LÓGICA PARA COMPUTAÇÃO

João Carlos Gluz
Mônica Xavier Py

Escola
Politécnica

COLEÇÃO EAD
EDITORA UNISINOS

LÓGICA PARA COMPUTAÇÃO

JOÃO CARLOS GLUZ
MÔNICA XAVIER PY

Editora Unisinos, 2014

SUMÁRIO

APRESENTAÇÃO

Este livro-texto foi escrito para possibilitar aos alunos dos cursos de graduação na área de Computação (Ciência da Computação, Sistemas de Informação, Análise de Desenvolvimento de Sistemas, Engenharia da Computação, dentre outros) o aprendizado dos conceitos da Lógica Proposicional e da Lógica de Predicados.

Os tópicos do livro foram definidos para serem abordados em uma disciplina oferecida em um semestre. Essa limitação é importante, pois existe na literatura uma gama enorme de outras lógicas. As duas lógicas abordadas neste texto servem como base para várias outras disciplinas da Computação.

Para reforçar os conceitos apresentados em cada capítulo, o livro contém no final uma seção com questões comentadas em diferentes níveis de aprendizado.

É importante salientar que diversos livros cobrem os tópicos abordados neste texto. Dentre eles, devemos destacar os livros de Nolt (1991), Gersting (2002), Mortari (2001), Hein (2004) e Alencar (1999) que serviram de base para o desenvolvimento do texto apresentado neste material.

O autor

CAPÍTULO 1
LÓGICA PROPOSICIONAL

Neste capítulo são apresentadas definições precisas sobre o que são *proposições*, fórmulas e tautologias que nos permitirão definir uma linguagem formal para a lógica das proposições, ou seja, nos permitirão criar uma *Lógica Proposicional*.

1.1 Proposições e operadores lógicos

Proposição lógica

Intuitivamente, uma proposição lógica (ou apenas uma **proposição**) é uma frase ou sentença da Língua Portuguesa (ou de qualquer outro idioma) que pode assumir apenas um de dois valores-verdade: ou a frase é *verdadeira* (ela diz uma verdade) ou ela é *falsa* (diz uma falsidade).

Observações:

- Proposições são bastante comuns na linguagem natural, porque, por exemplo, todas as afirmações são naturalmente proposições.
- Existem, entretanto, vários exemplos de frases que não se encaixam na definição de proposição: comandos e solicitações, por exemplo, não são exatamente proposições - quando uma ordem é dada ("feche a porta!") ou uma solicitação feita ("por favor, me passe o prato"), não há necessariamente um valor-verdade por trás destas frases.
- A linguagem natural também permite escrever sentenças cujo significado não pode ser logicamente determinado (são os famosos *paradoxos*). Por exemplo, a sentença, aparentemente inocente, "Esta sentença é falsa" não é (nem pode ser) nem verdadeira nem falsa (pense um pouco).

As proposições são usualmente simbolizadas (representadas) por letras maiúsculas do início do alfabeto: A, B, C, ... Os valores lógicos das proposições são representados de forma resumida usando V para verdadeiro e F para falso.

As proposições podem ser simples ou compostas. As proposições compostas são formadas de proposições simples conectadas através de **operadores** (ou **conetivos**) **lógicos**. Estes operadores ou conetivos representam as seguintes operações lógicas, que serão descritas mais adiante:

- Conjunção
- Disjunção
- Negação
- Implicação (ou condicional)
- Bi-implicação (ou bicondicional)

Conjunção de proposições

O conetivo "*e*" é a forma mais utilizada de se formarem proposições compostas através da **conjunção lógica de proposições**. Este conetivo é usado, por exemplo, em sentenças como "gatos são mamíferos e canários são aves", "$3 < 5$ e $2 + 3 = 5$" etc. A conjunção também pode ser representada por preposições como "mas", "também" e similares.

O símbolo $\wedge$ é usado para representar formalmente a conjunção lógica. Caso A e B sejam proposições, então a fórmula $A \wedge B$ representa a conjunção lógica das proposições A e B. Essa fórmula é lida simplesmente como "A e B".

Exercício 1.1

Agora responda as seguintes questões:

- Se A é verdadeira e B verdadeira, que valor você atribuiria a $A \wedge B$?
- Se A é verdadeira e B falsa, que valor você atribuiria a $A \wedge B$?
- Se A é falsa e B verdadeira, que valor você atribuiria a $A \wedge B$?
- Se A e B são falsas, que valor você atribuiria a $A \wedge B$?
- Construa uma tabela resumindo o resultado das questões (a) até (d). Use V para verdadeiro e F para falso. Mostre em cada linha da tabela a combinação de valores de A, B e de $A \wedge B$.

A tabela referente às questões acima, representada a seguir, é chamada de **tabela-verdade** do conetivo (ou operador) lógico da **conjunção (∧)**.

A	**B**	**A ∧ B**
V	*V*	*V*
V	*F*	*F*
F	*V*	*F*
F	*F*	*F*

Dica: Uma **conjunção** somente é **verdadeira** quando todas as proposições que a compõem são **verdadeiras**, e é falsa em todos os outros casos.

Disjunção de proposições

O símbolo ∨ será empregado para representar um dos significados usuais do conetivo "ou" em frases da linguagem natural. O significado assumido por este símbolo é o do "ou inclusivo" que somente será falso se ambas as sentenças sendo conectadas por ele forem falsas, isto é, A ∨ B será falso somente se A e B forem falsos. Diz-se que o símbolo ∨ representa a disjunção lógica das proposições A e B. A fórmula A ∨ B é lida como "A ou B".

Exercício 1.1.2

Construa a tabela-verdade do operador ∨.

Dica: Uma **disjunção** somente é **falsa** quando todas as proposições que a compõem são **falsas**, e é verdadeira em todos os outros casos.

Negação de uma proposição

Os símbolos ¬ ou ~ serão usados para representar a **negação** de uma proposição. Neste caso, se A é uma proposição verdadeira, então ¬A ou ~A será uma proposição falsa e vice-versa. Ou seja ¬A é a **negação lógica** de A (as vezes o símbolo ‘ (apóstrofo) também é usado para simbolizar a negação). A fórmula ¬A é lida simplesmente como "Não A" ou "Não é verdade que A".

Exercício 1.1.3

Construa a tabela da negação lógica.

1.2 Implicação e bi-implicação

Implicação

O símbolo → será usado para representar sentenças como “se chover, então a rua ficará molhada”, ou então “não estudar implica tirar notas baixas” ou também “não fui ao cinema porque o carro estragou” e sentenças similares. Geralmente, essas sentenças podem ser reescritas no formato “Se sentença A, então sentença B”, que simbolicamente fica apenas: A → B.

A noção que esse operador lógico pretende capturar é a de existência de implicação ou de consequência entre as sentenças. Dessa forma, a sentença B não poderia ser falsa se a sentença A fosse verdadeira, isto é, voltando aos exemplos, não faria sentido afirmar “se chover, então a rua ficará molhada” se (A) realmente choveu e (B) a rua não ficou molhada (!?). Isso significa que se considera que a sentença simbolizada por A→B seria falsa somente no caso em que A é verdadeira e B falsa. Nos outros casos a expressão A→B seria verdadeira.

O símbolo → é o operador de **implicação** (também denominado de **condicional**). Quando há chance de conflito com a relação de implicação lógica, esse operador também pode ser denominado **implicação material**.

A fórmula A → B é lida como "A implica em B" ou "se A então B". Neste caso a proposição A é denominada **condição** (ou **antecedente**) da implicação e a proposição B é denominada **conclusão** (ou

consequente) dessa implicação.

Nota: quando a sentença A na fórmula A → B é falsa, então o resultado de toda a fórmula A → B é verdadeiro independente de B. Isso simplesmente indica que, se assumimos falsidades como verdades, então qualquer coisa é possível.

Com base na discussão acima, a construção da tabela-verdade de A→B é representada a seguir.

A	**B**	**A → B**
V	*V*	*V*
V	*F*	F
F	*V*	V
F	*F*	*V*

Dica: Uma **implicação material** A→B somente é **falsa** quando a condição **A** for **verdadeira** e a conclusão **B** for **falsa**. Ela é verdadeira em todos os outros casos.

Bi-implicação ou equivalência

A operação de **bi-implicação** ou de **equivalência** (também denominada **bicondicional**) é representada pelo símbolo ↔ e verifica se duas proposições têm o mesmo valor lógico.

O uso de bi-implicações não é comum na linguagem natural, mas existe uma forma tradicional de se traduzir (transliterar) o operador ↔ para a linguagem natural, através da expressão *"se e somente se"*, assim a fórmula A↔B pode ser lida como "A bi-implica em B" ou "A se e somente se B".

A operação A↔B pode ser considerada uma abreviação da seguinte fórmula (A→B) ∧ (B→A), afinal ela é uma "bi-implicação", ou seja, uma implicação dupla. Com essa informação a construção da tabela-verdade de A↔B está representada na sequência.

A	**B**	**A ↔ B**

V	*V*	*V*
V	*F*	*F*
F	*V*	*F*
F	*F*	*V*

> **Dica**: Uma bi-implicação material A↔B é verdadeira quando A = B e falsa caso A ≠ B.

1.3 Fórmulas e precedência

Uma fórmula é construída pela composição de símbolos de sentenças simples (A, B, ...) e de conetivos lógicos binários (∧ , ∨ , → e ↔) e unários (¬, ~). Também podem ser usados parênteses. A precedência usual é:

1. Fórmulas dentro de parênteses (os mais internos primeiro)
2. ¬, ~ (a negação)
3. ∧ (conjunção)
4. ∨ (disjunção)
5. → (implicação material)
6. ↔ (bi-implicação ou equivalência lógica)
7. Da esquerda para a direita: ∧ , ∨
8. Da direita para a esquerda: → , ↔

Uma fórmula que não tenha erro de sintaxe em sua escrita (por exemplo, não tenha excesso nem falta de parênteses, operadores ou símbolos estranhos etc.) é chamada de "fórmula bem-formada" (ou fbf). Na literatura também é muito usado o termo em inglês "*well-formed formula*" (wff).

Neste livro, entretanto, quando nos referirmos a uma *fórmula*, estaremos assumindo que ela é bem-formada. Exceções são indicadas no texto.

Em termos de sintaxe, fórmula é uma estrutura que deve ser montada pelas seguintes regras de formação:

1. Qualquer sentença simples P, Q, R, ..., P_1, P_2, ... é uma fórmula.
2. Se P é uma fórmula, então $\neg P$ também é.
3. Se P e Q são fórmulas, então $P \land Q$, $P \lor Q$, $P \rightarrow Q$ e $P \leftrightarrow Q$ também são fórmulas.

Qualquer coisa que não possa ser montada pelas regras acima **não** é considerada uma fórmula. Essas regras mostram como fórmulas **compostas** ou **complexas** são construídas a partir das fórmulas simples, por aplicações repetidas das regras de formação. As fórmulas que entram na composição de uma fórmula complexa são sub-fórmulas desta fórmula.

Exemplos:

$$\neg (A \land B)$$

Pela regra 1 vemos que A e B são fórmulas simples. Vamos chamar a sentença simples A de fórmula P e a sentença simples B de fórmula Q. Seguindo pela regra 3 vemos que $P \land Q$ é uma fórmula e, portanto $A \land B$ também é uma fórmula. Para finalizar, basta chamar esta fórmula $A \land B$ récem formada de P, então, pela regra 2, $\neg P$ é uma fórmula e consequentemente $\neg (A \land B)$ é uma fórmula corretamente montada (formada) pelas regras de formação.

$$\neg B \land (C \rightarrow D)$$

Também podemos utilizar as regras de formação para "desmontar" uma fórmula, verificando enquanto isso se a fórmula foi corretamente formada. Se dividirmos a fórmula acima em duas partes, denominando $\neg B$ de P e $C \rightarrow D$ de Q, então podemos ver que a fórmula acima foi formada pela regra 3: $P \land Q$. Agora falta verificar se as partes P e Q são realmente fórmulas bem-formadas. No caso de P que é igual a $\neg B$ é fácil de ver que ela pode ser corretamente montada pela regra 1 para tanto basta considerar a sentença B como a fórmula P dessa regra, resultando em $\neg P$ e portanto em $\neg B$. O caso da parte Q que é igual a $C \rightarrow D$ é similar. Basta usar a regra 3 de novo, mas desta vez chamando a sentença C de fórmula P e a sentença D de fórmula Q, resultando em $P \rightarrow Q$ e então em $C \rightarrow D$. Não há mais o que "desmontar", portanto a fórmula $\neg B \land (C \rightarrow D)$ foi inteiramente desmontada pelas regras de formação, mostrando que ela é uma fórmula bem-formada.

Supondo que A, B e C são proposições lógicas, então as seguintes expressões são fórmulas:

1. $(A \rightarrow B) \leftrightarrow (B \rightarrow A)$
2. $(A \lor \neg A) \rightarrow (B \land \neg B)$
3. $(A \land \neg B) \rightarrow \neg C$
4. $(A \rightarrow B) \leftrightarrow B \rightarrow A$
5. $((A \land B \land C) \lor \neg(\neg B \text{ v } A) \text{ v } (A \land \neg C)) \rightarrow (C \text{ v } \neg A)$
6. $\neg\neg(A \land C)$

Contraexemplos (erros mais comuns):

As seguintes fórmulas apresentam alguns erros bastante comuns de escrita (sintaxe) que podem ocorrer nas fórmulas (do lado é indicado o tipo de erro):

1. $(A \rightarrow B \leftrightarrow (B \rightarrow A)$ falta um fecha parênteses
2. $(A \lor \neg) \rightarrow (B \land \neg B)$ a primeira negação não está seguida de uma proposição
3. $\neg((A \neg B) \rightarrow \neg C$ falta um operador lógico binário entre A e $\neg B$
4. $(\text{v } \neg A) \rightarrow (B \land \neg B)$ falta uma proposição no lado esquerdo do operador v
5. $((A\ B \land C) \lor \neg(\neg\text{v } A) \text{ v } (C \text{ v } \neg A)$ falta um operador lógico binário entre A e B e também falta uma proposição depois do operador de negação antes do operador v

1.4 Construção de tabelas-verdade para fórmulas

Uma tabela-verdade mostra, em suas colunas mais à esquerda, todas as combinações de valores lógicos que as proposições de uma determinada fórmula podem assumir. A partir desses valores de entrada pode-se “calcular” os valores que essa fórmula irá ter para cada uma dessas combinações de valores. Esse cálculo é feito passo a passo, criando-se colunas intermediárias que ficam posicionadas à direita das colunas de entrada e que contêm os valores das subfórmulas que compõem a fórmula principal. Na última coluna mais à direita se coloca a coluna que contém os valores finais dessa fórmula. Resumindo, para se construir a tabela-verdade de uma fórmula lógica pode-se seguir os passos:

(i) nas colunas à esquerda coloque os símbolos sentenciais simples (A, B, ...), depois,
(ii) se houver sentenças simples negadas (¬A, ¬B, ...), coloque-as nas próximas colunas e, por fim,
(iii) seguindo a precedência, crie uma coluna para cada fórmula composta (não é necessário repetir as sentenças simples negadas).

A última coluna à direita deve ser a expressão ou fórmula final.

As sentenças ou símbolos proposicionais simples pertencentes a uma fórmula definem o número de linhas da tabela-verdade para esta fórmula através de uma regra simples:

1 símbolo:A	2 linhas (2^1 combinações: V e F)
2 símbolos: A e B	4 linhas (2^2 combinações: VV, VF, FV, FF)
3 símbolos: A, B e C	8 linhas (2^3 combinações: VVV, VVF, VFV, VFF, FVV, FVF, FFV, FFF)
4 símbolos: A, B, C e D	16 linhas (2^4 combinações)
n símbolos: A, B, ...	2^n linhas (2^n combinações)

A última linha da tabela acima define a regra: para n símbolos proposicionais simples devem existir 2^n linhas na tabela para representar as 2^n combinações de valores-verdade possíveis.

Exemplo:

O operador de disjunção $\lor$ é aplicado sobre duas proposições A $\lor$ B. A tabela-verdade desse operador, usando V para indicar verdadeiro e F para indicar falso (que deveria ter sido construída no exercício 1.1.), é igual a:

A	**B**	**A $\lor$ B**
V	*V*	*V*
V	*F*	*V*
F	*V*	*V*
F	*F*	*F*

Outra forma de representar verdadeiro/falso é através de valores numéricos, 0 significa falso e 1 significa verdadeiro (esta é a forma mais comum usada em álgebra booleana e em circuitos lógicos). Usando essa notação, a tabela acima ficaria:

A	**B**	**A $\lor$ B**
0	0	0
0	1	1
1	0	1
1	1	1

Note que, quando se usa 0 e 1, a disposição dos valores verdadeiros e falsos muda. No caso de se usar V e F, geralmente se começa com a linha superior toda em V e as demais linhas vão aos poucos sendo preenchidas com F até que na linha inferior todos os valores são F. No caso de se usar 0 e 1 a disposição é exatamente contrária. O reflexo dessas diferentes disposições aparece claramente na última coluna.

Neste texto somente será usada a primeira representação, com V para verdadeiro e F para falso, que é a forma usual da lógica proposicional.

Exercício 1.4.1

Agora construa tabelas-verdade para as seguintes fórmulas:

1. $(A \rightarrow B) \leftrightarrow (B \rightarrow A)$
2. $(A \vee \neg A) \rightarrow (B \wedge \neg B)$
3. $\neg((A \wedge \neg B) \rightarrow \neg C)$
4. $(A \rightarrow B) \leftrightarrow (\neg B \rightarrow \neg A)$
5. $((A \wedge B \wedge C) \vee \neg\neg(\neg B \vee A) \vee (A \wedge \neg C)) \rightarrow (C \vee \neg A)$

1.5 Tautologias e contradições

Uma **tautologia** é uma fórmula que assume apenas o valor V, ou seja, que é sempre verdadeira. Uma tautologia é “intrinsecamente verdadeira” pela sua própria estrutura; ela é verdadeira independente de qualquer valor lógico atribuído às suas letras de proposição.

Uma **contradição** é o oposto de uma tautologia, ou seja, é uma fórmula que assume apenas o valor F independente de qualquer combinação de valores-verdade atribuída às proposições lógicas simples que entram em sua composição.

No caso da lógica proposicional, para demonstrar que uma fórmula é uma tautologia ou uma contradição basta construir sua tabela-verdade.

A questão $(A \rightarrow B) \leftrightarrow (\neg B \rightarrow \neg A)$ apresentada acima é um exemplo de tautologia (basta conferir sua tabela-verdade).

Exercício 1.5.1

Descobrir quais das seguintes fórmulas são tautologias, contradições ou fórmulas contingentes (fórmulas “simples” que não são tautologias ou contradições).

1. $(A \vee B) \leftrightarrow (B \vee A)$
2. $((A \vee B) \vee C) \leftrightarrow (A \vee (B \vee C))$
3. $\neg(A \wedge B) \leftrightarrow (\neg A \wedge \neg B)$
4. $(A \rightarrow B) \leftrightarrow (\neg B \rightarrow \neg A)$
5. $((A \wedge B) \wedge B) \rightarrow \neg(\neg B \vee A) \wedge \neg A)$
6. $\neg(\neg A \leftrightarrow \neg B) \rightarrow (B \wedge (C \vee \neg A))$

1.6 Equivalências tautológicas e Leis de DeMorgan

Equivalências tautológicas

Considere que P e Q sejam duas fórmulas lógicas quaisquer e que $P \leftrightarrow Q$ seja uma tautologia, então pela própria definição do conetivo $\leftrightarrow$, sempre que P for V em uma linha da tabela-verdade de $P \leftrightarrow Q$, a fórmula Q também deverá ser V nesta linha. O mesmo acontece para quando P tem valor F. Neste caso se diz que P e Q são **fórmulas equivalentes**.

Essa propriedade é denotada pelo operador $\Leftrightarrow$ de equivalência tautológica entre as fórmulas P e Q, simbolicamente fica $P \Leftrightarrow Q$.

Na tabela a seguir são apresentadas algumas equivalências tautológicas que definem propriedades importantes da disjunção e conjunção:

Tabela 1 – Equivalências da disjunção ($\vee$) e da conjunção ($\wedge$)

Propriedade	**Disjunção ($\vee$)**	**Conjunção ($\wedge$)**
Comutativa	$A \vee B \Leftrightarrow B \vee A$	$A \wedge B \Leftrightarrow B \wedge A$
Associativa	$(A \vee B) \vee C \Leftrightarrow A \vee (B \vee C)$	$(A \wedge B) \wedge C \Leftrightarrow A \wedge (B \wedge C)$
Distributiva	$A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$	$A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$
Elemento neutro	$A \vee F \Leftrightarrow A$	$A \wedge V \Leftrightarrow A$
Complemento	$A \vee \neg A \Leftrightarrow V$	$A \wedge \neg A \Leftrightarrow F$
Idempotência	$A \vee A \Leftrightarrow A$	$A \wedge A \Leftrightarrow A$

Fonte: elaborada pelos autores.

Na tabela a seguir são apresentadas algumas equivalências tautológicas que permitem reescrever ou redefinir os outros operadores:

Tabela 2 – Equivalências dos outros operadores

Dupla negação	$\neg\neg A \Leftrightarrow A$
Equivalência da implicação	$A \rightarrow B \Leftrightarrow \neg A \vee B$
Contraposição	$A \rightarrow B \Leftrightarrow \neg B \rightarrow \neg A$
Prova condicional	$A \rightarrow (B \rightarrow C) \Leftrightarrow (A \wedge B) \rightarrow C$

Fonte: elaborada pelos autores.

Exercício 1.6.1

Demonstrar, pelo uso da tabela-verdade, as equivalências tautológicas acima (não é necessário repetir as demonstrações para a equivalência comutativa, associativa e contraposição).

Leis de De Morgan

As equivalências vistas anteriormente permitem efetuar vários tipos de manipulações ou alterações em uma fórmula sem que ela altere seu significado. Além dessas fórmulas, entretanto, seria interessante que houvesse maneiras de se converter proposições conectadas pelo operador $\vee$ em proposições conectadas por $\wedge$. Essas equivalências são denominadas Leis de DeMorgan em homenagem ao matemático inglês do século XIX Augustus DeMorgan, que foi o primeiro a enunciá-las.

	Negação da disjunção	Negação da conjunção
DeMorgan	$\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$	$\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$

1.7 Simbolização de proposições

A seguir são apresentados alguns exemplos tanto de passagem de uma fórmula para texto quanto de texto para fórmula.

Supondo que A = "hoje está chovendo", B = "hoje faz frio" e C = "hoje vamos para a praia", então a fórmula:

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

Poderia ser versada para o português de diferentes maneiras:

Hoje está chovendo, hoje faz frio e não é verdade que hoje vamos para a praia.
Hoje está chovendo, faz frio e não vamos para a praia.
Hoje está chovendo e faz frio, todavia não vamos para a praia.
Hoje está chovendo e faz frio, mas não vamos para a praia hoje.

A frase (a) que é uma transliteração direta da fórmula $A \wedge B \wedge \neg C$, embora correta do ponto de vista gramatical, simplesmente não é usada pelo excessivo formalismo. As demais formas são mais usuais (e existem muitas outras equivalentes). Note que a negação quase nunca é colocada no início da proposição. Note também que são usadas várias formas de conjunções além do conetivo “e”: é usada a vírgula no caso (b) e as conjunções adversativas “todavia” e “mas” nos casos (c) e (d).

Usando os mesmos significados para os símbolos A, B e C, a fórmula:

$$(A \land B) \lor C$$

Poderia ser versada para o português das seguintes formas:

Hoje está chovendo e faz frio, ou hoje vamos para a praia.
Ou hoje está chovendo e faz frio, ou hoje vamos para a praia.
Ou hoje está chovendo e faz frio, ou então hoje vamos para a praia.

A fórmula:

$$\neg(A \land B)$$

é um pouco mais complexa de ser versada, por conta da negação de uma subfórmula em parênteses, algo simplesmente não usado na linguagem natural. A fórmula poderia ser transliterada no texto:

Não é verdade que hoje está chovendo e fazendo frio.

Mas este texto é ambíguo, podendo representar a fórmula $\neg A \land B$. Uma representação mais razoável da fórmula $\neg(A \land B)$ exigiria a retirada dos parênteses, algo que somente pode ser feito pelo uso das Leis de DeMorgan. Neste caso a fórmula se torna:

$$\neg A \lor \neg B$$

facilmente versada para:

Hoje não está chovendo ou hoje não está fazendo frio.

A seguir são apresentados exemplos de texto, com sua simbolização equivalente:

Se hoje está chovendo, então ou vamos para a praia, ou está fazendo frio.

$$A \rightarrow (C \vee B)$$

Hoje não está chovendo, nem fazendo frio, portanto vamos para a praia.

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

Hoje não vamos para a praia, porque está chovendo e fazendo frio.

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

1.8 Expressões em português e operadores lógicos

Devido à riqueza da língua portuguesa, palavras com significados semelhantes são representadas pelo mesmo operador lógico. A tabela a seguir mostra expressões comuns em português associadas a diversos operadores lógicos.

Tabela 3 – Expressões em português

Expressão em português	Operador lógico	Expressão lógica
E; mas; também	Conjunção	$A \wedge B$
Ou	Disjunção	$A \vee B$
Se A, então B A implica B A, logo B A somente se B	Implicação/ Condicional	$A \rightarrow B$
A se e somente se B	Bi-implicação/ Bicondicional	$A \leftrightarrow B$
Não A É falso que A... Não é verdade que A...	Negação	$\neg A$

Fonte: elaborada pelos autores.

1.9 Questões comentadas

Nas questões a seguir, é possível verificar a construção da tabela-verdade para três fórmulas. É importante lembrar dos passos para construção da tabela:

(i) colocar nas colunas à esquerda as proposições simples;

(ii) se houver proposições simples negadas, colocar nas colunas seguintes e

(iii) seguindo a precedência, crie uma coluna para cada fórmula composta.

Questão 1. (A → B) ↔ (B → A)

Neste exemplo, não temos o passo (ii).

A	**B**	**A → B**	**B → A**	**(A → B) ↔ (B → A)**
V	*V*	*V*	*V*	*V*
V	*F*	F	*V*	F
F	*V*	*V*	F	F
F	*F*	V	*V*	*V*

Questão 2. ¬((A ∧ ¬B) → ¬C)

Neste exemplo, é importante observar a precedência dos operadores para construir as colunas do passo (iii).

A	**B**	**C**	**¬B**	**¬C**	**A ∧ ¬B**	**(A ∧ ¬B) → ¬C**	**¬((A ∧ ¬B) → ¬C)**
V	*V*	*V*	*F*	*F*	*F*	*V*	*F*
V	*V*	*F*	*F*	*V*	*F*	*V*	*F*
V	*F*	*V*	*V*	*F*	*V*	*F*	*V*
V	*F*	*F*	*V*	*V*	*V*	*V*	*F*
F	*V*	*V*	*F*	*F*	*F*	*V*	*F*

F	*V*	*F*	*F*	*V*	*F*	*V*	*F*
F	*F*	*V*	*V*	*F*	*F*	*V*	*F*
F	*F*	*F*	*V*	*V*	*F*	*V*	*F*

Questão 3. ((A ∧ B ∧ C) ∨ ¬¬(¬B v A) v (A ∧ ¬C)) → (C v ¬A)

Como a construção da tabela-verdade dessa fórmula tem várias colunas, considerando que cada coluna tem uma fórmula com muitas proposições e operadores, então para auxiliar na representação da fórmula é utilizado uma legenda como segue, o número da coluna representa a fórmula abaixo:

1. (A ∧ B) ∧ C
2. (A ∧ B ∧ C) ∨ ¬¬(¬B v A)
3. (A ∧ B ∧ C) ∨ ¬¬(¬B v A) v (A ∧ ¬C)
4. ((A ∧ B ∧ C) ∨ ¬¬(¬B v A) v (A ∧ ¬C)) → (C v ¬A)

A	**B**	**C**	**¬A**	**¬B**	**¬C**	**A ∧ B**	**1.**	**A ∧ ¬C**	**¬B v A**	**¬(¬B v A)**	**¬¬(¬B v A)**	**C v ¬A**	**2.**	**3.**	**4.**
V	*V*	*V*	*F*	*F*	*F*	*V*	*V*	*F*	*V*	*F*	*V*	*V*	*V*	V	*V*
V	*V*	*F*	*F*	*F*	*V*	*V*	*F*	*V*	*V*	*F*	*V*	*F*	*F*	V	*F*
V	*F*	*V*	*F*	*V*	*F*	*F*	*F*	*F*	*V*	*F*	*V*	*V*	*V*	*V*	*V*
V	*F*	*F*	*F*	*V*	*V*	*F*	*F*	*V*	*V*	*F*	*V*	*F*	*F*	*V*	*F*
F	*V*	*V*	*V*	*F*	*F*	*F*	*F*	*F*	*F*	*V*	*F*	*V*	*V*	*V*	*V*
F	*V*	*F*	*V*	*F*	*V*	*F*	*F*	*F*	*F*	*V*	*F*	*V*	*V*	*V*	*V*
F	*F*	*V*	*V*	*V*	*F*	*F*	*F*	*F*	*V*	*F*	*V*	*V*	*V*	*V*	*V*
F	*F*	*F*	*V*	*V*	*V*	*F*	*F*	*F*	*V*	*F*	*V*	*V*	*V*	*V*	*V*

CAPÍTULO 2

DEDUÇÃO NA LÓGICA PROPOSICIONAL

As definições vistas até agora nos permitiram criar uma linguagem formal para a Lógica Proposicional. Essas definições também nos permitiram ver como se pode descobrir o valor-verdade de expressões nessas linguagens através de tabelas-verdade. Porém isso não é tudo que uma linguagem lógica pode nos fornecer. Ainda é necessário definir como são feitos raciocínios ou argumentações nessa linguagem. A lógica formal lida com um tipo particular de argumento, denominado *argumento dedutivo*, que nos permite deduzir uma conclusão Q, com base em um conjunto de proposições P_1 a P_n, onde Q e P_1 a P_n representam fórmulas inteiras bem-formadas da Lógica Proposicional (e não apenas proposições simples).

2.1 Argumentos

Quando afirmamos que uma proposição lógica é verdadeira em razão de (ou em consequência) de outras proposições lógicas também serem verdadeiras, estamos, na verdade, expressando um argumento lógico, ou simplesmente, um **argumento**.

Um argumento pode ser sempre verdadeiro, quando a proposição que é a **conclusão** do argumento sempre for verdadeira e quando as demais proposições que compõem o argumento (chamadas de **hipóteses** ou **premissas**) forem verdadeiras. Neste caso, temos um argumento **válido** que poderá ser utilizado para fazer raciocínios lógicos corretos. Por outro lado, nem sempre argumentos são válidos. Neste capítulo vamos estudar métodos para garantir ou verificar que um argumento é válido.

Um argumento pode ser representado de forma simbólica pela seguinte fórmula da Lógica Proposicional:

$$P_1 \land P_2 \land P_3 \land \ldots \land P_n \rightarrow Q$$

As proposições P_1 a P_n são as hipóteses ou premissas do argumento. A proposição Q é a conclusão do argumento. Em termos da Linguagem Natural esse tipo de simbolismo pode ser lido como:

"P_1, P_2, ... P_n **acarretam** Q" ou
"Q **decorre** de P_1, P_2, ... P_n" ou
"Q se **deduz** de P_1, P_2, ... P_n" ou ainda

“Q se **infere** de P_1, P_2, ... P_n”

Uma interpretação ingênua do argumento acima poderia pressupor que se houver uma situação que as premissas são verdadeiras e a conclusão também é verdadeira, isso seria suficiente para considerar o argumento como correto ou apropriado. Porém isso não é suficiente: a conclusão Q somente poderá ser considerada uma *conclusão lógica* das hipóteses P_1, P_2, ... P_n se e somente se a verdade das proposições P_1, P_2, ... P_n implicar na verdade Q, ou seja, apenas quando a fórmula.

$$P_1 \land P_2 \land P_3 \land \ldots \land P_n \to Q$$

for sempre verdadeira (for uma tautologia).

O problema com a interpretação ingênua é que ela poderia assumir como válido um argumento como:

$$A \land B \to C$$

onde A representa “um dia tem 24 horas”, B representa “bananas são frutas” e C representa “hoje é depois de ontem”. Embora as três sentenças sejam verdadeiras e portanto, neste caso, $A \land B \to C$ seja verdadeiro, não existe uma relação formal (ou real) entre elas e portanto não se pode dizer que um argumento na forma tão genérica quanto $A \land B \to C$ seja sempre válido, ou seja, verdadeiro independentemente do valor verdade das hipóteses ou da conclusão, mas em função apenas da sua forma.

Assim podemos definir que um **argumento válido** é um argumento onde a fórmula:

$$P_1 \land P_2 \land P_3 \land \ldots \land P_n \to Q$$

é uma tautologia.

O fato do argumento $P_1 \land P_2 \land P_3 \land \ldots \land P_n \to Q$ ser válido é simbolizado pela seguinte expressão:

$$P_1, P_2, P_3, \ldots, P_n \vdash Q$$

Esta expressão afirma que a fórmula Q é logicamente implicada pelas hipóteses P_1, P_2, P_3, ..., P_n, ou seja, há uma **relação de**

consequência lógica entre as hipóteses e a conclusão do argumento.

Em um argumento válido não interessam os valores-verdade das hipóteses nem da conclusão, porque somente a forma do argumento é capaz de garantir sua validade. Por isso ele é denominado **argumento formal** e esta é a razão por trás do poder de dedução da lógica formal, que pode verificar a validade ou correção de um argumento sem se ater às proposições que o compõem, isto é, sem se importar com seu significado.

Nota: às vezes pode-se encontrar na literatura o símbolo $\vdash$ em vez de $\vDash$ para representar a conclusão do argumento. Este símbolo, que será apresentado mais adiante, serve para indicar que existe uma *prova* ou *demonstração* do argumento.

2.2 Tabela-verdade para argumentos

Um argumento é válido se e somente se todas as suas instâncias são válidas. Uma instância é válida se todas as premissas forem verdadeiras e a conclusão também for verdadeira. A validade de um argumento da Lógica Proposicional pode ser testada através de uma versão simplificada da tabela-verdade, onde são colocadas em colunas separadas todas as premissas do argumento e na coluna final é colocada a conclusão do argumemto. Agora, cada uma das linhas dessa tabela mostra uma das instâncias possíveis do argumento. Assim, o argumento será válido se e somente se nas linhas onde todas as premissas são verdadeiras, então a conclusão também é verdadeira. Caso contrário, existirá nessa tabela-verdade pelo menos uma instância em que as premissas são verdadeiras e a conclusão é falsa, mostrando, então, que o argumento é inválido. Essa situação chama-se **contra-exemplo**. A existência de um contra-exemplo na tabela-verdade mostra que a forma do argumento é inválida.

Podemos nos basear na construção de tabelas-verdade para análise sintática (gramatical) para verificarmos se um argumento é válido. Buscamos examinar a sintaxe do cálculo proposicional mostrando como as formas de várias sentenças podem ser expressas como fórmulas (simbolização das sentenças).

Para determinar se a forma de um argumento da lógica proposicional é válida, vamos construir uma tabela-verdade para a expressão: $P \lor Q, \neg P \vdash Q$.

Tabela 4 – Tabela-verdade do argumento : $P \lor Q, \neg P \vdash Q$

	P	Q	¬P	P ∨ Q	Q
1	V	V	F	V	V
2	V	F	F	V	F
3	F	V	V	V	V
4	F	F	V	F	F

Fonte: elaborada pelos autores.

Esta tabela-verdade serve para testar a validade do argumento, sendo apenas uma versão simplificada da tabela-verdade da fórmula $((P \lor Q) \land \neg P) \rightarrow Q$. Nela podemos verificar que a forma do argumento é válida, pois onde as premissas são verdadeiras a conclusão também é verdadeira (destacado na linha 3). Assim, não há situação possível onde as premissas são verdadeiras e a conclusão é falsa, não importa qual a combinação de V e F atribuída a cada hipótese.

Outro exemplo de argumento onde vamos construir a tabela-verdade para verificarmos se sua forma é válida: $P \lor Q, Q \rightarrow \neg P \vdash Q \land P$.

Tabela 5 – Tabela-verdade do argumento: $P \lor Q, Q \rightarrow \neg P \vdash Q \land P$

	P	Q	¬P	P ∨ Q	Q → ¬P	Q ∧ P
1	V	V	F	V	F	V
2	V	F	F	V	V	F
3	F	V	V	V	V	F
4	F	F	V	F	V	F

Fonte: elaborada pelos autores.

Neste caso verificamos que nas linhas 2 e 3 as hipóteses são verdadeiras e a conclusão é falsa. Portanto, o argumento é inválido.

Se na construção da tabela-verdade verificarmos mais de uma instância com todas as premissas verdadeiras, então teremos que avaliar todas elas.

No argumento $P \rightarrow \neg Q, \neg Q \vdash P$ temos as linhas 2 e 4 em destaque para verificar. Na linha 2, as hipóteses e a conclusão são verdadeiras.

Com esse resultado, o argumento seria válido, entretanto, é necessário avaliar a linha 4 também. Na linha 4 temos o mesmo caso do exemplo anterior, as hipóteses são verdadeiras e a conclusão é falsa. Portanto, o argumento é inválido.

Tabela 6 – Tabela-verdade do argumento: $P \rightarrow \neg Q, \neg Q \vdash P$

	P	Q	¬Q	P → ¬Q	P
1	V	V	F	F	V
2	V	F	V	V	V
3	F	V	F	V	F
4	F	F	V	V	F

Fonte: elaborada pelos autores.

2.3 Demonstrações formais

Conforme visto anteriormente, para testar se o argumento $P_1 \wedge P_2 \wedge P_3 \wedge \ldots \wedge P_n \rightarrow Q$ é válido, poderíamos simplesmente construir a tabela-verdade correspondente ao argumento. Há, entretanto, um outro método para testar a validade de um argumento que é totalmente independente das tabelas-verdade. Este método é baseado na aplicação de **regras de dedução** (ou **regras de inferência**) que modificam fórmulas de modo a preservar seu valor lógico.

A ideia básica desse método é começar com as hipóteses $H_1, H_2, \ldots H_n$ (supostamente verdadeiras) e tentar aplicar regras de dedução até terminar com a conclusão Q. Essa conclusão teria que ser, então, verdadeira uma vez que o valor lógico verdadeiro é sempre preservado sob as regras de inferência.

Dessa forma, uma **prova** ou **demonstração** formal da lógica proposicional seria formada por uma série de linhas numeradas denominadas **passos** da prova, onde cada passo tem a seguinte estrutura:

1. H_1 Hipótese 1

2.	H_2	Hipótese 2
...		
n.	H_n	Hipótese n
n+1.	F1	Fórmula obtida aplicando uma regra sobre fórmulas dos passos anteriores
n+2.	F2	Fórmula obtida aplicando uma regra sobre fórmulas dos passos anteriores
...		
n+m.	Fm	Fórmula obtida aplicando uma regra sobre fórmulas dos passos anteriores
	Q	Fórmula obtida aplicando uma regra sobre fórmulas dos passos anteriores

Nesta demonstração, a conclusão Q é simplesmente a fórmula contida no último passo, que foi obtida através da aplicação de uma **regra de dedução** (também chamada de **regra de inferência**) sobre fórmulas contidas em passos anteriores. A própria forma da regra indica como ela pode ser utilizada e quais passos poderiam ser úteis. Os passos que foram efetivamente usados na aplicação da regra devem ser referenciados no fim do passo, para que se possa verificar se a regra foi utilizada corretamente.

A sequência de passos obtida por este processo é denominada **sequência de demonstração** da conclusão em função de suas hipóteses. Essa sequência é a **prova** (ou **demonstração**) que a conclusão Q se deduz das hipóteses H_1, H_2, ..., H_n. Nesse caso, provar um argumento $H_1 \land H_2 \land H_3 \land ... \land H_n \rightarrow Q$ é simbolizado através da seguinte expressão:

$$H_1, H_2, H_3, ..., H_n \vdash Q$$

Esta expressão afirma que a conclusão Q pode ser deduzida das hipóteses H_1, H_2, H_3, ..., H_n, ou seja, que existe uma **relação de dedução** entre as hipóteses e a conclusão do argumento.

Nota: Embora pareça muito mais simples aplicar o método de construção da tabela-verdade para verificar a validade de um argumento, o método da demonstração formal se justifica por duas razões:

(i) Quando o número de proposições simples é muito grande, por exemplo, com apenas 40 proposições simples seria necessária uma tabela-verdade com aproximadamente 1 trilhão de linhas.
(ii) No caso das lógicas mais expressivas como a lógica de predicados, simplesmente não é possível aplicar o método da tabela-verdade, ou seja, somente nos resta aplicar o método da demonstração formal.

Na verdade, na Lógica Proposicional é possível provar que qualquer um dos dois métodos, tabela-verdade ou demonstração formal, pode ser usado para comprovar a validade de um argumento, ou seja, sempre que a tabela-verdade mostrar que o argumento é válido, então haverá uma demonstração formal deste argumento e se houver uma demonstração formal de um argumento, então sua tabela-verdade irá mostrar que este argumento é válido.

2.4 Regras de dedução natural

O método de dedução utilizado neste livro é denominado Dedução Natural, por ser considerado um método que se aproxima da forma como deduções lógicas são feitas naturalmente. Esse método foi desenvolvido na década de 1930 por Gentzen para ajudar nas provas de consistência da Teoria dos Números (NOLT, 1991). O método de dedução empregado neste livro é muito similar aos métodos de dedução formal apresentados em (NOLT, 1991) e (BARONETT, 2005).

Todas as regras de dedução se baseiam em argumentos simples que já se tenha demonstrado por tabela-verdade serem válidos. A Tabela 7 apresenta as regras **básicas** de inferência da lógica proposicional.

Tabela 7 – Regras básicas de inferência

Inclusão de Operadores	**Exclusão de Operadores**
Redução ao absurdo (**raa**) $\dfrac{\begin{array}{\|l} P \\ \ldots \\ Q \wedge \neg Q \end{array}}{\neg P}$	Dupla negação (**dn**) $\dfrac{\neg\neg P}{P}$
Prova condicional (**pc**) $\dfrac{\begin{array}{\|l} P \\ \ldots \\ Q \end{array}}{P \rightarrow Q}$	*Modus Ponens* (**mp**) $\dfrac{P \quad P \rightarrow Q}{Q}$
Conjunção(**cj**) $\dfrac{P \qquad Q}{P \wedge Q}$	Simplificação (**sp**) $\dfrac{P \wedge Q}{P} \qquad \dfrac{P \wedge Q}{Q}$
Adição(**ad**) $\dfrac{P}{P \vee Q} \qquad \dfrac{P}{Q \vee P}$	Eliminação da disjunção - **∨E** $\dfrac{P \vee Q \quad P \rightarrow R \quad Q \rightarrow R}{R}$
Introdução da equivalência - **↔I** $\dfrac{P \rightarrow Q \quad Q \rightarrow P}{P \leftrightarrow Q}$	Eliminação da equivalência - **↔E** $\dfrac{P \leftrightarrow Q}{P \rightarrow Q} \qquad \dfrac{P \leftrightarrow Q}{Q \rightarrow P}$

Fonte: elaborada pelos autores.

Tabela 8 – Regras de inferência derivadas

Modus Tollens (**mt**)	Silogismo Hipotético (**sh**)
$\dfrac{P \rightarrow Q \quad \neg Q}{\neg P}$	$\dfrac{P \rightarrow Q \; Q \rightarrow R}{P \rightarrow R}$
Silogismo Disjuntivo (**sd**)	Dilema Construtivo (**dc**)
$\dfrac{P \vee Q \quad \neg P}{Q}$	$\dfrac{P \vee Q \quad P \rightarrow R \quad Q \rightarrow S}{R \vee S}$
Exportação (**exp**)	Inconsistência (**inc**)
$\dfrac{(P \wedge Q) \rightarrow R}{P \rightarrow (Q \rightarrow R)}$	$\dfrac{P \quad \neg P}{Q}$

Fonte: elaborada pelos autores.

Exemplos de aplicação do método

Excetuando-se as regras de redução ao absurdo (**RAA**) e prova condicional (**PC**), todas as demais regras de inferência e de equivalência apresentadas nas Tabelas 7, 8 e 9 são de aplicação **direta**. Nesta seção serão mostrados exemplos de uso dessas regras de aplicação direta.

Supondo que $A \rightarrow (B \wedge C)$ e A são duas hipóteses de um argumento, então a seguinte demonstração é válida:

1.	$A \rightarrow (B \wedge C)$	hip
2.	A	hip
3.	$B \wedge C$	1,2, mp

As fórmulas das 2 primeiras linhas são inseridas por conta das hipóteses, enquanto a fórmula da linha 3 é derivada das fórmulas das linhas 1 e 2 pela regra *modus ponens*.

Usando a lógica proposicional, prova-se o seguinte argumento:

$$A,\ B \rightarrow C,\ (A \wedge B) \rightarrow (C \rightarrow D),\ B \vdash D$$

Primeiro as hipóteses do argumento:

1.	A	hip
2.	$B \rightarrow C$	hip
3.	$(A \land B) \rightarrow (C \rightarrow D)$	hip
4.	B	hip

Alguns passos óbvios (que poderão ser úteis ou não):

5.	C	2, 4, mp
6.	$A \land B$	1, 4, cj
7.	$C \rightarrow D$	3, 6, mp

E agora, portanto:

8.	D	5, 7, mp

De acordo com o método, para provar que o argumento é válido, inicialmente enumera-se as hipóteses. O argumento para provar é:

$$P \land Q,\ P \rightarrow R,\ (R \land S) \rightarrow \neg T,\ Q \rightarrow S \vdash \neg T$$

1.	$P \land Q$	hip
2.	$P \rightarrow R$	hip
3.	$(R \land S) \rightarrow \neg T$	hip
4.	$Q \rightarrow S$	hip

A partir disso, como sugestão verifica-se onde está a conclusão ou parte dela nas hipóteses. Assim é possível aplicar as regras gerando fórmulas até que a fómula gerada seja a conclusão. A conclusão do argumento é $\neg T$, então é possível identificar $\neg T$ na linha 3.

3. $(R \land S) \to \neg T$ hip

Também é importante salientar que para aplicar uma regra na linha 3, temos que identificar o principal operador lógico da fórmula, nesse caso, a implicação. A regra que elimina a implicação (→) é a Modus Ponens (mp).

Modus Ponens (**mp**)

$$\frac{P \quad P \to Q}{Q}$$

Para aplicar a regra, precisa-se de duas fórmulas conforme indica a estrutura apresentada, isto é, uma fórmula é P e a outra é P → Q. Como as fórmulas estão acima da linha horizontal, então já devem existir na demonstração e o que está abaixo é a fórmula que será gerada a partir da aplicação da regra mp. Com isso, voltando à demonstração do argumento, para aplicar a regra mp na linha 3, temos que encontrar a outra fórmula P. Nesse caso, P deve ser exatamente igual ao antecedente da fórmula da linha 3, então devemos gerar R ∧ S.

Da mesma forma que foi discutido anteriormente para a fórmula da linha 3, onde a implicação é o operador principal e deve ser aplicada a regra mp, agora a linha 2 precisa-se liberar R e o operador principal é uma implicação.

1.	$P \land Q$	hip
2.	$P \to R$	hip
3.	$(R \land S) \to \neg T$	hip
4.	$Q \to S$	hip
5.	**P**	**1, sp**

Aplicar a regra mp é possível, pois temos duas fórmulas conforme mostra a imagem, então a fórmula R na linha 6 é gerada a partir da aplicação da regra mp na linha 2 e na linha 5.

1.	$P \wedge Q$	hip
2.	$P \rightarrow R$	hip
3.	$(R \wedge S) \rightarrow \neg T$	hip
4.	$Q \rightarrow S$	hip
5.	P	1, sp
6.	**R**	**2,5, mp**

Com esse raciocínio, é possível gerar a fórmulas da linha 7 e da linha 8.

1.	$P \wedge Q$	hip
2.	$P \rightarrow R$	hip
3.	$(R \wedge S) \rightarrow \neg T$	hip
4.	$Q \rightarrow S$	hip
5.	P	1, sp
6.	R	2,5, mp
7.	**Q**	**1, sp**
8.	**S**	**4,7, mp**

Pela ideia inicial de aplicar a regra mp na linha 3, foi necessário aplicar outras regras parar gerar fórmulas até gerar a fórmula da linha 9.

1.	$P \wedge Q$	hip
2.	$P \rightarrow R$	hip
3.	$(R \wedge S) \rightarrow \neg T$	hip
4.	$Q \rightarrow S$	hip
5.	P	1, sp
6.	R	2,5, mp
7.	Q	1, sp
8.	S	4,7, mp
9.	**$R \wedge S$**	**6,8, cj**

Por fim, aplica-se mp na linha 3 e na linha 9 para gerar a fórmula ¬T, que vem a ser a conclusão do argumento.

1.	$P \land Q$	hip
2.	$P \rightarrow R$	hip
3.	$(R \land S) \rightarrow \neg T$	hip
4.	$Q \rightarrow S$	hip
5.	P	1, sp
6.	R	2,5, mp
7.	Q	1, sp
8.	S	4,7, mp
9.	$R \land S$	6,8, cj
10.	$\neg \mathbf{T}$	**3, 9, mp**

2.5 Uso das regras hipotéticas

As seis regras básicas de inferência para a Lógica Proposicional que tratam da inclusão e exclusão dos operadores $\land$, $\lor$, $\neg$, $\rightarrow$ e $\leftrightarrow$, além da regras de exclusão da implicação (*Modus Ponens* - mp) e da negação (Dupla Negação - dn) são de aplicação direta, não sendo necessário um raciocínio hipotético para funcionar.

Por outro lado, as duas regras de inclusão da implicação (Prova Condicional - **pc**) e da negação (Redução Ao Absurdo - **raa**) exigem raciocínios hipotéticos para serem aplicadas. Esses raciocínios são incorporados na prova na forma de demonstrações auxiliares ou hipotéticas, que têm um caráter “temporário” e que deverão ser descartadas quando a conclusão da demonstração auxiliar for atingida.

A forma como os resultados obtidos com a demonstração auxiliar serão usados na prova principal depende da regra de inferência usada.

No caso da redução ao absurdo (**raa**), a demonstração auxiliar é usada para mostrar que a partir de uma fórmula P, usada como hipótese inicial na demonstração auxiliar, é possível demonstrar uma contradição lógica, isto é, uma fórmula do tipo $Q \land \neg Q$. Neste caso a nova fórmula que se pode adicionar na prova principal é $\neg P$, que pode ser assumida como válida, porque caso contrário se chega a uma contradição lógica.

Já no caso da prova condicional (**pc**), a demonstração auxiliar mostra que, caso se assuma a fórmula P como hipótese inicial da demonstração auxiliar, então é possível se deduzir uma fórmula Q. Se conseguirmos deduzir, a partir de P e das outras hipóteses, uma outra fórmula Q, então a fórmula P → Q pode ser adicionada na sequência normal de demonstração. Neste caso, a fórmula P → Q adicionada à prova principal resume o fato (provado) que de P se pode deduzir Q.

Quando as regras que utilizam raciocínio hipotético são concluídas, então todas as fórmulas que constituem a demonstração auxiliar têm que ser "descartadas" e não mais utilizadas na sequência normal de demonstração. Somente a fórmula que foi demonstrada através do artifício do raciocínio hipotético pode ser usada na demonstração normal.

Quando as regras hipotéticas são utilizadas em uma demonstração, algumas considerações devem ser esclarecidas:

Tabela 9 – Restrições sobre o uso de regras hipotéticas

1)	Cada hipótese de uma regra hipotética introduz em uma prova o início de uma nova linha vertical. Essa linha continua até a hipótese ser descartada pela finalização da regra hipotética, seja ela **PC** ou **RAA**.
2)	Nenhuma ocorrência de uma fórmula à direita de uma linha vertical pode ser citada em qualquer regra aplicada depois de terminar a linha vertical. Isso é para garantir que a fórmula derivada da hipótese não seja usada indevidamente depois que a hipótese foi descartada.
3)	Se duas ou mais hipóteses de regras hipotéticas são vigentes simultaneamente, então a ordem na qual elas são descartadas deve ser a ordem inversa na qual elas são introduzidas.
4)	Uma demonstração não está completa até que todas as hipóteses de regras hipotéticas seja descartadas (ou seja que todas as regras hipotéticas sejam finalizadas). As hipóteses que não são descartadas são premissas adicionais introduzidas ilicitamente na demonstração.

Fonte: adaptado de NOLT, 1991, p.121.

Exemplos:

Prove o seguinte argumento, que equivale à regra do silogismo hipotético:

$$P \to Q, Q \to R \vdash P \to R$$

A demonstração é a seguinte e usa a regra de prova condicional:

1	$P \to Q$	hip
2	$Q \to R$	hip
3	$\mid P$	hip-pc
4	$\mid Q$	1,3, mp
5	$\mid R$	2,4, mp
6	$P \to R$	3-5, pc

Observe que a hipótese da prova condicional (hip-pc) e as fórmulas obtidas a partir dela e das hipóteses normais foram escritas ao lado da barra vertical | , mais à esquerda que as fórmulas pertencentes à sequência normal de demonstração. Isso é para deixar claro o caráter temporário destas fórmulas.

Um outro exemplo de uso de prova condicional é usado na demonstração do argumento:

$$P \vdash (P \to Q) \to Q$$

A demonstração é:

1	P	hip
2	$\mid P \to Q$	hip-pc
3	$\mid Q$	1,2, mp
4	$(P \to Q) \to Q$	2-3, pc

A regra da prova condicional também é útil para provar o seguinte argumento:

$$A \rightarrow (A \rightarrow B) \vdash A \rightarrow B$$

Usando essa regras a demonstração fica:

1	$A \rightarrow (A \rightarrow B)$	hip
2	$\mid A$	hip-pc
3	$\mid A \rightarrow B$	1, 2, mp
4	$\mid B$	2, 3, mp
5	$A \rightarrow B$	2-4, pc

A regra de redução ao absurdo pode ser usada para provar o seguinte argumento (que é equivalente à regra de*Modus Tollens*):

$$P \rightarrow Q, \neg Q \vdash \neg P$$

A demonstração fica:

1	$P \rightarrow Q$	hip
2	$\neg Q$	hip
3	$\mid P$	hip-raa
4	$\mid Q$	1,3, mp
5	$\mid Q \land \neg Q$	2,4, cj
6	$\neg P$	3-5, raa

Regras hipotéticas podem ser "aninhadas" uma dentro da outra, se isso for necessário. Isso ocorre, por exemplo, na prova de:

$$(P \land Q) \rightarrow R \vdash P \rightarrow (Q \rightarrow R)$$

Cuja demonstração é a seguinte:

1	$(P \land Q) \rightarrow R$	hip

2	$\mid P$	hip-pc
3	$\mid \mid Q$	hip-pc
4	$\mid \mid P \land Q$	2,3, cj
5	$\mid \mid R$	1,4, mp
6	$\mid Q \rightarrow R$	3-5, pc
7	$P \rightarrow (Q \rightarrow R)$	2-6, pc

Importante:

Sempre que as regras hipotéticas **RAA** e **PC** forem usadas em uma demonstração elas devem ser finalizadas antes de se deduzir a conclusão do argumento.

Dicas de dedução para construção das provas

1. A regra de *Modus Ponens* é provavelmente a regra de inferência mais intuitiva. Tente usá-la muitas vezes.
2. Fórmulas na forma $\neg(P \lor Q)$ ou $\neg(P \land Q)$ dificilmente são úteis em uma sequência de demonstração. Tente usar as Leis de DeMorgan para convertê-las, respectivamente, em $\neg P \land \neg Q$ ou $\neg P \lor \neg Q$, separando os componentes individuais de cada fórmula.
3. Fórmulas na forma $P \lor Q$ dificilmente são úteis em uma sequência de demonstração, já que não implicam P nem Q. Tente usar a dupla negação para converter $P \lor Q$ em $\neg(\neg P) \lor Q$ e depois usar a regra do condicional para obter $\neg P \rightarrow Q$.
4. Não há maneira correta de se construir uma prova. Se uma forma pode ser provada, ele pode ser provada por diferentes trocas de regras. Entretanto, as provas mais curtas, mais simples e mais fáceis são obtidas por uma estratégia baseada na estrutura da conclusão. As sugestões da Tabela 10 são guias úteis para o planejamento de tais estratégias. Geralmente, elas levam a provas razoavelmente eficientes, embora alguns problemas requeiram habilidade e engenhosidade.

Tabela 10 – Estratégias para prova

Formato da Conclusão	**Estratégia de Prova**
Fórmula atômica	Se nenhuma estratégia é imediata, coloca-se como hipótese a negação da conclusão para **raa (redução ao absurdo)**. Se isso for bem-sucedido, então a conclusão pode ser obtida depois de raa, pela aplicação da regra **dupla negação** (**dn** – eliminação da ¬).
Fórmula negada	Coloca-se como hipótese a conclusão sem a negação para **raa**. Se resultar uma contradição, a conclusão pode ser obtida por **raa**.
Conjunção	Prove cada uma das sentenças separadamente e então faça a conjunção delas com a regra **conjunção (cj** – introdução do ∧).
Disjunção	Se uma premissa com uma disjunção faz parte do argumento, então tenta-se provar os condicionais necessários para obter a conclusão pela regra de **eliminação da disjunção** (**E**∨). Caso contrário, coloca-se como hipótese a negação da conclusão e tenta-se **raa**. Em algumas situações, é possível provarmos a conclusão diretamente, encontrando uma das premissas e aplicando a regra **adição** (**ad** – introdução do ∨).
Implicação	Coloca-se como hipótese o seu antecedente e deriva-se o seu consequente, aplicando a regra **prova condicional** (**pc** - introdução da →).
Bi-implicação	Use a regra **prova condicional** (**pc** – introdução do condicional) – duas vezes para provar os dois condicionais necessários para obter a conclusão pela regra **introdução da bi-implicação** (↔**Eq**)

Fonte: elaborada pelos autores.

2.6 Regras de equivalência

Além das regras de inferência (básicas e derivadas) vistas anteriomente, também é possível utilizar na dedução natural as regras baseadas em equivalências tautológicas apresentadas na Seção 1.6. Essas regras são denominadas **regras de equivalência**. A Tabela 11, apresentada a seguir, contém as principais regras de equivalência:

Tabela 11 – Regras de equivalência

Expressão	Equivale a	Nome (Abreviação) da Regra
$P \vee Q$ $P \wedge Q$	$Q \vee P$ $Q \wedge P$	Comutatividade (**com**)
$(P \vee Q) \vee R$ $(P \wedge Q) \wedge R$	$P \vee (Q \vee R)$ $P \wedge (Q \wedge R)$	Associatividade (**ass**)
$\neg(P \vee Q)$ $\neg(P \wedge Q)$	$\neg P \wedge \neg Q$ $\neg P \vee \neg Q$	De Morgan (**dmor**)
$P \to Q$	$\neg P \vee Q$	Condicional (**cond**)
P	$\neg(\neg P)$	Dupla negação (**dn**)
$P \to Q$	$\neg Q \to \neg P$	Contraposição (**cont**)
P	$P \wedge P$ $P \vee P$	Auto-referência (**auto**)
$P \wedge (Q \vee R)$ $P \vee (Q \wedge R)$	$(P \wedge Q) \vee (P \wedge R)$ $(P \vee Q) \wedge (P \vee R)$	Distributividade (**dist**)

Fonte: elaborada pelos autores.

Importante:

- Note que as propriedades de equivalência apresentadas na Seção 1.6 são “reversíveis”, isto é, pode-se passar da fórmula à esquerda do operador de equivalência ⇔ para a fórmula à direita deste operador e vice-versa. A Tabela 10 já inclui todas as possibilidades de "reversão" das equivalências, incluindo os dois "sentidos" possíveis de uso de cada regra de equivalência.
- Isso implica que, diferente das regras de inferência, uma regra de equivalência pode ser aplicada na modificação de partes de uma fórmula. Assim, as fórmulas de uma regra de equivalência são intercambiáveis: pode-se substituir uma

subfórmula de uma fórmula por outra equivalente sem alterar sua validade lógica.

- Porém as regras de inferência **não são reversíveis** e **não podem ser aplicadas a partes de uma fórmula**, somente pode-se passar da situação prevista no esquema de aplicação da regra, definido acima da barra horizontal, que deve ser "encaixado" em uma fórmula inteira de um passo da demonstração, para a fórmula situada abaixo desta barra. O oposto, pela própria natureza da regra, não é permitido.

Exemplos:

Provar o seguinte argumento:

$$\neg A \lor B,\ B \to C \vdash A \to C$$

Sem o uso das regras de equivalência, a prova da validade deste argumento é relativamente complexa:

1	$\neg A \lor B$	hip
2	$B \to C$	hip
3	$\mid A$	hip-pc
4	$\mid \mid B$	hip-pc
5	$\mid B \to B$	4-4, pc
6	$\mid \mid \neg A$	hip-pc
7	$\mid \mid B$	3,6, inc
8	$\mid \neg A \to B$	6 - 7, pc
9	$\mid B$	1,5,8, Ev
10	$\mid C$	2,9, mp
11	$A \to C$	3-10, pc

Por outro lado, se considerarmos a regra de equivalência do condicional (cond), então a fórmula da linha 1 $\neg A \lor B$ pode ser transformada em $A \to B$, tornando muito mais fácil a prova do argumento:

1	$\neg A \vee B$	hip
2	$B \rightarrow C$	hip
3	$A \rightarrow B$	1, cond
4	$\mid A$	hip-pc
5	$\mid B$	3, 4, mp
6	$\mid C$	2, 5, mp
7	$A \rightarrow C$	4-6, pc

Embora não pareça, esse argumento é muito similar ao argumento que prova a validade do silogismo hipotético (**SH**), que afirma que de $P \rightarrow Q$ e de $Q \rightarrow R$, pode-se inferir $P \rightarrow R$. Usando essa regra a prova desse argumento se torna ainda mais simples:

1	$\neg A \vee B$	hip
2	$B \rightarrow C$	hip
3	$A \rightarrow B$	1, cond
4	$A \rightarrow C$	2, 3, sh

Adaptado de: NOLT, 1991, p. 129.

Outro argumento para provar:

$$\neg P \vee \neg Q, \neg\neg Q, R \rightarrow P \vdash \neg R$$

Sem o uso das regras derivadas e de equivalência, a demonstração pode ser elaborada conforme a estratégia a seguir.

1	$\neg P \vee \neg Q$	hip
2	$\neg\neg Q$	hip
3	$R \rightarrow P$	hip
4	$\mid R$	hip-raa
5	$\mid P$	3, 4, mp
6	$\mid Q$	2, dn

7	| $P \land Q$	5, 6, cj
8	| $\neg(P \land Q)$	1, demor
9	| $(P \land Q) \land \neg(P \land Q)$	7, 8, cj
10	$\neg R$	4 - 9, pc

Aplicando o conjunto de regras básicas, derivadas e de equivalência, a demonstração pode ser mais simples.

1	$\neg P \lor \neg Q$	hip
2	$\neg\neg Q$	hip
3	$R \rightarrow P$	hip
4	$\neg Q \lor \neg P$	1, com
5	$\neg P$	2, 4, sd
6	$\neg R$	3, 5, mt

Por fim, uma solução usando as regras hipotéticas, as regras derivadas e de equivalência.

1	$\neg P \lor \neg Q$	hip
2	$\neg\neg Q$	hip
3	$R \rightarrow P$	hip
4	| P	hip-raa
5	| $\neg\neg P$	4, dn
6	| $\neg Q$	1, 5, sd
7	| $\neg Q \land \neg\neg Q$	2, 6, cj
8	$\neg P$	4-7, pc
9	$\neg R$	3, 8, mt

2.7 Prova de teoremas

Um teorema é simplesmente um argumento que não precisa de hipótese (ou premissa) para ser válido (ou seja, para ser sempre verdadeiro). Por exemplo, o seguinte argumento é um teorema:

$$\vdash P \rightarrow (P \lor Q)$$

A prova de um argumento sempre parte de uma regra hipotética (ou redução ao absurdo – **RAA** – ou prova condicional – **PC**).

Exemplos:

No caso do teorema a prova pode ser feita pela regra da prova condicional:

1	$\mid P$	hip-pc
2	$\mid P \lor Q$	1, ad
3	$P \rightarrow (P \lor Q)$	1-2, pc

Note que não há hipótese ou premissa para esse argumento, a linha 1 já começa pela regra de dedução da prova condicional.

Outro exemplo de teorema:

$$\vdash A \rightarrow (B \rightarrow A)$$

Este teorema pode ser provado da seguinte forma:

1	$\mid A$	hip-pc
2	$\mid \mid B$	hip-pc
3	$\mid \mid \mid \neg A$	hip-raa
4	$\mid \mid \mid A \land \neg A$	1,3, cj
5	$\mid \mid \neg\neg A$	3-4, raa
6	$\mid \mid A$	5, dn
7	$\mid B \rightarrow A$	2-6, pc
8	$A \rightarrow (B \rightarrow A)$	1-7, pc

2.8 Prova de equivalências

Uma equivalência é um teorema sobre uma bi-implicação lógica $P \leftrightarrow Q$. Neste caso, para provar que a bi-implicação é válida é necessário provar ambos os lados da implicação, ou seja, que de P pode-se deduzir Q e que de Q pode-se deduzir P. Neste caso P e Q são interderiváveis. Para provar a equivalência, seguimos a mesma estratégia usada para provar bi-implicações, criando-se provas separadas para cada lado da regra de introdução da equivalência ($\leftrightarrow$I).

Exemplo:

$$\vdash (P \vee Q) \leftrightarrow (\neg P \rightarrow Q)$$

1	$\mid P \vee Q$	hip-pc
2	$\mid \neg\neg P \vee Q$	1, dn (aplica-se a regra dn devido à exceção da regra indicada na Tabela 10.)
3	$\mid \neg P \rightarrow Q$	2, cond
4	$(P \vee Q) \rightarrow (\neg P \rightarrow Q)$	1-3, pc
5	$\mid \neg P \rightarrow Q$	hip-pc
6	$\mid \neg\neg P \vee Q$	5, cond
7	$\mid P \vee Q$	6, dn
8	$(\neg P \rightarrow Q) \rightarrow (P \vee Q)$	5-7, pc
9	$(P \vee Q) \leftrightarrow (\neg P \rightarrow Q)$	4,8, $\leftrightarrow$I

2.9 Simbolização de argumentos verbais

Considere o argumento:

"Se as taxas de juros caírem, o mercado vai melhorar. Ou os impostos federais vão cair, ou o mercado não vai melhorar. As taxas de juros vão cair. Portanto, os impostos vão cair."

Supondo que as proposições simples usadas no argumento acima são representadas pelos seguintes símbolos:

M = "O mercado vai melhorar"
J = "A taxa de juros vai cair"
I = "Os impostos federais vão cair"

Então o argumento verbal acima seria equivalente à seguinte fórmula da lógica proposicional:

$$(J \rightarrow M) \land (I \lor \neg M) \land J \vdash I$$

Que pode ser transformado no seguinte argumento formal:

$$J \rightarrow M, I \lor \neg M, J \vdash I$$

Uma demonstração possível da validade deste argumento formal é apresentada a seguir:

1.	$J \rightarrow M$	hip
2.	$I \lor \neg M$	hip
3.	J	hip
4.	$\neg M \lor I$	2, com
5.	$M \rightarrow I$	4, cond
6.	$J \rightarrow I$	1, 5, sh
7.	I	3, 6, mp

Em outro exemplo, considere o argumento:

"Se Linus Torvalds é o criador do Linux, então ele é um ídolo. Se Linus Torvalds é o criador do Linux, então o Bill Gates não pode superá-lo. Portanto, se Linus Torvalds é o criador do Linux, ele é um ídolo e Bill Gates não pode superá-lo."

As proposições usadas para representar o argumento são:

L = “Linus Torvalds é o criador do Linux”
I = “Linus Torvalds é um ídolo”
B = “Bill Gates não pode superá-lo”

O argumento verbal acima seria equivalente ao seguinte argumento formal:

$$L \rightarrow I, L \rightarrow B \vdash L \rightarrow (I \land B)$$

Uma demonstração possível da validade deste argumento formal é apresentada a seguir:

1	$L \rightarrow I$	hip
2	$L \rightarrow B$	hip
3	$\mid L$	hip-pc
4	$\mid I$	1,3, mp
5	$\mid B$	2,3, mp
6	$\mid I \land B$	4,5, cj
7	$L \rightarrow (I \land B)$	3-6, pc

2.10 Questões comentadas

Nas questões apresentadas a seguir, temos o argumento e uma sequência de demonstração. Em algumas questões, apresentamos mais de uma demonstração, pois em determinadas estratégias utiliza-se o conjunto de regras de inferência (básicas e derivadas) e o conjunto de regras de equivalência.

De forma geral, como dica para começarmos a elaboração da estratégia de demonstração, devemos listar as hipóteses e observar qual a fórmula da conclusão. A partir disso, procuramos nas hipóteses onde encontramos a fórmula da conclusão ou parte dela para considerarmos que regras aplicaremos a partir da identificação do operador principal da hipótese. Assim que identificamos o operador principal da hipótese, já selecionamos algumas regras possíveis de serem aplicadas na demonstração, bem como já percebemos a fórmula que será gerada. É importante relembrar que, pelo método, aplicamos uma regra por linha, além de identificarmos quais as linhas são utilizadas junto com a regra.

Com essa ideia em mente, acompanhe as estratégias elaboradas para cada demonstração.

Questão 1. $\neg(P \land Q) \rightarrow (R \rightarrow S), R \land \neg S, Q \rightarrow T \vdash T$

1	$\neg(P \land Q) \rightarrow (R \rightarrow S)$	hip
2	$R \land \neg S$	hip
3	$Q \rightarrow T$	hip
4	$\mid \neg(P \land Q)$	hip-raa
5	$\mid R \rightarrow S$	1, 4, mp
6	$\mid R$	2, sp
7	$\mid S$	5, 6, mp
8	$\mid \neg S$	2, sp
9	$\mid S \land \neg S$	7, 8, cj
10	$\neg\neg (P \land Q)$	4 - 9, raa
11	$P \land Q$	10, dn
12	Q	11, sp
13	T	3, 12, mp

A conclusão desse argumento é a fórmula **T**. Identificamos **T** na hipótese da linha 3 e constatamos que operador principal é uma implicação (→). Para inferir **T** aplicando a regra **mp** (modus pones), temos que encontrar outra fórmula igual ao antecedente da hipótese da linha 3, isto é, a fórmula **Q**. Consequentemente, identificamos **Q** na hipótese da linha 1 onde o operador principal também é uma implicação. Entretanto, **Q** faz parte do antecedente da hipótese, então a regra **mp** talvez não seja uma boa opção. Uma estratégia possível é a regra **mt** (modus tollens), que aplicada na linha 1, gera uma fórmula que é a negação do antecedente (**$\neg\neg(P \land Q)$**), basta termos também a fórmula **$\neg(R \rightarrow S)$**. Outra estratégia nesse caso será gerar a fórmula **$\neg\neg(P \land Q)$** a partir da regra **raa** (redução ao absurdo), contemplada na demonstração da linha 4 até a linha 10. A partir disso, precisamos apenas liberar a fórmula **Q** para gerarmos a fórmula da conclusão **T**.

Questão 2. P ∨ (¬Q → R), ¬(P ∨ S) ∧ ¬R ⊢ Q

1	P ∨ (¬Q → R)	hip
2	¬(P ∨ S) ∧ ¬R	hip
3	¬(P ∨ S)	2, sp
4	¬P ∧ ¬S	3, demor
5	¬P	4, sp
6	¬Q → R	1, 5, sd
7	¬R	2, sp
8	¬¬Q	6, 7, mt
9	Q	8, dn

A estratégia elaborada e desenvolvida na questão 2 é simples. A partir da identificação da fórmula da conclusão **Q** na linha 1 e do operador principal ser uma disjunção (∨), pensamos em aplicar a regra **sd** (silogismo disjuntivo). Para isso, é necessário gerarmos a fórmula **¬P**, etapa apresentada da linha 3 até a linha 5. O próximo passo segue a ideia geral, como liberar **Q** da linha 6, sendo que o operador principal é uma implicação(→). Para aplicar a regra **mt** (modus tollens) na linha 6, temos que gerar a fórmula **¬R**. Assim, geramos a fórmula da conclusão na linha 9.

Questão 3. (¬P ∨ ¬Q) → R, R → S ⊢ ¬S → P

1	(¬P ∨ ¬Q) → R	hip
2	R → S	hip
3	| ¬S	hip-pc
4	| ¬R	2, 3, mt

5	| $\neg(\neg P \lor \neg Q)$	1, 4, mt
6	| $\neg\neg P \land \neg\neg Q$	5, demor
7	| $\neg\neg P$	6, sp
8	| P	7, dn
9	$\neg S \rightarrow P$	3 - 8, pc

Essa demonstração baseia-se na dica de dedução sugerida na Tabela 10 para fórmulas da conclusão que têm a implicação como operador principal. A sugestão para elaborar a estratégia é aplicar a regra **pc** (prova condicional), então colocamos como hipótese o antecedente (**¬S**) e derivamos até o consequente (**P**).

Questão 4. $(\neg B \rightarrow \neg A) \leftrightarrow (C \lor D)$, D, ¬B, $A \lor (T \lor \neg\neg Q)$ $\vdash (T \lor \neg\neg Q) \lor R$

1	$(\neg B \rightarrow \neg A) \leftrightarrow (C \lor D)$	hip
2	D	hip
3	$\neg B$	hip
4	$A \lor (T \lor \neg\neg Q)$	hip
5	$(C \lor D) \rightarrow (\neg B \rightarrow \neg A)$	1, E↔
6	$C \lor D$	2, ad
7	$\neg B \rightarrow \neg A$	5, 6, mp
8	$\neg A$	3, 7, mp
9	$T \lor \neg\neg Q$	4, 8, sd
10	$(T \lor \neg\neg Q) \lor R$	9, ad

Essa demonstração também se baseia na dica sugerida na Tabela 10. Como o operador principal da fórmula da conclusão é uma disjunção (∨), então podemos encontrar uma das fórmulas e aplicar a regra **ad** (adição) para gerar a outra. Como a hipótese da linha 4 tem parte da conclusão (**T ∨ ¬¬Q**), a estratégia elaborada foi liberar essa fórmula. Para isso, foi necessário gerar a fórmula ¬A para aplicar a regra **sd** (silogismo disjuntivo), conforme mostram as linhas da demonstração.

Questão 5. $(\neg B \land \neg C) \to (D \to C), \neg B, C \to B \vdash \neg D$

1	$(\neg B \land \neg C) \to (D \to C)$	hip
2	$\neg B$	hip
3	$C \to B$	hip
4	$\neg C$	2, 3, mt
5	$\neg B \land \neg C$	2, 4, cj
6	$D \to C$	1, 5, mp
7	$\neg D$	4, 6, mt

Solução 2 para **$(\neg B \land \neg C) \to (D \to C), \neg B, C \to B \vdash \neg D$**

1	$(\neg B \land \neg C) \to (D \to C)$	hip
2	$\neg B$	hip
3	$C \to B$	hip
4	$\neg B \to \neg C$	3, cont
5	$\neg C$	2, 4, mp
6	$\neg B \land \neg C$	2, 5, cj
7	$D \to C$	1, 6, mp
8	$D \to B$	3, 7, sh
9	$\neg B \to \neg D$	8, cont
10	$\neg D$	2, 9, mp

Solução 3 para **$(\neg B \land \neg C) \to (D \to C), \neg B, C \to B \vdash \neg D$**

1	$(\neg B \land \neg C) \to (D \to C)$	hip
2	$\neg B$	hip
3	$C \to B$	hip
4	$\mid D$	hip-raa

5	| ¬C	2, 3, mt
6	| ¬B ∧ ¬C	2, 5, cj
7	| D → C	1, 6, mp
8	| C	4, 7, mp
9	| C ∧ ¬C	5, 8, cj
10	¬D	4 - 9, raa

Nessa questão apresentamos 3 demonstrações, sendo que a última faz uso da regra hipotética **raa** (redução ao absurdo). De acordo com a Tabela 10, a dica nesse caso é abrir a prova temporária colocando como hipótese a fórmula da conclusão sem a negação, pois se encontrarmos uma contradição, fechamos a prova e introduzimos a negação na fórmula da hipótese. Esses passos estão descritos na linha 4 (| **D**) quando abrimos a prova, na linha 9 (| **C ∧ ¬C**) quando conseguimos obter uma contradição e, por fim, na linha 10 (**¬D**) que nos permite fechar a prova temporária descartando a hipótese e inferindo a fórmula negada.

Questão 6. ¬P ∨ ¬Q, (R ∨ S) → P, Q ∨ ¬S, ¬R ⊢ ¬(R ∨ S)

1	¬P ∨ ¬Q	hip
2	(R ∨ S) → P	hip
3	Q ∨ ¬S	hip
4	¬R	hip
5	¬P → ¬(R ∨ S)	2, cont
6	| ¬Q	hip-pc
7	| ¬S	3, 6, sd
8	| ¬R ∧ ¬S	4, 7, cj
9	|¬(R ∨ S)	8, demor
10	¬Q → ¬(R ∨ S)	6 - 9, pc
11	¬(R ∨ S)	1, 5, 10, E∨

A estratégia para resolvermos essa questão poderia seguir a adotada na questão 5, pois a conclusão do argumento também é uma fórmula negada (**¬(R ∨ S)**). Entretanto, para apresentarmos outra estratégia e aplicarmos outro conjunto de regras, essa demonstração tem como base a regra **E**∨ (eliminação da disjunção). Pela dica de dedução da Tabela 10, se temos uma fórmula com uma disjunção, então tentamos provar os condicionais.

Questão 7. (P ∨ Q) → R, S → (¬R ∧ ¬Q), S ∨ U ⊢ P → U

1	(P ∨ Q) → R	hip
2	S → (¬R ∧ ¬Q)	hip
3	S ∨ U	hip
4	| P	hip-pc
5	| P ∨ Q	4, ad
6	| R	1, 5, mp
7	| | S	hip-raa
8	| | ¬R ∧ ¬Q	2, 7, mp
9	| | ¬R	8, sp
10	| | R ∧ ¬R	6, 9, cj
11	| ¬S	7 - 10, raa
12	| U	3, 11, sd
13	P → U	4 - 12, pc

A estratégia elaborada nessa demonstração usa regras hipotéticas aninhadas. Como a fórmula da conclusão é um condicional, toamos como base uma das dicas de construção de prova que é iniciar a demonstração aplicando a regra **pc** (prova condicional). Partimos da hipótese P, que é o antecedente da fórmula da conclusão.

Questão 8. P ∨ (Q ∧ R), ¬R ∨ S, S → ¬T ⊢ T → P

1	P ∨ (Q ∧ R)	hip

2	$\neg R \lor S$	hip
3	$S \rightarrow \neg T$	hip
4	$\mid T$	hip-pc
5	$\mid \neg\neg T$	4, dn
6	$\mid \neg S$	3, 5, mt
7	$\mid R \rightarrow S$	2, cond
8	$\mid \neg R$	6, 7, mt
9	$\mid \neg Q \lor \neg R$	8, ad
10	$\mid \neg(Q \land R)$	9, demor
11	$\mid (Q \land R) \lor P$	1, com
12	$\mid P$	10, 11, sd
13	$T \rightarrow P$	4 - 12, pc

A estratégia elaborada nessa demonstração usa a regra hipotética **pc** (prova condicional), pois pelas dicas de construção de prova da Tabela 10, quando a conclusão é uma fórmula com a implicação, a sugestão é abrir uma prova temporária colocando como hipótese o antecedente e deriva-se o consequente. Nessa demonstração partimos da fórmula **T**, então agora a estratégia é gerar a fórmula **P**. Para isso, identificamos **P** como parte da fórmula na linha 1 e seu operador principal é a disjunção ($\lor$). A regra utilizada para gerar **P** nessa solução é a regra **sd** (silogismo disjuntivo).

Questão 9. $P \rightarrow Q, \neg Q, P \lor R \vdash R$

1	$P \rightarrow Q$	hip
2	$\neg Q$	hip
3	$P \lor R$	hip
4	$\neg P$	1, 2, mt
5	R	3, 4, sd

Solução 2 para $P \rightarrow Q, \neg Q, P \lor R \vdash R$

1	$P \to Q$	hip
2	$\neg Q$	hip
3	$P \lor R$	hip
4	$\mid \neg R$	hip-raa
5	$\mid R \lor P$	3, com
6	$\mid P$	4, 5, sd
7	$\mid Q$	1, 6, mp
8	$\mid Q \land \neg Q$	2, 7, cj
9	$\neg\neg R$	4 - 8, raa
10	R	9, dn

Solução 3 para $P \to Q, \neg Q, P \lor R \vdash R$

1	$P \to Q$	hip
2	$\neg Q$	hip
3	$P \lor R$	hip
4	$\neg P \lor Q$	1, cond
5	$Q \lor \neg P$	4, com
6	$\neg P$	2, 5, sd
7	R	3, 6, sd

Apresentamos 3 soluções para essa questão. Vale ressaltar a solução 2 que usa a regra hipotética **raa**. Nesse caso, partimos de uma fórmula negada (**¬R**) e na sequência das derivações encontramos a contradição (| **Q ∧ ¬Q**). A dedução da contradição na linha 8 no permite descartar a hipótese e inferir a fórmula negada, o resultado é **¬¬R**.

Questão 10. $P \lor \neg Q, R \to \neg P, R \vdash \neg Q$

1	$P \lor \neg Q$	hip
2	$R \to \neg P$	hip

3	R	hip
4	$\neg P$	2,3, mp
5	$\neg Q$	1, 4, sd

Solução 2 para $P \lor \neg Q, R \rightarrow \neg P, R \vdash \neg Q$

1	$P \lor \neg Q$	hip
2	$R \rightarrow \neg P$	hip
3	R	hip
4	$\mid Q$	hip-raa
5	$\mid \neg\neg Q$	4, dn
6	$\mid \neg P$	2, 3, mp
7	$\mid \neg P \land \neg\neg Q$	5, 6, cj
8	$\mid \neg(P \lor \neg Q)$	7, demor
9	$\mid (P \lor \neg Q) \land \neg(P \lor \neg Q)$	1, 8, cj
10	$\neg Q$	4 - 9, raa

Solução 3 para $P \lor \neg Q, R \rightarrow \neg P, R \vdash \neg Q$

1	$P \lor \neg Q$	hip
2	$R \rightarrow \neg P$	hip
3	R	hip
4	$\mid Q$	hip-raa
5	$\mid \neg P$	2, 3, mp
6	$\mid \neg Q$	1, 5, sd
7	$\mid Q \land \neg Q$	4, 6, cj
8	$\neg Q$	4 - 7, raa

Novamente apresentamos 3 demonstrações possíveis para o argumento. Nesse momento, acreditamos que você já sabe o que deve considerar para elaborar a estratégia de prova. Então, fica como exercício você propor a resposta das 3 soluções.

Questão 11. $\neg P \lor Q, \neg S \rightarrow \neg R, P \lor (R \land T) \vdash Q \lor S$

1	$\neg P \lor Q$	hip
2	$\neg S \rightarrow \neg R$	hip
3	$P \lor (R \land T)$	hip
4	$P \rightarrow Q$	1, cond
5	$\mid P$	hip-pc
6	$\mid Q$	4, 5, mp
7	$\mid Q \lor S$	6, ad
8	$P \rightarrow (Q \lor S)$	5 - 7, pc
9	$\mid R \land T$	hip-pc
10	$\mid R$	9, sp
11	$\mid \neg\neg R$	10, dn
12	$\mid \neg\neg S$	2, 11, mt
13	$\mid S$	12, dn
14	$\mid Q \lor S$	13, ad
15	$(R \land T) \rightarrow (Q \lor S)$	9 - 14, pc
16	$Q \lor S$	3, 8, 15, E$\lor$

A estratégia elaborada tem como base a aplicação da regra **Ev** (eliminação da disjunção). Como precisamos gerar os condicionais, a ideia é que o consequente de cada fórmula seja a conclusão (**$Q \lor S$**). Com isso, temos a fórmula **$P \lor (R \land T)$** da linha 3 e teremos que gerar a fórmula **$P \rightarrow (Q \lor S)$** e a fórmula **$(R \land T) \rightarrow (Q \lor S)$**. Novamente, para aplicarmos a regra **pc** (prova condicional) colocamos como hipótese o antecedente da fórmula e derivamos até o consequente. Assim, a linha 16 é a aplicação da regra **Ev** nas linhas 3, 8 e 15.

Questão 12. $\neg P \land Q, R \rightarrow P \vdash \neg P \land \neg R$

1	$\neg P \land Q$	hip
2	$R \rightarrow P$	hip

3	$\neg P$	1, sp
4	$\neg R$	2,3, mt
5	$\neg P \land \neg R$	3, 4, cj

Solução 2 para $\neg P \land Q, R \rightarrow P \vdash \neg P \land \neg R$

1	$\neg P \land Q$	hip
2	$R \rightarrow P$	hip
3	$\neg P$	1, sp
4	$\mid R$	hip-raa
5	$\mid P$	2, 4, mp
6	$\mid P \land \neg P$	3, 5, cj
7	$\neg R$	4 - 6, raa
8	$\neg P \land \neg R$	3, 7, cj

A estratégia aplicada nas duas soluções é evidente, encontramos cada parte da fórmula separadamente e aplicamos a regra **cj** (conjunçao). Na solução 2 usamos a regra hipotética **raa** para gerar a fórmula **¬R**.

Questão 13. $A \leftrightarrow Q, F \leftrightarrow R, A \land R \vdash F \land Q$

1	$A \leftrightarrow Q$	hip
2	$F \leftrightarrow R$	hip
3	$A \land R$	hip
4	$A \rightarrow Q$	1, $E_{\leftrightarrow}$
5	$R \rightarrow F$	2, $E_{\leftrightarrow}$
6	A	3, sp
7	R	3, sp
8	Q	4, 6, mp
9	F	5, 7, mp

10	$F \wedge Q$	8, 9, cj

A estratégia elaborada é simples, encontramos cada parte da fórmula separadamente e aplicamos a regra **cj** (conjunção).

Questão 14. $P \wedge Q, P \rightarrow R, (R \wedge S) \rightarrow \neg T, Q \rightarrow S \vdash \neg T$

1	$P \wedge Q$	hip
2	$P \rightarrow R$	hip
3	$(R \wedge S) \rightarrow \neg T$	hip
4	$Q \rightarrow S$	hip
5	P	1, sp
6	Q	1, sp
7	R	2, 5, mp
8	S	4, 6, mp
9	$R \wedge S$	7, 8, cj
10	$\neg T$	3, 9, mp

Pela orientação geral, listamos as hipóteses, identificamos a conclusão **$\neg T$** como parte da hipótese da linha 3 e constatamos que o operador principal é uma implicação ($\rightarrow$). Para inferir **$\neg T$** aplicando a regra **mp** (modus pones), é necessário encontrar outra fórmula com o antecedente da hipótese da linha 3, isto é, a fórmula **$R \wedge S$**. A estratégia para gerar **$R \wedge S$**, é inferir separadamente a fórmula **R** e a fórmula **S** para aplicar a regra **cj** (conjunção) e inferir a fórmula esperada **$R \wedge S$**. Para finalizar a demonstração, aplicamos a regra **mp** (modus ponens) nas linhas 3 e 9 para inferir a conclusão.

Questão 15. $R \rightarrow (P \wedge Q), \neg P \vee \neg Q, R \vee S \vdash S$

1	$R \rightarrow (P \wedge Q)$	hip
2	$\neg P \vee \neg Q$	hip

3	$R \lor S$	hip
4	$\mid R$	hip-raa
5	$\mid P \land Q$	1, 4, mp
6	$\mid \neg(P \land Q)$	2, demor
7	$\mid (P \land Q) \land \neg(P \land Q)$	5, 6, cj
8	$\neg R$	4 - 7,raa
9	S	3, 8, sd

Apresentamos a demonstração, agora esperamos que você comente a estratégia elaborada e quem sabe nos indica outra solução? ;-)

Questão 16. $(P \land Q) \leftrightarrow \neg R, \neg R \rightarrow \neg P, \neg Q \rightarrow \neg R \vdash Q$

1	$(P \land Q) \leftrightarrow \neg R$	hip
2	$\neg R \rightarrow \neg P$	hip
3	$\neg Q \rightarrow \neg R$	hip
4	$\mid \neg R$	hip-raa
5	$\mid \neg P$	2, 4, mp
6	$\mid \neg P \lor \neg Q$	5, ad
7	$\mid \neg(P \land Q)$	6, demor
8	$\mid \neg R \rightarrow (P \land Q)$	1, $E_{\leftrightarrow}$
9	$\mid \neg\neg R$	7, 8, mt
10	$\mid R$	9, dn
11	$\mid R \land \neg R$	4, 10, cj
12	$\neg\neg R$	4 - 11, raa
13	$\neg\neg Q$	3, 12, mt
14	Q	13, dn

Apresentamos a demonstração, agora esperamos que você comente a estratégia elaborada e quem sabe nos indica outra solução? ;-)

Questão 17. $\neg(P \land Q) \rightarrow \neg T$, $T \lor (\neg\neg A \land \neg C)$, $A \rightarrow \neg E$, $\neg(P \land Q) \vdash (A \rightarrow \neg E) \land \neg E$

1	$\neg(P \land Q) \rightarrow \neg T$	hip
2	$T \lor (\neg\neg A \land \neg C)$	hip
3	$A \rightarrow \neg E$	hip
4	$\neg(P \land Q)$	hip
5	$\neg T$	1, 4, mp
6	$\neg\neg A \land \neg C$	2, 5, sd
7	$\neg\neg A$	6, sp
8	A	7, dn
9	$\neg E$	3, 8, mp
10	$(A \rightarrow \neg E) \land \neg E$	3, 9, cj

Apresentamos a demonstração, agora esperamos que você comente a estratégia elaborada e quem sabe nos indica outra solução? ;-)

CAPÍTULO 3
SENTENÇAS ABERTAS

Neste capítulo, serão apresentadas noções de sentenças abertas, aplicação dos operadores lógicos em sentenças abertas e sentenças abertas com n variáveis. Tais conceitos estão baseados na Teoria dos Conjuntos e são apresentados ao leitor para facilitar a compreensão da Lógica de Predicados, que será introduzida nos próximos capítulos.

3.1 Sentenças abertas com uma variável

Intuitivamente, uma **sentença aberta** pode ser considerada uma frase que possui "espaços em brancos" que devem ser preenchidos com "coisas" retiradas de algum conjunto predeterminado.

Quando algum elemento é retirado desse conjunto e "encaixado" na sentença aberta, então esta sentença deixa de ser aberta e passa a se comportar como uma proposição simples, tendo um valor lógico possível: ou ela é uma sentença que afirma algo verdadeiro (proposição verdadeira) ou uma sentença que afirma algo falso (uma proposição falsa). Diz-se que a sentença é **fechada** quando isso ocorre.

Construir sentenças abertas é similar a jogar um jogo de montar frases, em que as frases são formadas a partir de trechos sugeridos pelos participantes. Nesse tipo de jogo, por exemplo, um participante diz o início, um segundo diz o meio e um terceiro tem que sugerir um final engraçado para a frase (mas que também seja **consistente** com o que já foi dito). No nosso caso, podemos começar com algumas sentenças fechadas e depois identificar seus espaços em branco:

(a.1) "A mesa é branca."
(b.1) "Este computador está funcionando."
(c.1) "O cachorro comeu sua ração."

Estas poderiam se transformar nas seguintes sentenças abertas:

(a.2) " _ _ _ _ é branca."
(b.2) "_ _ _ _ está funcionando."
(c.2) " _ _ _ _ comeu sua ração."

Um problema com as frases acima é que cada espaço em branco é igual aos outros espaços em branco. Quando existe só uma sentença (só uma frase), então geralmente não há problema. Porém, quando várias sentenças são usadas em conjunto, então é necessário indicar claramente quem pode aparecer nos espaços em branco. Por exemplo, nas frases acima, embora a sentença que afirma que algo é branco ("_ _ _ _ é branca") pode ser usada tanto para cachorros quanto computadores, não faz muito sentido imaginar um cachorro "funcionando" ou um computador que "comeu sua ração". A solução é dar "nome" aos espaços em branco, que deixam de ser espaços e passam a ser variáveis:

(a.3) "x é branca."
(b.3) "y está funcionando."
(c.3) "z comeu sua ração."

Usar variáveis ajuda mas não resolve o caso, porque ainda é necessário escolher também de onde serão retirados os elementos que se encaixarão na frase aberta, ou seja, que poderão ser valores das variáveis. Isso ocorre também nos jogos de montar frases ou palavras, que recorremos às pessoas, coisas, objetos etc. conhecidos ou em que nos obrigamos a somente usar as palavras presentes em um dicionário. Não faz sentido ou, simplesmente não é engraçado falarmos sobre pessoas ou coisas que não conhecemos ou entendemos.

Para tanto é necessário definir o **domínio** das váriaveis utilizadas nas sentenças abertas. Esse domínio é apenas um conjunto que não pode ser vazio, que define de onde devem ser retirados os valores de uma variável. No exemplo, vamos supor que x pode ser retirada de um conjunto que contém móveis, como por exemplo: estantes, mesas, cadeiras etc. Os valores da variável y, por sua vez, pertencem a um conjunto que contém equipamentos como: computadores, calculadoras, telefones etc. Por fim, o domínio de z será um conjunto de animais domésticos.

Agora, para completar o processo de simbolização são atribuídos símbolos para as afirmações abertas e é definido a partir do domínio das variáveis:

(a.4) P(x) para $x \in$ A, onde P(x)= "x é branco" e A é um conjunto de móveis.
(b.4) Q(y) para x $\in$ B, onde Q(y)= "y funciona" e B é um

conjunto de equipamentos.

(c.4) R(z) para x $\in$ C, onde R(z)= "z comeu a ração" e C é um conjunto de animais domésticos.

Lembre-se que a gramática da Língua Portuguesa (e das demais Linguagens naturais) divide as frases em sujeito e predicado: o sujeito é a pessoa, objeto ou "coisa" a qual se refere o predicado da frase, enquanto o predicado diz qual ação este sujeito fez (ou sofreu). O espaço em branco das sentenças abertas (ou a variável correspondente) ocupa o lugar onde vai o sujeito da frase.

As sentenças ou orações simples da Língua Portuguesa servem para afirmar alguma propriedade (o predicado) sobre alguma pessoa, objeto ou coisa (o sujeito). Por essa razão sentenças abertas também são denominadas **predicados**.

Já as sentenças abertas **formais** são construídas considerando-se que o sujeito da frase é substituído por uma variável. Também é definido um domínio para essa variável, dizendo quem são os objetos, pessoas, entidades, coisas etc. que podem ser representados pela variável. O predicado restante passa a ser então a afirmação que está sendo feita sobre algum sujeito do domínio.

Em termos formais, uma **sentença aberta com uma variável em um domínio D** ou simplesmente uma **sentença aberta em D** é uma expressão P(x) tal que P(a) é verdadeira (V) ou falsa (F) para todo elemento a pertencente ao domínio D, ou seja, para todo $a \in D$. O domínio D é um conjunto não vazio que define os valores possíveis da variável x, ou seja, ele é o domínio da variável x.

Sentenças abertas podem ser aplicadas a qualquer tipo de domínio conhecido, porém é comum que essas sentenças sejam exemplificadas e caracterizadas através de proposições matemáticas, principalmente por causa da precisão e rigor que se consegue obter com exemplos matemáticos.

Na verdade, a definição teórica precisa sobre o significado de uma sentença aberta e sobre o significado das construções que podem ser feitas com essas sentenças é feita através da Teoria dos Conjuntos (que também fundamenta a Matemática).

Observações:

- Para ajudar, o Anexo 1 apresenta uma breve revisão dos conceitos e operações da Teoria dos Conjuntos.

- O significado (a *semântica*) que se atribuirá para as sentenças abertas será dado pela especificação de *conjuntos-verdade* correspondentes a essas sentenças. Esses conjuntos-verdade definirão a *extensão* correspondente da sentença na Teoria dos Conjuntos.
- Mais adiante será visto que a semântica dos predicados ou das operações lógicas pode ser definida sem a necessidade de se especificar conjuntos-verdade.
- Entretanto, por agora, é melhor utilizar os conceitos familiares desta teoria para estudar o significado das sentenças abertas e os efeitos dos operadores lógicos sobre essas sentenças.

Exemplos:

As seguintes expressões são exemplos de sentenças abertas em N:

- $x + 1 > 8$
- $x^2 - 5x + 6 = 0$
- x é primo
- x é divisor de 10

3.2 Conjunto-verdade de uma sentença aberta

Chama-se **conjunto-verdade** de uma sentença aberta P(x) em um domínio D o conjunto de todos os elementos $a \in$ D tais que P(a) é uma proposição **verdadeira**. Formalmente o conjunto-verdade pode ser definido como:

$$V_P = \{x \mid x \in D \text{ e } P(x)=V\}$$

ou, mais simplesmente como:

$$V_P = \{x \in D \mid P(x)\}$$

Exemplos:

Assumindo que o domínio das seguintes sentenças é $N = \{1, 2, 3, ...\}$, o conjunto dos números naturais, então:

(a) O conjunto-verdade de P(x) = "x + 1 > 8" é dado por:

$$V_P = \{x \in D \mid P(x)\} = \{x \in N \mid P(x)\} = \{x \in N \mid x + 1 > 8\} = \{8, 9, 10, ...\}$$

(b) O conjunto-verdade de Q(x) = "x + 7 < 8" é dado por:

$$V_Q = \{x \in D \mid Q(x)\} = \{x \in N \mid x + 7 < 8\} = \varnothing$$

(c) O conjunto-verdade de R(x) = "x é divisor de 10" é dado por:

$$V_R = \{x \in N \mid x \text{ é divisor de } 10\} = \{1, 2, 5, 10\}$$

(d) O conjunto-verdade de S(x) = "x + 5 > 3" é dado por:

$$V_S = \{x \in \mathrm{N} \mid x + 5 > 3\} = \{1, 2, 3, ...\} = N$$

Os exemplos acima indicam algumas conclusões importantes:

(i) O conjunto-verdade de uma sentença aberta com uma variável sempre está contido ou (no máximo) é igual ao domínio D da sentença:

$$V_P \subseteq D$$

(ii) Se P(x) é uma sentença aberta em D, então três casos podem ocorrer:

(ii.1) P(x) é verdadeira para todo $x \in D$. Neste caso o conjunto verdade de P(x) é igual ao próprio domínio A. Quando isso ocorre se diz que P(x) exprime uma **condição universal** ou **propriedade universal** no domínio D.
(ii.2) P(x) é verdadeira para alguns $x \in D$. Neste caso o conjunto verdade de P(x) é um subconjunto próprio do domínio D. Quando isso ocorre se diz que P(x) exprime uma **condição possível** ou **propriedade possível** no domínio

D.
(ii.3) P(x) não é verdadeira para nenhum $x \in D$. Neste caso o conjunto-verdade de P(x) é vazio ($V_P = \varnothing$). Quando isso ocorre se diz que P(x) exprime uma **condição impossível** ou **propriedade impossível** no domínio D.

Exercício 3.2.1

Determinar o conjunto-verdade em *N* (conjunto dos números naturais) de cada uma das sentenças abertas a seguir:

(a) $2x = 6$ **(b)** $x - 1 < 4$
(c) $x^2 - 5x + 6 = 0$ **(d)** $x^2 - x + 2 = 0$
(e) $x^2 - 5x = 0$ **(f)** $x - 5 \in N$

3.3 Conjunção sobre sentenças abertas (∧)

A conjunção lógica (a operação E lógico, representada pelo símbolo ∧) pode ser aplicada sobre sentenças abertas ou predicados.

Vamos começar a análise da conjunção de sentenças abertas, supondo duas sentenças abertas bastante simples:

“x é médico”, “x é professor”

que podem ser aplicadas sobre o domínio (conjunto) das pessoas vivas atualmente.

Agora, se conectarmos ambas as afirmações pelo conectivo E lógico (∧) fica-se com a expressão:

“x é médico” ∧ “x é professor”

que somente pode ser verdadeira (satisfeita) para as pessoas (os “x”) que são médico(a) e professor(a).

No caso das pessoas vivas (que é um conjunto finito) seria teoricamente possível montar uma tabela listando todas as pessoas e verificar quem é médico e quem é professor e, portanto, descobrir quem atende a ambas as condições:

x	x é médico	x é professor	x é médico ∧ x é professor
Pedro	V	F	F
Maria	V	V	V
Carlos	F	F	F
José	V	V	V
Beatriz	V	F	F
...	...	...	...

No caso o significado do operador ∧ é dado pela tabela-verdade deste operador que já foi usada na lógica proposicional. Quando os *x* são substituídos por elementos do conjunto das pessoas, então as proposições "*x* é médico" e "*x* é professor" se transformam em sentenças fechadas que são as proposições simples da lógica proposicional. No exemplo da tabela, quando *x* = Pedro, tem-se "Pedro é médico", que é uma proposição verdadeira, e "Pedro é professor", que é uma proposição falsa. A conjunção de ambas fica "Pedro é médico" ∧ "Pedro é professor", que é uma proposição composta falsa.

Em todas as conjunções de sentenças abertas em que os domínios são finitos pode-se teoricamente montar uma tabela similar à vista acima e verificar, usando as regras da lógica proposicional, qual o valor-verdade da conjunção. Porém o que se pode fazer quando os domínios são infinitos? Que tipo de significado se poderia atribuir para a conjunção de sentenças abertas sobre domínios infinitos?

A solução para esse problema é dada usando-se a Teoria Elementar dos Conjuntos para definir o significado da operação de conjunção lógica sobre duas sentenças abertas.

Para tanto é necessário definir qual poderia ser o significado da conjunção em termos de operações sobre conjuntos. Dessa forma, primeiro se deve definir que conjuntos poderão ser usados.

Uma sentença aberta P(x) já é definida em termos de dois conjuntos: o domínio A de suas variáveis e o conjunto-verdade V_P implicado por P(x).

Portanto, será sobre esses dois conjuntos que o conceito de conjunção deverá ser definido.

Em primeiro lugar vamos analisar a situação da conjunção de duas sentenças em termos de diagramas gráficos (os Diagramas de Venn) que conseguem expressar os conceitos da teoria dos conjuntos de uma forma muito mais intuitiva. Vamos supor as duas sentenças já vistas anteriormente:

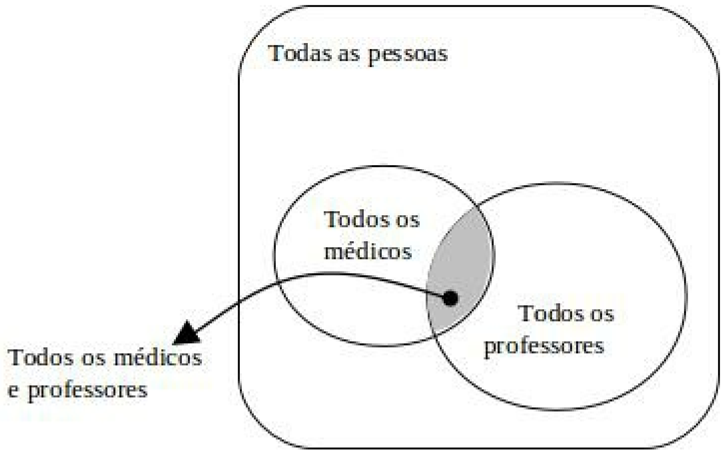

Figura 1 – Intersecção de conjuntos como conjunção lógica.
Fonte: elaborada pelos autores.

Neste desenho deve ficar claro que somente a **intersecção** das duas áreas (e portanto dos dois conjuntos) corresponde às pessoas que são médicos e professores. Genericamente, supondo duas sentenças abertas P(x) e Q(x) sobre um domínio A, tem-se que a conjunção de ambas somente pode ser satisfeita pelos elementos de A que satisfizerem ambas P(x) e Q(x), isto é, pela **intersecção** dos respectivos conjuntos-verdade.

Graficamente isso pode ser mostrado pelo seguinte diagrama:

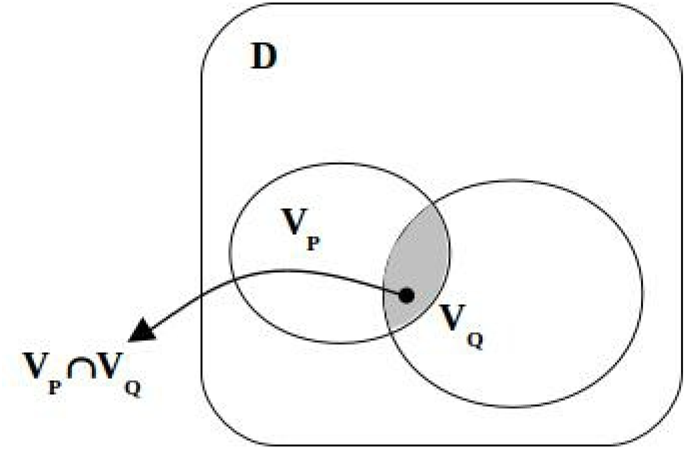

Figura 2 – Um exemplo de intersecção como conjunção.
Fonte: elaborada pelos autores.

Ou seja, o conjunto-verdade correspondente à conjunção de duas sentenças abertas é dado pela intersecção dos conjuntos-verdade de ambas as sentenças. Formalmente, este conjunto-verdade é definido como:

$$V_{P \wedge Q} = V_P \cap V_Q = \{x \in D \mid P(x)\} \cap \{x \in D \mid Q(x)\}$$

Exemplo:

Sejam as seguintes sentenças abertas em Z (conjunto dos números inteiros):

$P(x) = x^2 + x - 2 = 0 \quad Q(x) = x^2 - 4 = 0$

Neste caso o conjunto-verdade da sentença aberta P(x) $\wedge$ Q(x) pode ser calculado como segue:

$$
\begin{aligned}
V_{P \wedge Q} &= V_P \cap V_Q \\
&= \{x \in Z \mid P(x)\} \cap \{x \in Z \mid Q(x)\} \\
&= \{x \in Z \mid x^2 + x - 2 = 0\} \cap \{x \in Z \mid x^2 - 4 = 0\} \\
&= \{-2, 1\} \cap \{-2,2\} = \{-2\} \\
&= \{-2\}
\end{aligned}
$$

3.4 Disjunção sobre sentenças abertas ($\vee$)

A disjunção lógica (a operação OU lógico, representada pelo símbolo $\vee$) também pode ser aplicada sobre sentenças abertas ou predicados.

Para começar, vamos supor as duas sentenças abertas já usadas anteriormente:

“x é médico”, “x é professor”

sobre o domínio das pessoas vivas. Agora, se conectarmos ambas as afirmações pelo conectivo OU lógico ($\vee$) ficaremos com a seguinte expressão:

“x é médico” $\vee$ “x é professor”

que somente pode ser verdadeira (satisfeita) para as pessoas (os “x”) que são médicas ou pelas pessoas que são professoras ou pelas pessoas que têm ambas as profissões (somente não pode ser satisfeita pelas pessoas que não são nem médicas nem professoras).

Da mesma forma que no caso da conjunção também seria teoricamente possível montar uma tabela listando todas as pessoas e verificando quem é médico e quem é professor e, portanto, descobrindo quem atende a uma das condições ou a ambas. Porém, por generalidade, vamos partir direto para a interpretação gráfica em termos da Teoria Elementar dos Conjuntos, ou seja, vamos ver qual o diagrama de Venn apropriado para este caso:

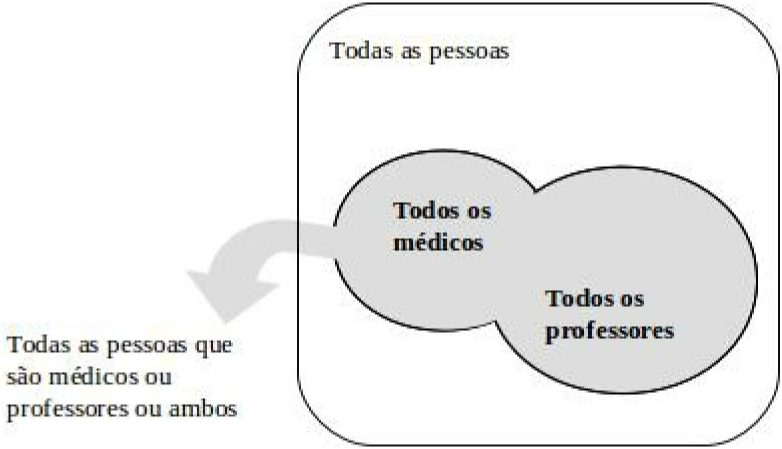

Figura 3 – União de conjuntos como disjunção lógica.
Fonte: elaborada pelos autores.

No diagrama deve ficar claro que a **união** das duas áreas (e portanto dos dois conjuntos-verdade) corresponde às pessoas que são médicos ou são professores ou ambos. Genericamente, supondo duas sentenças abertas P(x) e Q(x) sobre um domínio A, tem-se que a disjunção de ambas somente pode ser satisfeita pelos elementos de A que satisfizerem P(x) ou Q(x) ou ambas, isto é, pela **união** dos respectivos conjuntos-verdade. Graficamente isso pode ser mostrado pelo seguinte diagrama:

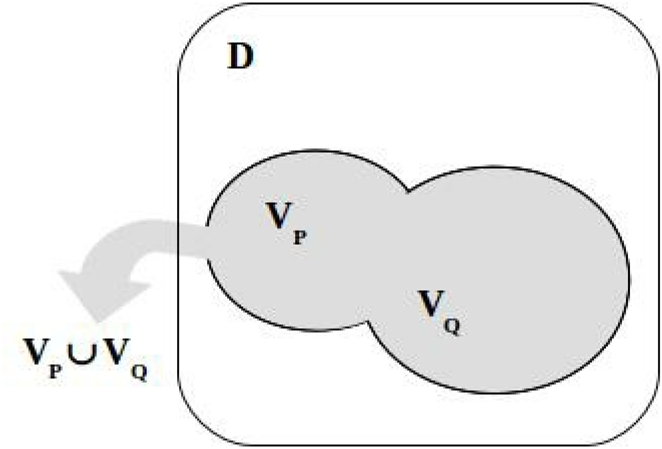

Figura 4 – Um exemplo de união como disjunção.
Fonte: elaborada pelos autores.

Ou seja, o conjunto-verdade correspondente à disjunção de duas sentenças abertas é dado pela união dos conjuntos-verdade de ambas as sentenças. Formalmente, este conjunto-verdade é definido como:

$$V_{P \vee Q} = V_P \cup V_Q = \{x \in D \mid P(x)\} \cup \{x \in D \mid Q(x)\}$$

Exemplo:

Sejam as seguintes sentenças abertas em Z (conjunto dos números inteiros):

$P(x) =$ "$x^2 + x - 2 = 0$" $\quad Q(x) =$ "$x^2 - 4 = 0$"

Neste caso o conjunto-verdade da sentença $P(x) \vee Q(x)$ pode ser calculado da seguinte forma:

$$V_{P \vee Q} = V_P \cup V_Q$$

$= \{x \in Z \mid P(x)\} \cup \{x \in Z \mid Q(x)\}$
$= \{x \in Z \mid x^2 + x - 2 = 0\} \cup \{x \in Z \mid x^2 - 4 = 0\}$
$= \{-2, 1\} \cup \{-2, 2\}$
$= \{-2, 1, 2\}$

Exercício 3.4.1

Determinar o conjunto-verdade em $D = \{1, 2, 3, ..., 9, 10\}$ de cada uma das seguintes sentenças abertas compostas:

(a) $P(x) = x < 7 \wedge Q(x) = x$ é ímpar

(b) $P(x) = x$ é par $\wedge Q(x) = x + 2 \leq 10$

(c) $P(x) = 3 \mid x \wedge Q(x) = x < 8$

(d) $P(x) = (x + 4) \in D \wedge Q(x) = (x^2 - 5) \notin D$

onde a | b é definida como a afirmação "a divide b sem resto" (por exemplo, 3 | 12 é uma afirmação verdadeira, porque 3 efetivamente divide 12 sem resto, porém 2 | 9 não é uma afirmação verdadeira porque 2 não consegue dividir 9 sem sobrar um resto).

3.5. Negação de uma sentença aberta (¬)

A negação lógica (a operação NÃO lógico, representada pelo símbolo ¬) também pode ser aplicada sobre sentenças abertas ou predicados.

Vamos começar considerando a sentença:

"x tem menos de 21 anos"

sobre o conjunto de todas as pessoas. Agora, antepondo a negação lógica sobre esta sentença, temos a expressão:

¬ "x tem menos de 21 anos"

que deve ser satisfeita somente pelas pessoas (os "x") que não tenham menos de 21 anos, ou seja, que tenham 21 anos ou mais.

Da mesma forma que nos conectivos vistos anteriormente, seria teoricamente possível montar a tabela listando todas as pessoas e verificar quem tem menos de 21 anos. Porém, por generalidade, vamos partir direto para a interpretação em termos de diagramas de Venn:

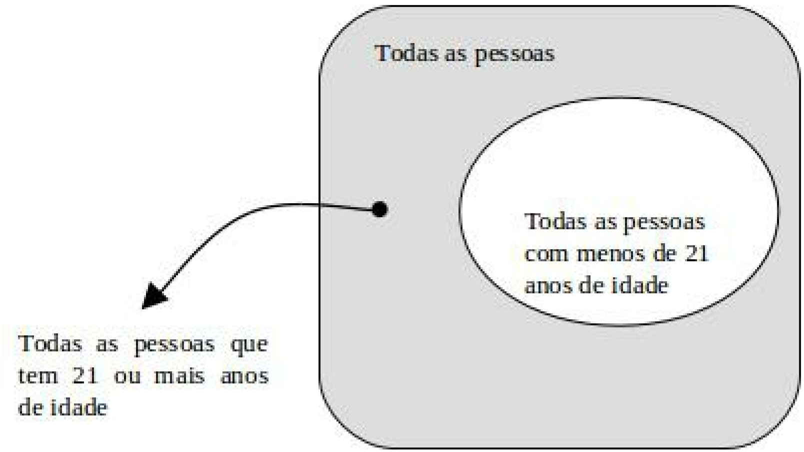

Figura 5 – Complementação de conjuntos e negação lógica.
Fonte: elaborada pelos autores.

No diagrama deve ficar claro que a expressão:

$$\neg \text{ "x tem menos de 21 anos"}$$

somente é satisfeita pelos elementos do conjunto de todas as pessoas que **não estão** no conjunto das pessoas com menos de 21 anos. Este conjunto, por sua vez, é definido como a diferença entre dois conjuntos: o domínio da sentença que é conjunto de todas pessoas subtraído do conjunto-verdade da sentença "x tem menos de 21 anos".

Portanto, supondo uma sentença aberta qualquer P(x) sobre um domínio D, tem-se que a negação desta sentença somente pode ser satisfeita pelos elementos de D que **não** estiverem no conjunto-verdade de P(x), isto é, que estiverem no conjunto definido pela diferença entre A e V_P. Graficamente isso pode ser mostrado pelo seguinte diagrama:

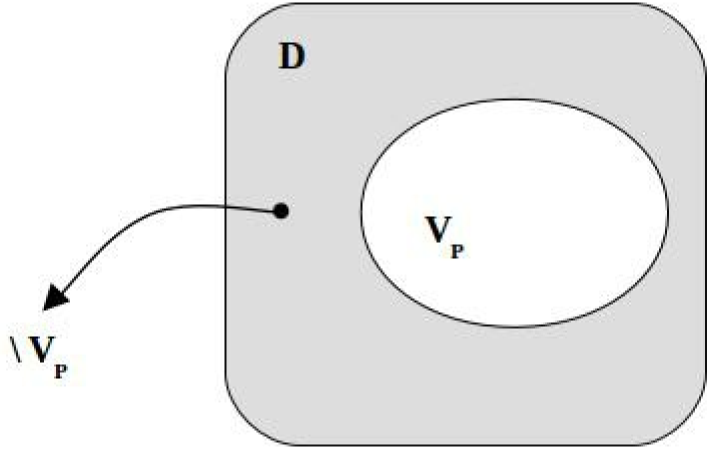

Figura 6 – Um exemplo de negação como complementação.
Fonte: elaborada pelos autores.

O conjunto-verdade correspondente à negação de uma sentença aberta é dado pela diferença entre o domínio da sentença e o conjunto-verdade desta. Formalmente, esse conjunto-verdade é definido como:

$$V_{\neg P} = D - V_P = D - \{x \in A \mid P(x)\}$$

onde a diferença entre conjuntos A - B é definida como o conjunto formado pelos elementos de A que não pertencem a B, isto é:

$$A - B = \{x \in A \mid x \notin B\}$$

Também é possível definir o conjunto-verdade correspondente à negação de uma sentença aberta, usando a operação de complemento de conjunto A em um determinado domínio D. Esta operação é indicada simplesmente por A é definida como a diferença entre o domínio D e o conjunto A:

$$A = D - A$$

Com a operação de complemento de conjuntos a negação de uma sentença aberta pode ser definida simplesmente como:

$$V_{\neg P} = D - V_P = \{x \in D \mid P(x)\}$$

Exemplo:

Seja a seguinte sentença aberta em $D = \{1, 2, 3, ..., 10\}$:

$$P(x) = x^2 \in D$$

cujo conjunto-verdade V_P é:

$$V_P = \{1, 2, 3\}$$

Sua negação fica:

$$\neg P(x) = \neg (x^2 \in D)$$

O conjunto-verdade dessa expressão é dado por:

$$\begin{aligned} V_{\neg P} &= D - V_P \\ &= D - \{1,2,3\} \\ &= \{1,2,3, ..., 10\} - \{1,2,3\} \\ &= \{4,5, ..., 10\} \end{aligned}$$

3.6 Implicação entre duas sentenças abertas (→)

Da mesma forma que no caso das outras operações lógica, a implicação lógica (o operador de implicação ou condicional, representado pelo símbolo →) também pode ser aplicada sobre sentenças abertas. Considerando as duas sentenças abertas já usadas anteriormente:

“x é médico”, “x é professor”

sobre o domínio das pessoas vivas. Agora, a implicação da primeira ("x é médico") sobre a segunda ("x é professor") é expressa da seguinte forma:

“x é médico” → “x é professor”

Esta implicação afirma que se uma pessoa é médica, então também deve ser professora. Ela será satisfeita por dois conjuntos de pessoas: pelas pessoas que efetivamente são médicas e professoras, mas também pelas pessoas que não são médicas. Somente as pessoas que são médicas, mas não são professoras tornam a implicação falsa. A Figura 7 mostra o diagrama de Venn apropriado para este caso:

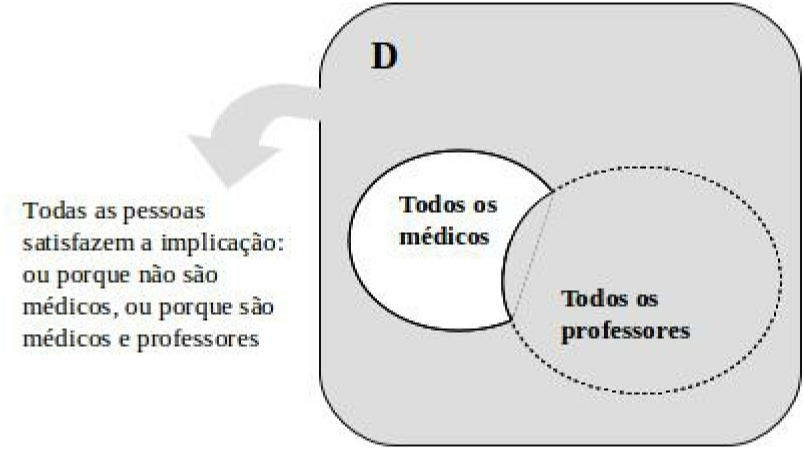

Figura 7 – Pessoas que satisfazem a implicação lógica.
Fonte: elaborada pelos autores.

O diagrama da Figura 7 também pode ser compreendido quando se leva em conta a propriedade de equivalência da implicação (ver Seção 1.6) que:

$$P \rightarrow Q \Leftrightarrow \neg P \vee Q$$

Esta propriedade também é válida quando aplicada às sentenças abertas, assumindo, neste caso, a seguinte forma:

$$P(x) \rightarrow Q(x) \Leftrightarrow \neg P(x) \vee Q(x)$$

Usando essa propriedade o conjunto-verdade para a expressão $P(x) \rightarrow Q(x)$ pode ser definido como:

$$V_{P \rightarrow Q} = V_{\neg P \vee Q} = V_P \cup V_Q$$

Graficamente isso pode ser mostrado pelo seguinte diagrama:

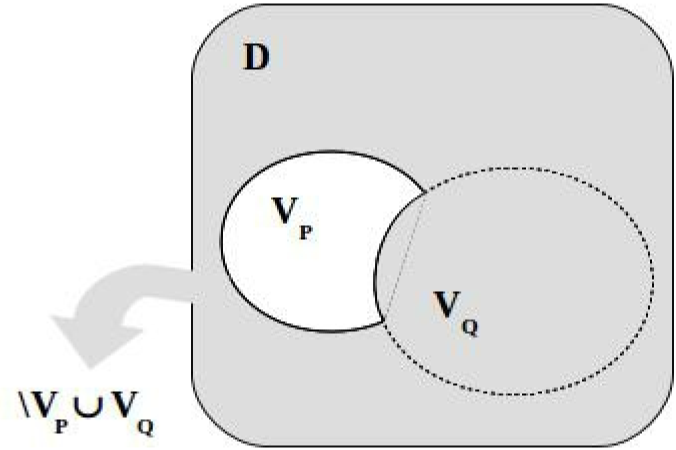

Figura 8 – Operação de conjuntos correspondente à implicação lógica.
Fonte: elaborada pelos autores.

Exemplo 1:

Supondo as seguintes sentenças sobre o domínio A = {-2, -1, 0, 1, 2}:

$P(x) = \text{"}x < 0\text{"} \quad Q(x) = \text{"}x^2 - x = 0\text{"}$

O conjunto-verdade para $P(x) \rightarrow Q(x)$ será dado por:

$$
\begin{aligned}
V_{P \rightarrow Q} &= V_{\neg P} \cup V_Q \\
&= \{x \in A \mid P(x)\} \cup \{x \in A \mid Q(x)\} \\
&= \{-2, -1\} \cup \{0, 1\} \\
&= \{0, 1, 2\} \cup \{0, 1\} \\
&= \{0, 1, 2\}
\end{aligned}
$$

Inicialmente, encontramos o conjunto-verdade da sentença P e da sentença Q, então $V_P = \{-2, -1\}$ e $V_Q = \{0, 1\}$. Após, é preciso encontrar os valores do conjunto-verdade da negação de P, então $V_{\neg P} = \{0, 1, 2\}$; e por fim, pela definição da disjunção faz-se a união dos conjuntos-verdade para encontrar o resultado .

Exemplo 2:

Supondo as seguintes sentenças sobre o domínio dos números naturais $N = \{1,2,3,...\}$:

P(x) = "x | 12" Q(x) = "x | 45"

para x | y indica a relação de x dividir y sem resto.

O conjunto-verdade para $P(x) \rightarrow Q(x)$ será dado por:

$$\begin{aligned} V_{P \rightarrow Q} &= V_{\neg P} \cup V_Q \\ &= \{x \in N \mid P(x)\} \cup \{x \in N \mid Q(x)\} \\ &= \{1,2,3,4,6,12\} \cup \{1,3,4,9,15,45\} \\ &= \{5,7,8,9,10,11,13,14,15,...\} \cup \{1,3,4,9,15,45\} \\ &= \{1,3,4,5,7,8,9,10,11,13,14,15,...\} \end{aligned}$$

Outra forma de representar o resultado desse exemplo é:

$$V_{P \rightarrow Q} = N - \{2, 6, 12\}$$

3.7 A bi-implicação lógica($\leftrightarrow$)

Seguindo a abordagem da seção anterior, vamos assumir que as regras de equivalência da lógica proposicional também valem para as sentenças abertas compostas. Neste caso pode-se usar a equivalência da bi-implicação para definir a semântica (o significado) desse operador. O operador de bi-implicação é definido em termos da implicação:

$$P \leftrightarrow Q \Leftrightarrow (P \rightarrow Q) \land (Q \rightarrow P)$$

que pode ser reescrito pela equivalência do condicional em:

$$P \leftrightarrow Q \Leftrightarrow (\neg P \lor Q) \land (\neg Q \lor P)$$

Agora, esta regra, quando aplicada às sentenças abertas, pode assumir a seguinte forma:

$$P(x) \leftrightarrow Q(x) \Leftrightarrow (\neg P(x) \lor Q(x)) \land (\neg Q(x) \lor P(x))$$

Dessa forma o conjunto-verdade para a expressão $P(x) \leftrightarrow Q(x)$ é dado por:

$$\begin{aligned} V_{P \leftrightarrow Q} &= V_{(P \rightarrow Q) \land (Q \rightarrow P)} \\ &= V_{(\neg P \lor Q) \land (\neg Q \lor P)} \\ &= V_{\neg P \lor Q} \cap V_{\neg Q \lor P} \\ &= (V_{\neg P} \cup V_{Q}) \cap (V_{\neg Q} \cup V_{P}) \end{aligned}$$

Exemplo:

Supondo as seguintes sentenças sobre o domínio dos números naturais $N=\{1,2,3,\ldots\}$:

$P(x) = $ "x | 6" $Q(x) = $ "x | 15"

para x | y indica a relação de x dividir y sem resto. Agora o conjunto-verdade de $P(x) \leftrightarrow Q(x)$ pode ser obtido da seguinte forma:

$$\begin{aligned}
V_{P \leftrightarrow Q} &= (V_{\neg P} \cup V_Q) \cap (V_{\neg Q} \cup V_P) \\
&= (\{x \in N \mid x \mid 6\} \cup \{x \in N \mid x \mid 15\}) \cap (\{x \in N \mid x \mid 15\} \cup \{x \in N \mid x \mid 6\}) \\
&= (\{1,2,3,6\} \cup \{1,3,5,15\}) \cap (\{1,3,5,15\} \cup \{1,2,3,6\}) \\
&= (\{4,7,8,9,...\} \cup \{1,3,5,15\}) \cap (\{2,4,6,7,..,14,16,17,...\} \cup \{1,2,3,6\}) \\
&= \{1,3,4,5,7,8,9,...\} \cap \{1,2,3,4,6,7,8,..,14,16,17,...\} \\
&= \{1,3,4,7,8, 9, 10, 11, 12, 13,14,16,17,...\}
\end{aligned}$$

Outra forma de representar o resultado desse exemplo é:

$$V_{P \leftrightarrow Q} = N - \{2,5,6,15\}$$

3.8 Equivalências tautológicas

Assim como assumimos as equivalências do condicional e do bicondicional, as outras equivalências da lógica proposicional se aplicam às sentenças abertas compostas:

- A **conjunção** e a **disjunção** continuam a ser **comutativas** e **associativas**, e cada uma delas é **distributiva** em relação à outra.
- A propriedade da **dupla negação** continua válida, assim como as **Leis de DeMorgan**.
- A **contraposição** e a **prova condicional** também continuam válidas.

Entretanto, as regras de **identidade** assumem um novo aspecto. Agora temos as seguintes regras:

(i) A conjunção de uma sentença aberta com uma outra que exprime uma **condição universal** (isto é, uma condição sempre verdadeira) é equivalente à primeira.
(ii) A **disjunção** de uma sentença aberta com uma outra que

exprime uma **condição impossível** (isto é, uma condição sempre falsa) é equivalente à primeira.

3.9 Predicados com múltiplas variáveis

Até agora foram apresentados e tratados apenas sentenças abertas como P(x) e Q(y) que utilizam somente uma variável. Tais tipos de sentenças abertas seguem a divisão gramatical usual da Língua Portuguesa (e das outras linguagens naturais) considerando o sujeito (representado pela variável x) e o predicado (representado pelo símbol P).

Porém, mesmo na linguagem natural, é comum existir mais subdivisões nas orações. Por exemplo, o predicado pode ainda ser dividido em verbo e objeto, ficando a oração formada por um sujeito, que faz uma ação, por um verbo que indica qual ação é efetuada pelo sujeito e por um objeto desta ação.

As frases a seguir são exemplos típicos dessa situação:

- "Pedro foi ao cinema"
- "Maria pintou a janela"
- "Beatriz comeu um lanche"

Essas frases podem ser simbolizadas da seguinte forma:

- "Pedro foi ao cinema" ⇒ F(p,c)
- "Maria abriu a janela" ⇒ A(m,j)
- "Beatriz comeu um lanche" ⇒ C(b,l)

onde F(x,y) simboliza que um sujeito x foi ao lugar y, A(x,y) simboliza x abriu uma coisa y e C(x,y) que x comeu um alimento y. As constantes p,m, e b representam, respectivamente, os sujeitos Pedro, Maria e Beatriz e as constantes c, j e l repreentam, respectivamente, os objetos cinema, janela e lanche.

F(x,y), A(x,y) e C(x,y) são agora denominados apenas de predicados porque já não faz muito sentido considerá-los sentenças abertas. Esses são exemplos de predicados de duas variáveis. Predicados com apenas uma variável, como os predicados P(x) e Q(x) que vínhamos trabalhando até agora são denominados predicados *unários*. Porém estes não são os únicos tipos de predicados possíveis. Na verdade, é possível se definir predicados que utilizem duas ou mais variáveis pertencentes ao mesmo domínio ou a domínios diferentes.

Predicados de duas variáveis são chamados *binários*. Predicados que utilizem duas variáveis também são denominados *relações* entre estas variáveis, porque definem uma relação entre os elementos de uma variável com a outra.

O significado atribuído para predicados de duas variáveis é definido através do *produto cartesiano* dos domínios de cada uma das variáveis.

O produto cartesiano de dois conjuntos: $A \times B$ é o conjunto formado por todas as *duplas* ordenadas (a, b) onde $a \in A$ e $b \in B$. Por exemplo, se $A=\{1,2,3\}$ e $B=\{a,b\}$ então $A \times B$, o produto cartesiano desses conjuntos é o seguinte conjunto de pares:

$$A \times B = \{(1,a), (1,b), (2,a), (2,b), (3,a), (3,b)\}$$

Note a diferença do produto $A \times B$ em relação ao produto $B \times A$ que resulta no seguinte conjunto de pares:

$$B \times A = \{(a,1), (a,2), (a,3), (b,1), (b,2), (b,3)\}$$

Como A e B são conjuntos distintos então o produto $A\times B$ e o produto $B\times A$ resultam em conjuntos distintos, apesar de terem a mesma cardinalidade (o mesmo número de elementos, quando A e B são conjuntos finitos). Isso ocorre porque pares ordenados são diferentes de conjuntos de dois elementos: o par (1,a) é diferente do par (a,1);

Supondo que D_1 e D_2 sejam usados como domínios individuais das duas variáveis do predicado, então pode-se considerar o domínio conjunto do predicado como o produto cartesiano destes domínios individuais: $D_1 \times D_2$.

Dessa forma um **predicado com duas variáveis em um domínio $D_1 \times D_2$**, ou simplesmente um **predicado em $D_1 \times D_2$**, é uma expressão P(x, y) tal que P(a,b) é verdadeira (*V*) ou falsa (*F*) para toda dupla $(a,b) \in D_1 \times D_2$.

O **conjunto-verdade** de uma sentença aberta P(x,y) no domínio $D_1 \times D_2$ é o conjunto de todas as duplas $(a,b) \in D_1 \times D_2$ tais que P(a,b) é uma proposição **verdadeira**. Formalmente esse conjunto-verdade pode ser definido como:

$$V_P = \{(x,y) \in D_1 \times D_2 \mid P(x,y)\}$$

Exemplo:

O conjunto-verdade de P(x,y) = “x + y ≤ 3” para x e y pertencentes ao conjunto $N = \{1, 2, 3, ...\}$ dos números naturais é dado por:

$$V_P = \{(x,y) \in N \times N \mid P(x,y)\} = \{(x,y) \in N \times N \mid x + y \leq 3\} = \{(1,1), (1,2), (2,1)\}$$

Predicados de mais de duas variáveis também são possíveis. Na verdade, basta generalizar a ideia de produto cartesiano usada para definir os domínios dos predicados de duas variáveis para o caso em que são usadas três ou mais variáveis, ou seja, é possível se definir predicados que utilizem quaisquer números de variáveis (pertencentes ao mesmo domínio ou a domínios diferentes).

Predicados de duas variáveis são chamados *binários*. Predicados de três variáveis são predicados *ternários* e assim por diante. No caso geral, predicados que utilizem **n** variáveis são chamados de predicados *n-ários*. Da mesma forma que no caso dos predicados binários, quaisquer predicados que utilizem duas ou mais variáveis também são denominados *relações* entre estas variáveis.

O significado atribuído para predicados n-ários é definido através do *produto cartesiano* dos domínios das suas variáveis.

O produto cartesiano de dois conjuntos: A × B é o conjunto formado por todos as *duplas* ordenadas (a, b) onde $a \in A$ e $b \in B$. Generalizando para o caso de *n* conjuntos $A_1 \times A_2 \times \ldots \times A_n$, tem-se como produto cartesiano o conjunto das *n-uplas* $(a_1, a_2, \ldots, a_n)$ onde cada $a_i \in A_i$ para $1 \leq i \leq n$.

Supondo *n* conjuntos $D_1, D_2, \ldots, D_n$ que serão usados como domínios individuais das variáveis de um predicado $P(x_1, x_2, \ldots, x_n)$ com *n* variáveis. Neste caso, pode-se considerar como o domínio de todas as variáveis o conjunto resultante do produto cartesiano destes domínios individuais: $D_1 \times D_2 \times \ldots \times D_n$

Dessa forma, um predicado com *n* variáveis em um domínio $D_1 \times D_2 \times \ldots \times D_n$, é uma expressão $P(x_1, x_2, \ldots, x_n)$ tal que $P(a_1, a_2, \ldots, a_n)$ será verdadeira (*V*) ou falsa (*F*) para toda a *n*-upla $(a_1, a_2, \ldots, a_n) \in D_1 \times D_2 \times \ldots \times D_n$.

O conjunto-verdade de uma sentença aberta $P(x_1, x_2, \ldots, x_n)$ no domínio $D_1 \times D_2 \times \ldots \times D_n$ é o conjunto de todas as n-uplas $(a_1, a_2, \ldots, a_n) \in D_1 \times D_2 \times \ldots \times D_n$ tais que $P(a_1, a_2, \ldots, a_n)$ é uma proposição **verdadeira**. Formalmente esse conjunto-verdade pode ser definido de uma forma muito similar ao conjunto verdade para predicados de duas variáveis, apenas que agora generalizado para o caso de *n* variáveis:

$$V_P = \{(x_1, x_2, \ldots, x_n) \in D_1 \times D_2 \times \ldots \times D_n \mid P(x_1, x_2, \ldots, x_n)\}$$

Exemplo:

O conjunto-verdade de $Q(x,y,z) = $ "$x + y \leq z$" para $x \in A$, $y \in B$ e $z \in C$, com $A=\{1,2\}$, $B=\{2,3\}$ e $C=\{3,4\}$ é dado por:

$$V_Q = \{(x,y) \in A \times B \times C \mid Q(x,y,z)\} = \{(x,y,z) \in A \times B \times C \mid x + y \leq z\}$$

Mas como o domínio de $Q(x,y,z)$ é:

$$A \times B \times C = \{1,2\} \times \{2,3\} \times \{3,4\} = \{(1,2,3),(1,2,4),(1,3,3),(1,3,4),\\ (2,2,3),(2,2,4),(2,3,3),(2,3,4)\}$$

Temos:

$$\{(x,y,z) \in A \times B \times C \mid x + y \leq z\} = \{(1,2,3), (1,2,4), (2,2,4)\}$$

Portanto:

$$V_Q = \{(1,2,3), (1,2,4), (2,2,4)\}$$

3.10 Questões comentadas

Encontrar o conjunto-verdade considerando as sentenças abertas $P(x) = x^2 \in C$ e $Q(x) = 2x + 2 < 15$ sobre o domínio $C = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

O conjunto-verdade de $P(x)$, representado por V_P, é construído a partir dos valores de C que substituído na sentença faz com que ela fique verdadeira. Da mesma forma, constrói-se o conjunto-verdade V_Q.

$$V_P = \{1, 2, 3\}$$
$$V_Q = \{1, 2, 3, 4, 5, 6\}$$

$$\begin{aligned} V_{P \wedge Q} &= V_P \cap V_Q \\ &= \{1, 2, 3\} \cap \{1, 2, 3, 4, 5, 6\} \\ &= \{1, 2, 3\} \end{aligned}$$

$$\begin{aligned} V_{P \vee Q} &= V_P \cup V_Q \\ &= \{1, 2, 3\} \cup \{1, 2, 3, 4, 5, 6\} \\ &= \{1, 2, 3, 4, 5, 6\} \end{aligned}$$

$$\begin{aligned} V_{\neg P} &= C - V_P \\ &= C - \{1,2,3\} \\ &= \{1,2,3, \ldots, 10\} - \{1,2,3\} \\ &= \{4,5, 6, 7, 8, 9, 10\} \end{aligned}$$

$$\begin{aligned} V_{\neg Q} &= C - V_Q \\ &= C - \{1,2,3, 4, 5, 6\} \\ &= \{1,2,3, \ldots, 10\} - \{1,2,3, 4, 5, 6\} \\ &= \{7, 8, 9, 10\} \end{aligned}$$

$$\begin{aligned} V_{P \rightarrow Q} &= V_{\neg P} \cup V_Q \\ &= \{4, 5, 6, 7, 8, 9, 10\} \cup \{1, 2, 3, 4, 5, 6\} \\ &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \end{aligned}$$

$$\begin{aligned} V_{Q \rightarrow P} &= V_{\neg Q} \cup V_P \\ &= \{7, 8, 9, 10\} \cup \{1, 2, 3\} \\ &= \{1, 2, 3, 7, 8, 9, 10\} \end{aligned}$$

$$
\begin{aligned}
V_{P \leftrightarrow Q} &= (V_{\neg P} \cup V_Q) \cap (V_{\neg Q} \cup V_P) \\
&= (\{4, 5, 6, 7, 8, 9, 10\} \cup \{1, 2, 3, 4, 5, 6\}) \cap (\{7, 8, 9, 10\} \cup \{1, 2, 3\}) \\
&= (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) \cap (1, 2, 3, 7, 8, 9, 10) \\
&= \{1, 2, 3, 7, 8, 9, 10\}
\end{aligned}
$$

CAPÍTULO 4

QUANTIFICADORES

Quantificadores são *operadores* lógicos aplicados a *uma variável* e a *uma expressão* (uma sentença aberta simples ou uma fórmula composta de várias sentenças).

Os quantificadores foram definidos para capturar conceitos da linguagem natural como:

- Para todo mundo...
- Não tem ninguém aqui que...
- Todos aqui...
- Tem alguém que poderia...
- Qualquer um que...
- Existe pelo menos um de nós...

Estas orações exprimem afirmações que são verdadeiras para vários elementos do domínio. No caso da lógica de predicados somente são considerados dois tipos de afirmações gerais sobre os elementos de um domínio:

- Afirmações universais, que devem ser válidas para todos os elementos de um domínio.
- Afirmações existenciais, que devem ser válidas para pelo menos um dos elementos do domínio.

Para cada um destes tipos de afirmações, corresponde um diferente tipo de quantificador:

- Quantificadores universais, para representar as afirmações universais.
- Quantificadores existenciais, para representar as afirmações existenciais.

A combinação das sentenças abertas (predicados) com quantificadores dará origem à Lógica de Predicados.

4.1 Quantificador universal

O *quantificador universal* é usado para representar orações similares a:

- Todos os cidadãos devem respeitar a lei.
- Educação de qualidade para todos os brasileiros.

- Quaisquer que sejam as consequências, deve-se resolver o problema.
- Para qualquer número primo, sempre haverá um número primo maior.

Este tipo de oração tem um formato similar ao das seguintes expressões:

- Todos os ...
- ... para todos ...
- Qualquer um que ...
- Para qualquer ...
- Quaisquer que ...

Todas estas orações expressam *afirmações universais* que são verdadeiras para todos os sujeitos envolvidos na oração. Assim o quantificador universal é utilizado para representar essas afirmações universais. Para tanto, esse quantificador deve ser aplicado sobre a variável x de uma sentença aberta P(x) cujo domínio é D. Supondo que V_P seja o conjunto-verdade de P(x). Dessa forma, quando V_P for igual ao domínio D (isto é, V_P=D), então **todos** os elementos de D irão obrigatoriamente satisfazer P(x), ou seja, para todos os elementos de D, P(x) deve ser verdadeira. Isso pode ser expresso por afirmações como:

Para todo $x \in$ D , tem-se que P(x) é verdadeira, ou,
Qualquer que seja o $x \in$ D, tem-se que P(x) é verdadeira, ou na sua forma mais simples:
Para todo x $\in$ A , P(x).

O símbolo $\forall$ é usado para representar o quantificador universal. Com esse símbolo a quantificação universal da variável x do predicado P(x) com domínio D é formalizada pela seguinte expressão (fórmula):

$$(\forall x \in D)\ (P(x))$$

que pode ser simplificado para:

$$(\forall x)\ (P(x))$$

quando o domínio D está claro pelo contexto ou é desnecessário.

O quantificador universal é um novo operador que deve ser aplicado a uma variável, um domínio (opcional) e um predicado. Nos exemplos foi usada a variável x e o predicado $P(x)$ mas qualquer variável ou predicado podem ser utilizados.

Pela definição de quantificação universal deve ter ficado claro que o significado desse operador, em termos do domínio D e do conjunto-verdade de uma sentença $P(x)$, é o de afirmar uma igualdade entre ambos conjuntos, isto é, afirmação:

$$(\forall x \in D)\ (P(x))$$

é equivalente a dizer que:

$$V_P = D$$

ou seja,

$$(\forall x \in D)\ (P(x)) \Leftrightarrow V_P = D$$

Graficamente esta relação pode ser representada como:

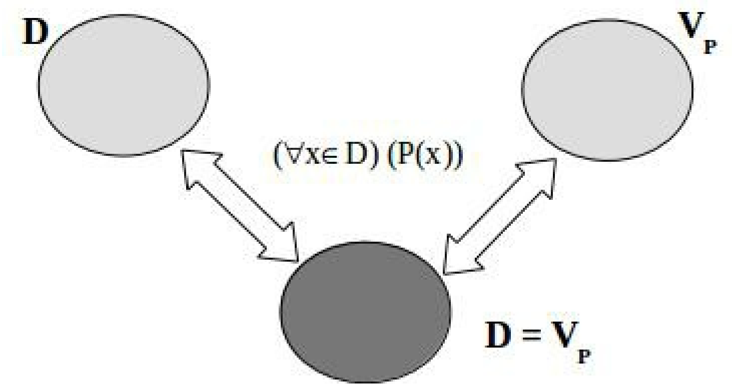

Figura 9 – Quantificação universal, domínio e conjunto verdade.
Fonte: elaborada pelos autores.

É importante salientar que enquanto P(x) é uma sentença aberta, a **sentença quantificada** $(\forall x \in A)\ (P(x))$ **não é mais uma sentença aberta**. A quantificação “fecha” uma sentença aberta, transformando-a em uma proposição simples que pode ser verdadeira ou falsa no domínio D, dependendo do conjunto-verdade V_P ser ou não igual ao domínio D.

Em outras palavras, dada uma sentença aberta P(x) em um domínio, o operador ∀ representa uma operação lógica que transforma a sentença aberta P(x) em uma proposição que é verdadeira ou não, dependendo de P(x) ser ou não uma condição universal sobre o domínio.

Em particular, quando o número de elementos do domínio D é finito, com $D=\{a_1,a_2,...,a_n\}$, então a proposição $(\forall x \in A)(P(x))$ é equivalente à conjunção das *n* proposições $P(a_1)$, $P(a_2)$, ..., $P(a_n)$, ou seja:

$$(\forall x \in D)(P(x)) \Leftrightarrow P(a_1) \wedge P(a_2) \wedge \ldots \wedge P(a_n)$$

Exemplos:

Afirmações universais verdadeiras:

$(\forall x \in H)$ (x é mortal), no domínio H do conjunto de seres humanos.
$(\forall x \in N)$ $(x + 2 > x)$, no domínio N do conjunto dos números naturais.
$(\forall x \in A)$ $(x < 7)$, para $A = \{0,1,2,3,4\}$

Afirmações universais falsas:

$(\forall x \in H)$ (x é mãe), no domínio H do conjunto de seres humanos.
$(\forall x \in N)$ $(x + 2 > 2x)$, para N o conjunto dos números naturais.
$(\forall x \in A)$ $(x \in N)$, para $A = \{0, 1, 2, 3, -3, 2.5, 4, 0.999, \pi\}$ e N o conjunto dos números naturais.

4.2 Quantificador existencial

O *quantificador existencial* é usado para representar as *afirmações existenciais* que no português são expressas por orações similares a:

- Alguém deve abrir a porta.
- Existe algo melhor que isto.
- Este produto deve ser bom, para algum móvel de madeira.
- Existe pelo menos um que sabe a verdade.

Estas afirmações têm formato similares aos seguintes:

- Alguém ...
- ... para algum ...
- Existe pelo menos um ...

Uma afirmação existencial diz que algum possível sujeito da frase deve tornar essa afirmação verdadeira, ou seja, deve existir pelo menos um sujeito (objeto, elemento ou "coisa") que satisfaça a condição expressa pelo predicado da frase. Da mesma forma que no caso da quantificação universal, também existe um operador de quantificação existencial, que pode ser aplicado sobre uma variável x, um domínio D e uma sentença aberta P(x) para representar afirmações existenciais nesse domínio.

Supondo que V_P seja o conjunto-verdade de P(x). Dessa forma, quando V_P não for igual ao conjunto vazio $\varnothing$ (isto é, $V_P \neq \varnothing$), então com certeza **existe algum** elemento de D que irá satisfazer P(x), ou seja, para algum elemento de D, P(x) deve ser verdadeira. Isso pode ser expresso um pouco mais formalmente como:

Para algum x $\in$ A, P(x) é verdadeira, ou ,
Existe pelo menos um x $\in$ A, no qual P(x) é verdadeira, ou simplesmente por,
Existe x $\in$ A, tal que P(x).

Usando o símbolo $\exists$ para o quantificador existencial, estas afirmações são expressas pela seguinte fórmula:

$$(\exists x \in D) (P(x))$$

que às vezes é simplificada para:

$$(\exists x)\ (P(x))$$

quando o domínio D está claro pelo contexto ou é desnecessário.

Pela definição da quantificação existencial deve estar claro que o significado desse operador, em termos do domínio e do conjunto-verdade de uma sentença P(x), é o de afirmar que o conjunto-verdade não pode ser vazio, isto é, afirmação:

$$(\exists x \in D)\ (P(x))$$

é equivalente a dizer que:

$$V_P \neq \varnothing$$

ou seja,

$$(\exists x \in D)\ (P(x)) \Leftrightarrow V_P \neq \varnothing$$

Graficamente esta relação pode ser representada como:

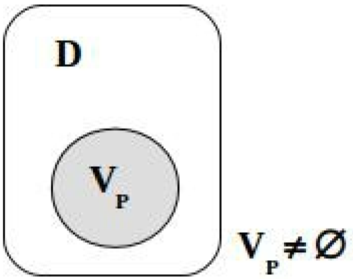

Figura 10 – Quantificação existencial, domínio e conjunto vazio.
Fonte: elaborada pelos autores.

Da mesma forma que no caso do quantificador universal, também no caso do quantificador existencial tem-se que, embora P(x) seja uma sentença aberta, a **sentença quantificada** $(\exists x \in A)\ (P(x))$ **não é mais uma sentença aberta**. A quantificação “fecha” uma sentença aberta, transformando-a em uma proposição simples que pode ser verdadeira ou falsa no domínio A, dependendo de V_P ser ou não vazio.

Em outras palavras, dada uma sentença aberta P(x) em um domínio, o operador $\exists$ representa uma operação lógica que transforma a sentença aberta P(x) em uma proposição que é verdadeira ou não, dependendo de P(x) ser ou não uma condição possível sobre o domínio.

Em particular, quando o número de elementos do domínio D é finito, com $A=\{a_1,a_2,...,a_n\}$, então é óbvio que a proposição $(\exists x \in A)$ P(x) é equivalente à disjunção das n proposições $P(a_1)$, $P(a_2)$, ..., $P(a_n)$:

$$(\exists x \in D)\ P(x) \Leftrightarrow P(a_1) \vee P(a_2) \vee \ldots \vee P(a_n)$$

Exemplos:

Afirmações existenciais verdadeiras:

$(\exists x \in H)$ (x é pai), para H o conjunto de seres humanos.
$(\exists x \in N)$ $(x + 2 > 2x)$, para N o conjunto dos números naturais.
$(\exists x \in A)$ $((500x + 3) \in A)$, para $A = \{0,1,2,3,4\}$

Afirmações existenciais falsas:

$(\exists x \in H)$ (x tem mais de 300 anos de idade), para H o conjunto de seres humanos.
$(\exists x \in N)$ $(x + 1 = x)$, para N o conjunto dos números naturais.

Exercício 4.2.1

Sendo R o conjunto dos números reais, determinar o valor lógico das seguintes expressões:

a) $(\forall x \in R)$ $(|\ x\ | = x)$
b) $(\exists x \in R)$ $(x^2 = x)$
c) $(\exists x \in R)$ $(|\ x\ | = 0)$
d) $(\exists x \in R)$ $(x + 2 = x)$
e) $(\forall x \in R)$ $(x + 1 > x)$
f) $(\forall x \in R)$ $(x^2 = x)$

Para $|x|$ a função **módulo de x**, que é calculada como:

$| x | = x$, se $x \geq 0$
$| x | = -x$, se $x < 0$

4.3 Precedência e escopo dos quantificadores

Um quantificador existencial ou universal é um operador lógico aplicado a uma variável, a um domínio e a uma fórmula. Tanto a variável quanto o domínio (que pode opcionalmente ser omitido) estão claramente indicados na fórmula porque são definidos dentro dos parênteses do quantificador. Por exemplo, o quantificador existencial ($\exists x \in D$) se aplica sobre a variável x pertencente a um domínio D, enquanto o quantificador universal ($\forall y$) se aplica sobre a variável y cujo domínio deve ser inferido pelo contexto.

Porém, quando um quantificador ocorre em uma fórmula, esse influencia a parte da fórmula onde foi aplicado. A extensão dessa influência é chamada **escopo** do quantificador e depende da **precedência** do quantificador.

A **precedência** tanto do quantificador existencial quanto do universal é **alta**, **equivalente** à precedência do operador de **negação** ($\neg$).

A seguir são mostrados exemplos de escopo de quantificadores:

- Na fórmula $\exists x\ P(x)$, onde $P(x)$ é uma sentença aberta ou predicado simples, então $P(x)$ é o escopo do quantificador $\exists x$.
- Da mesma forma, na fórmula $\forall x\ P(x)$, onde $P(x)$ é uma sentença aberta ou predicado simples, então $P(x)$ é o escopo do quantificador $\forall x$.
- Porém no caso da fórmula $\exists x\ P(x,y) \rightarrow Q(x)$, o escopo do quantificador $\exists x$ é $P(x,y)$, pois pelas regras de precedência esta fórmula é equivalente a $(\exists x\ P(x,y)) \rightarrow Q(x)$.
- Por outro lado, por conta do uso dos parênteses o escopo do quantificador $\exists x$ na fórmula $\exists x\ (P(x,y) \rightarrow Q(x))$ é $P(x,y) \rightarrow Q(x)$.

4.4 Variáveis quantificadas (aparentes) e variáveis livres

Quando uma variável x está dentro do escopo de um quantificador sobre esta mesma variável, então se diz que é uma **variável quantificada** (as vezes esta variável é denominada **variável aparente**). O termo variável aparente dadas as variáveis quantificadas vem do fato de uma variável quantificada não se comportar realmente como uma variável, ou seja, ela está comprometida pelo quantificador a uma determinada associação universal ou existencial com os elementos do domínio. Não se esqueça que uma sentença aberta quantificada não é realmente uma sentença aberta, mas uma proposição lógica fechada que pode ser apenas verdadeira ou falsa.

Por outro lado, se uma variável em uma fórmula não estiver sobre o escopo de nenhum quantificador, então se diz que ela é uma variável livre.

Uma ocorrência da variável x em uma fórmula também é dita **ligada** se ela está no escopo de quantificador $\exists x$ ou $\forall x$. Caso contrário, uma ocorrência de x é dita livre na fórmula.

Exemplos:

1) $\exists x\ P(x,y) \rightarrow Q(x)$

As duas primeiras ocorrências de x são ligadas ou quantificadas, pois o escopo de $\exists x$ é $P(x,y)$. A única ocorrência de y é livre e a terceira ocorrência de x é livre.

2) $\forall x\ (((P(x) \land Q(y)) \rightarrow \exists y\ Q(y))$

As duas ocorrências de x são ligadas. O escopo de $\forall x$ é $(P(x) \land Q(y)) \rightarrow \exists y\ Q(y)$ e o escopo do $\exists y$ é $Q(y)$. A primeira ocorrência de y é livre.

4.5 Quantificação múltipla e parcial

Uma fórmula pode ter tantos quantificadores quanto o número de variáveis diferentes dentro da fórmula. Assim, para R o conjunto dos números reais são possíveis fórmulas como:

$$(\exists x \in R)\ (\exists y \in R)\ (x^2 + y^2 + 2x + xy > 0)$$
$$(\exists x \in R)\ (\forall y \in R)\ (\exists z \in R)\ (x^2 + y^2 + z^3 - yz + x = 0)$$

que estão totalmente quantificadas, isto é, que não têm variável sem quantificação.

Contudo, nem todas as variáveis de uma fórmula precisam estar quantificadas. Quando nem todas as variáveis de uma fórmula estão quantificadas, se diz que esta fórmula está **parcialmente quantificada**. Por exemplo, as seguintes fórmulas em R, o conjunto dos número reais:

$$(\exists x \in R)\ (x^2 + y^2 = 0)$$
$$(\exists x \in R)\ (\forall y \in R)\ (x^2 + y^2 + 2z = 0)$$

estão parcialmente quantificadas uma vez que existe pelo menos uma variável em cada fórmula que não foi previamente quantificada.

Importante: Uma fórmula **parcialmente quantificada** continua sendo uma **sentença aberta** nas variáveis que não foram quantificadas.

4.6 Propriedades dos quantificadores

Equivalência das variáveis quantificadas ou aparentes

Um princípio simples, mas válido para a manipulação de expressões lógicas ou fórmulas compostas de sentenças quantificadas, afirma que todas as vezes que uma variável quantificada é substituída em todos os lugares onde aparece em uma expressão por outra variável que não apareça nesta mesma expressão, então a expressão resultante é equivalente.

Esse princípio garante a equivalência das seguintes fórmulas lógicas.

$\forall x$ (x é mortal) $\Leftrightarrow$ $\forall y$ (y é mortal) $\Leftrightarrow$ ($\forall$ pessoa) (pessoa é mortal) $\Leftrightarrow$ ...
$\exists x$ (x foi à Lua) $\Leftrightarrow$ $\exists y$ (y foi à Lua) $\Leftrightarrow$ ($\exists$ pessoa) (pessoa foi à Lua) $\Leftrightarrow$ ...

Negação de fórmulas com quantificadores

Qualquer expressão ou fórmula lógica quantificada também pode ser precedida do operador de negação ($\neg$). Por exemplo, considerando o domínio das pessoas atualmente vivas as expressões formais:

$\forall x$ (x fala inglês)
$\neg \forall x$ (x fala inglês)
$\exists x$ (x foi à Antártida)
$\neg \exists x$ (x foi à Antártida)

poderiam ser enunciadas, respectivamente, como:

Todas as pessoas falam inglês.
Nem todas as pessoas falam inglês.
Alguém foi a Antártida.
Ninguém foi a Antártida.

Analisando estas expressões, pode-se inferir algumas equivalências intuitivas. Por exemplo, afirmar que nem todas as pessoas falam inglês é claramente equivalente a afirmar que existe alguém que não fala inglês, ou seja:

$\neg \forall x$ (x fala inglês) $\Leftrightarrow$ $\exists x \neg$ (x fala inglês)

Afirmar que ninguém foi à Antártida também é claramente equivalente a afirmar que para qualquer pessoa não é verdade que essa pessoa tenha ido à Antártida:

$\neg \exists x$ (x foi à Antártida) $\Leftrightarrow$ $\forall x \neg$(x foi à Antártida)

Esta análise pode ser generalizada pelas seguintes regras:

(i) A **negação** da fórmula $\forall x\ P(x)$ é equivalente à afirmação de que **pelo menos para um** $x \in D$ tem-se que $P(x)$ é falsa, ou então que $\neg P(x)$ é verdadeira. Portanto devem valer as seguintes equivalências:

$$\neg \forall x \in D\ P(x) \Leftrightarrow \exists x \in D\ \neg P(x)$$

$$\neg \forall x \in D\ \neg P(x) \Leftrightarrow \exists x \in D\ P(x)$$

(ii) Da mesma forma, negar a fórmula $\exists x\ P(x)$ equivale a afirmar que para todos os $x \in D$ a sentença $P(x)$ deve ser falsa, ou então que a sentença $\neg P(x)$ deve ser verdadeira, o que nos leva às seguintes equivalências:

$$\neg \exists x \in D\ P(x) \Leftrightarrow \forall x \in D\ \neg P(x)$$

$$\neg \exists x \in D\ \neg P(x) \Leftrightarrow \forall x \in D\ P(x)$$

As duas regras acima expressam relações entre os dois quantificadores, são conhecidas como **regras de equivalência dos quantificadores** (**equiv**).

Comutatividade de quantificadores

Os quantificadores de uma fórmula somente podem ser comutados de acordo com as seguintes regras:

(i) Quantificadores de **mesmo tipo podem ser comutados**. Portanto, a seguinte equivalência é válida:

$$\exists x\ \exists y\ P(x,y) \Leftrightarrow \exists y\ \exists x\ P(x,y)$$

e também é válida a equivalência:

$$\forall x\ \forall y\ P(x,y) \Leftrightarrow \forall y\ \forall x\ P(x,y)$$

e outras equivalências similares.

(ii) Quantificadores de **tipos distintos não podem ser comutados**.

4.7. Simbolização de predicados e nomes na Lógica de Predicados

A combinação das sentenças abertas (predicados) com quantificadores dá origem à **Lógica de Predicados** (também chamada de **Lógica de Primeira Ordem**) que será estudada com mais detalhes no próximo capítulo. Apesar de ser inteiramente formal, a Lógica de Predicados deve muito de suas características à Lógica tradicionalmente estudada nos textos de Filosofia. Essa lógica tradicional tem suas origens na Lógica Aristotélica e também na Lógica Escolástica, criada pelos filósofos medievais. Tais Lógicas trabalham com sentenças em linguagem natural e com a noção de enunciados categóricos. Assim vale a pena começar a entender como é possível passar de orações e sentenças na Língua Portuguesa (ou em outras linguagens naturais) para fórmulas da Lógica de Predicados e vice-versa. O objetivo desta seção e das demais seções deste capítulo é justamente introduzir esse tema de estudo.

Para começar, nem todas as sentenças possuem quantificadores, mas ainda assim podem ser representadas (simbolizadas) na Lógica de Predicados. Por exemplo, na sentença:

"*Morpheus é o deus grego dos sonhos*".

a palavra "Morpheus" não é um predicado, e sim um nome próprio. Os nomes próprios devem ser representados por símbolos de **constantes**. Diferente das variáveis, que são representadas por letras minúsculas do fim do alfabeto (x,y,z), as constantes (ou nomes próprios) são representados por letras minúsculas do início e do meio do alfabeto (a,b,c,...).

Diferente da linguagem natural onde o sujeito normalmente vem antes do predicado, na Lógica de Predicados se escreve o sujeito depois do predicado, entre parênteses. Assim, a sentença vista acima pode ser formalizada simplesmente por:

$$D(m)$$

Onde a constante m é interpretada como o nome próprio Morpheus e D(x) como o predicado "x é deus grego dos sonhos".

No exemplo a seguir:

"*Se Nemo é um peixe, então é capaz de nadar*".

Nemo também é um nome próprio. Assumindo *n* como a constante para Nemo, P(x) como o predicado "x é um peixe" e N(x) como o predicado "x é capaz de nadar", então este exemplo pode ser simbolizado pela seguinte fórmula:

$$P(n) \rightarrow N(n)$$

É importante salientar que existem enunciados que aplicam verbos transitivos, como amar, bater, odiar, que exigem sujeito e objeto. Neste caso, a formalização considera a ordem "predicado-sujeito-objeto". Os nomes (sujeito e objeto) são representados por letras minúsculas, enquanto os predicados são representados por letras maiúsculas. Vejamos alguns exemplos:

- Afrodite é deusa grega $D(a)$
- Afrodite não ama Zeus $\neg A(a,z)$
- Zeus ama Afrodite $A(z,a)$
- Afrodite ama a si mesma $A(a,a)$
- Zeus deu joias para Afrodite $D(z,j,a)$

Nestes exemplos, as constantes a, z e j representam, respectivamente, Afrodite, Zeus e joias, enquanto o predicado $D(x)$ representa o fato de x ser um deus grego, $A(x,y)$ indica a relação "x ama y" e $D(x,y,z)$ a relação de três elementos: "x deu um objeto y para z". Um predicado que exige somente um nome (um sujeito) é chamado de predicado unário. Predicado que exige dois nomes (um sujeito e um objeto) é chamado de predicado binário. Já o predicado que exige três nomes (um sujeito, um objeto e um complemento) é chamado de predicado ternário e assim por diante. Diferente da linguagem natural, a Lógica de Predicados pode ter predicados com quatro, cinco ou mais variáveis.

A formalização de expressões com quantificadores existenciais como "alguém", "algum", "existe" etc., ou quantificadores universais como "todos", "quaisquer" etc. exige muita atenção por conta da necessidade de identificar a posição correta das variáveis quantificadas. Na próxima seção serão vistas as formas mais gerais de simbolização deste tipo de expressão, denominadas Enunciados Categóricos. Porém, em alguns casos mais simples é possível identificar qual a fórmula correspondente à expressão. A seguir são apresentados alguns exemplos de formalização de expressões com quantificadores, relacionados aos predicados sobre deuses gregos vistos anteriormente:

- Alguém ama Afrodite $\exists x\, A(x,a)$
- Alguém ama a si mesmo $\exists x\, A(x,x)$

- Zeus ama ninguém $\forall x \quad \neg A(z,x)$ ou $\neg \exists x$
- Alguém ama alguém $A(z,x)$
- Existe um x tal que para todo y x $\exists x\ \exists y\ A(x,y)$ ama y $\exists x\ \forall y\ A(x,y)$
- Todos amam todos $\forall x\ \forall y\ A(x,y)$

4.8 Simbolização de enunciados categóricos na Lógica de Predicados

Enunciado categórico é uma afirmação (uma frase) da linguagem natural, mas que possui uma estrutura mais ou menos formal. Esses enunciados permitem definir em linguagem natural expressões formadas por quantificadores e predicados. Os componentes básicos que formam um enunciado categórico são os tradicionais elementos estudados na gramática das linguagens naturais: o *sujeito* e o *predicado* da frase. Também compõem um enunciado categórico palavras que definem o tipo de quantificação sendo usada no enunciado.

Existem quatro tipos clássicos de enunciados categóricos, tradicionalmente identificados pelas letras A, E, I e O:

Tipo	**Formato geral do enunciado**
A	Todo *S* é *P*
E	Nenhum *S* é *P*
I	Algum *S* é *P*
O	Algum *S* não é *P*

Nestes enunciados S representa o sujeito da frase e P o predicado. O predicado representa sempre uma classe de elementos ou de entidades onde o sujeito deve pertencer. Assim, são exemplos válidos de enunciados as seguintes frases, uma para cada tipo de enunciado:

A: Todo *ser humano* é *mortal*.
E: Nenhuma *fruta* é *de gosto amargo*.
I: Algum *pássaro* é *capaz de nadar*.
O: Algum *móvel da sala* não é *feito de madeira*.

A transformação desse tipo de enunciado para a lógica simbólica é feita levando em conta primeiro que todos os enunciados categóricos forçam que entidades que atendem as características ou propriedades do sujeito devem também atender as características ou propriedades do predicado.

Por exemplo, ao se enunciar que Todo *S* é *P*, na verdade se está afirmando que toda entidade *S* também é *P*. Isso leva a uma primeira transformação dos enunciados acima em um formato um pouco mais formal, onde estas entidades são indicadas pela variável *x*:

A_1: Todo *x* que é *ser humano* também é *mortal*.

E_1: Nenhum *x* que é *fruta* também é *de gosto amargo*.

I_1: Algum *x* que é *pássaro* também é *capaz de nadar*.

O_1: Algum *x* que é *móvel da sala* também não é *feito de madeira*.

As afirmações anteriores estão mais próximas da lógica simbólica, mas ainda falta esclarecer alguns detalhes relativos ao tipo de quantificador empregado em cada tipo de enunciado e também ao significado atribuído à palavra “também” usada nas afirmações.

As palavras “todo” e “algum” são referências diretas, respectivamente, aos quantificadores universais (“para todo”) e existencial (“existe pelo menos um”). Porém a forma (E) “Nenhum S é P” não designa diretamente um quantificador, mas corresponde, na verdade, a afirmar que todos os *x* que são *S* não podem ser *P*. Isso permite a utilização do quantificador universal para representar o tipo (E).

A palavra “também” indica que há uma relação lógica entre a afirmação relativa ao sujeito e a afirmação relativa ao predicado. Neste caso, considera-se que esta relação seja de consequência (implicação) lógica no caso das quantificações universais e de conjunção lógica no caso das quantificações existenciais.

Levando isso em conta, os exemplos acima podem ser transformados nas seguintes afirmações, quase simbólicas:

A_2: Para todo *x*, se *x* é *ser humano*, então *x* é *mortal*.

E_2: Para todo *x*, se *x* é *fruta*, então *x* não é *de gosto amargo*.

I_2: Existe pelo menos um *x* tal que *x* é *pássaro* e *x* é *capaz de nadar*.

O_2: Existe pelo menos um *x* tal que *x* é *móvel da sala* e *x* não é *feito* *de madeira*.

O resultado final pode ser totalmente simbolizado, se assumirmos como domínio D dos enunciados categóricos vistos acima, o conjunto formado pelos seres vivos e pelas "coisas" (seres inanimados, objetos, móveis, etc.) existentes na Terra. Neste caso os predicados simbólicos $H(x)$, $M(x)$, $F(x)$, $A(x)$, $R(x)$, $N(x)$, $S(x)$, $D(x)$ e $P(x)$ representam, respectivamente, as sentenças abertas "*x* é *ser humano*", "*x* é *mortal*", "*x* é *fruta*", "*x* é *de gosto amargo*", "*x* é *réptil*", "*x* é *capaz de nadar*", "*x* é *móvel da sala*", "*x* não é *feito de madeira*" e "*x* é *peixe*". Assim, os enunciados categóricos seriam formalizados pelas seguintes fórmulas da Lógica de Predicados:

A_3: $\forall x \in D\ (H(x) \rightarrow M(x))$
E_3: $\forall x \in D\ (F(x) \rightarrow \neg A(x))$
I_3: $\exists x \in D\ (P(x) \wedge N(x))$
O_3: $\exists x \in D\ (S(x) \wedge \neg D(x))$

4.9 Exemplos e dicas de simbolização

Nos exemplos a seguir vamos formalizar as sentenças de acordo com as formas apresentadas anteriormente.

Exemplo 1:

As sentenças:

"Todo gato é mimoso" ou
"Cada gato é mimoso".

Têm o formato do enunciado categórico A e significam, portanto, que, "dada uma coisa, se esta coisa é gato, então esta coisa é mimosa". Usando os predicados $G(x)$ = "x é um gato" e $M(x)$ = "x é mimoso" essas sentenças podem ser simbolizadas por:

$$\forall x\ (G(x) \rightarrow M(x))$$

Exemplo 2:

As sentenças:

"Existe um gato mimoso",
"Alguns gatos são mimosos" ou
"Existem gatos mimosos".

Seguem o formato do enunciado categórico I e significam que, "existe alguma coisa que é, ao mesmo tempo, gato e mimoso". Usando os predicados G(x) e M(x) do exemplo anterior, todos esses enunciados são simbolizados por:

$$\exists x\ (G(x) \land M(x))$$

Exemplo 3:

A afirmação:

"Todo número inteiro par é divisível por 2".

Pode ser formalizada por:

$$\forall x\ (P(x) \rightarrow D(x))$$

usando os predicados P(x) = "x é um número inteiro par" e D(x) = "x é divisível por 2".

Exemplo 4:

Usando os predicados:

- A(x) = "x é um aluno",
- E(x) = "x é estudioso" e
- P(x) = "x é aprovado na disciplina de Lógica".

A sentença:

"Todo aluno estudioso é aprovado na disciplina de Lógica"

pode ser formalizada por:

$$\forall x\ (A(x) \land E(x) \rightarrow P(x))$$

Exemplo 5:

Os predicados:

- $G(x)$ = "x é um gato".
- $I(x)$ = "x é independente".
- $C(x)$ = "x é um cachorro".
- $B(x)$ = "x é bagunceiro".
- $P(x)$ = "x é peludo".
- $E(x,y)$ = "x é mais esperto do que y".

E a constante:

- g = "Garfield".

São usados para formalizar as seguintes sentenças:

Exemplo 5.1:

Os gatos são independentes e os cachorros são bagunceiros.

$$\forall x\ (G(x) \rightarrow I(x)) \land \forall x\ (C(x) \rightarrow B(x))$$

Exemplo 5.2:

Se todos os gatos são independentes, então não existem gatos baguncieros.

$$\forall x\ (G(x) \rightarrow I(x)) \rightarrow \neg \exists x\ (G(x) \land B(x))$$

Exemplo 5.3:

Nem todos os gatos são independentes, nem há gatos mais espertos que o Garfield que sejam peludos.

$$\neg \forall x\ (G(x) \rightarrow I(x)) \land \neg \exists x\ (G(x) \land E(x,g) \land P(x))$$

Exemplo 5.4:

Todo gato peludo que é baguncieiro é mais esperto do que algum gato independente.

$$\neg \forall x\ (G(x) \land P(x) \land B(x) \rightarrow \neg \exists y\ (G(y) \land I(y) \land E(x,y))$$

Exemplo 6:

Nesse exemplo, considere o predicado G(x) = "x é um gato".

Todos são gatos.	para todo x, x é gato	$\forall x\ G(x)$
Existem gatos.	existe um x, tal que x é um gato	$\exists x\ G(x)$
Não existem gatos.	não existe x tal que x é um gato	$\neg \exists x\ G(x)$
Não existem gatos.	para todo x, x não é um gato	$\forall x\ \neg G(x)$
Nem todos são gatos.	não é verdade que para todo x, x é um gato	$\neg \forall x\ G(x)$
Nem todos são	existe um x, tal que x não é um	$\exists x$

gatos.	gato	$\neg G(x)$

A seguir são apresentadas algumas dicas importantes para a formalização de enunciados. Tais dicas foram adaptadas do livro de Lógica de Nolt e Rohatyn (1991).

1. **Variáveis diferentes não designam, necessariamente, objetos diferentes**.
   A fórmula $\forall x \forall y\ A(x,y)$ afirma que para qualquer x e qualquer y (não necessariamente distintos) x ama y. Afirma, em outras palavras, não somente que qualquer pessoa ama uma outra pessoa, como também que qualquer pessoa ama a si mesma.
2. **Escolha de variáveis não faz diferença para o significado**.
   A fórmula "$\exists x\ A(x,x)$", por exemplo, poderia também ser escrita com as simbolizações "$\exists y\ A(y,y)$" ou "$\exists z\ A(z,z)$". Entretanto, quando dois ou mais quantificadores justapõem-se em uma mesma parte da fórmula (como em "$\forall y\ \exists x\ A(x,y)$"), uma variável diferente deve ser usada para cada quantificador. Essa fórmula pode ser escrita, equivalentemente como "$\forall x\ \exists y\ A(y,x)$"ou "$\forall z\ \exists x\ A(x,z)$", mas não como "$\forall x\ \exists y\ A(x,x)$". Neste último caso não podemos distinguir qual quantificador governa as duas últimas ocorrências de "x". Devemos considerar essas fórmulas como malformadas.
3. **A mesma variável usada com vários quantificadores não designa, necessariamente, o mesmo objeto, em cada caso**.
   Assim, "$\exists x\ A(b,x) \land \exists x\ A(c,x)$" é uma formalização correta de "Existe alguém que Bob ama e existe alguém que Carmelita ama"; um enunciado que nem afirma nem nega que cada um ama a mesma pessoa. Mas algumas pessoas acham mais natural escrever "$\exists x\ A(b,x) \land \exists y\ A(c,y)$ uma formalização equivalente, e igualmente correta, do mesmo enunciado. Ela é legítima porque usa uma mesma variável em cada quantificador, os quais não se sobrepõem em uma mesma parte da fórmula (cada um é aplicado para o conjunto do qual é uma parte).

4. **As sentenças que misturam os quantificadores universal e existencial são, geralmente, ambíguas**. À primeira vista, a sentença – "Alguém ama todo mundo" significa:

    (a) Existe alguém que ama todo mundo.

    (b) Todo mundo é amado por alguém. (Não necessariamente a mesma pessoa em cada caso.)

    Essas sentenças mais prolixas, mas menos ambíguas, têm formalizações não equivalentes; a diferença no significado é revelada, formalmente, pela ordem diferente dos quantificadores. As formalizações dessas sentenças não são ambíguas.

5. **A ordem dos quantificadores consecutivos afeta o significado somente quando quantificadores universal e existencial são misturados**. Ocorrências consecutivas do quantificador universal podem ser permutadas sem, contudo, mudar o significado; o mesmo acontece com ocorrências consecutivas do quantificador existencial. Por exemplo, $\exists x\ \forall y\ A(x,y)$" e "$\forall y\ \exists x\ A(x,y)$" têm significados diferentes, ao passo que "$\exists x\, \exists y\ A(x,y)$" e "$\exists y\, \exists x\ A(x,y)$" significam, ambas, simplesmente e não ambiguamente, "alguém ama alguém".

Adaptado de: NOLT e ROHATYN, 1991. p. 246.

CAPÍTULO 5

A LÓGICA DE PREDICADOS

Todos os elementos que vimos até agora nos permitem construir uma nova linguagem lógica bastante distinta daquela que foi vista anteriormente, isto é, bastante distinta da lógica das proposições ou lógica proposicional.

O que vimos até agora foram os elementos que nos permitem construir uma lógica sobre predicados (ou uma lógica sobre sentenças abertas, já que sentenças abertas são essencialmente equivalentes aos predicados). Essa lógica de predicados também é denominada Lógica de Primeira Ordem, porque permite falar sobre as propriedades dos elementos pertencentes a um determinado domínio (ou conjunto). A lógica proposicional seria uma lógica de "ordem zero", porque não permitiria falar sobre elementos ou entidades, mas somente sobre frases fechadas (as proposições), que podem ser verdadeiras ou não. Por outro lado, seriam possíveis, embora não sejam estudadas neste texto, lógicas de ordens mais altas, que são lógicas que poderiam falar sobre os domínios (os conjuntos) em si, sobre conjuntos de conjuntos etc.

5.1 Significado das fórmulas

Até agora, o significado atribuído às fórmulas da Lógica de Predicados foi considerado indissociavelmente ligado aos domínios de variáveis e aos conjuntos-verdade dos predicados que ocorrem na fórmula. Dessa forma, para qualquer fórmula da Lógica de Predicados apresentada até agora, seu significado sempre foi dado pela definição dos domínios de suas variáveis e pela definição das propriedades que, nestes domínios, correspondem os símbolos dos predicados (sentenças abertas) existentes na fórmula. Essa abordagem derivada da Lógica Matemática define as propriedades de uma Lógica através do significado que possa ser atribuído às suas construções em termos de um modelo matemático baseado na Teoria dos Conjuntos.

Porém, essa não é a única abordagem possível para se tratar da Lógica de Predicados (ou de outros tipos de Lógica). Na verdade, de agora em diante os significados atribuídos para fórmulas da Lógica de Predicados serão tratados de um ponto de vista mais abstrato, buscando-se garantir que uma fórmula possa ser considerada verdadeira (ou falsa) somente de acordo com a sua forma, com o seu "formato".

Embora isso possa parecer difícil e pouco intuitivo, neste capítulo mostraremos que é perfeitamente possível definir significados dessa maneira. Mas primeiro é necessário definir de forma precisa a sintaxe das fórmulas da Lógica de Predicados.

5.2 Fórmulas bem-formadas

Para começar a definir o significado que uma fórmula da Lógica de Predicados pode assumir, primeiro é necessário considerar possível a existência das fórmulas puramente simbólicas sem associação com domínio ou conjunto-verdade. Isto é, são possíveis fórmulas como:

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

além de infinitas outras, onde não é necessário definir domínios para as variáveis ou conjuntos-verdade para os predicados.

Como já apresentado no Capítulo 1 que mostra a sintaxe das fórmulas da Lógica Proposicional, agora iremos mostrar quais são as regras de formação destas fórmulas puramente simbólicas da Lógica de Predicados. Para isso, são utilizados os seguintes conjuntos de símbolos:

- As variáveis individuais são as letras do fim do alfabeto, indexadas se necessário: u, v, w, x, y, z, x_1, x_2, x_3, ...
- As constantes individuais são letras do início e do meio do alfabeto, inxadas se necessário: a, b, c, d, e, g, h, a_1, a_2, a_3, ...
- Os símbolos para os predicados são letras predicativas maiúsculas: P, Q, R, S, T, A, B, C, ...
- Os símbolos para os conetivos lógicos são: $\neg$, $\rightarrow$, $\land$, $\lor$, $\leftrightarrow$
- Os símbolos para os operadores de quantificação são: $\forall$, $\exists$
- Também pode-se utilizar parênteses como símbolos de pontuação: (,)

O número de argumentos para um predicado normalmente é verificado no contexto. Um predicado sem argumento é considerado uma proposição simples.

Um *termo* é uma variável ou uma constante. Uma *fórmula atômica* é um predicado aplicado a argumentos que são termos. Por exemplo, P(x, a). De maneira geral, se P é símbolo de um predicado que tem *n* argumentos e $t_1, t_2, \ldots, t_n$ são termos, então $P(t_1, t_2, \ldots, t_n)$ é uma fórmula atômica.

Utilizando termos e fórmulas atômicas, toda a fórmula da Lógica de Predicados deve ser formada de acordo com as seguintes regras:

- Toda fórmula atômica é uma fórmula.
- Se *P* é uma fórmula, então $\neg P$ e (P) também é uma fórmula.
- Se *P* é uma fórmula e x é uma variável, então $(\forall x)P$ e $(\exists y)P$ também são fórmulas.
- Se *P* e *Q* são fórmulas e x é uma variável, então $P \wedge Q$, $P \vee Q$, $P \rightarrow Q$ e $P \leftrightarrow Q$ também são fórmulas.
- Nada mais é fórmula!

Note que nas regras acima os identificadores *P* e *Q* são utilizados como esquemas de fórmulas, que podem ser substituídos tanto por predicados simples ou fórmulas atômicas, quanto por fórmulas mais complexas. Para escrever fórmulas sem parênteses deve-se ter mais atenção devido à precedência dos operadores e quantificadores. Os quantificadores estão na mesma hierarquia de precedência que o operador de negação. A precedência de operadores lógicos está descrita no Capítulo 1 e a precedência dos quantificadores (que é igual a do operador de negação "¬ ") é definida no Capítulo 4.

5.3 Domínios e interpretações para fórmulas

Pelo que vimos até agora, uma fórmula é somente uma sequência (*string*) de símbolos sem significado. Para uma fórmula ter significado, é preciso dar uma *interpretação* para os seus símbolos de predicados e de constantes, além de definir o *domínio* de suas variáveis, para que seja possível descobrir qual o valor-verdade V ou F de todas as expressões que compõem a fórmula.

Por exemplo, suponha que a interpretação do predicado P(x) seja a propriedade “x é um número inteiro” e suponha que a interpretação da constante *c* seja o número 236. Com essas interpretações a fórmula P(c) torna-se a sentença "236 é um número inteiro", que é verdadeira.

Por outro lado, se mudarmos as interpretações essa fórmula pode se tornar falsa. Considere que somente a interpretação do predicado P(x) seja alterada para "x é um número primo", neste caso P(c) se torna na afirmação "236 é um número primo" que é falsa.

Agora vamos ver outro exemplo com a fórmula (∀ x)(∃ y) S(x,y). Assumindo que a interpretação de S(x,y) é "o sucessor de x é y", onde os valores das variáveis x e y pertencem ao conjunto dos números naturais *N*. Com essa interpretação e nesse domínio, a fórmula torna-se a sentença "Para todo número natural x existe um número natural y tal que o sucessor de x é y", que é verdadeira.

Da mesma forma que no exemplo anterior, se a interpretação ou o domínio forem alterados essa fórmula poderá ter outro valor lógico. Se o domínio das variáveis *x* e *y* for alterado para o conjunto das letras do alfabeto, mas S(x,y) ainda for interpretada como "o sucessor de x é y", apenas que neste caso levando em conta a sequência alfabética, então a fórmula (∀ x)(∃ y) S(x,y) torna-se falsa.

Dos exemplos acima é possível ver a importância da definição dos domínios e da interpretação dos símbolos de predicados e constantes. Essa é a base da semântica abstrata (chamada de *Semântica de Modelos*) que pode ser atribuída para as fórmulas puramente simbólicas da Lógica de Pedicados.

5.4 Semântica de Modelos

A *Semântica de Modelos* é uma generalização da semântica baseada na Teoria de Conjuntos que vem sendo empregada para definir as fórmulas com sentenças abertas. A semântica de modelos é, entretanto, mais geral, mais abstrata e, principalmente, mais rigorosa que a semântica de conjuntos, podendo ser aplicada a qualquer fórmula da Lógica de Predicados. A sêmantica de modelos foi desenvolvida apenas na década de 1950 com base nos trabalhos de Tarski, permitindo completar a análise formal da semântica da Lógica de Predicados e da Lógica de Primeira Ordem.

Como o próprio nome indica, a base da semântica de modelos são os **modelos** para as fórmulas lógicas. Neste contexto um modelo é uma estrutura matemática capaz de fazer uma fórmula da Lógica de Predicados se tornar verdadeira, ou seja, um modelo é capaz de

satisfazer uma fórmula.

Assim, para começar a definir a semântica de modelos de uma fórmula P, vamos precisar definir as estuturas matemáticas que formam os possíveis modelos para essa fórmula. As estrutura matemática utilizadas como modelos para as fórmulas da Lógica de Predicados são denominadas *estruturas semânticas de primeira-ordem* (ou apenas de **estruturas semânticas**). Uma estrutura semântica $\boldsymbol{S}$ é composta de dois componentes distintos:

- Um conjunto não vazio $\boldsymbol{D}$, que fornece o **domínio** para as variáveis da fórmula.
- Uma função ou mapeamento $\boldsymbol{I}$ que define a **interpretação** de cada predicado e constante da fórmula.

Formalmente as estruturas semânticas $\boldsymbol{S}$ são pares ordenados formados pela combinação do domínio $\boldsymbol{D}$ com a interpretação $\boldsymbol{I}$:

$$\boldsymbol{S} = < \boldsymbol{D}, \boldsymbol{I} >$$

A interpretação $\boldsymbol{I}$ deve definir para cada símbolo de predicado que aparece na fórmula P qual o conjunto-verdade correspondente para este predicado dentro do domínio $\boldsymbol{D}$ e para cada constante que aparece em P, qual o elemento correspondente para esta constante dentro do domínio $\boldsymbol{D}$. Assim uma interpretação $\boldsymbol{I}$ faz um mapeamento de cada um dos símbolos de predicados existentes em P em subconjuntos do domínio $\boldsymbol{D}$ e, se houver símbolos de constantes, eles também devem ser mapeados em elementos apropriados do domínio $\boldsymbol{D}$.

Dessa forma o domínio $\boldsymbol{D}$ também pode ser dividido em uma série subconjuntos D_1, D_2, ...,D_n que formarão os **conjuntos-verdade** correspondentes aos predicados contidos em P, de acordo com a interpretação $\boldsymbol{I}$. Da mesma forma, também podem ser identificados os elementos d_1, d_2, ..., d_m do domínio $\boldsymbol{D}$ correspondentes às constantes existentes em P.

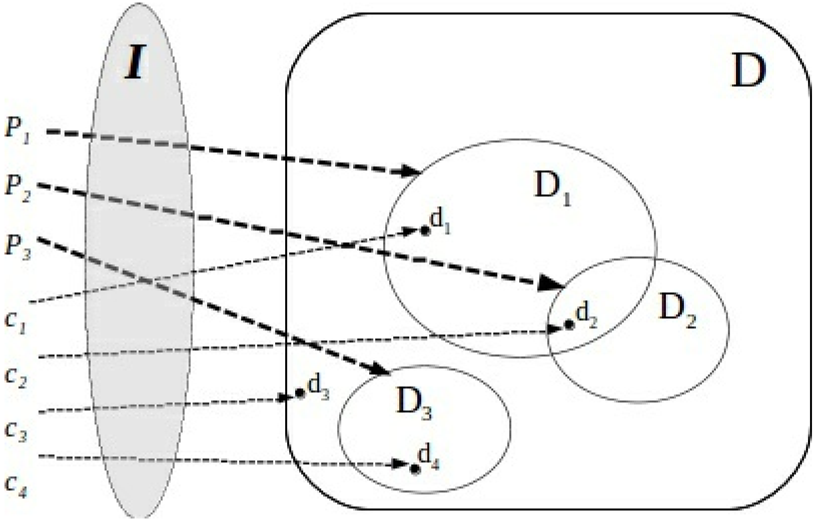

Figura 11 – Mapeamento de símbolos de predicados e constantes em subconjuntos e elementos do domínio.
Fonte: elaborada pelos autores.

Formalmente se P_1, P_2, ..., P_n são os símbolos de predicados na fórmula P e c_1, c_2, ..., c_m são as constantes da fórmula P, então uma interpretação $\boldsymbol{I}$ irá atribuir para cada predicado P_i um subconjunto A_i e para cada constante c_j um elemento d_j:

$$\boldsymbol{I}(P_i) = D_i \text{ para } 1 \leq i \leq n$$
$$\boldsymbol{I}(c_j) = d_j \text{ para } 1 \leq j \leq m$$

Assim, no exemplo da Figura 11, a interpretação **I** é definida da seguinte forma:

$$\boldsymbol{I}(P_1) = \mathbf{D}_1, \boldsymbol{I}(P_2) = \mathbf{D}_2, \boldsymbol{I}(P_3) = \mathbf{D}_3$$
$$\boldsymbol{I}(c_1) = d_1, \boldsymbol{I}(c_2) = d_2, \boldsymbol{I}(c_3) = d_3, \boldsymbol{I}(c_4) = d_4$$

5.5 Função de Avaliação

Intuitivamente, uma interpretação define os significados dos símbolos de uma fórmula dentro do domínio. Assim, quando possuímos uma fórmula P e uma estrutura $\boldsymbol{S}$ estamos "quase" prontos para descobrir o significado (valor-verdade) dessa fórmula. Esse valor-verdade pode ser "calculado" por meio de uma função capaz de "desmontar" a fórmula em suas sub-fórmulas componentes, de acordo com as regras de formação vistas anteriormente, obtendo o valor-verdade de cada um desses componentes e utilizando esses valores para calcular o valor-verdade final da fórmula.

Porém, para que isso seja possível é necessário definir como serão tratadas as variáveis da fórmula. A estrutura $\boldsymbol{S}$ apenas define os domínios destas variáveis mas não diz qual o valor armazenado em cada variável. Para resolver isso será utilizada uma **memória** $\boldsymbol{M}$ que é um vetor (ou *array*) que guarda os valores atribuídos para as variáveis da fórmula P. Essa memória é utilizada da mesma forma que os vetores ou *arrays* das linguagens de programação, utilizando os nomes das variáveis x, y,... como índices do vetor e retornando o valor correspondente à variável que está armazenado no vetor de memória. Dessa forma $\boldsymbol{M}[x]$ tem o valor atual da variável x, $\boldsymbol{M}[y]$ o valor de y e assim por diante para as demais variáveis em P.

Agora, usando uma estrutura $\boldsymbol{S}$ e uma memória $\boldsymbol{M}$ é possível calcular o valor-verdade da fórmula P, através da função de avaliação $\boldsymbol{Eval}(P, \boldsymbol{S}, \boldsymbol{M})$ que tem a fórmula P, a estrutura $\boldsymbol{S}$ e a memória $\boldsymbol{M}$ como parâmetros de entrada e fornece o valor-verdade V ou F da fórmula P como saída.

A função $\boldsymbol{Eval}(P, \boldsymbol{S}, \boldsymbol{M})$ pode ser definida recursivamente para cada um dos formatos de fórmula previstos pelas regras de formação definidas na Seção 5.2. A tabela a seguir mostra como cada um dos formatos de fórmula possíveis da Lógica de Predicados é tratado pela função de avaliação:

Tabela 12 – Função de avaliação

$$1)\ \boldsymbol{Eval}(P \wedge Q, \boldsymbol{S}, \boldsymbol{M}) = \begin{cases} V, \text{ se } \boldsymbol{Eval}(P, \boldsymbol{S}, \boldsymbol{M}) = V \text{ e } \boldsymbol{Eval}(Q, \boldsymbol{S}, \boldsymbol{M}) = V \\ F, \text{ caso contrário} \end{cases}$$

$$2)\ \boldsymbol{Eval}(P \vee Q, \boldsymbol{S}, \boldsymbol{M}) = \begin{cases} V, \text{ se } \boldsymbol{Eval}(P, \boldsymbol{S}, \boldsymbol{M}) = V \text{ ou } \boldsymbol{Eval}(Q, \boldsymbol{S}, \boldsymbol{M}) = V \\ F, \text{ caso contrário} \end{cases}$$

$$3)\ \boldsymbol{Eval}(P \rightarrow Q, \boldsymbol{S}, \boldsymbol{M}) = \begin{cases} F, \text{ se } \boldsymbol{Eval}(P, \boldsymbol{S}, \boldsymbol{M}) = V \text{ e } \boldsymbol{Eval}(Q, \boldsymbol{S}, \boldsymbol{M}) = F \\ V, \text{ caso contrário} \end{cases}$$

$$4)\ \boldsymbol{Eval}(P \leftrightarrow Q, \boldsymbol{S}, \boldsymbol{M}) = \begin{cases} V, \text{ se } \boldsymbol{Eval}(P, \boldsymbol{S}, \boldsymbol{M}) = \boldsymbol{Eval}(Q, \boldsymbol{S}, \boldsymbol{M}) \\ F, \text{ caso contrário} \end{cases}$$

$$5)\ \boldsymbol{Eval}(\neg P , \boldsymbol{S}, \boldsymbol{M}) = \begin{cases} V, \text{ se } \boldsymbol{Eval}(P, \boldsymbol{S}, \boldsymbol{M}) = F \\ F, \text{ caso contrário} \end{cases}$$

$$6)\ \boldsymbol{Eval}(P(x) , \boldsymbol{S}, \boldsymbol{M}) = \begin{cases} V, \text{ se } \boldsymbol{M}[\mathrm{x}] \in \boldsymbol{I}(\mathrm{P}) \\ F, \text{ caso contrário} \end{cases}$$

$$7)\ \boldsymbol{Eval}(P(c) , \boldsymbol{S}, \boldsymbol{M}) = \begin{cases} V, \text{ se } \boldsymbol{I}(c) \in \boldsymbol{I}(\mathrm{P}) \\ F, \text{ caso contrário} \end{cases}$$

$$8)\ \boldsymbol{Eval}((\exists\, x)P, \boldsymbol{S}, \boldsymbol{M}) = \begin{cases} V, \text{ se existe algum d} \in \boldsymbol{D} \text{ tal que } \boldsymbol{Eval}(P, \boldsymbol{S}, \boldsymbol{M}')=V, \\ \qquad \text{onde } \boldsymbol{M}' \text{ é igual a } \boldsymbol{M} \text{ exceto para } x\text{: } \boldsymbol{M}'[x]=\text{d} \\ F, \text{ caso contrário} \end{cases}$$

$$9)\ \boldsymbol{Eval}((\forall\, x)P, \boldsymbol{S}, \boldsymbol{M}) = \begin{cases} V, \text{ se para todo d} \in \boldsymbol{D} \text{ tem-se que } \boldsymbol{Eval}(P, \boldsymbol{S}, \boldsymbol{M}')=V, \\ \qquad \text{onde } \boldsymbol{M}' \text{ é igual a } \boldsymbol{M} \text{ exceto para } x\text{: } \boldsymbol{M}'[x]=\text{d} \\ F, \text{ caso contrário} \end{cases}$$

Fonte: elaborada pelos autores.

Para facilitar a compreensão da operação desta função, a definição acima mostra como são tratados os predicados de uma variável (predicados de múltiplas-variáveis são tratados através do produto cartesiano, mas isso complica desnecessariamente a explicação do funcionamento da função de avaliação.

Uma estrutura semântica ***S*** para uma fórmula *P* é chamada de **modelo** para *P* se *P* é verdade em relação a esta estrutura, ou seja, se ***Eval***(*P*, ***S***, ***M***) = *V*, para ***M*** uma memória inicial "vazia", que não atribui valor para qualquer variável de *P*. Caso contrário, a interpretação é um **contramodelo** da fórmula *P*.

Exemplo 1:

Qual o valor-verdade da fórmula $\exists x\ (P(x) \land \neg Q(x))$ na estrutura $\boldsymbol{S}=<\boldsymbol{D},\boldsymbol{I}>$, com domínio $\boldsymbol{D}=\{1,2,3\}$ e interpretação $\boldsymbol{I}$ definida como segue:

$$\boldsymbol{I}(P) = \{1,2\} \text{ e } \boldsymbol{I}(Q) = \{2,3\}$$

Para calcular o valor-verdade desta fórmula basta utilizar a função de avaliação passando como parâmetros para essa função a fórmula em questão, a estrutura $\boldsymbol{S}$ e uma memória $\boldsymbol{M}$ "vazia", isto é, uma memória que não tem valor atribuído para as variáveis x e y:

$$\boldsymbol{Eval}((\exists x)\ (P(x) \land \neg Q(x)), \boldsymbol{S}, \boldsymbol{M})$$

Esta chamada é tratada pelo caso 8 da Tabela 12. Esse caso implica em um processo de busca que irá tentar localizar algum elemento do domínio $\boldsymbol{D}$ que faça a fórmula P dentro do escopo do quantificador existencial se tornar verdadeira, ou seja, irá verificar se existe algum elemento $d \in \boldsymbol{D}$ que torna $P(x) \land \neg Q(x)$ verdadeira, quando $\boldsymbol{M}'[x]=d$.

Teste com d=3:

Para tanto, vamos começar os testes com d=3 e, portanto, $\boldsymbol{M}'[x]=3$. Para descobrir o valor-verdade de $P(x) \land \neg Q(x)$ nessas condições a função de avaliação é chamada novamente:

$$\boldsymbol{Eval}(P(x) \land \neg Q(x), \boldsymbol{S}, \boldsymbol{M}')$$

Esta nova chamada é atendida pelo caso 1 da Tabela. Para atender este caso são geradas duas novas chamadas da função de avaliação:

$\boldsymbol{Eval}(P(x), \boldsymbol{S}, \boldsymbol{M}')$ e $\boldsymbol{Eval}(\neg Q(x), \boldsymbol{S}, \boldsymbol{M}')$

A chamada $\boldsymbol{Eval}(P(x), \boldsymbol{S}, \boldsymbol{M}')$ será tratada pelo caso 6 da Tabela, resultando no valor-verdade F porque $\boldsymbol{M}'[x]$ que é igual a 3 não pertence a $\boldsymbol{I}(P)$, ou seja, $\boldsymbol{M}'[x] \notin \boldsymbol{I}(P)$. Com esse resultado, o resultado da chamada $\boldsymbol{Eval}(P(x) \land \neg Q(x), \boldsymbol{S}, \boldsymbol{M}')$ também será $\boldsymbol{F}$, mostrando que o elemento d=3 não torna a fórmula $P(x) \land \neg Q(x)$ verdadeira.

Teste com d=2:

Assim, deve-se testar outro elemento. Considerando que d=2 e ***M'***[x]=2, o tratamento da função de avaliação procede de maneira similar ao caso onde d=3. Porém agora a chamada ***Eval***(P(x), ***S***, ***M'***) irá retornar o valor-verdade *V,* porque ***M'***[x]=2 e, portanto, ***M'***[x] $\in$ ***I***(P). O problema ocorre com a chamada ***Eval***($\neg$Q(x), ***S***, ***M'***). Essa chamada é tratada pelo caso 5 que verifica o valor retornado por ***Eval***(Q(x), ***S***, ***M'***). De maneira similar a ***Eval***(P(x), ***S***, ***M'***), a chamada ***Eval***(Q(x), ***S***, ***M'***) será tratada pelo caso 6 da Tabela, resultando no valor-verdade *V* porque ***M'***[x] $\in$ ***I***(Q). Mas então, o valor-verdade retornado por ***Eval***($\neg$Q(x), ***S***, ***M'***) será *F* (ver caso 5) mostrando que o elemento d=2 também não torna a fórmula P(x) $\wedge$ $\neg$Q(x) verdadeira.

*Teste com d=*1:

O último elemento que se pode testar de D é 1. Fazendo d=3 e ***M'***[x]=1, que os resultados de ***Eval***(P(x), ***S***, ***M'***) e Eval($\neg$Q(x), ***S***, ***M'***) serão ambos verdadeiros. No caso de ***Eval***(P(x), ***S***, ***M'***) com ***M'***[x]=1, tem-se que ***M'***[x] $\in$ ***I***(P) e no caso de ***Eval***($\neg$Q(x), ***S***, ***M'***) tem-se que ***M'***[x] $\notin$ ***I***(Q) o que torna ambas expressões verdadeiras. Assim há pelo menos um elemento do domínio ***D*** (no caso o valor 1) que torna a fórmula P(x) $\wedge$ $\neg$Q(x) verdadeira e portanto a fórmula $\exists$x (P(x) $\wedge$ $\neg$Q(x)) também deve ser verdadeira.

Exemplo 2:

Considere uma estrutura semântica ***S*** com domínio formado pelo conjunto de todas as pessoas e com uma interpretação que atribui ao predicado P(x,y) a relação "x é pai de y", e as constantes *a* e *l*, respectivamente, as pessoas "Anakin" e "Luke". Assuma também que Anakin é pai de Luke. Veja o resultado da aplicação da função de avaliação sobre as seguintes fórmulas (considere que ***M*** é uma memória vazia, que não atribui valor inicial para x ou y):

- ***Eval***(P(*a*, *l*), ***S***, ***M***) = *V*
- ***Eval***($\exists$x) P(x, *l*), ***S***, ***M***) = *V* (porque existe uma pessoa, no caso Anakin, que é pai de Luke)
- ***Eval***($\forall$x P(x, *l*), ***S***, ***M***) = *F* (pois existem pessoas que não são pais de Luke)
- ***Eval***($\forall$x $\exists$y P (x, y), ***S***, ***M***) = *F* (pois existem pessoas que não

são pais)

- ***Eval***($\forall$y $\exists$x P(x, y), ***S***, ***M***) = *V* (pois todos têm pais)
- ***Eval***($\exists$x $\forall$y P(x, y), ***S***, ***M***) = *F* (não existe um pai de todos)
- ***Eval***($\exists$y $\forall$x P(x, y), ***S***, ***M***) = *F* (não existe um filho de todos)
- ***Eval***($\exists$x $\exists$y P(x, y), ***S***, ***M***) = *V*
- ***Eval***($\exists$y $\exists$x P(x, y), ***S***, ***M***) = *V*

5.6 Validade

No contexto da Semântica de Modelos é possível a ocorrência de três situações distintas para uma fórmula qualquer:

(i) Às vezes será possível encontrar interpretações e domínios que a farão verdadeira e também será possível encontrar interpretações e domínios que a tornarão falsa.
(ii) Para certas fórmulas, entretanto, todas as interpretações e domínios que forem encontradas tornarão a fórmula verdadeira.
(iii) Por fim, para outras fórmulas, somente será possível encontrar interpretações e domínios que a tornarão falsa.

Exemplos da primeira afirmação foram vistos na seção anterior. Porém as afirmações (ii) e (iii) podem ser um pouco mais difíceis de aceitar. Por exemplo, a afirmação (ii) levanta a questão de como seria possível que uma fórmula fosse sempre verdadeira em qualquer estrutura semântica possível? Essa dúvida é normal e pode ser traduzida no questionamento de como seria possível criar uma frase que sempre fosse verdadeira, independente de quem a está proferindo ou sobre o quê ela está falando ou a quem ela está se referindo?

Apesar de parecer difícil, veremos que é perfeitamente possível construir fórmulas que tenham tal propriedade. Na verdade, um dos principais objetivos de estudo da Lógica de Predicados é encontrar e usar estas fórmulas que são sempre verdadeiras. Tais fórmulas são as equivalentes, em termos da Lógica de Predicados, das tautologias da Lógica Proposicional. Além desse tipo de fórmulas, outros tipos de fórmulas também são estudados na Lógica de Predicados, de acordo com a classificação a seguir:

- Uma fórmula é **válida** quando ela é verdadeira para todas as estruturas semânticas possíveis.

- Porém, se existir uma estrutura semântica onde essa fórmula é falsa, então se diz que a fórmula é **inválida**.
- Uma fórmula é **satisfazível** quando é verdadeira em alguma estrutura semântica, ou seja, quando é **satisfeita** por alguma estrutura semântica.
- Portanto, uma fórmula é **não satisfazível** se ela é falsa para todas as estruturas semânticas possíveis. Assim para uma fórmula não-satisfazível todas as estruturas semânticas são contramodelos.

Dessas definições, temos que todas as fórmulas da Lógica de Predicados devem ter exatamente um dos seguintes pares de propriedades:

- A fórmula é *válida* e *satisfazível*: equivale a uma tautologia da Lógica Proposicional.
- A fórmula é *satisfazível* e *inválida*: equivale a uma fórmula contingente da Lógica Proposicional.
- A fórmula é *não-satisfazível* e *inválida*: equivale a uma contradição da Lógica Proposicional.

A validade de uma fórmula da Lógica de Predicados deve ser deduzida de sua forma, já que a validade é independente de qualquer interpretação ou domínio em particular: uma fórmula é válida se ela é verdadeira para todas as estruturas semânticas possíveis, ou seja, uma fórmula válida é "intrinsecamente verdadeira".

Na Lógica Proposicional podemos utilizar tabelas-verdade para verificar se qualquer fórmula proposicional é uma tautologia. E como podemos mostrar que uma fórmula é válida na Lógica de Predicados? Existe um algoritmo análogo ao da Lógica Proposicional que permite decidir se uma fórmula da Lógica de Predicados é válida?

Na verdade, não podemos verificar as infinitas estruturas semânticas para ver se cada uma é um modelo. Acontece que não existe um algoritmo para decidir a validade das fórmulas da Lógica de Predicados. Isso não significa que ainda não se encontrou um algoritmo, significa que já foi provado que não existe tal algoritmo.

A tabela a seguir apresenta um resumo comparando as fórmulas proposicionais e as fórmulas predicadas.

Tabela 13 – Validade das fórmulas

	Fórmulas proposicionais	Fórmulas predicadas
Valores lógicos	V ou F, dependendo dos valores lógicos atribuídos às proposições.	V, F ou sem valor lógico (se a fórmula conter uma variável livre), dependendo da interpretação.
Intrinsecamente verdadeiro	Tautologia, verdadeiro para todas as atribuições de valores lógicos.	Fórmula válida – verdadeira para todas as interpretações.
Metodologia	Algoritmo (tabela-verdade) para determinar se uma fórmula é uma tautologia.	Não existe um algoritmo para determinar se uma fórmula é válida.

Fonte: GERSTING: 2004. p. 32.

A seguir são apresentados exemplos que analisam a validade e a satisfazibilidade de algumas fórmulas.

Exemplo 1:

Considere a fórmula: (∀ x) P(x). E as seguintes estruturas semânticas:

- As estruturas S_1, S_2 e S_3 formadas pelo domínio N igual ao conjunto dos números naturais e pelas interpretações $\boldsymbol{I}_1$, $\boldsymbol{I}_2$ e $\boldsymbol{I}_3$ que mapeiam P(x) em subconjuntos dos números naturais que satisfazem, respectivamente, as propriedades: "x é primo", "x + 1 > x" e "x < 0".
- As estruturas S_4, S_5 e S_6 formadas pelo domínio M composto pelos móveis de nossa sala de aula com as interpretações $\boldsymbol{I}_4$, $\boldsymbol{I}_5$ e $\boldsymbol{I}_6$ que mapeiam P(x) nos subconjuntos destes móveis que

atendem, respectivamente, às propriedades: "*x* é preto", "*x* é feito de ouro" e "*x* é de propriedade da universidade".

Agora vamos verificar o valor desta fórmula em todas essas estruturas:

Estrutura Semântica	**Domínio**	**Interpretação de P(x)**	**Valor-verdade de (∀ x)P(x)**
S_1	N={1,2,...}	I_1="x é primo"	*F*
S_2	N={1,2,...}	I_2="x+1>x"	*V*
S_3	N={1,2,...}	I_3="x<0"	*F*
S_4	M=Móveis da sala de aula	I_4="x é preto"	*F*
S_5	M=Móveis da sala de aula	I_5="x é feito de ouro"	*F*
S_6	M=Móveis da sala de aula	I_6="x é de propriedade da universidade"	*V*

Pode-se ver claramente que para algumas estruturas semânticas a fórmula é verdadeira enquanto para outras a fórmula é falsa, ou seja, a fórmula (∀ x) P(x) se encaixa no caso (i) visto anteriormente.

Exemplo 2:

Agora vamos analisar como a fórmula

$$\forall x \mathrm{P}(x) \rightarrow \forall x \mathrm{P}(x)$$

se comporta para as mesmas estruturas e interpretações:

Estrutura Semântica	**Domínio**	**Interpretação de P(x)**	**Valor-verdade de (∀ *x*)P(*x*) → (∀**

			x)P(x)
S_1	N={1,2,...}	I_1="x é primo"	V
S_2	N={1,2,...}	I_2="x+1>x"	V
S_3	N={1,2,...}	I_3="x<0"	V
S_4	M=Móveis da sala de aula	I_4="x é preto"	V
S_5	M=Móveis da sala de aula	I_5="x é feito de ouro"	V
S_6	M=Móveis da sala de aula	I_6="x é de propriedade da universidade"	V

Ao analisarmos a situação, veremos que a fórmula ∀xP(x)) → ∀xP(x) sempre terá que ser verdadeira, independente de que interpretação ou domínio seja utilizado. Isso ocorre porque se ∀xP(x) é *F* em alguma estrutura semântica, então a fórmula se reduz a F → F, que é verdadeiro, da mesma forma se ∀xP(x) for V para outra estrutura semântica, então temos V → V, que também é verdadeiro pela definição do operador de implicação. Como P(x) somente pode ser V ou F, tem-se que ∀x P(x) → ∀xP(x) será sempre verdadeira. Ou seja, a fórmula ∀x P(x)) → ∀x(P(x) é um exemplo claro de fórmula que se encaixa no caso (ii) e, portanto, a fórmula ¬ (∀x P(x) → ∀xP(x)) é um exemplo que se encaixa no caso (iii).

Exemplo 3:

A fórmula do exemplo 2 tinha o formato da tautologia P → P da Lógica Proposicional e, portanto, se transformou em uma fórmula válida da Lógica de Predicados. Na verdade, todas as tautologias da Lógica Proposicional se transformam em fórmulas válidas na Lógica de Predicados. Porém, é importante ressaltar, a Lógica de Predicados tem muitas outras fórmulas que são válidas, mas que não têm uma tautologia equivalente em termos da lógica de predicados. Por exemplo, a fórmula

$$\forall x\ (P(x) \rightarrow P(x))$$

não tem uma fórmula tautológica da Lógica Proposicional que possa ser “casada” termo a termo, já que o quantificador ($\forall$ x) simplesmente não tem um termo ou operador equivalente na lógica proposicional. A fórmula acima simplesmente não pode ser casada com uma fórmula composta da Lógica Proposicional, por causa da inexistência da noção de quantificadores e variáveis nesta lógica (ela casaria apenas com uma proposição simples como P que poderia ser tanto verdadeira quanto falsa). Dessa forma, seria impossível dizer se ela é válida ou não através da Lógica Proposicional.

Mas na Lógica de Predicados essa fórmula é válida. Para entender isso vamos verificar o valor dessa fórmula nas estruturas semânticas que usamos nos exemplos anteriores:

Estrutura Semântica	**Domínio**	**Interpretação de P(x)**	**Valor-verdade de ($\forall$ x)(P(x) → P(x))**
S_1	N={1,2,...}	I_1="x é primo"	V
S_2	N={1,2,...}	I_2="x+1>x"	V
S_3	N={1,2,...}	I_3="x<0"	V
S_4	M=Móveis da sala de aula	I_4="x é preto"	V
S_5	M=Móveis da sala de aula	I_5="x é feito de ouro"	V
S_6	M=Móveis da sala de aula	I_6="x é de propriedade da universidade"	V

A justificativa para que $\forall x(P(x) \rightarrow P(x))$ seja sempre verdadeira é similar ao exemplo anterior: em qualquer domínio ou interpretação, a fórmula P(x) será ou verdadeira ou falsa, para qualquer valor escolhido para x. Assim se P(x) for falsa, a expressão $P(x) \rightarrow P(x)$ se reduz a $F \rightarrow F$, que é verdadeiro, da mesma forma se P(x) for verdadeira então P(x) → P(x) se reduz a V → V, que também é verdadeiro. Logo ∀x(P(x) → P(x)) será sempre verdadeira. Ou seja, esta fórmula, apesar de não ter equivalente em termos de fórmulas da Lógica Proposicional, também é válida.

Exemplo 4:

A situação pode ser ainda mais complexa, porque pode acontecer de existir uma fórmula da Lógica de Predicados que "casa" com uma fórmula da Lógica Proposicional, sendo que esta última não é uma tautologia, enquanto que a fórmula da Lógica de Predicados é válida (é sempre verdadeira). Por exemplo, a fórmula:

$$\forall x\ P(x) \rightarrow P(a)$$

onde a é uma constante que designa um elemento particular de qualquer domínio que escolhamos, claramente se “encaixa” com a fórmula:

$$P \rightarrow Q$$

da lógica proposicional, que obviamente não é uma tautologia. Entretanto, o problema aqui é que a fórmula:

$$\forall x\ P(x) \rightarrow P(a)$$

é uma fórmula válida na lógica de predicados. Vamos ver por que: se ∀x P(x) é verdadeiro em alguma estrutura semântica, então para qualquer elemento x que escolhamos do domínio dessa estrutura, P(x) será verdadeira, portanto também será verdadeira para um elemento a arbitrário deste domínio e, logo, P(a) também será verdadeiro, sendo a expressão reduzida a V → V, que é verdadeiro. Por outro lado, se ∀x P(x) é falso em alguma estrutura, então não precisamos nos preocupar com P(a), porque tanto F → V quanto F → F resultam verdadeiros.

De maneira similar ao caso da fórmula ∀x P(x) → P(a), a fórmula ∀x P(x) → ∃x P(x) também será válida na Lógica de Predicados. O raciocínio é similar: em qualquer estrutura semântica, se todo elemento do domínio tem uma determinada propriedade, então existe um elemento desse domínio que tem essa propriedade. Logo, sempre que o antecedente for verdadeiro, o consequente também o é e, portanto, a implicação é verdadeira.

Exercício 5.1

Em cada uma das fórmulas a seguir encontre um domínio e uma interpretação onde a fórmula é verdadeira e uma outra onde a fórmula é falsa (pode usar o mesmo domínio para todas as fórmulas):

(a) $\forall x\ ((A(x) \lor B(X)) \land \neg(A(x) \land B(x)))$
(b) $\forall x \forall y\ (P(x,y) \rightarrow P(y,x))$
(c) $\forall x\ (P(x) \rightarrow \exists y\, Q(y))$

5.7 Regras de dedução para a Lógica de Predicados

Alguns aspectos devem ser relembrados antes de introduzirmos as regras de dedução da Lógica de Predicados. As regras de inferência mostram como gerar fórmulas verdadeiras a partir de outras fórmulas. Na Lógica Proposicional, essas regras estão organizadas na forma de um sistema de dedução (o Sistema de Dedução Natural) que é capaz de gerar todas as formas de argumento válidas desta Lógica. Porém, a Lógica Proposicional é, na verdade, uma parte da Lógica de Predicados, assim todas as regras de inferência, incluindo as regra básicas, derivadas e de equivalência da Lógica Proposicional também podem ser utilizadas nas demonstrações da Lógica de Predicados, ou seja, o Sistema de Dedução Natural também é usado na Lógica de Predicados. Sendo assim, podemos usar essas regras da mesma forma que na Lógica Proposicional desde que consigamos casar perfeitamente os símbolos de fórmulas lógicas que constam nessas regras com fórmulas predicadas com ou sem quantificadores.

Por exemplo, um argumento da forma: $P,\ P \rightarrow Q \vdash Q$, continua sendo válido por *Modus Ponens*, mesmo que as fórmulas envolvidas estejam predicadas (quantificadas ou não). O argumento predicado abaixo é um exemplo disso:

$$\forall x\, R(x),\ \forall x R(x) \rightarrow \forall x S(x) \vdash \forall\, x S(x)$$

Este argumento pode ser provado pela seguinte demonstração:

1	$\forall x\ R(x)$	Hip
2	$\forall x\ R(x) \rightarrow (\forall\ x)S(x)$	Hip
3	$\forall x\ S(x)$	1,2, mp

O argumento a seguir é outro exemplo de argumento válido da Lógica de Predicados que pode ser provado apenas com as regras já conhecidas da Lógica Proposicional.

$$\neg F(a) \vee \exists x\ F(x),\ \exists x\ F(x) \rightarrow P \vdash F(a) \rightarrow P$$

1	$\neg F(a) \vee \exists x\ F(x)$	Hip
2	$\exists x\ F(x) \rightarrow P$	Hip
3	$\mid F(a)$	Hip-pc
4	$\mid \mid \neg F(a)$	Hip-raa
5	$\mid \mid F(a) \wedge \neg F(a)$	3,4, cj
6	$\mid \neg\neg F(a)$	4-5, raa
7	$\mid (\exists\ x)F(x)$	1,6, sd
8	$\mid P$	2,7, mp
9	$F(a) \rightarrow P$	3-8, pc

Por outro lado, como já vimos anteriormente, existem argumentos com fórmulas da Lógica de Predicados que não são tautologias da Lógica Proposicional, mas ainda assim são válidos devido à sua estrutura e ao significado dos quantificadores universal e existencial.

Seguindo a ideia do Sistema de Dedução Natural, a abordagem geral para provar esses argumentos é retirar os quantificadores, manipular as fórmulas sem eles e depois recolocá-los no lugar, se necessário for. Algumas propriedades que levam em conta a manipulação de quantificadores já foram vistas, notadamente as propriedades relativas a:

- negação de fórmulas quantificadas, e

- comutação de quantificadores de mesmo tipo.

Porém, estas propriedades, que eventualmente se transformarão em regras derivadas do sistema de dedução da Lógica de Predicados, não retiram nenhum quantificador do lugar nem inserem um quantificador em uma fórmula que não o possuía anteriormente. Assim, são necessárias novas regras de inferência que permitam manipular os quantificadores, isto é, que forneçam mecanismos para inseri-los e retirá-los quando necessário. Isso implica que serão necessárias quatro regras novas: uma para retirada de cada um dos dois tipos de quantificadores e uma para a inserção de cada um deles.

As regras que inserem um quantificador são denominadas regras de **generalização**, porque transformam uma fórmula representando uma afirmaçãou específica para algum indivíduo (mesmo que seja representado por uma variável livre) em uma fórmula representando uma afirmação geral para um conjunto de indivíduos, indicados pela variável quantificada. Por outro lado, as regras de eliminação de quantificadores são chamadas de regras de **particularização** porque fazem o processo oposto: transformam uma fórmula com uma afirmação genérica, em uma fórmula com uma afirmação sobre um indivíduo (constante ou variável livre) em particular. Essas regras são apresentadas na Tabela 14 a seguir:

Tabela 14 – Regras de inferência da Lógica de Predicados

Regras	Restrição de Uso
Particularização Universal (**PU**) $(\forall x)P(x)$ ———————— $P(t)$ $[x/t]$	Se o novo termo *t* que substituirá a variável x em P(x) for uma variável diferente de x, então esta variável não deve estar quantificada na fórmula P(*t*).
Particularização Existencial (**PE**) $(\exists x)\ P(x)$ ———————— $P(t)$ $[x/t]$	O termo *t* que substituirá a variável x em P(x) não deve ter sido usado anteriormente na demonstração como constante ou variável livre
Generalização Universal (**GU**) $P(x)$ ———————— $(\forall x)P(x)$ $[x/x]$	A variável x em P(x) não pode ter sido deduzida de nenhuma hipótese ativa onde x era originalmente uma variável livre. A fórmula P(x) também não pode ter sido deduzida por Particularização Existencial (PE) sobre a variável x.
Generalização Existencial (**GE**) $P(t)$ ———————— $(\exists x)\ P(x)$ $[t/x]$	O termo *t* em P(*t*) deve ser uma constante ou variável livre. A variável x que substituirá o termo *t* deve ser nova e não pode ter aparecido anteriormente na fórmula P(*t*).

Fonte: elaborada pelos autores.

Características da notação empregada na regras:

- Como no caso das regras da Lógica Proposicional, a notação P(x) **não** indica que P é um predicado unário (com apenas uma variável) tendo x com sua única variável. Essa notação representa um *esquema de fórmula* da Lógica de Predicados, indicando, simplesmente, que x é uma das variáveis na fórmula representada por P. Portanto P pode ser uma expressão como $\exists y\ (\forall z)\ (Q(x,y,z))$.
- O termo *t* representa variáveis ou símbolos para constantes do domínio.
- As notações [x/t] e [t/x] indica o elemento da esquerda da "/", constante na fórmula original, foi trocado pelo elemento da direita da "/" na nova fórmula gerada após a aplicação de uma regra de particularização ou generalização.
- A notação [x/x] usada na GU mostra apenas que nessa regra

não se faz substituição ou troca de nome de variável.

A generalização e particularização também podem ser encontradas na literatura como regras de introdução e eliminação dos quantificadores, respectivamente. Mas, neste texto a notação adotada para tratar as regras é generalização e particularização dos quantificadores.

A seguir são analisados em detalhes casos de utilização de cada uma dessas regras.

5.8 Particularização universal

$\dfrac{(\forall x)P(x)}{P(t)\ [x/t]}$	Se o novo termo t que substituirá a variável x em P(x) for uma variável diferente de x, então esta variável não deve estar quantificada na fórmula P(t).

A regra de particularização universal (**PU**) diz que podemos deduzir P(x), P(y), P(a) etc. de (∀ x)P(x), retirando seu quantificador universal. A justificativa intuitiva para essa regra é que, se o predicado (ou fórmula) P é verdadeiro para todos os elementos do domínio, então podemos nomear um elemento qualquer desse domínio por um nome arbitrário de variável ou por um símbolo de constante que P continuará sendo verdadeiro para essa nova variável ou constante.

A particularização universal pode ser usada para demonstrar um dos silogismos clássicos da Lógica Aristotélica, que foi a primeira lógica sistematizada na história da humanidade, pelo filósofo grego Aristóteles, que viveu de 384 a 322 a.C. Todos os argumentos clássicos (silogismos), similares ao argumento:

Todos os seres humanos são mortais.
Sócrates é um ser humano.
Logo, Sócrates é mortal.

podem ser "semiformalizados" pelo seguinte esquema:

Todos os A são B.

a é um A.
Logo, a é um B.

Estes argumentos semiformais podem, então, ser aplicados a uma enorme gama de casos. A formalização completa desse tipo de argumento, em termos da Lógica de Predicados, pode ser feita de acordo com a seguinte fórmula:

$$\forall x\ (A(x) \rightarrow B(x)),\ A(a) \vdash B(a)$$

Este argumento pode ser provado pela seguinte demonstração:

1	$\forall x\ (A(x) \rightarrow B(x))$	Hip
2	$A(a)$	Hip
3	$A(a) \rightarrow B(a)$	1, pu [x/a]
4	$B(a)$	2,3, mp

Exemplos:

A seguir são apresentados dois exemplos de provas de argumentos que fazem uso da regra de particularização do quantificador universal (**PU**).

Demonstração do argumento : $\forall x\ (F(x) \rightarrow G(x)),\ \forall x\ F(x) \vdash G(a)$

1	$(\forall\ x)(F(x) \rightarrow G(x))$	Hip
2	$(\forall\ x)F(x)$	Hip
3	$F(a) \rightarrow G(a)$	1, pu [x/a]
4	$F(a)$	2, pu [x/a]
5	$G(a)$	3,4, mp

Demonstração do argumento: $\neg F(a) \vdash \neg \forall x F(x)$

1	$\neg F(a)$	Hip

2	$\mid \forall x F(x)$	Hip-raa
3	$\mid F(a)$	2, pu [x/a]
4	$\mid F(a) \land \neg F(a)$	1,3, cj
5	$\neg \forall x F(x)$	2-4, raa

A restrição para a regra de particularização universal define que se *t* for uma variável, então não deve estar dentro do escopo de um quantificador para *t*. Esta evita, por exemplo, que fórmulas similares a $\forall x \exists y P(x,y)$ possam ser particularizadas como $\exists y\ P(y,y)$, o que não seria válido. É fácil demonstrar que esse tipo de inferência é inválido, através da apresentação de um contraexemplo. Por exemplo, se assumirmos que $P(x,y)$ significa $x < y$ no domínio dos números naturais, então $\forall x \exists y P(x,y)$ é verdade (para todo natural sempre existe um número natural maior), enquanto $\exists y P(y,y)$ é obviamente falsa, já que não existe número natural y tal que $y < y$.

Podemos analisar o impacto dessa restrição através de alguns casos. Por exemplo, dada a fórmula $\forall x\ (P(x,y) \land Q(x))$, podemos utilizar a regra **PU** para inferir as fórmulas:

- $P(c,y) \land Q(c)$
- $P(x,y) \land Q(x)$
- $P(y,y) \land Q(y)$

Então, podemos inferir $P(t,y) \land Q(t)$ para qualquer termo *t*, sem restrição, pois a fórmula não contém quantificadores (não tem variáveis ligadas). Por outro lado, se a fórmula for $\forall x \forall y\ (P(x,y) \land Q(x))$, podemos utilizar a regra **PU** para inferir as fórmulas:

- $\forall y (P(c,y) \land Q(c))$
- $\forall y (P(x,y) \land Q(x))$

Entretanto, devido à restrição da regra, não é possível inferir a fórmula $\forall y (P(y,y) \land Q(y))$, pois y não é livre na fórmula.

5.9 Particularização existencial

$(\exists x)\ P(x)$ ———————— $P(t)\ [x/t]$	O termo t que substituirá a variável x em P(x) não deve ter sido usado anteriormente na demonstração como constante ou variável livre

A regra de particularização existencial (**PE**) nos permite retirar o quantificador existencial de uma fórmula. Essa regra diz que, a partir de $\exists x\ P(x)$, podemos deduzir P(y), P(z), P(a), P(b) etc. desde que y, z, a, b etc. sejam essencialmente símbolos novos. A justificativa intuitiva para essa regra é que se P é verdadeira para algum elemento do domínio, então podemos dar um nome específico para esse elemento, mas não podemos supor nada mais a seu respeito, isto é, nada nos impede de dar um (novo) nome a este suposto elemento "x" que satisfaz P(x).

Para exemplificar, vamos considerar uma reescrita do argumento similar ao silogismo clássico:

Todos os A são B.
Existe algum A.
Logo, um fulano é B.

onde "fulano" indica alguém que não conhecemos mas que sabemos certamente que "é B". Este argumento pode ser formalizado por:

$$\forall x\ (A(x) \rightarrow B(x)),\ \exists y\ A(y) \vdash B(a)$$

e pode facilmente ser demonstrado por:

1	$\forall x\ (A(x) \rightarrow B(x))$	Hip
2	$\exists y\ A(y)$	Hip
3	$A(a)$	2, pe [y/a]
4	$A(a) \rightarrow B(a)$	1, pu [x/a]
5	$B(a)$	3,4, mp

Um detalhe importante em relação a essa demonstração é que os passos 3 e 4 **não podem ser trocados** de ordem por causa da restrição de aplicação da regra de particularização existencial. Se assim o fosse, ou seja, se **PE** tivesse sido usada primeiro sobre a hipótese 1, então não haveria razão para supor que este termo *a* particular é o que tem a propriedade P, como na hipótese 2. Portanto, o efeito básico da restrição de uso dessa regra é que você é obrigado, primeiro, a olhar todas as hipóteses e, se quiser usar a **PE** em alguma delas, tem que fazer isso primeiro.

Outro exemplo que esclarece a restrição da regra é dado pela tentativa de resolução do argumento: $\exists x\ P(x) \land \exists x\ Q(x) \vdash P(c) \land Q(c)$

1	$\exists x\ P(x) \land \exists x\ Q(x)$	Hip
2	$\exists x\ P(x)$	1, sp
3	$\exists x\ Q(x)$	1, sp
4	$P(c)$	2, pe [x/c]
~~5~~	~~Q(c)~~	~~2, pe [x/c]~~ <<== Problema!! **Erro**!

O problema na demonstração deste argumento é o reúso da constante c na linha 5: isso é proibido pela restrição de uso da particularização existencial. E por que essa restrição existe? Considere a interpretação para P(x) = "x é par" e Q(x) = "x é ímpar" no domínio dos números naturais. Então, a fórmula $\exists x\ P(x) \land \exists x\ Q(x)$ é verdadeira porque existe um número que é par e existe um número que é ímpar dentro do conjunto dos números naturais. Entretanto, com essa interpretação a conclusão é falsa, pois nenhum número natural pode ser par e ímpar ao mesmo tempo. O problema está na linha 5, quando inferimos a regra pe e utilizamos o mesmo termo c já utilizado na linha 4. Dessa forma, sempre que for aplicar a particularização do quantificador existencial, lembre-se da restrição dessa regra.

5.10 Generalização universal

$\dfrac{P(x)}{(\forall x)P(x)\ [x/x]}$	A variável x em P(x) não pode ter sido deduzida de nenhuma hipótese ativa onde x era originalmente uma variável livre. A fórmula P(x) também não pode ter sido deduzida por Particularização Existencial (PE) sobre a variável x.

A regra generalização universal (**GU**) permite que se insira um quantificador universal. No entanto, isso precisa ser feito muito cuidadosamente. Essa inserção somente pode ser feita se estivermos seguros que a sentença aberta P(x) é verdadeira e que a variável x, usada nesta sentença, indica um elemento realmente arbitrário, isto é, **x pode realmente ser qualquer** elemento do domínio. Neste caso, então, nada nos impede de afirmar (∀ x)(P(x)). Porém, se existir alguma **pressuposição** na demonstração de que **x é algum elemento específico** do domínio (por exemplo, P(x) foi obtido por particularização existencial), então **não podemos generalizar** P(x) para (∀ x)(P(x)).

Para exemplificar, vamos provar mais um argumento similar ao silogismo clássico:

$$\forall x\ (P(x) \rightarrow Q(x)),\ \forall y\ P(y) \vdash \forall x\ Q(x)$$

através da seguinte demonstração:

1	∀x (P(x) → Q(x))	Hip
2	∀y P(y)	Hip
3	P(x) → Q(x)	1, pu [x/x]
4	P(x)	2, pu [y/x] (note que não existe restrições na PU sobre continuar a usar o mesmo nome da variável)
5	Q(x)	3,4, cp
6	∀x Q(x)	5, gu [x/x]

A utilização da generalização universal no passo 6 é correta, uma vez que x não era uma variável livre em nenhuma hipótese, nem a particularização existencial (**PE**) foi utilizada para se chegar até Q(x).

A primeira restrição da generalização universal evita que a utilização de alguma variável livre nas hipóteses possa ser usada como base para inferir uma propriedade universal. A afirmação em alguma hipótese de uma sentença aberta P(x) com a variável x pode tanto indicar que exista pelo menos um elemento do domínio que satisfaça P(x) quanto indicar que todos satisfazem P(x), mas não dá informação adicional, portanto não pode ser usada como base de uma generalização correta. Note que a restrição se aplica sobre **hipóteses ativas**, ou seja, sobre qualquer hípotese que esteja valendo no ponto onde a regra vai ser aplicada na demonstração. Isso inclui, além das hipóteses ou premissas iniciais, qualquer hipótese introduzida por uma regra hipotética como RAA ou PC.

A segunda restrição apenas evita que o formalismo desconsidere o significado por trás da operação de generalização, isto é, se chegamos a um P(x) em uma demonstração com base em uma particularização de um existencial, isso implica que estamos seguros que existe pelo menos um elemento do domínio no qual vale P(x). Como já vimos anteriormente, não faz mal chamarmos este elemento de "x", mas não podemos, daí, inferir que qualquer outro elemento do domínio também atenda à sentença P(x), logo, é impossível generalizar este P(x) para ∀x P(x). Do ponto de vista puramente formal, isso é evitado pela restrição que obriga que P(x), para ser generalizado universalmente, não possa ter sido previamente demonstrado por uma particularização existencial.

Exemplos:

Para exemplificar o uso de regras hipotéticas no caso da lógica de predicados, vamos provar o seguinte argumento:

$$P(x) \rightarrow \forall y\ Q(x,y) \vdash \forall y\ (P(x) \rightarrow Q(x,y))$$

pela seguinte demonstração:

1	$P(x) \rightarrow \forall y\ Q(x,y)$	Hip
2	$\mid P(x)$	Hip-PC
3	$\mid \forall y\ Q(x,y)$	1,2, mp
4	$\mid Q(x,y)$	2, pu [y/y]
5	$P(x) \rightarrow Q(x,y)$	2-4, pc
6	$\forall y\ (P(x) \rightarrow Q(x,y))$	5, gu [y/y]

Outro exemplo relevante para compreendermos as restrições da regra GU considera a interpretação para P(x) = “x é um número primo” no domínio dos números naturais. Nesse exemplo, fica claro que não podemos aplicar GU e gerar a fórmula ∀x P(x), pois não faz sentido afirmar que todo x é um número primo. A demonstração a seguir está errada devido à restrição da regra GU que impede a generalização da fórmula P(x) porque x é uma variável livre na hipótese. O passo 2 em destaque é onde foi feito o erro:

1	P(x)	Hip
~~2~~	~~∀x P(x)~~	~~1, gu [x/x]~~ <== **Erro**!

5.11 Generalização existencial

P(*t*) ——————— (∃ x) P(x) [*t*/x]	O termo *t* em P(*t*) deve ser uma constante ou variável livre. A variável x que substituirá o termo *t* deve ser nova e não pode ter aparecido anteriormente na fórmula P(*t*).

A regra generalização existencial (**GE**) permite a inserção de um quantificador existencial. De P(x) ou P(a), podemos deduzir ∃x P(x). A justificativa intuitiva para esta regra é que, se alguma já foi identificada como tendo a propriedade P no decorrer da demonstração, então podemos afirmar que existe alguma coisa com a propriedade P, logo ∃x P(x). Para exemplificar, vamos provar o argumento:

$$\forall x\, P(x) \vdash \exists x\, P(x)$$

pela seguinte demonstração:

1	∀x P(x)	Hip
2	P(y)	1, pu [x/y]
3	∃x P(x)	2, ge [y/x]

A restrição da generalização existencial tem uma base similar à da restrição empregada na particularização universal. Ela serve para evitar que, por exemplo, de fórmulas similares a P(a,y) possamos deduzir ∃y P(y,y). Como já vimos anteriormente, isso não seria válido. Novamente é fácil encontrar um contraexemplo que mostre isso porque, por exemplo, se assumirmos que P(x,y) significa x < y no domínio dos números naturais, então P(a,y) pode ser verdade (para algum a e para algum y), enquanto ∃y (P(y,y)) é obviamente falsa, já que é impossível que y < y para qualquer y pertencente aos números naturais.

Esses problemas são evitados quando se utiliza uma nova variável que não foi utilizada antes na fórmula original P(t). Na verdade essa é uma restrição forte que pode ser diminuída: é possível deduzir ∃x P(x) de P(t), somente se a igualdade P(*t*) = P(x)[x/*t*] se mantém, isto é, se substituirmos todas as ocorrências de x em P(x) pelo termo *t* originalmente usado em P(*t*), a fórmula resultante deve voltar a ser igual a P(*t*). A igualdade P(t) = P (x)[x/t] é o teste do passo da volta pois P(t) aparece anteriormente na demonstração.

No caso de P(a,y) se deduzirmos por PE [a/y] resultando em ∃y P(y,y) então o teste do passo de volta não é bem-sucedido porque P(y,y) [y/a] = P(a,a) que **não é igual** a P(a,y). Por outro lado, se usarmos uma variável diferente, como z, então PE [a/z] resulta em ∃z P(z,y) e o teste de passo de volta P(z,y)[z/a] resulta em P(a,y) que é igual à fórmula original.

Faz sentido ressaltar que, se uma propriedade vale para uma coisa em particular, então a propriedade vale para alguma coisa. Voltando ao exemplo da interpretação P(x) = “x é um número primo”. Por exemplo, sabemos que o número 5 é um número primo, então podemos concluir que existe um número primo. A demonstração do exemplo segue:

1	P(5)	Hip
2	∃x P(x)	1, ge [5/x]

5.12 Teoremas e equivalências

Assim como na lógica proposicional, é possível provar fórmulas sem utilizarmos premissas. Essas fórmulas são conhecidas como teoremas, já definidos na Seção 2.7. As equivalências também são empregadas na lógica de predicados.

Por exemplo, a forma E dos enunciados categóricos é “Nenhum S é P”, simbolizada por $\forall x(S(x) \rightarrow \neg P(x))$. Entretanto, também é possível simbolizar o enunciado com o quantificador existencial $\neg \exists x\ (S(x) \wedge P(x))$. Como fórmulas equivalentes podem ser substituídas por outra, acompanhe a demonstração da equivalência $\neg \forall x\ P(x) \leftrightarrow \exists x\ \neg P(x)$. A prova da equivalência restante da negação e da equivalência da comutação de quantificadores, apresentadas no Capítulo 4, Seção 4.6, são sugeridas como exercício.

Prova do teorema: $\vdash \neg \forall x\ P(x) \leftrightarrow \exists x \neg P(x)$

1	$\mid \neg \forall x P(x)$	Hip-PC
2	$\mid \mid \neg \exists x \neg P(x)$	Hip-RAA
3	$\mid \mid \mid \neg P(x)$	Hip-RAA
4	$\mid \mid \mid \neg \exists\ x \neg P(x)$	3, ge [x/x]
5	$\mid \mid \mid (\exists\ x)\neg P(x) \wedge \neg(\exists\ x)\neg P(x)$	2,4, cj
6	$\mid \mid \neg\neg P(x)$	RAA 3-5
7	$\mid \mid P(x)$	DN 6
8	$\mid \mid (\forall\ x)P(x)$	GU 7 [x/x] ***
9	$\mid \mid (\forall\ x)P(x) \wedge \neg(\forall\ x)P(x)$	Cj 1,8
10	$\mid \neg\neg(\exists\ x)\neg P(x)$	RAA 2-9
11	$\mid (\exists\ x)\neg P(x)$	DN 10
12	$\neg(\forall\ x)P(x) \rightarrow (\exists\ x)\neg P(x)$	PC 1-11
13	$\mid (\exists\ x)\neg P(x)$	Hip-PC
14	$\mid \neg P(a)$	PE 13 [x/a]
15	$\mid \mid (\forall\ x)P(x)$	Hip-RAA
16	$\mid \mid P(a)$	PU 15 [x/a]
17	$\mid \mid P(a) \wedge \neg P(a)$	Cj 14,16
18	$\mid \neg(\forall\ x)P(x)$	RAA 15-17
19	$(\exists\ x)\neg P(x) \rightarrow \neg(\forall\ x)P(x)$	PC 13-18
20	$\neg(\forall\ x)P(x) \leftrightarrow (\exists\ x)\neg P(x)$	+Eq 12,19

***Note que x em P(x) não foi derivada por PE e x também não aparece em nenhuma hipótese ativa como variável livre - x foi usada como variável livre na RAA, mas esta RAA já está fechada e deve, portanto, ser desconsiderada.

Com a demonstração anterior e com as demais demonstrações de equivalências deixadas como exercícios, podemos assumir como válidas as seguintes regras de equivalência da Lógica de Predicados:

Equivalência da Negação de Quantificadores (**EqNQ**)	$\dfrac{\neg(\forall x)P(x)}{(\exists x)\neg P(x)}$	$\dfrac{(\exists x)\neg P(x)}{\neg(\forall x)P(x)}$
Comutação dos Quantificadores (**CmtQ**)	$\dfrac{(\forall x)(\forall y)P(x,y)}{(\forall y)(\forall x)P(x,y)}$	$\dfrac{(\exists y)(\exists x)P(x,y)}{(\exists x)(\exists y)P(x,y)}$

A equivalência das duas simbolizações da forma E é provada a seguir, fazendo uso tanto das regras de equivalência da Lógica Proposicional quanto das novas regras de equivalência dos quantificadores.

Prova do argumento: $\forall x\,(S(x) \rightarrow \neg P(x)) \vdash \neg \exists x\,(S(x) \wedge P(x))$

1	$\forall x\,(S(x) \rightarrow \neg P(x))$	Hip
2	$\forall x\,(S(x) \rightarrow \neg P(x))$	1, pu [x/a]
3	$S(a) \rightarrow \neg P(a)$	2, cond
4	$\neg(S(a) \wedge P(a))$	3, dmor
5	$\forall x \neg(S(x) \wedge P(x))$	4, gu [a/x]
6	$\neg \exists x\,(S(x) \wedge P(x))$	5, equiv

Como sugestão, desenvolva outra demonstração, entretanto, sem o uso das regras de equivalência.

5.13 Prova por contraexemplo

Normalmente não é fácil demonstrar a validade de uma afirmação universal, uma vez que se é obrigado a demonstrar a validade desta afirmação para todos os elementos de um domínio (possivelmente infinito). Embora existam formas de se demonstrar afirmações universais, principalmente sobre domínios matemáticos (indução matemática), ainda assim isso não é uma tarefa muito simples.

Por outro lado, afirmações existenciais podem ser demonstradas pela apresentação de (pelo menos) um elemento que satisfaça a afirmação, o que, às vezes, é muito mais fácil do que tentar demonstrar uma propriedade universal dos elementos de um conjunto.

Assim, se houvesse uma forma de se transformar um argumento composto de sentenças com quantificadores universais em um argumento equivalente, mas composto de sentenças existenciais, então, muitas vezes, este segundo argumento teria uma demonstração mais fácil.

Na verdade, existe este método que é chamado de *prova por contraexemplo*. Vamos ver como pode ser aplicado. O método é aplicável sobre argumentos com afirmações puramente universais na seguinte forma:

$$\forall x P(x) \rightarrow (\forall x) Q(x)$$

Uma forma direta de demonstrar a veracidade desse argumento seria demonstrar primeiro a validade da afirmação universal $(\forall x)P(x)$ e então deduzir $(\forall x)Q(x)$ por *modus ponens*. Entretanto, a demonstração de $(\forall x)(P(x))$ poderia ser uma tarefa muito difícil.

Para facilitar um pouco as coisas, pode-se considerar que um argumento nessa forma pode, pela regra de equivalência da contraposição, ser transformado em:

$$\neg \forall x Q(x) \rightarrow \neg \forall x P(x)$$

Além disso, pela regra da negação dos quantificadores, o argumento acima pode ser transformado na seguinte fórmula equivalente:

$\exists x \neg Q(x) \rightarrow \exists x P(x)$

Uma prova para o argumento acima, pelo fato dele ser equivalente ao argumento original, também é uma prova para o argumento original.

O que nos deixa com a necessidade de demonstrar a afirmação $\exists x \neg Q(x)$, que é uma afirmação existencial.

Agora, basta encontrar um elemento que faça Q(x) se tornar falsa, e portanto tornar $\neg Q(x)$ verdadeira, para garantir a validade de $\exists x \neg Q(x)$.

Esse elemento passa a ser, então, o **contraexemplo** da afirmação $\forall x Q(x)$, ou seja, o exemplo contrário que faz $\forall x Q(x)$ ficar falsa e, portanto, faz $\neg \forall x Q(x)$ se tornar válida.

Caso esse elemento seja encontrado, então a fórmula $\exists x P(x)$ é deduzida por *modus ponens* da fórmula $\exists x \neg Q(x) \rightarrow \exists x P(x)$, provando este argumento e, portanto, provando o argumento original.

5.14 Lógica de Predicados com igualdade

Até agora apenas trabalhamos com a Lógica de Predicados clássica, sem permitir o operador de igualdade entre termos lógicos. A possibilidade de afirmar que dois termos são iguais ou não, em uma demonstração, pode ser bastante útil em determinadas circunstâncias.

Para tratar dessas situações a Lógica de Predicados pode ser estendida com o operador de igualdade entre termos. Neste caso é possível criar fórmulas lógicas no formato:

$$t_1 = t_2$$

onde tanto t_1 quanto t_2 são termos, ou seja, são constantes ou variáveis da Lógica de Predicados.

Para que essas fórmulas possam ser manipuladas nas demonstrações são utilizadas as regras de inferência da *identidade* e da *substituição*:

Tabela 15 – Regras de inferência da igualdade

Inserção da Igualdade	**Eliminação da Igualdade**
Identidade (**Id**)	Substituição (**Subs**)
$\dfrac{\quad}{t = t}$	$\dfrac{P(x) \quad x=t}{P(t)\ [x/t]} \qquad \dfrac{P(x) \quad t=x}{P(t)\ [x/t]}$

Fonte: elaborada pelos autores.

APÊNDICES

APÊNDICE A

TABELAS-VERDADE DOS OPERADORES LÓGICOS

Tabelas-verdade das operações lógicas binárias

A	**B**	$\mathbf{A \vee B}$	$\mathbf{A \wedge B}$	$\mathbf{A \rightarrow B}$	$\mathbf{A \leftrightarrow B}$
V	*V*	*V*	*V*	*V*	*V*
V	*F*	*V*	*F*	*F*	*F*
F	*V*	*V*	*F*	*V*	*F*
F	*F*	*F*	*F*	*V*	*V*

Tabela-verdade da operação lógica unária de negação

A	$\neg$**A**
V	*F*
F	*V*

APÊNDICE B

PROPRIEDADES DAS OPERAÇÕES

Equivalências da disjunção ($\vee$) e da conjunção ($\wedge$)

Propriedade	Disjunção (∨)	Conjunção (∧)
Comutativa	$A \vee B \Leftrightarrow B \vee A$	$A \wedge B \Leftrightarrow B \wedge A$
Associativa	$(A \vee B) \vee C \Leftrightarrow A \vee (B \vee C)$	$(A \wedge B) \wedge C \Leftrightarrow A \wedge (B \wedge C)$
Distributiva	$A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$	$A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$
Elemento neutro	$A \vee F \Leftrightarrow A$	$A \wedge V \Leftrightarrow A$
Complemento	$A \vee \neg A \Leftrightarrow V$	$A \wedge \neg A \Leftrightarrow F$
Idempotência	$A \vee A \Leftrightarrow A$	$A \wedge A \Leftrightarrow A$
DeMorgan:	$\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$	$\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$

Equivalências dos demais operadores

Dupla negação	$\neg\neg A \Leftrightarrow A$
Equivalência da implicação	$A \rightarrow B \Leftrightarrow \neg A \vee B$
Contraposição	$A \rightarrow B \Leftrightarrow \neg B \rightarrow \neg A$
Prova condicional	$A \rightarrow (B \rightarrow C) \Leftrightarrow (A \wedge B) \rightarrow C$

APÊNDICE C

SIMBOLIZAÇÃO DE EXPRESSÕES EM PORTUGUÊS

Expressão em português	**operador lógico**	**Expressão lógica**
e; mas; também;	conjunção	$A \wedge B$
ou	disjunção	$A \wedge B$
Se A, então B A implica B A, logo B A só se B; A somente se B B segue de A A é uma condição suficiente para B B é uma condição necessária para A B se A	implicação (condicional)	$A \rightarrow B$
A se e somente se B	bi-implicação	$A \leftrightarrow B$

A é condição necessária e suficiente para B	(bicondicional)	
não A É falso que A... Não é verdade que A...	negação	$\neg A$ $\sim A$

APÊNDICE D

REGRAS DE DEDUÇÃO DE EQUIVALÊNCIA E INFERÊNCIA

Regras básicas de inferência

Inclusão de Operadores	**Exclusão de Operadores**
Redução ao absurdo (**raa**) $\dfrac{\begin{array}{\|l} P \\ \ldots \\ Q \land \neg Q \end{array}}{\neg P}$	Dupla negação (**dn**) $\dfrac{\neg\neg P}{P}$
Prova condicional (**pc**) $\dfrac{\begin{array}{\|l} P \\ \ldots \\ Q \end{array}}{P \rightarrow Q}$	*Modus Ponens* (**mp**) $\dfrac{P \quad P \rightarrow Q}{Q}$
Conjunção(**cj**) $\dfrac{P \quad Q}{P \land Q}$	Simplificação (**sp**) $\dfrac{P \land Q}{P}$ $\dfrac{P \land Q}{Q}$
Adição(**ad**) $\dfrac{P}{P \lor Q}$ $\dfrac{P}{Q \lor P}$	Eliminação da disjunção - **$\lor$E** $\dfrac{P \lor Q \quad P \rightarrow R \quad Q \rightarrow R}{R}$
Introdução da equivalência - **$\leftrightarrow$I** $\dfrac{P \rightarrow Q \quad Q \rightarrow P}{P \leftrightarrow Q}$	Eliminação da equivalência - **$\leftrightarrow$E** $\dfrac{P \leftrightarrow Q}{P \rightarrow Q}$ $\dfrac{P \leftrightarrow Q}{Q \rightarrow P}$

Regras de inferência derivadas

Modus Tollens (**mt**) $\dfrac{P \to Q \quad \neg Q}{\neg P}$	Silogismo Hipotético (**sh**) $\dfrac{P \to Q \; Q \to R}{P \to R}$
Silogismo Disjuntivo (**sd**) $\dfrac{P \lor Q \quad \neg P}{Q}$	Dilema Construtivo (**dc**) $\dfrac{P \lor Q \quad P \to R \quad Q \to S}{R \lor S}$
Exportação (**exp**) $\dfrac{(P \land Q) \to R}{P \to (Q \to R)}$	Inconsistência (**inc**) $\dfrac{P \quad \neg P}{Q}$

Regras de equivalência

Expressão	**Equivale a**	**Nome (Abreviação) da Regra**
$P \lor Q$ $P \land Q$	$Q \lor P$ $Q \land P$	Comutatividade (**com**)
$(P \lor Q) \lor R$ $(P \land Q) \land R$	$P \lor (Q \lor R)$ $P \land (Q \land R)$	Associatividade (**ass**)
$\neg(P \lor Q)$ $\neg(P \land Q)$	$\neg P \land \neg Q$ $\neg P \lor \neg Q$	De Morgan (**dmor**)
$P \to Q$	$\neg P \lor Q$	Condicional (**cond**)
P	$\neg(\neg P)$	Dupla negação (**dn**)
$P \to Q$	$\neg Q \to \neg P$	Contraposição (**cont**)
P	$P \land P$ $P \lor P$	Auto-referência (**auto**)
$P \land (Q \lor R)$ $P \lor (Q \land R)$	$(P \land Q) \lor (P \land R)$ $(P \lor Q) \land (P \lor R)$	Distributividade (**dist**)

APÊNDICE E

REGRAS DE INFERÊNCIA DA LÓGICA DE PREDICADOS

Regras de inferência da lógica de predicados

Regra	Restrições de uso
Particularização universal (**pu**) $\dfrac{(\forall x)(P(x))}{P(t)}$	Se o novo termo t que substituirá a variável x em P(x) também for uma variável, então esta nova variável deve ser livre dentro da fórmula P(x) original.
Particularização existencial (**pe**) $\dfrac{(\exists x)\ (P(x))}{P(t)}$	O novo termo t que substituirá a variável x em P(x), quer seja variável ou constante, não deve ter sido usado anteriormente na demonstração.
Generalização universal (**gu**) $\dfrac{P(x)}{(\forall x)(P(x))}$	A fórmula P(x) não pode ter sido deduzida de nenhuma hipótese onde x é uma variável livre. A fórmula P(x) também não pode ter sido deduzida por particularização existencial (**pe**) de uma fórmula onde x é uma variável livre.
Generalização existencial (**ge**) $\dfrac{P(t)}{(\exists x)\ (P(x))}$	Se o termo t da fórmula original P(t) for um símbolo de uma constante do domínio, então a nova variável x que o substituirá não pode ter aparecido anteriormente na fórmula P(t).

Regras de equivalência dos quantificadores

Expressão	**Equivale a**	**Nome (abreviação) da regra**
$\neg\ \forall x\ P(x)$ $\neg\ \forall x \neg\ P(x)$ $\forall x\ \neg P(x)$ $\forall x\ P(x)$	$\exists x\ \neg\ P(x)$ $\exists x\ \ P(x)$ $\neg \exists x\ \ P(x)$ $\neg \exists x\ \ \neg P(x)$	Equivalência dos quantificadores (**equiv**)

Regras de inferência da igualdade

Inclusão da igualdade	**Exclusão da igualdade**
Identidade (id) – = I $\dfrac{}{t = t}$	Substituição (**subs**) - = E $\dfrac{P(x) \quad x = t}{P(t)}$ $\qquad$ $\dfrac{P(x) \quad t = x}{P(t)}$

ANEXOS

Anexo 1 - Revisão da Teoria dos Conjuntos

Um conjunto pode ser considerado como uma coleção de objetos, os *elementos* ou *membros* do conjunto. Normalmente são usadas letras maiúsculas para denotar conjuntos: A, B, C, X,Y, ... e letras minúsculas: a, b, c, x, y, z, ... para denotar elementos dos conjuntos.

A1.1 Relação de Pertinência

A afirmação "x é um elemento do conjunto A" (ou equivalentemente "x **pertence a** A") é simbolizada como:

$$x \in \mathrm{A}$$

Inversamente, a afirmação "x não é um elemento do conjunto A" (ou que "x **não pertence a** A") é simbolizada por:

$$x \notin \mathrm{A}$$

A1.2 Especificação de Conjuntos

Existem duas maneiras básicas de se especificar um conjunto:

(1) Listando seus elementos:

$$\mathrm{A} = \{a, b, c, d, e, f\}$$

Note que listagens por enumerações são possíveis: neste caso deve-se listar um número suficiente de elementos para que seja possível perceber a ordem de formação dos elementos, seguidos de três pontos "..." para indicar a continuação da sequência de elementos. Caso a sequência de elementos seja finita, basta indicar o último elemento. Assim o conjunto das letras minúsculas pode ser especificado como:

$$\mathrm{B} = \{a, b, c, \ldots, z\}$$

Caso a sequência seja infinita, basta deixar os "..." sem indicar um último elemento. Os números pares podem ser especificados como:

$$C = \{2, 4, 6, 8, \dots\}$$

(2) Definindo a propriedade que caracteriza os elementos do conjunto:

$$D = \{x \mid x \text{ é um número natural maior que 3 e menor que 8}\}$$

que deve ser lido como "D é o conjunto dos x tal que x é um número natural maior que 3 e menor que 8". Note que no segundo caso pode-se usar os operadores e símbolos usuais das fórmulas matemáticas. Assim o exemplo acima é escrito como:

$$D = \{x \mid x\square\mathbb{N} \text{ e } 3 < x < 8\}$$

Observações:

- Note que não há repetição de elementos iguais em conjuntos, assim os conjuntos:
  {a, b, c}, {a, a, b, c} e {a, b, b, b, c, c } são todos iguais ao conjunto {a, b, c}, contendo apenas 3 elementos.
- Também não há ordem em um conjunto, assim os conjuntos:
  {1, 2, 3 }, {3, 1, 2} e {3, 2, 1} também são todos iguais, contendo apenas os elementos 1, 2 e 3.

A1.3 Conjuntos Matemáticos

Os conjuntos usados na matemática são os seguintes:

- $\mathbb{N} = \{1, 2, 3, \dots\}$: o conjunto dos números naturais (sem o zero);
- $\mathbb{Z} = \{ \dots, -3, -2, -1, 0, 1, 2, 3, \dots \}$: o conjunto dos números inteiros;
- $\mathbb{Q}$ = o conjunto dos números racionais;
- $\mathbb{R}$ = o conjunto dos números reais;
- $\mathbb{C}$ = o conjunto dos números complexos.

A1.4 Conjunto Universo e Vazio

Em qualquer aplicação da Teoria dos Conjuntos os elementos de todos os conjuntos considerados devem pertencer a algum conjunto maior, conhecido como **conjunto universo**, representado pelo símbolo *U*. O conjunto que não contém elementos é denominado **conjunto vazio**, representado por Ø .

A1.5 Relação de Subconjunto

Se todo elemento de um conjunto A também é elemento de um conjunto B, então diz-se que A é um **subconjunto** de B. Também pode-se dizer que A está contido em B, ou que B contém A. A relação de subconjunto é escrita como:

$$A \subseteq B$$

Se A **não é um subconjunto** de B, isso é escrito como:

$$A \nsubseteq B$$

Um conjunto é **igual** a outro somente se ambos forem subconjuntos um do outro, ou seja, A = B se e somente se $A \subseteq B$ e $B \subseteq A$.

Quando um conjunto A é subconjunto de B, mas B tem elementos que não estão em A (B é "maior" que A), então se diz que A é um *subconjunto próprio* de B, simbolizado como:

$$A \subset B.$$

A1.6 Operações de União e Intersecção

A **união** de dois conjuntos A e B, denotada por $A \cup B$, é o conjunto de todos os elementos que pertencem a A ou a B:

$$A \cup B = \{ x \mid x \in A \text{ ou } x \in B\}$$

Note que o ou é usado no sentido do ou inclusivo ("e/ou").

A **intersecção** dos conjuntos A e B, denotada por A ∩ B, é conjunto de todos os elementos que pertence a A e a B:

$$A \cap B = \{\, x \mid x \in A \text{ e } x \in B\}$$

A1.7 Operações de Complementação e Diferença

O **complemento absoluto** de um conjunto A, definido como \A (ou também por –A), é o conjunto de todos os elementos que pertencem ao conjunto universo U, mas não pertencem a A:

$$\backslash A = \{\, x \mid x \in U \text{ e } x \notin A\}$$

O **complemento relativo** ou **diferença de conjuntos**, denotada por A\B (ou também por A – B), é o conjunto de todos os elementos que pertencem a A, mas não pertencem a B:

$$A \backslash B = \{x \mid x \in A \text{ e } x \notin B\}$$

A1.8 Conjuntos Finitos e Cardinalidade

Um conjunto A é **finito** se possui apenas n elementos distintos, onde n é algum número natural. Netse caso é possível se definir a **cardinalidade** (número de elementos) desse conjunto, representada po #A (ou às vezes por |A|), como igual a n:

$$\#A = n$$

REFERÊNCIAS

ALENCAR FILHO, Edgard. *Iniciação à lógica matemática*. São Paulo: Nobel, 1999.

AZEREDO, Vânia Dutra de (Org.). *Introdução à lógica*. Ijuí: Unijuí, 2000.

DAGHLIAN, Jacob. *Lógica e algebra de Boole*. São Paulo: Atlas, 1995.

DEL PICCHIA, Walter. *Métodos numéricos para resolução de problemas lógicos*. São Paulo: Edgard Blücher, 1993.

GERSTING, Judith L. *Fundamentos matemáticos para a Ciência da Computação*. Rio de Janeiro: LTC, 2001 (trad. 4 ed.).

HEGENBERG, Leônidas. *Lógica – O cálculo de predicados*. São Paulo: USP, 1973.

HEIN, James L. *Discrete structures, logic, and computability*. 2nd. ed. Boston: Jones and Bartlett, 2002.

HUTH, Michael; RYAN, Mark. *Logic in computer science: modelling and reasoning about systems*. 2nd ed. Cambridge: Cambridge University, 2004..

LIPSCHUTZ, Seymour. *Teoria dos conjuntos*. São Paulo: McGraw Hill, 1976.

MENDELSON, Elliott. *Álgebra Booleana e circuitos de chaveamento*. São Paulo: Mc Graw Hill, 1977.

MORTARI, Cezar A. *Introdução à lógica*. São Paulo: UNESP, 2001.

NOLT, John; ROHATYN, Dennis. *Lógica*. São Paulo: McGraw-Hill, 1991 (Coleção Schaum).

POFFAL, Cristiana Andrade; RENZ, Sandra Pacheco. *Fundamentos de lógica matemática*. Porto Alegre: La Salle, 2001.

YAGLOM, I. M. *Álgebra Booleana*. São Paulo: Atual, 1999.

SOBRE OS AUTORES

JOÃO CARLOS GLUZ

Doutor e mestre em Ciências da Computação pela Universidade Federal do Rio Grande do Sul (UFRGS). Professor adjunto da UNISINOS. Tem experiência de ensino nas áreas de Arquitetura de Computadores, Sistemas Operacionais, Redes de Computadores, Teoria da Computação, Engenharia de Software, Algoritmos e Estruturas de Dados.

MÔNICA XAVIER PY

Mestre em Ciência da Computação pela UFRGS. Graduada em Ciência da Computação pela Universidade Católica de Pelotas (UCPel). Professora da UNISINOS. Tem experiência de ensino nas áreas de Paradigmas e Linguagens de Programação, Projeto e desenvolvimento de Sistemas para Web, Linguagens Formais e Teoria da Computação.

UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS

Reitor: Pe. Marcelo Fernandes de Aquino, SJ
Vice-reitor: Pe. José Ivo Follmann, SJ
Diretor da Editora Unisinos: Pe. Pedro Gilberto Gomes

Editora Unisinos
Avenida Unisinos, 950, 93022-000, São Leopoldo, Rio Grande do Sul, Brasil
editora@unisinos.br
www.edunisinos.com.br

G567 Gluz, João Carlos.
Lógica para computação / João Carlos Gluz, Mônica Xavier Py. – São Leopoldo : Ed. UNISINOS, 2014. (EaD)

Disponível apenas em versão e-book
ISBN 978-85-7431-690-1

1. Lógica simbólica e matemática. 2. Computação Matemática. I. Py, Mônica Xavier. II. Título. III. Série.

CDD 511.3

CDU 510.6

Dados Internacionais de Catalogação na Publicação (CIP)
(Bibliotecário: Flávio Nunes – CRB 10/2145)

Coleção EAD
Editor: Carlos Alberto Gianotti
Acompanhamento editorial: Jaqueline Fagundes Freitas
Revisão: Bianca Basile Parracho
Editoração: Guilherme Hockmüller

A coleção EaD, de que faz parte este livro, é uma produção da Universidade do Vale do Rio dos Sinos para apoiar os processos de ensino e aprendizagem dos seus cursos de graduação a distância. Entretanto, o uso dessa obra não fica restrito apenas a essa modalidade de ensino, uma vez que pode servir como orientador no estudo de qualquer acadêmico.
Os exemplares foram elaborados a partir da experiência de professores de reconhecido mérito acadêmico da Universidade, e traduzem a excelência dos cursos de graduação ofertados na modalidade presencial e a distância da Instituição.

AUSJAL
MEMBRO DA
REDE DE EDITORAS UNIVERSITÁRIAS DA AUSJAL
www.ausjal.org

COLEÇÃO EAD
EDITORA UNISINOS

UNISINOS