres

Sentença atómica Uma sentença atómica é uma sentença que não pode ser dividida em outras sentenças mais simples.

Fórmula Fórmulas são idênticas ás sentenças com a diferença de possuírem variáveis livres.

Formula atómica Uma fórmula atómica é uma fórmula que não contém conectivos lógicos nem quantificadores, ou seja, uma fórmula que não contém sub-fórmulas.

$$
\begin{array}{ccccccc}
 & & & \text{Possui variáveis livres?} & & & \\
 & & \swarrow & & \searrow & & \\
 & & \text{Sim} & & \text{Não} & & \\
 & & \downarrow & & \downarrow & & \\
 & & & \text{Pode ser dividida em fragmentos?} & & & \\
 & \swarrow & \searrow & & \swarrow & \searrow & \\
 & \text{Sim} & \text{Não} & & \text{Sim} & \text{Não} & \\
 & \downarrow & \downarrow & & \downarrow & \downarrow & \\
 & \text{Fórmula} & \text{Fórmula atómica} & & \text{Sentença} & \text{Sentença atómica} &
\end{array}
$$

* Uma variável diz-se livre se não está sobre o efeito de nenhum quantificador ($\exists$ ou $\forall$).

símbolos predicativos Servem para exprimir relações entre objetos de acordo com as as seguintes regras:

- cada símbolo predicativo tem uma aridade igual ao número de argumentos da relação correspondente;

- a relação expressa deve ser bem determinada.

Por exemplo: $Tet(a), Samesize(a, b), Between(a, b, c)$ de aridade 1, 2 e 3, respetivamente.

símbolos funcionais São símbolos, cada um com uma aridade, usados para construir os termos, em particular, os termos fechados.

termos fechados Os termos fechados permitem obter outras designações para objetos de acordo com regras similares às regras para as constantes. São construidos de acordo com as seguintes regras:

- todas as constantes são termos fechados e todas as variáveis são termos;
- se f é um símbolo funcional de aridade n e $t_1, ..., t_n$ são n termos fechados, então a expressão $f(t_1, ..., t_n)$ é um termo fechado;
- apenas as expressões obtidas aplicando as regras anteriores um número finito de vezes são termos fechados.
- Um termo diz-se fechado se nele não ocorrem variáveis.

Nota: Conectivos booleanos $(<, >, \neg)$ são usados para formar sentenças novas a partir de sentenças validas pré existentes. Não podem, no entanto, ser usados em termos.
Em geral, num exercício de classificação deve-se:

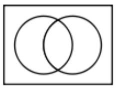
- Verificar a sintaxe (colocação de virgulas, parênteses, etc.)
- Verificar se as aridades foram respeitadas
- Verificar se os símbolos predicativos ocorrem sempre fora dos funcionais e de outros predicativos
- Verificar se foram usados conectivos booleanos sem a presença de predicativos

Capítulo 2

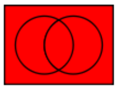
Valor de verdade em mundos

sentença atómica	**interpretação**
Tet(a)	a é um tetraedro
Cube(a)	a é um cubo
Dodec(a)	a é um dodecaedro
Small(a)	a é pequeno
Medium(a)	a é médio
Large(a)	a é grande
SameSize(a, b)	a e b têm o mesmo tamanho
SameShape(a, b)	a e b têm a mesma forma
Larger(a, b)	a é maior do que b
Smaller(a, b)	a é mais pequeno do que b
SameCol(a, b)	a e b estão na mesma coluna
SameRow(a, b)	a e b estão na mesma linha
Adjoins(a, b)	a e b estão em quadrículas contíguas (mas não diagonalmente)
LeftOf(a, b)	a está mais próxima da esquerda do quadriculado do que b
RightOf(a, b)	a está mais próxima da direita do quadriculado do que b
FrontOf(a, b)	a está mais próxima da frente do quadriculado do que b
BackOf(a, b)	a está mais próxima de trás do quadriculado do que b
Between(a, b, c)	a está entre b e c, na mesma coluna, linha ou diagonal

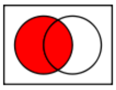
Contradição/Falso

Notação	Fórmulas equivalentes	Tabela de verdade	Diagrama de Venn
$\bot$ "inferior"	$P \wedge \neg P$ Opq	Q: 0 1 P 0: 0 0 P 1: 0 0	

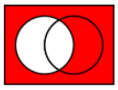
Tautologia/Verdade

Notação	Fórmulas equivalentes	Tabela de verdade	Diagrama de Venn
$\top$ "topo"	$P \vee \neg P$ Vpq	Q: 0 1 P 0: 1 1 P 1: 1 1	

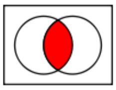
Proposição P

Notação	Fórmulas equivalentes	Tabela de verdade	Diagrama de Venn
P	p Ipq	Q: 0 1 P 0: 0 0 P 1: 1 1	

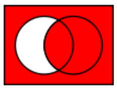
Negação de P

Notação	Fórmulas equivalentes	Tabela de verdade	Diagrama de Venn
$\neg P$ $\sim P$	Np Fpq	Q: 0 1 P 0: 1 1 P 1: 0 0	

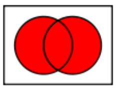
Conjunção

Notação	Fórmulas equivalentes	Tabela de verdade	Diagrama de Venn
$P \wedge Q$ $P \,\&\, Q$ $P \cdot Q$ P AND Q	$P \nrightarrow \neg Q$ $\neg P \nleftarrow Q$ $\neg P \downarrow \neg Q$ Kpq	Q: 0 1 P 0: 0 0 P 1: 0 1	

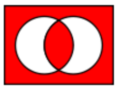
Implicação material

Notação	Fórmulas equivalentes	Tabela de verdade	Diagrama de Venn
$P \rightarrow Q$ $P \supset Q$ $P \leq Q$	$P \uparrow \neg Q$ $\neg P \vee Q$ $\neg P \leftarrow \neg Q$ Cpq	Q: 0 1 P 0: 1 1 P 1: 0 1	

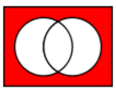
Disjunção

Notação	Fórmulas equivalentes	Tabela de verdade	Diagrama de Venn
$P \vee Q$ P OR Q	$P \leftarrow \neg Q$ $\neg P \rightarrow Q$ $\neg P \uparrow \neg Q$ $\neg(\neg P \wedge \neg Q)$ Apq	Q: 0 1 P 0: 0 1 P 1: 1 1	

Bicondicional

Notação	Fórmulas equivalentes	Tabela de verdade	Diagrama de Venn
$P \leftrightarrow Q$ $P \equiv Q$ P XNOR Q P IFF Q	$P \nleftrightarrow \neg Q$ $\neg P \nleftrightarrow Q$ $\neg P \leftrightarrow \neg Q$ Epq	Q: 0 1 P 0: 1 0 P 1: 0 1	

Negação conjunta

Notação	Fórmulas equivalentes	Tabela de verdade	Diagrama de Venn
$P \downarrow Q$ P NOR Q	$P \nleftarrow \neg Q$ $\neg P \nrightarrow Q$ $\neg P \wedge \neg Q$ Xpq	Q: 0 1 P 0: 1 0 P 1: 0 0	

Capítulo 3

tt

3.1 Satisfazibilidade

Uma sentença diz-se tt-satisfazível se há pelo menos uma linha da tabela de verdade em que o seu valor é V ou 1 (verdadeiro). Não confundir com tautologia, que é verdadeiro em todas as linhas. Por exemplo P:

P
0
1

3.2 Equivalência

Duas sentenças P e Q são tt-equivalentes (ou tautologicamente equivalentes) se, na tabela de verdade conjunta para P e Q, em cada uma das linhas as sentenças P e Q têm o mesmo valor lógico. Por exemplo, as sentenças P e $P \wedge (P \vee Q)$:

P	Q	P ∨ Q	P ∧ (P ∨ Q)	P
0	0	0	0	0
0	1	1	0	0
1	0	1	1	1
1	1	1	1	1

3.3 Consequência

Uma sentença Q é tt-consequência de P se e só se $P \rightarrow Q$ é uma tautologia. Se Q é tt-consequência de P escreve-se $P \Rightarrow Q$. Numa tabela de verdade conjunta verifica-se que Q é tt-consequência de P e S se Q não é falsa quando P e S são verdadeiras, i.e. Q só não é tt-consequência de P e S se existir uma linha em que P e S são verdadeiras e Q é falsa.

Capítulo 4

Formas

4.1 Forma normal disjuntiva

Uma fórmula está na forma normal disjuntiva se é apenas composta por disjunções.

4.1.1 Conversão para fnd

- Remoção das equivalências:

$$A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A)$$

- Remoção das implicações:

$$A \rightarrow B = \neg A \vee B$$

- Aplicar as leis de De Morgan:

$$\neg(A \vee B) = (\neg A \wedge \neg B), \neg(A \wedge B) = (\neg A \vee \neg B)$$

- Eliminação das negações duplas:

$$\neg\neg A = A$$

- Utilizar as leis distributivas para colocar a fórmula resultante na fnd:

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

4.2 Forma normal conjuntiva

Uma fórmula está na forma normal conjuntiva se é apenas composta por conjunções.

4.2.1 Conversão para fnc

O algoritmo de conversão para fnc é exatamente igual ao para fnd, porém, usam-se leis distributivas para se obter uma conjunção de disjunções, por exemplo:

$$((p \land q) \to r) \land s = \tag{4.1}$$
$$= (\neg(p \land q) \lor r) \land s = \tag{4.2}$$
$$= (\neg p \lor \neg q \lor r) \land s \tag{4.3}$$

4.3 Forma normal prenexa

Uma fórmula de uma LPO está na forma normal prenexa (fnp) se e só se é livre de quantificadores ou é da forma:

$$Q_1x_1Q_2x_2...Q_nx_nS$$

onde $Q \in \{\exists, \forall\}, x_1, ..., x_n$ são variáveis e S é uma fórmula da LPO livre de quantificadores.

4.3.1 Conversão para fnp

Conjunção e disjunção:

$$(\forall x\phi) \land \psi = \forall x(\phi \land \psi) \tag{4.4}$$
$$(\forall x\phi) \lor \psi = \forall x(\phi \lor \psi) \tag{4.5}$$
$$(\exists x\phi) \land \psi = \exists x(\phi \land \psi) \tag{4.6}$$
$$(\exists x\phi) \lor \psi = \exists x(\phi \lor \psi) \tag{4.7}$$

Negação:

- $\neg\exists x\phi$ é equivalente a $\forall x\neg\phi$, uma vez que se não há ao menos um x onde ϕ é verdade, então, para todo x, ϕ não é verdade.
- $\neg\forall x\phi$ é equivalente a $\exists x\neg\phi$, uma vez que, se nem para todo x, ϕ é verdadeiro, então para algum x, ϕ é falso.

Implicação:

Remover quantificadores dos antecedentes:

- $(\forall x\phi) \to \psi$ é equivalente a $\exists x(\phi \to \psi)$
- $(\exists x\phi) \to \psi$ é equivalente a $\forall x(\phi \to \psi)$

Remover quantificadores dos consequentes:

- $\phi \to (\exists x\psi)$ é equivalente a $\exists x(\phi \to \psi)$
- $\phi \to (\forall x\psi)$ é equivalente a $\forall x(\phi \to \psi)$

4.4 Skolemização

Uma fórmula na forma normal prenexa diz-se que está na forma de skolem se todos os seus quantificadores são universais ($\forall$). A skolemização assenta no facto de que $\exists = \neg\forall$ e $\neg\exists = \forall$.

4.4.1 Skolemização I

A skolemização I aplica-se quando a formula se inicia com um quantificador existencial, $\exists$. A skolemização I consiste em substituir as variáveis por objectos concretos, por exemplo:

$$\exists x \exists y (Cube(x) \wedge Larger(x, y)) \tag{4.8}$$

$$\exists y (Cube(c) \wedge Larger(c, y)) \qquad skolem.I \tag{4.9}$$

$$Cube(c) \wedge Larger(c, d) \qquad skolem.I \tag{4.10}$$

4.4.2 Skolemização II

A skolemização II aplica-se quando a formula se inicia com um quantificador universal, $\forall$. A skolemização II consiste em substituir as variáveis por simbolos, por exemplo:

$$\forall u \exists u \exists z (Tet(c) \wedge Tet(u) \rightarrow Between(u, y, z))) \tag{4.11}$$

$$\forall u \exists z (Tet(c) \wedge (Tet(u) \rightarrow Between(u, f(u), z))) \qquad skolem.II \tag{4.12}$$

$$\forall u (Tet(c) \wedge (Tet(u) \rightarrow Between(u, f(u), g(u)))) \qquad skolem.II \tag{4.13}$$

Capítulo 5

Algoritmos

5.1 Horn

5.1.1 Fórmulas de Horn

Uma fórmula de Horn é uma sentença na fnc tal que em cada disjunção de literais há no máximo um literal positivo.

5.1.2 Algoritmo de Horn

O algoritmo de Horn serve para verificar se uma fórmula de Horn S é tt-satisfazível.

Passo 1 - Fazer a lista das implicações que ocorrem em S na forma condicional.

Passo 2 - Algoritmo

Input: lista das implicações que ocorrem em S na forma condicional

Output: S é tt-satisfazível ou S não é tt-satisfazível

5.1.3 Condições de paragem

a) Se ocorre em S uma implicação do tipo $(A_1 \wedge ... \wedge A_k) \rightarrow \top$com $A_1, ..., A_k \in V$, terminar e concluir que S não é tt-satisfazível.

b) Enquanto ocorrerem em S implicações do tipo $(A_1 \wedge ... \wedge A_k) \rightarrow B$ com $A_1, ..., A_k \in V$, fazer:

$$V = V \cup \{B\}$$

e regressar a a)

c) Terminar e concluir que S é tt-satisfazível, atribuindo às sentenças atómicas em V o valor lógico 1 e às restantes o valor lógico 0.

5.1.4 Exemplo

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

Passo 1: Fazer a lista das implicações que ocorrem em S na forma condicional:

$$\top \to A, (C \wedge D) \to B, (A \wedge D) \to C, B \to D, A \to B, E \to \bot$$

Passo 2: Executar o Algoritmo:

$$1.A \tag{5.1}$$
$$- - - - - \tag{5.2}$$
$$2.B \tag{5.3}$$
$$3.D \tag{5.4}$$
$$4.C \tag{5.5}$$

Logo S é tt-satisfazível

5.2 Resolução

5.2.1 Algoritmo

A sentença S é tt satisfazível? Passo 1 - Pôr S na fnc. Formar a lista C das cláusulas de S. Passo 2 - Algoritmo:

Input: lista C das cláusulas de S

Output: S é tt-satisfazível ou S não é tt-satisfazível

5.2.2 Condições de paragem

a) Se $\{\} \in C$, terminar e concluir que S não é tt-satisfazível.

b) Enquanto existirem cláusulas $C_1, C_2 \in C$ tais que para alguma sentença atómica P temos:

   - $P \in C_1$
   - $\neg P \in C_2$
   - O resolvente
     $$C_2 = (C_1\{P\}) \cup (C_2\{\neg P\})$$
     de C_1 e C_2 não está em C, fazer:
     $$C = C \cup \{C_2\}$$
     e regressar a a).

c) Terminar e concluir que S é tt-satisfazível.

5.2.3 Exemplo

$$S = \neg A \wedge (B \vee C \vee B) \wedge (\neg C \vee \neg D) \wedge (A \vee D) \wedge (\neg B \vee \neg D)$$

Passo 1: S já está na fnc, logo, formar a lista das cláusulas:

$$C = \{\neg A\}, \{B, C\}, \{\neg C, \neg D\}, \{A, D\}, \{\neg B, \neg D\}$$

Passo 2: Executar o algoritmo:

$$1.\{\neg A\} \tag{5.6}$$
$$2.\{B, C\} \tag{5.7}$$
$$3.\{\neg C, \neg D\} \tag{5.8}$$
$$4.\{A, D\} \tag{5.9}$$
$$5.\{\neg B, \neg D\} \tag{5.10}$$
$$- - - - - - - - - - - - - \tag{5.11}$$
$$6.\{B, \neg D\} \qquad Res(2, 3) \tag{5.12}$$
$$7.\{\neg D\} \qquad Res(5, 6) \tag{5.13}$$
$$8.\{A\} \qquad Res(4, 7) \tag{5.14}$$
$$9.\{\} \qquad Res(1, 8) \tag{5.15}$$

Logo S não é tt-satisfazível