Lógica
Notas de aula

Marcus Ritt

5 de Junho de 2007

Universidade Federal do Rio Grande do Sul
Instituto de Informática
Departamento de Informática Teórica

Conteúdo

1. Introdução

A lógica é a ciência do raciocínio correto. Ela é base da matemática e de muitas áreas da ciência de computação. Um exemplo importante é a especificação e verificação de sistemas computacionais.

Importância da lógica

"Que coisa imbecil, o Amor!" resmungou o estudante, afastando-se. "Nem vale a utilidade da Lógica, porque não prova nada, está sempre prometendo o que não cumpre e fazendo acreditar em mentiras. Nada tem de prático e como neste século o que vale é a prática, volto à Filosofia e vou estudar metafísica."

Oscar Wilde, O Rouxinol e a Rosa.

A lógica é a ciência do raciocínio. Por isso, ela é omnipresente: Encontramos-a em todas as ciências bem como a dia-a-dia.
Na informática ela tem muitas aplicações, por exemplo,

- na construção de circuitos digitais,
- em linguagens de programação,
- no estudo teórico de linguagens de programação,
- na inteligência artificial,
- em bancos de dados, e
- na teoria de complexidade.

Dia-a-dia: É lógico que...

Em que estado o senhor encontra a instituição hoje?

> Encontramos a universidade em um estado muito bom. É lógico que não é o ideal, mas a situação atual é muito boa.

Entrevista de José Carlos Ferraz Hennemann falando dos projetos e das prioridades neste início de mandato à frente da UFRGS. Universia 14/10/2004.

O que quer "é lógico" dizer?

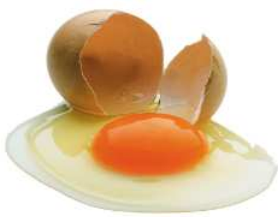
Mais adiante vamos ver, que isso pode ser modelado usando uma proposição p que significa "O estado da universidade não é ideal". Como a definição do "ideal" é: "Um conceito perfeito, que não da pra chegar", p sempre é falso e a sua negação $\neg p$ é uma tautologia (i.e. sempre é verdadeiro). Portanto a afirmação acima "é lógico".

É lógico que...

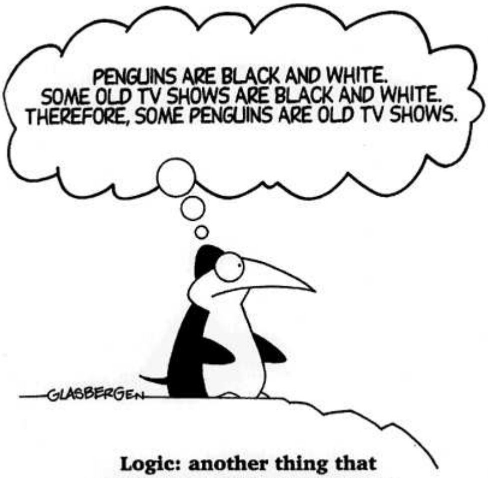
- Se as "circunstâncias" permitem "creer" um fato "é lógico" que esse fato é verdadeiro.
- Na lógica chamamos
  - os fatos que supomos as *premissas* (ou *antecedentes*).
  - o resultado do raciocínio a *conclusão* (ou *sucedente*).
- Na lógica pesquisamos as regras que permitem chegar das premissas às conclusões.

Raciocínio: Exemplo

$$\text{Se o ovo cai, então o ovo quebra.} \tag{1.1}$$
$$\text{O ovo cai.} \tag{1.2}$$
$$\text{Logo, o ovo quebra.} \tag{1.3}$$

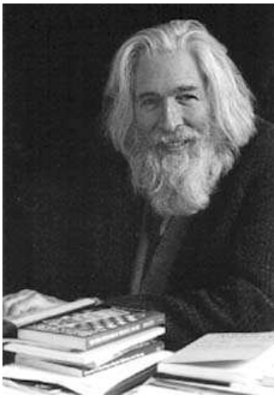


Raciocínio: Outro exemplo



Logic: another thing that penguins aren't very good at.

Raciocínio: Outro exemplo



Some cars rattle. My car is some car. Therefore, my car rattles.

Outra brincadeira atribuído a Smullyan.

> "I'll make a statement. If the statement is true, you give me your autograph. It doesn't have to be on a check, it can be on a blank piece of paper. If the statement is false, you don't give me your autograph,"Smullyan sets up the puzzle. "Well, my statement is, 'You will give me neither your autograph nor a kiss.'

> ”If it were true, you’d have to give me your autograph as agreed, but that would falsify the statement. You’d have a contradiction. So therefore the statement must be false. Since it’s false that you’ll give me neither, it means you’ll have to give me either. But you can’t give me your autograph for a false statement, so you owe me a kiss.”

2. Lógica proposicional

2.1. Introdução

Uma lógica simples, todavia importante é a lógica proposicional. O nome é devido as proposições que são os componentes atômicos da lógica proposicional. Uma proposição é uma declaração sobre algum sistema em consideração. Uma característica fundamental é que a lógica proposicional é uma lógica dual: uma proposição pode ser verdadeira ou falsa, mas não tem terceira possibilidade (um fato conhecido como o "lei do terceiro excluído", tertium non datur, law of the excluded middle). Verdadeiro (V) e falso (F) são *valores lógicos* ou *valores de verdade*. As proposições são denotadas com variáveis proposicionais.

Proposições

Para começar, abstraímos das frases particulares. Em geral, temos

- *Proposições* elementares ou atômicas (ou sentenças declarativas), que denotamos com símbolos $p, q, r, \ldots$. Por exemplo:

$$p : \text{“O ovo cai”}$$
$$q : \text{“O ovo quebra”}$$

Definição 2.1 (Proposição)

Uma *proposição* (frase declarativa, sentença declarativa) é uma afirmação na linguagem natural que tem um valor de verdade (que pode ser verdadeiro ou falso). Denotamos proposições com *variáveis proposicionais*. Por convenção, usamos letras minúsculas $p, q, r, \ldots$ para elas (com alterações como $p_1, q', r'_2, \ldots$).

Exemplo 1 (Proposições)

Exemplos de proposições são

1. p: Hoje é um dia lindo.

2. q: Meu computador é quebrado.

3. r: Francesco gosto churrasco.

4. s: Brasil ganha o copo do mundo.

(Os exemplos mostram também nossa convenção de declarar variáveis proposicionais.)
Contra-exemplos de proposições são

1. Oi! Bom dia! Obrigado!

2. Silêncio!

3. Vamos!

4. Cuidado! Socorro!

$\Diamond$

Conectivos

Temos

- *Conectivos* entre as proposições: Se p então q.

- Isso é um exemplo da *implicação* $p \to q$. Lê: "p implica q".

- Assim, em nosso exemplo temos as premissas p, $p \to q$ e a conclusão q. Escrevemos

$$p, p \to q \vdash q$$

O seqüente

Mais geral: Dado as premissas $\Phi_1, \ldots, \Phi_n$ e a conclusão Ψ, escrevemos

$$\Phi_1, \ldots, \Phi_n \vdash \Psi$$

Lê: "Das premissas $\Phi_1, \ldots, \Phi_n$ se pode concluir Ψ." Ou: "q segue da $\Phi_1, \ldots, \Phi_n$", "$\Phi_1, \ldots, \Phi_n$ portanto Ψ"

- $\Phi_1, \ldots, \Phi_n \vdash q$ se chama *seqüente*.

- Um seqüente é *válido*, se a conclusão é o resultado das premissas (i.e. pode ser provado a partir das premissas).

- Se $\vdash \Psi$, i.e. a conclusão não depende de premissas, Ψ é um *teorema*.

Observe que as letras gregas Φ e Ψ denotam fórmulas lógicas arbitrárias. Não é para confundir com as variáveis p, q, etc. que denotam proposições. O símbolo $\vdash$ (barra de inferência, inglês: turnstile) é uma relação (de dedutibilidade ou demonstrabilidade) entre as premissas e a conclusão. Com a com $\Phi_1, \ldots, \Phi_n \vdash \Psi$ queremos afirmar, que a partir das n premissas Φ_1 até Φ_n, um raciocínio lógico, i.e. a aplicação da regras lógicas, permite chegar na conclusão Ψ. Observe, que um seqüente não é uma regra lógica: vamos definir-lhes no capítulo 2.3. O seqüente acima foi justificado intuitivamente.

Seqüente: Exemplo
Premissas:

Francesco gosta de jogar ou de estudar (ou ambas).
Francesco não gosta de estudar.

Conclusão? Francesco gosta de jogar.

- Escrevemos $p \lor q$ se p, q ou ambas são verdadeiras.
- Escrevemos $\neg p$ se a negação de p é verdadeira.
- Escrevemos $p \land q$ se p é q são verdadeiras.

p : Francesco gosta de jogar.
q : Francesco gosta de estudar.

Então $p \lor q, \neg q \vdash p$ é válido.

Exemplo 2
Se a janela está aberto, vento entra.
Não entra vento.
Logo, a janela está fechado. $\diamondsuit$

2.2. Sintaxe

Usando as proposições, podemos construir *fórmulas* com *operadores* ou *conectivos*. Os operadores mais comuns são: a negação de uma proposição e a conjunção ou disjunção de duas proposições, com a notação, $\neg p$, $p \land q$ e $p \lor q$, respectivamente. Uma conjunção afirme que ambas proposições p e q são verdadeiros e uma disjunção afirme que ao menos uma das proposições (talvez ambas) são verdadeiras.

Exemplo 3 (Operadores "não", "e" e "ou")
Com as proposições do exemplo 1 temos:

1. $\neg p$: Hoje *não é* uma dia lindo.
2. $p \land s$: Brasil ganha o copo do mundo *e* hoje é um dia lindo.
3. $p \lor q$: Brasil ganha o copo do mundo *ou* meu computador é quebrado (ou ambas).

$\Diamond$

Fórmulas na lógica proposicional

Temos todos os ingredientes para construir *fórmulas* arbitrárias na lógica proposicional.

1. Um conjunto de átomos $\mathrm{Atom} = \{p, q, r, \dots\}$
2. O conjunto de fórmulas (bem formadas) $\mathcal{L}$ (com $\Phi \in \mathcal{L}$)

$$\Phi ::= p \mid (\neg\Phi) \mid (\Phi \lor \Psi) \mid (\Phi \land \Psi) \mid (\Phi \to \Psi) \mid \top \mid \bot$$

   com $p \in \mathrm{Atom}$.

Nomes, nomes, nomes:

- Um *literal* é um átomo (p) ou a negação dele ($\neg p$).
- Os conectivos $\neg$, $\lor$, $\land$, $\to$ são a *negação*, *disjunção*, *conjunção* e *implicação*.

Observe que em nossa definição o conjunto de átomos é infinito, mais cada fórmula só precisa um número finito deles.
Como diferençar entre cadeias de letras arbitrárias e fórmulas? Caso uma dada cadeia de letras (um string) é uma fórmula, podemos provar isso mostrando uma derivação dessa fórmula na gramática. Por exemplo o string "$(p \to (\neg q))$" é uma fórmula, porque temos a derivação

$$\Phi \Rightarrow (\Phi \to \Psi) \Rightarrow (p \to \Psi) \Rightarrow (p \to (\neg\Phi)) \Rightarrow (p \to (\neg q))$$

Do outro lado, se queremos mostrar que uma dado string não é uma fórmula, temos que argumentar, que esse string não tem uma derivação na gramática da lógica proposicional. Por exemplo o string "$p \to$" não é uma fórmula porque o lado direto do $\to$ é vazio, mas a gramática não permite derivar o string vazio. Isso é um exemplo de um argumento informal; formalmente temos que provar fatos desse tipo com indução sobre as fórmulas (veja também exercício 2.2).

Notação simplificada

$$((p \lor q) \to (r \land (\neg s)))$$

é uma fórmula bem formada. Na prática, o número das parênteses incomoda. Usamos algumas convenções para abreviar fórmulas:

- Prioridade: $\neg$ tem mais prioridade que $\lor$ e $\land$, quais tem mais prioridade que $\to$.

- Associação: $\to$ associa para direita, $p \to q \to r$ denota $(p \to (q \to r))$; para $\lor$ e $\land$ usamos os parênteses!

Fórmula completa	Fórmula abreviada
$(\neg(\neg(\neg(\neg p))))$	$\neg\neg\neg\neg p$
$((p \lor q) \to (r \land (\neg s)))$	$p \lor q \to r \land \neg s$

Árvores de parse

Considere $p \lor q \to r \land \neg s$. A correspondente *árvore de parse* é

$$\begin{tikzcd}
 & & & \to \arrow[dll] \arrow[drr] & & & \\
 & \lor \arrow[dl] \arrow[dr] & & & & \land \arrow[dl] \arrow[dr] & \\
p & & q & & r & & \neg \arrow[d] \\
 & & & & & & s
\end{tikzcd}$$

Uma *subfórmula* de uma fórmula é cada sub-árvore de sua árvore de parse. (Veja exercício 2.3).

2.2.1. Indução

Indução

Para provar uma proposição $P(n)$ sobre $\mathbb{N}$

1. Base: Prova que $P(0)$

2. Passo: Prova que $P(n) \Rightarrow P(n+1)$

Essa técnica se chama *indução matemática* ou *indução natural*.
Exemplo $P(n) = \left(\sum_{0 \le i \le n} i = n(n+1)/2\right)$:

1. Base: $P(0) = \left(\sum_{0 \le i \le 0} i = 0(0+1)/2\right)$

2. Passo: Suponha $P(n)$.

$$\begin{aligned}\sum_{0\le i\le n+1} i &= \sum_{0\le i\le n} i + (n+1)\\ &= n(n+1)/2 + (n+1) = (n+1)(n+2)/2.\end{aligned}$$

Logo $P(n+1)$.

Exemplo
Cada número natural par não igual 0 é a soma de dois números ímpares?

$$P(n) = (2n \text{ é a soma de dois números impares})$$

- Base: $2 = 1 + 1$
- Passo: Suponha $P(n)$: Existem i, j tal que $2n = (2i + 1) + (2j + 1)$. Logo

$$\begin{aligned}2(n+1) &= 2n + 2 = (2i+1) + (2j+1) + 2\\ &= (2i+1) + (2j+3) = (2i+1) + (2(j+1)+1)\end{aligned}$$

- Observe: A indução começa com $n = 1$!

Indução completa

- Com indução natural, provamos $P(n+1)$ usando $P(n)$.
- As vezes uma prova de $P(n+1)$ só é possível usando algumas (ou todos) $P(k)$ com $k < n$.
- Esse tipo de argumento também é possível e se chama *indução completa*.
- Para provar $P(n)$ prove
  - Se $P(k)$ para qualquer $k < n$, então $P(n)$.
- E o caso $P(0)$?

Observe que o caso $P(0)$ já está incluído na prova. Se $n = 0$, $P(k)$ é verdadeiro para qualquer $k < n$, porque não tem $k < n$. Logo, a prova da implicação "Se $P(k)$ para qualquer $k < n$, então $P(n)$" inclui a prova que $P(0)$ é verdadeiro.

Exemplo
Sejam os números f_i definido como $f_0 = 0$, $f_1 = 1$ e $f_n = f_{n-1} + f_{n-2}$ para $n \geq 2$.

$$P(n) = (f_n \leq 2^n)\,?$$

Prova com indução completa: Seja $P(k)$ para qualquer $k < n$. Objetivo: Provar $P(n)$. Base: Se $n = 0$ ou $n = 1$

$$f_0 = 0 < 1 = 2^0; \qquad f_1 = 1 < 2^1.$$

Passo: Se $n \geq 2$

$$\begin{aligned} f_n &= f_{n-1} + f_{n-2} && \text{por definição de } f_n \\ &\leq 2^{n-1} + 2^{n-2} && \text{usando a hipótese da indução} \\ &= 3(2^{n-2}) && \text{distributividade} \\ &< 2^n \end{aligned}$$

2.3. Teoria de provas

Provas são a base da matemática. A partir de *axiomas*, que são proposições ou conjuntos de proposições supostas verdadeiras, um raciocínio correto, uma prova, justifica conseqüências. Dependente do objetivo, os resultados matemáticas tem nomes diferentes:

Proposição Uma proposição é um resultado simples. Não se confunde com as proposições da lógica proposicional.

Lema Um lema (de grego: gancho) é um resultado intermediário, que ajuda a prova de outros resultados.

Teorema Um teorema é um resultado central ou importante.

Corolário Um corolário é uma conseqüência simples de outros resultados.

A seguir vamos estudar uma versão formalizada de prova chamada *dedução natural* ou *sistema de prova do tipo Gentzen* (o nome é devido ao inventor Gerhard Gentzen).

Como concluir?

Ainda não sabemos como chegar na conclusão: Os exemplos foram analisados intuitivamente.
Regras de prova nos liberam dessa situação! Lembre-se do exemplo:

$$p \to q, p \vdash q$$

Esse seqüente pode ser justificado usando a regra *eliminação da implicação*:

$$\frac{\Phi \quad \Phi \to \Psi}{\Psi} \to_{\mathrm{e}}$$

Regras como essa tentam de modelar o nosso raciocínio; por isso, um raciocínio seguindo essa regra (e as outras que nos vamos ver em breve) se chama *dedução natural*.

Notação para regras

Em geral, as regras tem a notação

$$\frac{P_1 \quad P_2 \cdots P_n}{C} \text{nome}$$

com premissas $P_1, P_2, \ldots, P_n$ e conclusão C. Para referir-se a uma regra em provas, ela tem um nome.
Observe que o caso $n = 0$ é possível. Uma regra desse tipo não tem premissas é esta chamada *axioma*. Escrevemos

$$\frac{\emptyset}{C} \text{nome}$$

Teoremas e fórmulas equivalentes

- Um seqüente que não depende de premissas

  $$\vdash \Phi$$

  se chama *teorema*.

- Se temos duas fórmulas Φ e Ψ tal que

  $$\Phi \vdash \Psi$$
  $$\Psi \vdash \Phi$$

  eles são equivalentes (em termos de provas). Escrevemos também

  $$\Phi \dashv\vdash \Psi$$

Falta um conectivo?

- Quais são os conectivos da lógica proposicional?
- $\wedge, \vee, \neg, \rightarrow$
- E $p \leftrightarrow q$ (o bicondicional)?
- É uma abreviação para $(p \rightarrow q) \wedge (q \rightarrow p)$

Observe que nossa definição da linguagem formal da lógica proposicional não contém $\leftrightarrow$. Por isso, $p \leftrightarrow q$ formalmente não é uma fórmula, mas uma abreviação na meta-linguagem, i.e. na linguagem que nos estamos usando para trabalhar com a lógica. Essa diferença é devido objetivos diferentes no uso da lógica: Se queremos aplicar a lógica na modelagem de sistemas, é conveniente de ter uma sintaxe rica, que simplifica a descrição. Do outro lado, se queremos provar teoremas *sobre a lógica*, o trabalho é menos se a definição da lógica é a mais breve possível.

Regras para a conjunção

- Introdução da conjunção
  $$\frac{\Phi \quad \Psi}{\Phi \wedge \Psi} \wedge_{\mathrm{i}}$$
  O tanque está vazio. O motor funciona. Logo, o tanque está vazio **e** o motor funciona.
- Eliminação da conjunção
  $$\frac{\Phi \wedge \Psi}{\Phi} \wedge_{\mathrm{e}_1} \qquad \frac{\Phi \wedge \Psi}{\Psi} \wedge_{\mathrm{e}_2}$$
  O tanque está vazio e o motor funciona. (a) Logo, o tanque está vazio. (b) Logo, o motor funciona.

Árvores de prova

Provamos um seqüente composto

$$p, q, r, s \vdash (p \wedge q) \wedge (r \wedge s)$$

Usando a introdução de conjunção múltiplas vezes, obtemos uma árvore de prova:

$$\frac{\dfrac{p \quad q}{p \wedge q} \wedge_{\mathrm{i}} \quad \dfrac{r \quad s}{r \wedge s} \wedge_{\mathrm{i}}}{(p \wedge q) \wedge (r \wedge s)} \wedge_{\mathrm{i}}$$

Notação linear

Para provas mais complicados, árvores de prova ficam complicado. Uma alternativa é uma prova linear, com referências:

1	p	premissa
2	q	premissa
3	r	premissa
4	s	premissa
5	$p \wedge q$	$\wedge_{\mathrm{i}} 1, 2$
6	$r \wedge s$	$\wedge_{\mathrm{i}} 3, 4$
7	$(p \wedge q) \wedge (r \wedge s)$	$\wedge_{\mathrm{i}} 5, 6$

Exemplo: Associatividade

$$p \wedge (q \wedge r) \vdash (p \wedge q) \wedge r \tag{2.1}$$

1	$p \wedge (q \wedge r)$	premissa
2	p	$\wedge_{\mathrm{e}_1}$ 1
3	$q \wedge r$	$\wedge_{\mathrm{e}_2}$ 1
4	q	$\wedge_{\mathrm{e}_1}$ 3
5	r	$\wedge_{\mathrm{e}_2}$ 3
6	$p \wedge q$	$\wedge_{\mathrm{i}} 2, 4$
7	$(p \wedge q) \wedge r$	$\wedge_{\mathrm{i}} 6, 5$

- Logo, $p \wedge (q \wedge r) \vdash (p \wedge q) \wedge r$ é válido.
- E a inversa $(p \wedge q) \wedge r \vdash p \wedge (q \wedge r)$? Também (Exercício!).

$$p \wedge (q \wedge r) \dashv\vdash (p \wedge r) \wedge r$$

- Por isso, "$\wedge$ é associativo."

Exemplo: Comutatividade

$$p \wedge q \vdash q \wedge p \tag{2.2}$$

1	$p \wedge q$	premissa
2	p	$\wedge_{\mathrm{e}_1} 1$
3	q	$\wedge_{\mathrm{e}_1} 2$
4	$q \wedge p$	$\wedge_{\mathrm{i}} 2, 3$

- Logo $p \wedge q \vdash q \wedge p$.
- A inversa obviamente é válido também; brevemente: "$\wedge$ é comutativo".

Regras para negação dupla

- Eliminação da negação dupla

$$\frac{\neg\neg\Phi}{\Phi}\ \neg\neg_{\mathrm{e}}$$

Não é que o motor não funciona. Logo, o motor funciona.

- Introdução da negação dupla

$$\frac{\Phi}{\neg\neg\Phi}\ \neg\neg_{\mathrm{i}}$$

O tanque está vazio. Logo, não é que o tanque não está vazio.

Exemplo

1	$\neg\neg(p \wedge q)$	premissa
2	$p \wedge q$	$\neg\neg_{\mathrm{e}}1$
3	p	$\wedge_{\mathrm{e}_1}2$
4	q	$\wedge_{\mathrm{e}_2}2$
5	$\neg\neg p$	$\neg\neg_{\mathrm{i}}3$
6	$\neg\neg q$	$\neg\neg_{\mathrm{i}}4$
7	$\neg\neg p \wedge \neg\neg q$	$\wedge_{\mathrm{i}}5, 6$

Logo, $\neg\neg(p \wedge q) \vdash \neg\neg p \wedge \neg\neg q$ é valido.

Eliminação da implicação

A eliminação da implicação também é conhecida como *modus ponens*: Sabendo que Φ implica Ψ e Φ é correto, Ψ tem que ser correto também.

$$\frac{\Phi \quad \Phi \to \Psi}{\Psi}\ \to_{\mathrm{e}}$$

O motor funciona. Se o motor funciona, o carro anda. Logo, o carro anda.

Exemplo

O seqüente $p, p \to q, p \to (q \to r) \vdash r$ é válido?

1	p	premissa
2	$p \to q$	premissa
3	$p \to (q \to r)$	premissa
4	$q \to r$	$\to_{\mathrm{e}}$ 1,3
5	q	$\to_{\mathrm{e}}$ 1,2
6	r	$\to_{\mathrm{e}}$ 5,4

Modus tollens

Uma outra possibilidade da eliminação da implicação é o raciocínio seguinte: Sabendo que Φ implica Ψ e também Ψ é correto, Φ não pode ser correto.

$$\frac{\Phi \to \Psi \quad \neg\Psi}{\neg\Phi}\,\mathrm{MT}$$

Se o motor funciona, o carro anda. O carro não anda. Logo, o motor não funciona.

Exemplo

$$p \to \neg q, q \vdash \neg p$$

1	$p \to \neg q$	premissa
2	q	premissa
3	$\neg\neg q$	$\neg\neg_{\mathrm{i}}$ 2
4	$\neg p$	MT 1,3

Introdução da implicação

O raciocínio da implicação $p \to q$ e

Se p é verdadeiro, **então** q é verdadeira.

Nesta situação não sabemos se p é verdadeiro. Então, como introduzir uma implicação?
Se suponhamos *temporariamente* que p é verdadeira, e, usando essa hipótese podemos justificar q, a introdução da implicação $p \to q$ é justificada também.
Temos que diferenciar entre premissas é hipóteses. Para isso, usamos uma caixa:

$$\frac{\boxed{\begin{matrix}\Phi \\ \vdots \\ \Psi\end{matrix}}}{\Phi \to \Psi}\,\to_{\mathrm{i}}$$

As hipóteses são proibidos de fugir da caixa!
Compare a regra da introdução da implicação com um raciocínio comum na matemática. Por exemplo, queremos provar a proposição "se $x > 0$ então $x > -1$", i.e. a implicação $x > 0 \to x > -1$. Um raciocínio típico e **Prova.** Suponha $x > 0$. Como $0 > -1$ (isso é um axioma) e com a transitividade de $>$ temos $x > 0 > -1$. ∎

Exemplo: Distribuição

1	$p \to q \wedge r$	premissa
2	p	hipótese
3	$q \wedge r$	$\to_e$ 2, 1
4	q	$\wedge_{e_1}$ 3
5	$p \to q$	$\to_i$ 2 − 4
6	p	hipótese
7	$q \wedge r$	$\to_e$ 2, 1
8	r	$\wedge_{e_2}$ 7
9	$p \to r$	$\to_i$ 6–8
10	$(p \to q) \wedge (p \to r)$	$\wedge_i$ 5, 9

Logo, $p \to q \wedge r \vdash (p \to q) \wedge (p \to r)$ é valido ou brevemente "$\to$ distribui sobre $\wedge$".

Exemplo: Transitividade

O seqüente $p \to q, q \to r \vdash p \to r$ é válido?

1	$p \to q$	premissa
2	$q \to r$	premissa
3	p	hipótese
4	q	$\to_e$ 3,1
5	r	$\to_e$ 4,2
6	$p \to r$	$\to_i$ 3–5

Logo, $p \to q, q \to r \vdash p \to r$ é válido ("a implicação é transitivo").

Exemplo

$p \to (q \to p)$ é um teorema?

1	p	hipótese
2	q	hipótese
3	p	cópia 1
4	$q \to p$	$\to_i$ 2–3
5	$p \to (q \to p)$	$\to_i$ 1–4

- Como a introdução de implicação no passo 3 tem que acabar com p, precisamos uma cópia.
- É permitido de copiar fórmulas de fora de uma caixa para dentro (mas *não* na outra direção!)
- Anotamos "cópia" com uma referência da linha fonte.

Introdução da disjunção

$$\frac{\Phi}{\Phi \vee \Psi}\ \vee_{\mathrm{i}_1}$$

O motor funciona. Logo, o motor funciona ou o tanque é vazio.

$$\frac{\Psi}{\Phi \vee \Psi}\ \vee_{\mathrm{i}_2}$$

O motor funciona. Logo, o motor funciona ou o mundo é um disco.

Eliminação da disjunção

$$\frac{\Phi \vee \Psi \quad \boxed{\begin{array}{c}\Phi\\ \vdots\\ \chi\end{array}} \quad \boxed{\begin{array}{c}\Psi\\ \vdots\\ \chi\end{array}}}{\chi}\ \vee_{\mathrm{e}}$$

Se eu ganho no loto, eu fico rico. Se eu herdo muito dinheiro, eu fico rico. Ganho no loto ou herdo muito dinheiro.

Supondo, eu ganho no loto, logo, eu fico rico. Supondo, eu herdo muito dinheiro, eu fico rico.

Logo, eu fico rico.

Exemplo: Distribuição

$$p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r) \tag{2.3}$$

1	$p \wedge (q \vee r)$	premissa
2	p	$\wedge_{\mathrm{e}_1} 1$
3	$q \vee r$	$\wedge_{\mathrm{e}_1} 2$
4	q	hipótese
5	$p \wedge q$	$\wedge_{\mathrm{i}} 2, 4$
6	$(p \wedge q) \vee (p \wedge r)$	$\vee_{\mathrm{i}_1} 5$
7	r	hipótese
8	$p \wedge r$	$\wedge_{\mathrm{i}} 2, 7$
9	$(p \wedge q) \vee (p \wedge r)$	$\vee_{\mathrm{i}_2} 8$
10	$(p \wedge q) \vee (p \wedge r)$	$\vee_{\mathrm{e}} 3, 4-6, 7-9$

Logo $p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$, ou brevemente "$\wedge$ distribui sobre $\vee$".

Introdução da negação

- Se alguma hipótese permite deduzir uma contradição, podemos concluir que a negação dessa hipótese tem que ser válido.
- Esse tipo de raciocínio se chama *reductio ad absurdum*.

$$\frac{\boxed{\begin{array}{c}\Phi\\ \vdots\\ \bot\end{array}}}{\neg\Phi}\ \neg_{\mathrm{i}}$$

Eliminação da negação

- Qualquer fórmula de forma $\Psi \land \neg\Psi$ é uma *contradição*.
- Escrevemos $\bot$ para a contradição.
- Se encontramos uma fórmula e a negação dela, podemos concluir uma contradição.

$$\frac{\Psi \quad \neg\Psi}{\bot}\ \neg_{\mathrm{e}}$$

O carro anda. O carro não anda. Logo, temos uma contradição.

Exemplo: Contraposição

1	$p \rightarrow q$	premissa
2	$\neg q$	hipótese
3	p	hipótese
4	q	$\rightarrow_{\mathrm{e}}$ 3,1
5	$\bot$	$\neg_{\mathrm{e}}$ 4,2
6	$\neg p$	$\neg_{\mathrm{i}}$ 3–5
7	$\neg q \rightarrow \neg p$	$\rightarrow_{\mathrm{i}}$ 2–6

Logo

$$p \rightarrow q \vdash \neg q \rightarrow \neg p \tag{2.4}$$

O objetivo da prova acima e mostrar o uso de $\neg_{\mathrm{i}}$ e $\bot_{\mathrm{e}}$. Usando o modus tollens obtemos a prova mais simples

1	$p \rightarrow q$	premissa
2	$\neg q$	hipótese
3	$\neg p$	MT
4	$\neg q \rightarrow \neg p$	$\rightarrow_{\mathrm{i}}$ 2–3

Eliminação da contradição

- Dado uma contradição podemos concluir qualquer coisa.
- A eliminação da contradição é uma das regras pouco intuitiva.

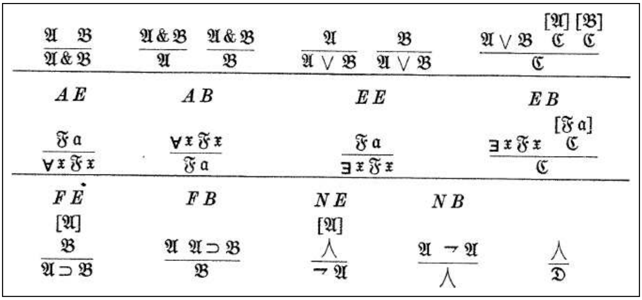
$$\frac{\perp}{\Phi}\,\perp_e$$

O mundo é redondo. O mundo não é redondo. Logo, a grama é azul.
Essa regra também está chamada “o lei de Duns Scotus”. Ela é uma das regras menos intuitivas. Um jeito de pensar sobre a regra é que se existe um “mundo” que permite a corretude de uma fórmula e a sua negação, esse mundo é “cheio” demais: Com uma hipótese dessa, podemos concluir tudo. Por isso, uma lógica que tem essa característica as vezes é chamada *explosiva*. O exemplo também mostra a falta de relevância entre as premissas e a conclusão. Lógicas *relevantes* exigem um vínculo desse tipo.

A arte de construir provas

- Primeiro, escrevemos as premissas em cima e a conclusão em baixo.
- Agora, o objetivo é de encher o espaço em branco entre os dois.
- Quais regras de prova podemos aplicar às premissas?
  - Tem um $\wedge$? Usa $\wedge_{e_1}$ ou $\wedge_{e_2}$.
  - É provável de precisar uma conjunção das premissas? Usa $\wedge_i$.
  - Tem um $\vee$? Da para eliminar com $\vee_e$? Qual seria uma conclusão que ajuda?
  - Ajuda de introduzir um $\vee$? Qual seria a outra fórmula que ajuda?
- Quais regras de prova ajudam em chegar na conclusão?
  - A conclusão é $\Phi \to \Psi$? Usa $\to_i$ e tenta de chegar a Ψ de Φ.
  - A conclusão é $\vee$? Talvez $\vee_e$ ajuda; ou: supõe a negação e tenta PBC.

Gerhard Gentzen



Untersuchungen über das logische Schließen I, Mathematische Zeitschrift, 39(1934), 176–210.

Gerhard Gentzen (*1909, +1945)

Regras derivadas

Modus tollens

O modus tollens é uma regra derivada:

1	$\Phi \to \Psi$	premissa
2	$\neg\Psi$	premissa
3	Φ	hipótese
4	Ψ	$\to_{\mathrm{e}}$ 1,3
5	$\perp$	$\neg_{\mathrm{e}}$ 4,2
6	$\neg\Phi$	$\neg_{\mathrm{i}}$ 3–5

Introdução da negação dupla

Também a introdução da negação dupla pode ser provada:

1	Φ	premissa
2	$\neg\Phi$	hipótese
3	$\perp$	$\neg_{\mathrm{e}}$ 1,2
4	$\neg\neg\Phi$	$\neg_{\mathrm{i}}$

Eliminação da contradição

Até $\perp_{\mathrm{e}}$ podemos provar:

1	$p \land \neg p$	premissa
2	p	$\land_{e_1}$
3	$\neg p$	$\land_{e_2}$
4	$\neg q$	hipótese
5	p	cópia 2
6	$\neg q \to p$	$\to_i$ 4–5
7	$\neg p \to q$	Lema 2.4
8	q	$\to_e$ 3,7

Prova por contradição

O que podemos concluir, se a negação de alguma fórmula implica uma contradição: $\neg\Phi \to \bot$?

1	$\neg\Phi \to \bot$	premissa
2	$\neg\Phi$	hipótese
3	$\bot$	$\to_e$ 1,2
4	$\neg\neg\Phi$	$\neg_i$ 2-3
5	Φ	$\neg\neg_e$

Se uma fórmula negada implica uma contradição, a fórmula tem que ser válido. Isso é um lei importante, que justifica uma regra de "prova por contradição"

$$\dfrac{\boxed{\begin{array}{c}\neg\Phi\\ \vdots\\ \bot\end{array}}}{\Phi}\ \text{PBC}$$

O exemplo também mostra que com uma prova $\bot$ a partir de $\neg\Phi$ podemos concluir $\neg\Phi \to \bot$ (usando $\to_i$). Isso é um fato mais geral, que vamos usar mais adiante: $\Phi \vdash \Psi$ significa que tem prova de Ψ usando Φ, e com isso, $\vdash \Phi \to \Psi$ tem que ser válido também. Em geral, com $\Phi_1, \ldots, \Phi_n \vdash \Psi$ podemos concluir $\vdash \Phi_1 \to (\Phi_2 \to (\cdots(\Phi_n \to \Psi)\cdots))$.

Lei do terceiro excluído

O lei do terceiro excluído ("law of the excluded middle", "tertium non datur") afirma que sempre sabemos, sem premissas, que uma proposição ou fórmula tem que ser correto ou não.

1	$\neg(\Phi \lor \neg\Phi)$	hipótese
2	Φ	hipótese
3	$\Phi \lor \neg\Phi$	$\lor_{i_1}$ 2
4	$\bot$	$\neg_e$ 3,1
5	$\neg\Phi$	$\neg_i$ 2–4
6	$\Phi \lor \neg\Phi$	$\lor_{i_2}$ 5
7	$\bot$	$\neg_e$ 6,1
8	$\neg\neg(\Phi \lor \neg\Phi)$	$\neg_i$ 1–7
9	$\Phi \lor \neg\Phi$	$\neg\neg_e$ 8

$$\frac{}{\Phi \lor \neg\Phi}\ \text{LEM}$$

2.3.1. Exemplos e teoremas importantes

Exemplo 1

$q \rightarrow r \vdash (p \rightarrow q) \rightarrow (p \rightarrow r)$ é válido?

1	$q \rightarrow r$	premissa
2	$p \rightarrow q$	hipótese
3	p	hipótese
4	q	$\rightarrow_e$ 3,2
5	r	$\rightarrow_e$ 4,1
6	$p \rightarrow r$	$\rightarrow_i$ 3–5
7	$(p \rightarrow q) \rightarrow (p \rightarrow r)$	$\rightarrow_i$ 2–7

Exemplo 2

$$\neg(\neg p \lor q) \vdash p \tag{2.5}$$

é valido?

1	$\neg(\neg p \lor q)$	premissa
2	$\neg p$	hipótese
3	$\neg p \lor q$	$\lor_{i_1}$ 2
4	$\bot$	$\neg_e$ 3,1
5	p	PBC 2–4

Exemplo 3

$\vdash p \rightarrow (q \rightarrow p)$ é valido?

1	p	hipótese
2	q	hipótese
3	p	cópia 1
4	$q \to p$	$\to_i$ 2–3
5	$p \to (q \to p)$	$\to_i$ 1–4

Exemplo 4

$$p \to q \dashv\vdash \neg p \vee q?$$

1	$p \to q$	premissa
2	$\neg(\neg p \vee q)$	hipótese
3	p	Lema 2.5
4	q	$\to_e$ 3,1
5	$\neg p \vee q$	$\vee_{i_2}$ 4
6	$\bot$	$\neg_e$ 5,2
7	$\neg p \vee q$	

1	$\neg p \vee q$	premissa
2	p	hipótese
3	$\neg p$	hipótese
4	$\bot$	$\neg_e$ 2,3
5	q	$\bot_e$ 4
6	q	hipótese
7	q	cópia 6
8	q	$\vee_e$ 1,3–5,6–7
9	$p \to q$	$\to_i$ 2–8

Comutação, distribuição e idempotência

- Comutação

$$p \wedge q \dashv\vdash q \wedge p \tag{2.6}$$
$$p \vee q \dashv\vdash q \vee p \tag{2.7}$$

- Contraposição

$$p \to q \dashv\vdash \neg q \to \neg p \tag{2.8}$$

- Distribuição

$$p \wedge (q \vee r) \dashv\vdash (p \wedge q) \vee (p \wedge r) \tag{2.9}$$
$$p \vee (q \wedge r) \dashv\vdash (p \vee q) \wedge (p \vee r) \tag{2.10}$$

- Idempotência

$$p \land p \dashv\vdash p \tag{2.11}$$
$$p \lor p \dashv\vdash p \tag{2.12}$$

Exemplo: Comutação

- $p \land q \dashv\vdash q \land p$: Lema 2.2.
- $p \to q \dashv\vdash \neg q \to \neg p$: Lema 2.4 (contraposição).

1	$p \lor q$	premissa
2	p	hipótese
3	$q \lor p$	$\lor_{i_2}$ 2
4	q	hipótese
5	$q \lor p$	$\lor_{i_1}$ 4
6	$q \lor p$	$\lor_e$ 2–3,4–5

Exemplos: Distribuição

- $p \land (q \lor r) \vdash (p \land q) \lor (p \land r)$: Lema 2.3.
- O contrário

1	$(p \land q) \lor (p \land r)$	premissa
2	$p \land q$	hipótese
3	p	$\land_{e_1}$ 2
4	q	$\land_{e_2}$ 2
5	$q \lor r$	$\lor_{i_1}$ 4
6	$p \land (q \lor r)$	$\land_i$ 3,5
7	$p \land r$	hipótese
8	p	$\land_{e_1}$ 7
9	r	$\land_{e_2}$ 7
10	$q \lor r$	$\lor_{i_2}$
11	$p \land (q \lor r)$	$\land_i$ 8,10
12	$p \land (q \lor r)$	$\lor_e$ 1,2–6,7–11

Exemplos: Idempotência

1	$p \land p$	premissa
2	p	$\land_{e_1}$

1	p	premissa
2	$p \land p$	$\land_i$ 1,1

Absorção, associatividade e de Morgan

- Absorção

$$(p \wedge q) \vee p \dashv\vdash p \tag{2.13}$$
$$(p \vee q) \wedge p \dashv\vdash p \tag{2.14}$$

- Associatividade

$$p \vee (q \vee r) \dashv\vdash (p \vee q) \vee r \tag{2.15}$$
$$p \wedge (q \wedge r) \dashv\vdash (p \wedge q) \wedge r \tag{2.16}$$
$$\tag{2.17}$$

- Leis de De Morgan

$$\neg(p \wedge q) \dashv\vdash \neg p \vee \neg q \tag{2.18}$$
$$\neg(p \vee q) \dashv\vdash \neg p \wedge \neg q \tag{2.19}$$

Exemplo: Absorção

1	$(p \wedge q) \vee p$	premissa
2	$p \wedge q$	hipótese
3	p	$\wedge_{e_1}$ 2
4	p	hipótese
5	p	$\vee_e$ 1,2–3,4

Exemplo: Associatividade

- $p \wedge (q \wedge r) \dashv\vdash (p \wedge q) \wedge r$: prova da equação 2.1 e exercício 2.7.

1	$p \vee (q \vee r)$	premissa
2	p	hipótese
3	$p \vee q$	$\vee_{i_1}$ 2
4	$(p \vee q) \vee r$	$\vee_{i_1}$ 3
5	$q \vee r$	hipótese
6	q	hipótese
7	$p \vee q$	$\vee_{i_2}$ 6
8	$(p \vee q) \vee r$	$\vee_{i_1}$ 7
9	r	hipótese
10	$(p \vee q) \vee r$	$\vee_{i_2}$ 9
11	$(p \vee q) \vee r$	$\vee_e$ 5,6–8,9–10
12	$(p \vee q) \vee r$	$\vee_e$ 1,2–4,5–11

Exemplo: Leis de De Morgan

1	$\neg(p \wedge q)$	premissa
2	$\neg(\neg p \vee \neg p)$	hipótese
3	$\neg p$	hipótese
4	$\neg p \vee \neg q$	$\vee_{\mathrm{i}_1}$ 3
5	$\perp$	$\neg_{\mathrm{e}}$ 4,2
6	p	PBC 3–5
7	$\neg q$	hipótese
8	$\neg p \vee \neg q$	$\vee_{\mathrm{i}_2}$ 7
9	$\perp$	$\neg_{\mathrm{e}}$ 8,2
10	q	PBC 7–9
11	$p \wedge q$	$\wedge_{\mathrm{i}}$ 6,10
12	$\perp$	$\neg_{\mathrm{e}}$ 11,1
13	$\neg p \vee \neg q$	PBC 2–12

Exemplo de uma prova alternativa:

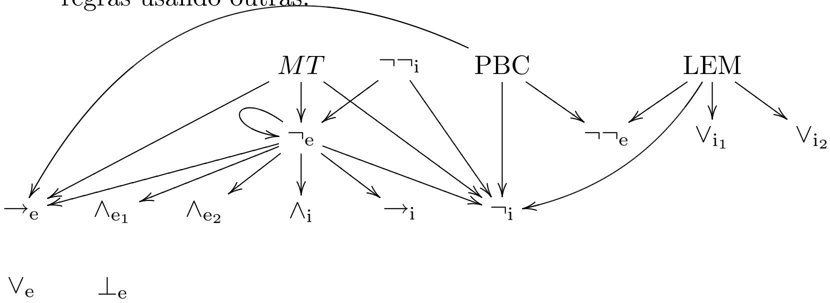
1	$\neg(p \wedge q)$	premissa
2	$p \vee \neg p$	LEM
3	p	hipótese
4	q	hipótese
5	$p \wedge q$	$\wedge_{\mathrm{i}}$ 3,4
6	$\perp$	$\neg_{\mathrm{e}}$ 5,1
7	$\neg q$	PBC 4–6
8	$\neg p \vee \neg q$	$\vee_{\mathrm{i}_2}$ 7
9	$\neg p$	hipótese
10	$\neg p \vee \neg q$	$\vee_{\mathrm{i}_1}$ 9
11	$\neg p \vee \neg q$	$\vee_{\mathrm{e}}$ 2,3–8,9–10

Prova do seqüente inversa:

1	$\neg p \vee \neg q$	premissa
2	$p \wedge q$	hipótese
3	p	$\wedge_{\mathrm{i}}1$ 2
4	q	$\wedge_{\mathrm{i}}2$ 2
5	$\neg p \vee \neg q$	cópia 1
6	$\neg p$	hipótese
7	p	cópia 3
8	$\perp$	$\neg_{\mathrm{e}}$ 7,6
9	$\neg q$	hipótese
10	q	cópia 4
11	$\perp$	$\neg_{\mathrm{e}}$ 10,9
12	$\perp$	$\vee_{\mathrm{e}}$ 5,6–8,9–11
13	$\neg(p \wedge q)$	PBC 2–12

Regras básicas e regras derivadas

- Nosso sistema de regras não é mínimo: Foi possível de deduzir algumas regras usando outras.



Quantas regras são suficientes?

- Talvez mais das nossas regras são regras derivadas? Não é obviou.
- Porém, foram propostas sistemas diferentes com o objetivo de achar um conjunto mínimo da regras.
- Um exemplo é o sistema $\mathcal{H}$ do matemático David Hilbert.
- Também se chama o *cálculo* de Hilbert.

David Hilbert
(*1862, +1943)

2.3.2. Sistemas do tipo Hilbert

A lógica permite varias formalizações, e historicamente foram inventados diferentes sistemas de prova. Sistemas de tipo Hilbert em geral tem como única regra o modus ponens. Eles são sistemas para fórmulas simples e eles variam na escolha de conectivos e axiomas. Exemplos são

1. Sistema $\mathcal{H}_1$ com três axiomas

$$\vdash A \rightarrow (B \rightarrow A) \tag{2.20}$$
$$\vdash (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)) \tag{2.21}$$
$$\vdash (\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B) \tag{2.22}$$

2. Sistema $\mathcal{H}_2$ com um axioma (Meredith):

$$((((A \rightarrow B) \rightarrow (\neg C \rightarrow \neg D)) \rightarrow C) \rightarrow E) \rightarrow ((E \rightarrow A) \rightarrow (D \rightarrow A)) \tag{2.23}$$

Historicamente, esse tipo de sistema foi inventado primeiramente e serviu para a formalização da lógica. A desvantagem dessas formas é que eles são pouco intuitivas: eles não modelam o raciocínio usado na matemática e pelos humanos em geral.

O sistema $\mathcal{H}$

$$\frac{}{A \to (B \to A)}\,\mathrm{ax}_1$$

$$\frac{}{(A \to (B \to C)) \to ((A \to B) \to (A \to C))}\,\mathrm{ax}_2$$

$$\frac{}{(\neg B \to \neg A) \to (A \to B)}\,\mathrm{ax}_3$$

$$\frac{A \qquad A \to B}{B}\,\to_{\mathrm{e}}$$

- Observe: Não tem regras para $\wedge$ e $\vee$!
- O sistema usa $A \wedge B =_{\mathrm{def}} \neg(A \to \neg B)$ e $A \vee B =_{\mathrm{def}} \neg A \to B$ (Exercício: Prove a equivalência!)

Observações sobre $\mathcal{H}$

- Como saber que $\mathcal{H}$ é equivalente à nosso sistema?
- Usando as regras de $\mathcal{H}$, prova as nossas regras.
- Usando as nossas regras, prova as regras de $\mathcal{H}$

Poder de nosso sistema

- Reconhecemos algumas regras: $\to_{\mathrm{e}}$ é o modus ponens,
- ax_1 é um teorema em nosso sistema.
- ax_3 é parecido com a regra 2.8. Usando $\neg\neg_{\mathrm{i}}$ e $\neg\neg_{\mathrm{e}}$ obtemos uma prova.
- E ax_2?

1	$A \to (B \to C)$	hipótese
2	$A \to B$	hipótese
3	A	hipótese
4	B	$\to_e$ 3,2
5	$B \to C$	$\to_e$ 3,1
6	C	$\to_e$ 4,5
7	$A \to C$	$\to_i$ 3–6
8	$(A \to B) \to (A \to C)$	$\to_i$ 2–7
9	$(A \to (B \to C)) \to ((A \to B) \to (A \to C))$	$\to_i$ 1–8

Poder do $\mathcal{H}$

- É possível de obter o contrario também: Usando os axiomas de $\mathcal{H}$ podemos provar as nossas regras.

Exemplos: Transitividade e auto-implicação

Trans:	1	$A \to B$	premissa
	2	$B \to C$	premissa
	3	$(B \to C) \to (A \to (B \to C))$	ax_1
	4	$A \to (B \to C)$	$\to_e$ 2,3
	5	$(A \to (B \to C)) \to ((A \to B) \to (A \to C))$	ax_2
	6	$(A \to B) \to (A \to C)$	$\to_e$ 4,5
	7	$A \to C$	$\to_e$ 1,6

Auto-implicação:	1	$(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$	ax_2
	2	$A \to ((A \to A) \to A)$	ax_1
	3	$(A \to (A \to A)) \to (A \to A)$	$\to_e$ 2
	4	$A \to (A \to A)$	ax_1
	5	$A \to A$	$\to_e$ 4

Exemplo: A regra $\neg\neg_e$

1	$\neg\neg A \to (\neg\neg\neg\neg A \to \neg\neg A)$	ax_1
2	$(\neg\neg\neg\neg A \to \neg\neg A) \to (\neg A \to \neg\neg\neg A)$	ax_3
3	$(\neg A \to \neg\neg\neg A) \to (\neg\neg A \to A)$	ax_3
4	$\neg\neg A \to (\neg A \to \neg\neg\neg A)$	Trans 1,2
5	$\neg\neg A \to (\neg\neg A \to A)$	Trans 4,3
6	$\neg\neg A \to \neg\neg A$	Auto-implicação
7	$(\neg\neg A \to (\neg\neg A \to A)) \to ((\neg\neg A \to \neg\neg A) \to (\neg\neg A \to A))$	ax_2
8	$(\neg\neg A \to \neg\neg A) \to (\neg\neg A \to A)$	$\to_e$ 5,7
9	$\neg\neg A \to A$	$\to_e$ 6,8

2.3.3. Árvores de refutação

Provas

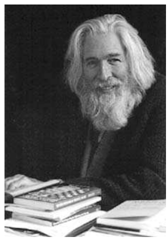
Como provar ou refutar um seqüente? Temos duas possibilidades:

- Na teoria das provas: Busca uma prova. Ou: Busca uma contradição.
  - Vantagem: Fórmulas arbitrárias podem ter uma prova curta.
  - Desvantagem: Não podemos construir provas (curtas) só mecanicamente: as vezes precisamos criatividade.
- Na semântica: Construí uma tabela de verdade.
  - Vantagem: Construção mecânica – a lógica proposicional é *decidível*.
  - Desvantagem: Trabalho exponencial.

Introdução

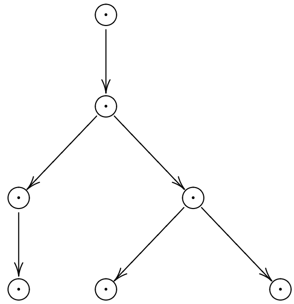
- *Árvores de refutação* ou *tableaux* é um sistema de prova alternativa.
- Idéia: Se $\Phi_1, \ldots, \Phi_n \vdash \Psi$, então $\Phi_1, \ldots, \Phi_n, \neg\Psi \vdash \bot$.
- Logo, vamos procurar sistematicamente para uma contradição.
- Se encontramos uma contradição em todos os casos (todos os casos são *inconsistentes*), o argumento é válido.
- Se encontramos um caso que é consistente (sem contradição) o argumento não pode ser válido.

Evert Willem Beth (*1908,+1964)



Raymond Merrill Smullyan (*1919)

Árvores



- Uma árvore consiste em nós (internos ou folhas), arestas, e ramos.
- Em árvores de refutação usamos os nós para fórmulas.

O algoritmo

Para testar um seqüente, procedemos assim:

T1. Init Construí uma árvore inicial, que consiste em um ramo só. Cada premissa e a negação da conclusão é um nó.

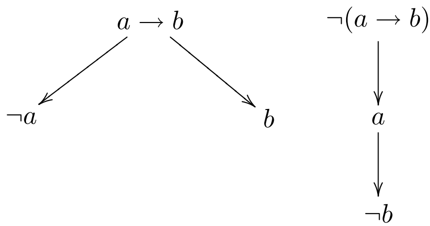
T2. Expansão Enquanto existe uma fórmula, que não foi expandida seguinda as regras, expande ela (e marca ela “expandida”).

T3. Inválido? Se um ou mais ramos são consistentes: Imprime “O argumento não é válido” e para.

T4. Válido? (Aqui, todos ramos são inconsistentes) Imprime “O argumento é válido” e para.

Exemplo: Regras para a implicação

Por exemplo, a implicação tem os seguintes regras:



- A regra na esquerda diz: Se temos um nó com fórmula $a \to b$, expande cada ramo em baixo na folha com dois ramos; um com uma folha para $\neg a$ e um com uma folha para b.
- A regra na direita diz: Se temos um nó com fórmula $\neg(a \to b)$, expande cada ramo em baixo dessa fórmula com dois nós a e $\neg b$.

Implicação: Exemplo

$$a \to b, b \to c \vdash a \to c?$$

$$\begin{array}{c} a \to b \\ b \to c \\ \neg(a \to c) \\ a \\ \neg c \end{array}$$

$$\begin{array}{cccc} \neg a & & b & \\ \neg b \quad c & & \neg b & c \\ \times \quad \times & & \times & \times \end{array}$$

Sim!

Implicação: Modus ponens

$$p \to q, p \vdash q?$$

$$\begin{array}{c} p \to q \\ p \\ \neg q \end{array}$$

$$\begin{array}{cc} \neg p & q \\ \times & \times \end{array}$$

Sim!

Implicação: Modus ponens

Funciona sem premissas também?

$$\vdash p \to p?$$

$$\begin{array}{c} \neg(p \to p) \\ p \\ \neg p \\ \times \end{array}$$

Sim!

Observações

- As regras para construir árvores de refutação aplicam-se somente a fórmulas inteiras, e não a subfórmulas.
- O resultado não depende da ordem de aplicação das regras.
- Intuitivamente, se aplicamos uma regra a uma fórmula, se essa fórmula é verdadeira em uma interpretação, ao menos um ramo contém uma fórmula verdadeira.

Também temos as seguintes técnicas práticas:

- Se nos encontramos a e $\neg a$ em um ramo, podemos fechar o ramo imediatamente.
- Uma refutação se torna mais eficiente se aplicarmos primeiramente as regras que não levam a uma bifurcação.

Noções

- Um ramo é *fechado*, se tem fórmulas a e $\neg a$ em dois nós. Marcamos um ramo fechado com x em baixo dele.
- Senão o ramo é *em aberto*. Marcamos um ramo em aberto com $\odot$.
- Um tableau é *completa*, se todas regras que se podem aplicar são aplicados.
- Um tableau é *fechado*, se todos seus ramos são fechados.

Assim, nosso algoritmo lê-se:

T1. Init Construí uma árvore inicial, que consiste em um ramo só. Cada premissa e a negação da conclusão é um nó.

T2. Expansão Enquanto a tableau não é *completa*, escolhe uma fórmula e aplica a regra correspondente (e marca ela "expandida").

T3. Inválido? Se o tableau não é fechado: Imprime "O argumento não é válido" e para.

T4. Válido? (O tableau é fechado) Imprime "O argumento é válido" e para.

Regras

Regra para a negação

$$\begin{tikzcd} \neg\neg a \arrow[d] \\ a \end{tikzcd}$$

Exemplo: Negação

$$p \to q \vdash \neg q \to \neg p?$$

$$\begin{array}{ccc} & p \to q & \\ & \neg q \to \neg p & \\ & \neg q & \\ & \neg\neg p & \\ & p & \\ \neg p & & \neg q \\ \times & & \times \end{array}$$

Sim!

Regras para a conjunção

$$\begin{tikzcd} & \neg(a \wedge b) \arrow[dl] \arrow[dr] & \\ \neg a & & \neg b \end{tikzcd} \qquad \begin{tikzcd} a \wedge b \arrow[d] \\ a \arrow[d] \\ b \end{tikzcd}$$

Exemplo: Conjunção

$$p \wedge \neg p \vdash q?$$

$$\begin{array}{c} p \wedge \neg p \\ \neg q \\ p \\ \neg p \\ \times \end{array}$$

Sim!

Regras para a disjunção

$$
\begin{tikzcd}
 & a \vee b \arrow[dl] \arrow[dr] & \\
a & & b
\end{tikzcd}
\qquad
\begin{tikzcd}
\neg(a \vee b) \arrow[d] \\
\neg a \arrow[d] \\
\neg b
\end{tikzcd}
$$

Exemplo: Lei de De Morgan

$$\neg(a \wedge b) \vdash \neg a \vee \neg b?$$

$$
\begin{tikzcd}[row sep=small]
 & \neg(a \wedge b) & \\
 & \neg(\neg a \vee \neg b) & \\
 & \neg\neg a & \\
 & \neg\neg b & \\
 & a & \\
 & b \arrow[dl] \arrow[dr] & \\
\neg a & & \neg b \\
\times & & \times
\end{tikzcd}
$$

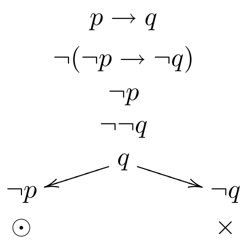
Sim!

Achar contra-exemplos

- Os ramos abertos de uma árvore completa mas não fechada ("não válida") contém contra-exemplos.
- Cada ramo em aberto corresponde a um contra-exemplo.
- Para construir um contra-exemplo usando um ramo aberto
  1. Se ele contém um literal p, define p verdadeiro.
  2. Se ele contém um literal $\neg p$, define p falso.
  3. Se, para alguma proposição p, ele contém nem p nem $\neg p$, define p arbitrário.

Exemplo 1

$$p \rightarrow q \vdash \neg p \rightarrow \neg q?$$

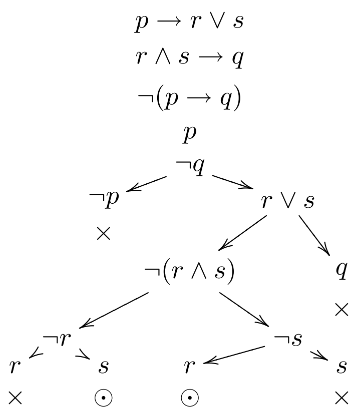


Não:

p	q	$p \rightarrow q$	$\neg p \rightarrow \neg q$
f	v	v	f

Exemplo 2

$p \rightarrow r \vee s, r \wedge s \rightarrow q \vdash p \rightarrow q?$



Não:

p	q	r	s	$p \rightarrow r \vee s$	$r \wedge s \rightarrow q$,	$p \rightarrow q$
v	f	f	v	v	v	f
v	f	v	f	v	v	f

Consistência e completude?

O método de árvores de refutação é

- consistente: se um seqüente foi refutado (o tableau fecha), ele é valido semanticamente.

- completo: se um seqüente é valido, tem refutação.

Veja capítulo 2.5.

Exemplo 1

$$p \to q \vdash \neg(p \land q)?$$

1	$p \to q$		premissa
2	$\neg\neg(p \land q)$		negação da conclusão
3	$p \land q$		$\neg\neg 2$
4	p		$\land 3$
5	q		$\land 3$
6	$\neg p$	q	$\to 1$
7	$\times$	$\odot$	

O seqüente não é válido. Um contra-exemplo é $p = q = v$.

Exemplo 2

$$p \lor (q \land r) \vdash (p \lor q) \land (p \lor r)?$$

1	$p \lor (q \land r)$			
2	$\neg((p \lor q) \land (p \lor r))$			
3	p		$q \land r$	
4	$\neg(p \lor q)$	$\neg(p \lor r)$	q	
5	$\neg p$	$\neg p$	r	
6	$\neg q$	$\neg r$	$\neg(p \lor q)$	$\neg(p \lor r)$
7	$\times$	$\times$	$\neg p$	$\neg p$
8			$\neg q$	$\neg r$
			$\times$	$\times$

O seqüente é válido.

Exemplo 3

$$q \to r \vdash (p \to q) \to (p \to r)?$$

1		$q \to r$			premissa
2		$\neg((p \to q) \to (p \to r))$			negação da conclusão
3		$p \to q$			$\neg\to$ 2
4		$\neg(p \to r)$			$\neg\to$ 2
5		p			$\neg\to$ 4
6		$\neg r$			$\neg\to$ 4
7	$\neg p$		q		$\to$ 3
8	$\times$				
9		$\neg q$		r	$\to$ 1
10		$\times$		$\times$	

A seqüente é válido.

2.4. Teoria de modelos

Introdução

- Na primeira parte vimos regras de dedução para provar seqüentes $p_1, p_2, \ldots, p_n \vdash c$.
- As regras funcionam sintaticamente só, sem "saber" o que os símbolos significam (mas as provas precisam de criatividade).
- Intuitivamente, as fórmulas tem uma interpretação com de *valores de verdade*
  - Uma proposição atômica pode ser verdadeira (v) ou falsa (f).
  - Podemos definir a verdade (ou falsidade) de uma fórmula usando os valores de verdade das proposições atômicas.
- A *relação de conseqüência semântica*

  $$p_1, p_2, \ldots, p_n \models c$$

  afirma que a premissas justificam uma conclusão baseado nesses valores de verdade (c é uma conseqüência de $p_1, \ldots, p_n$).

Semântica da conjunção

- Com proposições p e q que podem ser verdadeira ou falso, o que significa $p \land q$?

- A intuição é que $p \land q$ é verdadeira se ambos, p e q são verdadeiros e falso senão.

- Assim, definimos o seguinte *tabela de verdade* para $\land$

Φ	Ψ	$\Phi \land \Psi$
f	f	f
f	v	f
v	f	f
v	v	v

- Cada combinação de valores de verdade para as proposições se chama uma *atribuição* ou *valoração*.

- Quantas atribuições tem com n proposições?

Com três proposições temos $2^3 = 8$ atribuições possíveis. Em geral, n proposições permitem 2^n atribuições, porque para cada proposição podemos escolher verdadeira ou falso independentemente.

Exemplo: Fórmula composta

p	q	r	$q \land r$	$p \land (q \land r)$
f	f	f	f	f
f	f	v	f	f
f	v	f	f	f
f	v	v	v	f
v	f	f	f	f
v	f	v	f	f
v	v	f	f	f
v	v	v	v	v

Exemplo: Notação alternativa

Um jeito mais compacto de escrever a mesma coisa (Quine):

p	$\wedge$	$(q$	$\wedge$	$r)$
f	f	f	f	f
f	f	f	f	v
f	f	v	f	f
f	f	v	v	v
v	f	f	f	f
v	f	f	f	v
v	f	v	f	f
v	v	v	v	v

Semântica da disjunção

- Uma disjunção de p e q é verdadeira, se p ou q (ou ambas) são verdadeiras.
- A tabela de verdade correspondente é

Φ	Ψ	$\Phi \vee \Psi$
f	f	f
f	v	v
v	f	v
v	v	v

Exemplo

$p \wedge (q \vee r) \models (p \wedge q) \vee (p \wedge r)$?

p	q	r	$q \vee r$	$p \wedge q$	$p \wedge r$	$p \wedge (q \vee r)$	$(p \wedge q) \vee (p \wedge r)$
f	f	f	f	f	f	f	f
f	f	v	v	f	f	f	f
f	v	f	v	f	f	f	f
f	v	v	v	f	f	f	f
v	f	f	f	f	f	f	f
v	f	v	v	f	v	v	v
v	v	f	v	v	f	v	v
v	v	v	v	v	v	v	v

Semântica da implicação

- Suponhamos $p \rightarrow q$.
- Se p é verdadeira, q tem que ser verdadeira também para $p \rightarrow q$ ser verdadeira; senão $p \rightarrow q$ tem que ser falso.

- O que podemos dizer quando p é falso?

Φ	Ψ	$\Phi \to \Psi$
f	f	v
f	v	v
v	f	f
v	v	v

Semântica da negação e das constantes

Φ	$\neg\Phi$
f	v
v	f

$\bot$
f

$\top$
v

Exemplo 1

$(p \to \neg q) \to (q \lor \neg p)$.

p	q	$\neg p$	$\neg q$	$p \to \neg q$	$q \lor \neg p$	$(p \to \neg q) \to (q \lor \neg p)$
f	f	v	v	v	v	v
f	v	v	f	v	v	v
v	f	f	v	v	f	f
v	v	f	f	f	v	v

Exemplo 2

Teorema conhecido: $\neg(p \land q) \dashv\vdash \neg p \lor \neg q$

p	q	$\neg p$	$\neg q$	$p \land q$	$\neg(p \land q)$	$\neg p \lor \neg q$
f	f	v	v	f	v	v
f	v	v	f	f	v	v
v	f	f	v	f	v	v
v	v	f	f	v	f	f

Observe as últimas duas colunas!

O exemplo acima mostra, que fórmulas equivalentes tem o mesmo valor de verdade para todas atribuições. Por isso, as últimas duas colunas são idênticas.

Exemplo 3
Seqüente conhecido: $p \wedge p \to q \vdash q$

p	q	$p \to q$	$p \wedge p \to q$
f	f	v	f
f	v	v	f
v	f	f	f
v	v	v	v

Esse exemplo mostra, o que acontece, se não temos uma equivalência. q é uma conseqüência de $p \wedge p \to q$, mas o contrário não é verdadeiro. Por isso, para cada atribuição tal que $p \wedge p \to q$ é verdadeiro, q é verdadeiro também. Mas nem sempre quando q é verdadeiro, $p \wedge p \to q$ tem que ser verdadeiro também.

Atribuição e interpretação

- Temos valores de verdade $\mathbb{B} = \{v, f\}$.
- Uma *atribuição* mapa átomos para valores de verdade

$$A : \text{Atom} \to \mathbb{B}$$

- Uma atribuição pode ser estendida para uma (única) interpretação

$$[\![\cdot]\!]_A : \mathcal{L} \to \mathbb{B}.$$

Definição da interpretação
Para fórmulas $\Phi, \Psi \in \mathcal{L}$ e proposição $p \in$ Atom arbitrários

$$\begin{aligned}
[\![p]\!]_A &= A(p) \\
[\![\top]\!]_A &= v \\
[\![\bot]\!]_A &= f \\
[\![\neg\Phi]\!]_A &= \neg[\![\Phi]\!]_A \\
[\![\Phi \wedge \Psi]\!]_A &= [\![\Phi]\!]_A \wedge [\![\Psi]\!]_A \\
[\![\Phi \vee \Psi]\!]_A &= [\![\Phi]\!]_A \vee [\![\Psi]\!]_A \\
[\![\Phi \to \Psi]\!]_A &= [\![\Phi]\!]_A \to [\![\Psi]\!]_A
\end{aligned}$$

Observe que os conectivos na definição do $[\![\cdot]\!]_A$ tem dois significados diferentes. No lado esquerdo, eles ocorrem como símbolos em fórmulas da lógica de predicados. No lado direito eles denotam funções sobre valores de verdade que foram definidos com tabelas de verdade. Isso é um tipo de sobrecarregamento (inglês: overloading), que não causa problemas, porque geralmente é claro do contexto se um conectivo denota um símbolo ou uma função. Em caso de dúvidas podemos diferenciar com nomes diferentes, por exemplo escrevendo $\wedge$ para o símbolo, e $\dot{\wedge}$ para a função.

Exemplo 4
Com A tal que $A(p) = v$ e $A(q) = f$ temos

$$\llbracket p \wedge q \rrbracket_A = f$$
$$\llbracket \neg p \vee (p \to q) \rrbracket_A = f$$

$\Diamond$

Uma outra notação comum é $A \models \Phi$ (lê: a atribuição A é um modelo de Φ). Ela se chama *relação de satisfação* é significa que dado a atribuição A a fórmula Φ é verdadeiro.

Definição 2.2 (Relação de satisfação)

$$A \models \Phi \iff \llbracket \Phi \rrbracket_A = v$$

A relação de conseqüência semântica

- Os exemplos sugerem a seguinte definição de $\models$
- Se, para cada atribuição A, tal que $\Phi_1, \dots, \Phi_n$ são verdadeiras ($\llbracket \Phi_i \rrbracket_A = v$), Ψ também é verdadeiro ($\llbracket \Psi \rrbracket_A = v$), escrevemos

  $$\Phi_1, \dots, \Phi_n \models \Psi$$

  (lê: $\Phi_1, \dots, \Phi_n$ modelam Ψ)
- $\models$ se chama relação de conseqüência semântica.
- Uma fórmula é verdadeira em todas interpretações é uma *tautologia* (escreve: $\models \Phi$).
- Se temos $\Phi \models \Psi$ e $\Psi \models \Phi$, as fórmulas são *semanticamente equivalentes* (escreve: $\Phi \equiv \Psi$).

Revisão da terminologia (Referência)

Símbolo $\square$	Dedução natural $\vdash$	Semântica $\models$
$\square$	Relação de dedutibilidade	Relação de conseqüência (semântica)
	Seqüente	
$p_1, \ldots, p_n \square q$	p_i portanto q	p_i modelam q
	é valido	é correto
$\square q$	q é um teorema	q é uma tautologia

Operadores binárias

Em geral, uma tabela de verdade para *operadores binárias* (com dois argumentos) é determinada com quatro valores de verdade:

p	q	$p \circ q$
f	f	a
f	v	b
v	f	c
v	v	d

Quantos combinações de a,b,c e d tem? Ou, equivalente, quantos operadores binárias são possíveis?

Operadores binárias (Referência)

Tabela abcd	Notação	Símbolo	Nomes
$ffff$	f		Contradição, falsidade, constante f
$fffv$	$pq, p \wedge q, p\&q$	$\wedge$	conjunção, e
$ffvf$	$p \wedge \bar{q}, p \not\supset q, [p > q], p \dot{-} q$		Não-implicação, diferença, mas não
$ffvv$	p		Projeção à esquerda
$fvff$	$\bar{p} \wedge q, p \not\subset q$		Não-implicação inversa, não... mas
$fvfv$	q		Projeção à direita
$fvvf$	$p \oplus q, p \not\equiv q, p\hat{\ }q$	$\oplus$	Disjunção exclusiva, Não-equivalência, “xor”
$fvvv$	$p \vee q, p \mid q$	$\vee$	Disjunção, ou, e/ou
$vfff$	$\bar{p} \wedge \bar{q}, p \bar{\vee} q, p\bar{\vee}q, p \downarrow q$	$\bar{\vee}$	Não-disjunção, joint denial, nem ... nem
$vffv$	$p \equiv q, p \leftrightarrow q, p \Leftrightarrow q$	$\leftrightarrow$	Equivalência, se e somente se
$vfvf$	$\bar{q}, \neg q, !q, \sim q$		Complementação à direita
$vfvv$	$p \vee \bar{q}, p \subset q, p \Leftarrow q, [p \geq q], p^q$	$\leftarrow$	Implicação inversa, se
$vvff$	$\bar{p}, \neg p, !p, \sim p$		Complementação à esquerda
$vvfv$	$\bar{p} \vee q, p \supset q, p \Rightarrow q, [p \leq q], q^p$	$\rightarrow$	Implicação, somente se, se ... então
$vvvf$	$\bar{p} \vee \bar{q}, p \bar{\wedge} q, p\bar{\wedge}q, p \mid q$	$\bar{\wedge}$	Não-conjunção, não ... e, “nand”
$vvvv$	v		Afirmação, validade, tautologia, constante v

Fonte: [6]

Sobre dois proposições p e q com dois valores de verdade $2^{2\times 2} = 16$ conectivos são possíveis. Os seis operadores $ffff, ffvv, fvfv, vfvf, vvff, vvvv$ são de pouco interesse: eles dependem só de um ou nenhum argumento. Não todos os outros dez operadores são independentes, por exemplo os conjuntos $\{\wedge, \vee, \neg\}$ ou $\{\bar{\vee}\}$ são suficientes para definir as outras operações.

Exemplos

- $\bar{\wedge}$ e $\bar{\vee}$ são importantes, porque uma única operação é suficiente para definir as outras.

- Por exemplo $\neg p = p \bar{\wedge} p$, $p \wedge q = (p \bar{\wedge} q) \bar{\wedge} (p \bar{\wedge} q)$, $p \vee q = (p \bar{\wedge} p) \wedge (q \bar{\wedge} q)$
- $\oplus$ ("ou exclusivo, xor") tem varias aplicações:
  - "Brincadeiras" conhecidas são a troca de duas variáveis ($x := x \oplus y; y := y \oplus x; x := x \oplus y$) ou criptografia ingênua (com chave c: $x_i := x_i \oplus c$).
  - Outras aplicações usam que $x \oplus y = (x + y) \mod 2$.

Discussão
Tabeles de verdade

- Com tabelas de verdade é mais fácil de analisar a relação de conseqüência (semântica) porque a avaliação funciona mecanicamente.
- Podemos escrever um programa para avaliar seqüentes.
- Do outro lado, o tamanho de trabalho é exponencial no número das proposições: Testando 10^9 proposições por segundo, a análise de uma fórmula com 60 proposições já demora mais que 30 anos...
- Em comparação usando dedução natural, provas com 60 proposições são possíveis.
- Mas ainda não é claro: $\vdash$ e $\models$ são a mesma coisa?

2.5. Adequação e decibilidade

Temos duas noções de conseqüência: a dedutibilidade $\vdash$ da teoria de provas e a conseqüência semântica $\models$. Uma característica desejável é que essas noções fornecem os mesmos resultados: digamos que um sistema de prova tem que ser *adequado* para uma lógica. A adequação é a combinação de duas características. (i) Uma conseqüência semântica deve ter uma prova e (ii) se temos uma prova, o que foi provado deve ser semanticamente correto. A primeira característica se chama *completude*, a segunda *consistência*. Podemos definir-los como

Definição 2.3
Um sistema de provas que define uma relação de dedutibilidade $\vdash$ é chamado

$$\begin{array}{lc} \text{completo, sse} & \models A \Rightarrow \vdash A \\ \text{consistente, sse} & \vdash A \Rightarrow \models A \\ \text{adequado, sse} & \vdash A \iff \models A \end{array}$$

(Observe que "sse" significa "se e somente se".)
É importante de enfatizar a diferença entre as duas noções. A semântica formalize nossa intuição da verdade. O que chamamos uma conseqüência semântica é a observação que, sempre quando as premissas são verdadeiras, a conclusão também é. Do outro lado, um sistema de prova tenta de capturar essa intuição em um conjunto de regras sintáticas. *A priori* não é obviou que é possível de achar tal sistema. Nesse capítulo vamos ver que a dedução natural de fato é um sistema de prova adequado para a lógica proposicional.

Relação entre as relações

- Duas características importantes da lógica proposicional estão em aberto:
- Uma prova corresponde com a semântica? A dedução natural é *consistente* se
$$\Phi_1, \ldots, \Phi_n \vdash \Psi \Rightarrow \Phi_1, \ldots, \Phi_n \models \Psi$$
- Cada conseqüência semântica correta tem uma prova? A dedução natural é *completa* se
$$\Phi_1, \ldots, \Phi_n \models \Psi \Rightarrow \Phi_1, \ldots, \Phi_n \vdash \Psi$$

Duas características

Primeiro, vamos provar dois teoremas que são importantes para estabelecer a consistência e a completude da lógica proposicional mais adiante.

Teorema de dedução

Teorema 1 (Teorema de dedução, Herbrand)
Para premissas $p_1, \ldots, p_n \in \mathcal{L}$ e conclusão $c \in \mathcal{L}$

$$(p_1, p_2, \ldots, p_n \vdash c) \Rightarrow (\vdash p_1 \rightarrow (p_2 \rightarrow \cdots (p_n \rightarrow c) \cdots)) \tag{2.24}$$

Prova. Seja

$$P(n) = ((p_1, p_2, \ldots, p_n \vdash c) \Rightarrow (\vdash p_1 \rightarrow (p_2 \rightarrow \cdots (p_n \rightarrow c) \cdots))).$$

Base: $P(0)$ significa $\vdash c \Rightarrow \vdash c$.
Passo: Suponha $P(n)$. Suponha mais que $(p_1, p_2, \ldots, p_{n+1} \vdash c)$. Logo, temos uma prova com premissas $p_1, \ldots, p_{n+1}$ e com conclusão c. Com isso, podemos construir a seguinte prova:

1	p_1	premissa
.	$\cdots$	premissas
n	p_n	premissa
n+1	p_{n+1}	hipótese
.	$\cdots$	(cópia da prova)
m-1	c	
m	$p_{n+1} \to c$	

Logo, $p_1, \ldots, p_n \vdash p_{n+1} \to c$, e usando a hipótese da indução, obtemos $P(n+1)$. ■

(Veja também exercício 2.13.)

Observe que nessa prova usamos a implicação $\to$ da lógica proposicional e também uma implicação $\Rightarrow$ na meta-linguagem (a linguagem em que nos estamos discutindo a lógica proposicional).

O segundo teorema é o equivalente semântico do teorema 1.

Relação da conseqüência lógica e implicação

Teorema 2

Para premissas $p_1, \ldots, p_n \in \mathcal{L}$ e conclusão $c \in \mathcal{L}$

$$(p_1, p_2, \ldots, p_n \models c) \Rightarrow (\models p_1 \to (p_2 \to \cdots (p_n \to c) \cdots)) \tag{2.25}$$

Prova. Queremos provar a propriedade

$$P(n) = ((p_1, p_2, \ldots, p_n \models c) \Rightarrow (\models p_1 \to (p_2 \to \cdots (p_n \to c) \cdots)))$$

com indução.

Base: $P(0)$ significa $\models c \Rightarrow \models c$, que obviamente é verdadeira.

Passo: Temos que provar $P(n) \Rightarrow P(n+1)$. Suponha $P(n)$. Para provar $P(n+1)$ suponha mais

$$(*)\; p_1, \ldots, p_{n+1} \models c.$$

Então, $p_1, \ldots, p_n \models (p_{n+1} \to c)$ é correto, por análise de casos: Sejam $p_1, \ldots p_n$ verdadeiras em alguma interpretação. Caso (i): Se p_{n+1} não é verdadeira nessa interpretação, a implicação é verdadeira. Caso (ii) Se p_{n+1} é verdadeira nessa interpretação, usando $(*)$, c é verdadeira e logo $p_{n+1} \to c$ também. Por isso, aplicando a hipótese da indução para $p_1, \ldots, p_n \models (p_{n+1} \to c)$ obtemos $\models p_1 \to (p_2 \to \cdots (p_n \to c) \cdots)$. ■

2.5.1. Consistência

Teorema da consistência

Para premissas $\Phi_1, \ldots, \Phi_n \in \mathcal{L}$ e conclusão $\Psi \in \mathcal{L}$: Se $\Phi_1, \ldots, \Phi_n \vdash \Psi$ é válido,

Rascunho da prova:

- Vamos usar indução completa sobre o comprimento da prova.
- Analisando um prova do tamanho n, sabemos que todas as provas de tamanho menos que n produzem seqüentes que também são semanticamente corretos.
- Por isso, nos vamos analisar caso a caso a última regra aplicada na prova e argumentar que, como as premissas da regra são conseqüências semânticas, a sua conclusão também é.
- Em seguido, vamos formalizar essa idéia e mostrar três casos.

Prova (1)
A propriedade $P(k)$ que queremos provar é

> Para cada seqüente $\Phi_1, \Phi_2, \ldots, \Phi_n \vdash \Psi$ com uma prova de k linhas, $\Phi_1, \Phi_2, \ldots, \Phi_n \models \Psi$ é correto.

A prova é com indução sobre o comprimento $k \geq 1$ de uma prova.
Base: Temos uma prova com comprimento $k = 1$. A única prova possível é

1	Φ	premissa

porque todas regras não-derivadas tem premissas, e o seqüente em consideração tem que ser $\Phi \vdash \Phi$. Neste caso temos também $\Phi \models \Phi$.

Prova (2)
Suponha $P(i)$ para $i < k$: queremos provar $P(k)$. Suponha, que temos uma prova de $\Phi_1, \ldots, \Phi_n \vdash \Psi$ com k linhas da seguinte forma

1	Φ_1	premissa
2	Φ_2	premissa
	$\ldots$	
n	Φ_n	premissa
	$\ldots$	
k	Ψ	(justificação)

Qual é a última regra aplicada? Nossa análise vai considerar todos casos.

Prova: O caso $\wedge_i$
Suponha a última regra foi

$$\text{k} \qquad \Psi_1 \wedge \Psi_2 \quad \wedge_i \; k_1, k_2$$

- Por definição das regras: $k_1 < k$ e $k_2 < k$. Logo temos provas parciais

$$\Phi_1, \ldots, \Phi_n \vdash \Psi_1; \quad \Phi_1, \ldots, \Phi_n \vdash \Psi_2.$$

- Aplica a hipótese da indução

$$\Phi_1, \ldots, \Phi_n \models \Psi_1; \quad \Phi_1, \ldots, \Phi_n \models \Psi_2.$$

- Em outras palavras: Se $\Phi_1, \ldots, \Phi_n$ são verdadeiras, Ψ_1 e Ψ_2 também.
- Pela definição de $\wedge$: $\Psi = \Psi_1 \wedge \Psi_2$ é verdadeira também, i.e.

$$\Phi_1, \ldots, \Phi_2 \models \Psi.$$

Prova: O caso $\rightarrow_i$
Suponha a última regra foi

$$\text{k} \qquad \Psi_1 \rightarrow \Psi_2 \quad \rightarrow_i \; k_1\text{–}k_2$$

- Por definição das regas: $k_1 < k$ e $k_2 < k$.
- A caixa das linhas k_1–k_2 começa com Ψ_1 e termina com Ψ_2. Com Ψ_1 premissa adicional obtemos (em $< k$ linhas)

$$\Phi_1, \ldots, \Phi_n, \Psi_1 \vdash \Psi_2.$$

- Aplica a hipótese da indução

$$\Phi_1, \ldots, \Phi_n, \Psi_1 \models \Psi_2. \qquad (*)$$

- Agora seja A uma atribuição tal que $\Phi_1, \ldots, \Phi_n$ são verdadeiras. Se $[\![\Psi_1]\!]_A = f$: Pela definição da implicação $[\![\Psi_1 \rightarrow \Psi_2]\!]_A = v$. Se $[\![\Psi_1]\!]_A = v$: Com (*) temos $[\![\Psi_2]\!] = v$ e logo $[\![\Psi_1 \rightarrow \Psi_2]\!]_A = v$.
- Isso mostra:

$$\Phi_1, \ldots, \Phi_n \models \Psi_1 \rightarrow \Psi_2.$$

Prova: O caso $\neg_{\mathrm{i}}$
Suponha a última regra foi

$$\mathrm{k} \qquad \neg\Psi_1 \quad \neg_{\mathrm{i}}\ k_1\text{–}k_2$$

- Por definição das regras $k_1 < k$ e $k_2 < k$.
- A caixa das linha k_1 até k_2 começa com Ψ_1 e termina com $\perp$. Como no caso do $\rightarrow_{\mathrm{i}}$ temos (em $< k$ linhas)

$$\Phi_1, \ldots, \Phi_n, \Psi_1 \vdash \perp.$$

- Aplica a hipótese da indução

$$\Phi_1, \ldots, \Phi_n, \Psi_1 \models \perp. \qquad (*)$$

- Seja A uma atribuição tal que $\Phi_1, \ldots, \Phi_n$ são verdadeiras.
- Ψ_1 não pode ser verdadeiro: isso contradiria (*): $\perp$ teria que ser verdadeiro.
- Logo $\neg\Psi_1$ é verdadeira:

$$\Phi_1, \ldots, \Phi_n \models \neg\Psi_1.$$

Os casos restantes

- Nossa sistema tem 12 regras (não-derivadas).
- Provamos a consistência para 3 deles.
- Os 9 casos restantes permitem uma prova semelhante (crê ou tenta: exercício!).

Discussão

- Concluindo a prova, obtemos a primeira característica importante: A dedução natural não prova seqüentes, que não são corretos na semântica: *ela é consistente.*
- A consistência fornece uma técnica para provar que uma prova na dedução natural não existe:
  - Na semântica, busca uma atribuição tal que as premissas são verdadeiras, mas a conclusão é falso (um contra-exemplo).

  - Logo, não existe uma prova do seqüente correspondente, porque ela implica a conseqüência semântica.

- A dedução natural fica consistente com menos regras. Até podemos retirar todas as regras (quais seqüentes podemos provar sem regras?). Logo a consistência vale pouco sem completude!

2.5.2. Completude

Teorema da completude
Sejam $\Phi_1, \ldots, \Phi_n \in \mathcal{L}$ e $\Psi \in \mathcal{L}$:

$$\text{Se } \Phi_1, \ldots, \Phi_n \models \Psi \text{ é correto, } \Phi_1, \ldots, \Phi_n \vdash \Psi \text{ é válido.}$$

Rascunho da prova: Dado uma conseqüência semântica, temos que mostrar que ela é uma conseqüência lógica também (i.e. que existe uma prova).

1. Converta $\Phi_1, \ldots, \Phi_n \models \Psi$ para tautologia $\models \eta$ com $\eta = \Phi_1 \rightarrow (\Phi_2 \rightarrow (\cdots(\Phi_n \rightarrow \Psi)\cdots))$.
2. Sejam $p_1, \ldots, p_m$ as proposições atômicas dessa tautologia. Obtemos a seguinte tabela de verdade

p_1	p_2	$\cdots$	p_m	η
f	f	$\cdots$	f	v
f	f	$\cdots$	v	v
		$\cdots$		
v	v	$\cdots$	v	v

3. Codifique cada linha da tabela usando uma prova $\hat{p}_1, \ldots, \hat{p}_m \vdash \eta$ com $\hat{p}_i = p_i$ se p_i é verdadeiro, e $\hat{p}_i = \neg p_i$ senão.
4. Usando as provas da cada linha, construí uma prova para η.
5. Converta $\vdash \eta$ para $\Phi_1, \ldots, \Phi_n \vdash \Psi$.

Exemplo

- $\neg(p \wedge q) \models \neg p \vee \neg q$
- Com $\eta = \neg(p \wedge q) \rightarrow (\neg p \vee \neg q)$ temos $\models \eta$.
- Construí a tabela de verdade

p	q	η
f	f	v
f	v	v
v	f	v
v	v	v

- Codifique cada linha

$$\neg p, \neg q \vdash \eta \tag{2.26}$$
$$\neg p, q \vdash \eta \tag{2.27}$$
$$p, \neg q \vdash \eta \tag{2.28}$$
$$p, q \vdash \eta \tag{2.29}$$

Exemplo...

- Junta as provas

1	$p \vee \neg p$	LEM
2	p	hipótese
3	$q \vee \neg q$	LEM
4	q	hipótese
5	$\cdots$	(prova 4)
6	η	
7	$\neg q$	hipótese
8	$\cdots$	(prova 3)
9	η	
10	η	$\vee_e$ 3,4–6,7–9
11	$\neg p$	hipótese
12	$q \vee \neg q$	LEM
13	q	hipótese
14	$\cdots$	(prova 2)
15	η	
16	$\neg q$	hipótese
17	$\cdots$	(prova 1)
18	η	
19	η	$\vee_e$ 3,4–5,6–7
20	η	$\vee_e$ 1,2-10,11-19

- Finalmente converta $\vdash \eta$ para $\Phi_1, \ldots, \Phi_n \vdash \Psi$.

Como chegar numa prova formal?

- O primeiro passo já provamos.
- A conversão final também pode ser provada com indução natural.
- O núcleo é provar que uma fórmula Φ com proposições $p_1, \ldots, p_m$.

$$\hat{p}_1, \ldots, \hat{p}_m \vdash \Phi \qquad \text{se o valor correspondente é verdadeiro}$$
$$\hat{p}_1, \ldots, \hat{p}_m \vdash \neg\Phi \qquad \text{se o valor correspondente é falso}$$

- Isso pode ser provado com indução sobre o tamanho da árvore de parse de Φ.

Conclusão

- A lógica proposicional é consistente e completa.
- Podemos escolher entre a dedução natural e tabelas de verdade para provar uma relação de conseqüência.

2.5.3. Decibilidade

Noções

Para uma fórmula $\phi \in \mathcal{L}$, as seguintes perguntas ocorrem freqüentemente:

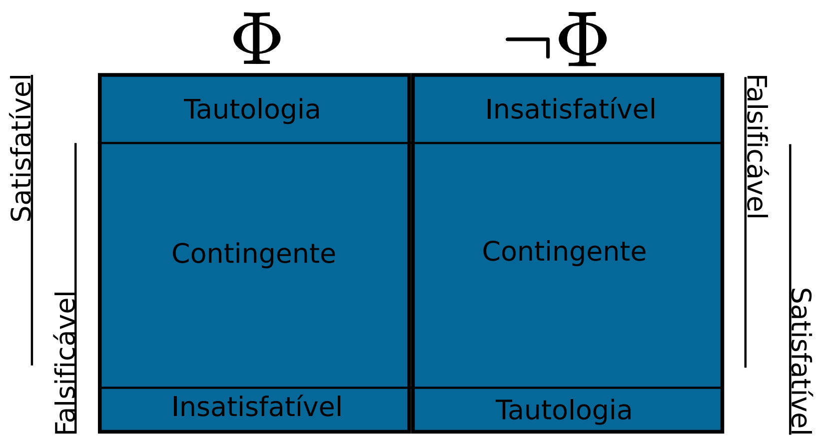
ϕ é	se
satisfatível	existe uma atribuição A, tal que $[\![\phi]\!]_A = v$ (A é um *modelo* de ϕ)
válida	para todas atribuições A, $[\![\phi]\!]_A = v$
falsificável	existe uma atribuição A, tal que $[\![\phi]\!]_A = f$
insatisfatível	para todas atribuições $[\![\phi]\!]_A = f$
contingente	ela é satisfatível e falsificável

Temos os seguintes relações entre essas noções:

- Uma fórmula válida ou contingente é satisfátivel.
- Uma fórmula insatisfatível ou contingente é falsificável.
- Uma fórmula ϕ é insatisfatível, se e somente se a sua negação é válida.

Também observe que introduzimos outro uso da noção *válido*. Na dedução natural um seqüente é valido, se a partir da premissas tem uma prova da conclusão.

A relação entre as noções



Analisar uma fórmula

- Como decidir as características de uma fórmula ϕ?
- Com árvores de refutação: considere $\models \phi$.
- ϕ é
  - Válida, se a árvore fecha (e satisfatível também).
  - Insatisfatível, se todos os ramos ficam em aberto.
  - Satisfatível (e falsificável e contingente) se alguns ramos ficam em aberto, e outros fecham.

Formas normais

Notação

- A disjunção e a conjunção são associativos:

$$p \land (q \land r) \equiv (p \land q) \land r \qquad p \lor (q \lor r) \equiv (p \lor q) \lor r.$$

- Por isso é justificado de escrever

$$p \land q \land r \qquad p \lor q \lor r$$

  e, em geral

$$p_1 \land p_2 \land \cdots \land p_n \qquad p_1 \lor p_2 \lor \cdots \lor p_n$$

  são fórmulas não ambíguas.

- Notação:

$$\bigwedge_{1 \le i \le n} p_i \qquad \bigvee_{1 \le i \le n} p_i.$$

Forma normal conjuntiva

- Um *literal* é uma proposição atômica simples ou negada.
- Uma *claúsula* é uma disjunção de literais.
- Uma fórmula em forma normal conjuntiva (FNC) e uma *conjunção* de cláusulas.

$$\bigwedge_{1 \le j \le m} \bigvee_{1 \le k \le s_j} u_{jk} = (u_{11} \lor \cdots \lor u_{1s_1}) \land \cdots \land (u_{m1} \lor \cdots \lor u_{ms_m})$$

- Simetricamente, um *implicante* é uma conjunção de literais.
- Uma fórmula em forma normal disjuntiva (FND) e uma *disjunção* de implicantes.

$$\bigvee_{1\le j\le m}\ \bigwedge_{1\le k\le s_j} u_{jk} = (u_{11}\land\cdots\land u_{1s_1})\lor\cdots\lor(u_{m1}\land\cdots\land u_{ms_m})$$

As formas normais não são únicas. Por exemplo a fórmula $p \to q$ tem formas normais disjuntivas $\neg p\lor q$, $q\lor\neg p$, $(p\land p)\lor\neg q$. (A falta de é devido: (i) Termos equivalentes com p e $p\land p$ e (ii) a ordem dos termos. É possível de obter uma forma normal única só permitindo os assim-chamados *termos mínimos* (no caso da FNC) ou *termos máximos* (no caso da FND) e definir uma ordem dos termos.)

Questões de decisão

Formais normais facilitem decisões:

- Qual cláusula é válida?

$$p\lor q\lor\neg r$$
$$\neg p\lor q\lor p?$$

- Em geral uma cláusula é válida sse ela contém um literal e sua negação.
- Qual implicante é insatisfatível?

$$p\land q\land\neg r$$
$$\neg p\land r\land p$$

- Em geral um implicante é insatisfatível sse ele contém um literal e a sua negação.

Questões de decisão...

		Fórmula em FND é satisfatível.
Generalização para formas normais:	$\iff$	Existe um implicante satisfatível.
	$\iff$	Existe um implicante que não contém u

		Fórmula em FNC é válida.
Analogamente:	$\iff$	Todas cláusulas são válidas.
	$\iff$	Nenhuma cláusula contém um p e $\neg p$.

Exemplo 5
Considere o seqüênte $p \to q \models \neg q \to \neg p$ ou equivalente

$$\models (p \to q) \to (\neg q \to \neg p).$$

Temos as formas normais

$$\begin{aligned} (p \land \neg q) \lor q \lor \neg p & \qquad \text{FND} \\ (p \lor q \lor \neg p) \land (q \lor \neg q \lor p) & \qquad \text{FNC} \end{aligned}$$

Na FNC é simples de ver que a fórmula é satisfatível (escolhe, por exemplo, $q = v$); na FNC é simples de ver que ela é válida: todas cláusulas contém um literal e a negação dele. ♢

Computação de formas normais

- Se for possível de transformar uma fórmula arbitrária em uma forma normal, a decisão da satisfatibilidade (no caso do FND) ou validade (no caso da FNC) se torna simples.
- Uma transformação é possível usando as equivalências

$$\begin{aligned} p \to q &\equiv \neg p \lor q \\ \neg(p \land q) &\equiv \neg p \lor \neg q \\ \neg(p \lor q) &\equiv \neg p \land \neg q \\ p \lor (q \land r) &\equiv (p \lor q) \land (p \lor r) \\ p \land (q \lor r) &\equiv (p \land q) \lor (p \land r) \\ \neg\neg p &\equiv p. \end{aligned}$$

Computação de formas normais

O seguinte algoritmo constrói uma forma normal:

Passo 1 Elimine a implicação usando $a \to b \equiv \neg a \lor b$.

Passo 2 Produz uma fórmula que contém somente negações de proposições usando os leis de De Morgan (elimina negações duplas).

Passo 3 Produz uma fórmula em forma normal conjuntiva ou disjuntiva usando os leis da distribuição.

Exemplo

Initial $(a \to (b \to c)) \to ((a \to b) \to (a \to c))$

Passo 1 $(a \to (\neg b \lor c)) \to ((\neg a \lor b) \to (\neg a \lor c))$

Passo 1 $(\neg a \lor \neg b \lor c) \to (\neg(\neg a \lor b) \lor \neg a \lor c)$

Passo 1 $\neg(\neg a \lor \neg b \lor c) \lor \neg(\neg a \lor b) \lor \neg a \lor c$

Passo 2 $(a \land b \land \neg c) \lor (a \land \neg b) \lor \neg a \lor c$ (FND)

Passo 3 $(a \lor (a \land \neg b) \land ((b \land \neg c) \lor (a \land \neg b))) \lor (\neg a \lor c)$

Passo 3 $((a \lor a) \land (a \lor \neg b) \land (b \lor (a \land \neg b)) \land (\neg c \lor (a \land \neg b))) \lor (\neg a \lor c)$

Passo 3 $((a \lor a) \land (a \lor \neg b) \land (b \lor a) \land (b \lor \neg b) \land (\neg c \lor a) \land (\neg c \lor \neg b)) \lor (\neg a \lor c)$

Passo 3 $(a \lor a \lor \neg a \lor c) \land (a \lor \neg b \lor \neg a \lor c) \land (b \lor a \lor \neg a \lor c) \land (b \lor \neg b \lor \neg a \lor c) \land (\neg c \lor a \lor \neg a \lor c) \land (\neg c \lor \neg b \lor \neg a \lor c)$

Finalmente $\top$ (porque todas cláusulas contém uma proposição e sua negação e $p \lor \neg p \equiv \top$)

Problema resolvido?

- Conhecemos vários métodos (provas, árvores de refutação, tabelas de verdade) para decidir fórmulas da lógica proposicional.
- Infelizmente, todas tem uma complexidade exponencial no caso pior. (Muitos casos tem uma decisão simples; inclusive a satisfatibilidade de uma função randômica.)
- Usando esse algoritmo a decisão da validade ou satisfatibilidade é mais eficiente?
- Não: neste caso o trabalho de produzir uma forma normal pode ter custo alto!

O problema SAT

- E se o fórmula for dado em forma normal?
- Por exemplo, se for a FNC é fácil de decidir a validade.

- Mas, nesse caso a satisfatibilidade se torna complicado. Por exemplo

$$(x_2 \vee x_3 \vee \neg x_4) \wedge (x_1 \vee x_3 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge$$
$$(\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4) \wedge$$
$$(x_1 \vee \neg x_2 \vee \neg x_4) \wedge (x_1 \vee x_2 \vee \neg x_3)$$

  é satisfatível?

- Pergunta mais famosa em aberto na informática: Tem um algoritmo eficiente de decidir a satisfatibilidade de uma fórmula de lógica proposicional? (Eficiente significa uma fórmula com n proposições pode ser decidida em tempo polinomial n^k.)

2.6. Tópicos

2.6.1. Cláusulas de Horn

O que podemos decidir?

- Restrição para FNC com três literais por cláusula (3-SAT): O problema da satisfatibilidade fica complicado.
- Restrição para FNC com dois literais por cláusula (2-SAT): Existe algoritmo eficiente.
- Restrição para fórmulas de Horn: Existe algoritmo eficiente.

Cláusulas de Horn

- Uma cláusula é *Horn* se ela contém nenhum ou um literal positivo (não negado).
- Exemplos: $\neg p \vee \neg q \vee \neg r$, $p \vee \neg q \vee \neg r$.
- Contra-exemplo: $\neg p \vee q \vee r$, $\neg(p \vee q) \vee \neg r \vee s$.
- Usando a implicação, cláusulas de Horn podem ser escritos

$$p_1 \wedge \cdots \wedge p_n \rightarrow q$$

- Uma fórmula é Horn se ela é uma conjunção de cláusulas Horn.

Por que cláusulas de Horn?

- Varias situações podem ser formalizadas com eles.
- Cláusulas de Horn são a base da representação de conhecimento em Prolog.
- Prolog representa elas na forma `q :- p1,p2,...,p`

Algoritmo

Núcleo Horn

Entrada Um conjunto de cláusulas Horn H.

Saída Uma lista de proposições que têm ser verdadeiras para satisfazer H ou $\bot$ caso H é insatisfatível.

```
Substitui cada p_1 ... p_n → com p_1 ... p_n → ⊥
Marca todas proposições → p.
while (existe p_1 ... p_n → q
       com p_i marcadas mas não q) do
  marca q
end while
if ⊥ marcado then
  return ⊥
else
  return proposições marcadas
end if
```

- Pode ser implementado em tempo linear.

Aplicação: Jogo Nim

- Para dois jogadores.
- Começa com um número n de fósforos.
- Cada um pega $1, 2, 3$ deles.
- Quem pega o último fósforo ganha.

Aplicação: Jogo Nim

- Associa quatro proposições com cada estado (=número de fósforos)
- $A^+(n)$: A ganha jogando com n fósforos.
- $A^-(n)$: A perde jogando com n fósforos.
- $B^+(n)$: B ganha jogando com n fósforos.
- $B^-(n)$: B perde jogando com n fósforos.

Aplicação: Jogo Nim

- Para $n \geq 3$ temos

$$\begin{gathered} A^-(n-3) \to B^+(n) \\ A^-(n-2) \to B^+(n) \\ A^-(n-1) \to B^+(n) \\ A^+(n-3) \wedge A^+(n-2) \wedge A^+(n-1) \to B^-(n) \end{gathered}$$

- E vice versa. Para $n = 1, 2$ uma restrição desses condiçãoes se aplica.
- Para $n = 0$

$$\begin{gathered} \to A^-(0) \\ \to B^-(0) \end{gathered}$$

Resultados: Jogo Nim
A-0 A+1 A+2 A+3 A-4 A+5 A+6 A+7 A-8 A+9 A+10 A+11 A-12 A+13 A+14 A+15 A-16 A+17 A+18 A+19 A-20

2.6.2. Resolução e Prolog

Sistemas automatizados de provas

- Considere a lógica proposicional.
- Podemos automatizar os sistemas de prova (dedução tipo Hilbert ou Gentzen, árvores de refutação).
- A complexidade da busca depende da forma e número de regras.

  - Dedução natural: 12 regras, várias com premissas arbitrárias, algumas introduzem novas fórmulas (p.ex. eliminação do "ou")
  - Árvores de refutação: 8 regras, ordem da aplicação arbitrária.

- Um sistema de prova automatizada preferencialmente tem poucas regras eficientes.

Resolução

- Resolução é um sistema de refutação.
- Para provar que uma fórmula Φ é válida a resolução começa com $\neg\Phi$ em formal normal conjuntiva.
- A prova aplica respectivamente a *regra da resolução*. (A resolução tem a vantagem que só tem uma única regra.)
- Se a prova chega numa contradição, $\neg\Phi$ é insatisfatível e logo Φ é válido.
- Observação: Para provar um seqüente $\Phi_1, \ldots, \Phi_n \vdash \Psi$ podemos
  - usar o teorema equivalente $\vdash (\Phi_1 \land \cdots \land \Phi_n) \to \Psi$ e
  - provar que $\Phi_1 \land \cdots \land \Phi_n \land \neg\Psi$ leva a uma contradição.

Lembrança: FNC

- Uma cláusula é uma disjunção de literais $\bigvee l_i$.
- Uma cláusula $l_1 \lor l_2 \lor \cdots \lor l_n$ é representada pelo conjunto de literais $\{l_1, \ldots, l_n\}$.
- Uma fórmula em forma normal conjuntiva é uma conjunção de *cláusulas* $\bigwedge C_i$.
- Uma fórmula $C_1 \land C_2 \cdots \land C_n$ é representada pelo conjunto de cláusulas $\{C_1, \ldots, C_n\}$.
- A cláusula vazia $\Box$ representa uma contradição (sempre falso).
- A fórmula vazia $\emptyset$ represente uma tautologia (sempre verdadeiro).
- Exemplo:
$$(p \to q) \land \neg(p \land r) \cong \{\{\neg p, q\}, \{\neg p, \neg r\}\}$$

Regra da resolução

- A resolução é baseado na seguinte observação:
- Se duas cláusulas $C_1 = \{l\} \cup C_1'$, $C_2 = \{\neg l\} \cup C_2'$ são satisfatíveis, $C_1' \cup C_2'$ também é satisfatível.
- $C_1' \cup C_2'$ é o *resolvente* de C_1 e C_2 ao respeito do literal l.
- C_1 e C_2 são os *pais* do resolvente.
- Exemplos
  - $\{q, r\}$ é resolvente de $\{p, q\}$ e $\{\neg p, r\}$ (em p)
  - $\{p, q\}$, $\{\neg p, \neg q\}$ tem resolventes $\{q, \neg q\}$ (em p) e $\{p, \neg p\}$ (em q).

A observação acima pode ser justificado da seguinte maneira: $C_1 = l \vee \Phi$, $C_2 = \neg l \vee \Psi$. Logo se C_1 e C_2 são satisfatíveis temos um modelo (uma atribuição) A tal que $A \models C_1$ e $A \models C_2$. Como $A \models l$ ou $A \models \neg l$ temos ou $A \models C_1'$ ou $A \models C_2'$ também. Logo $A \models C_1' \vee C_2'$.

Aplicar a resolução

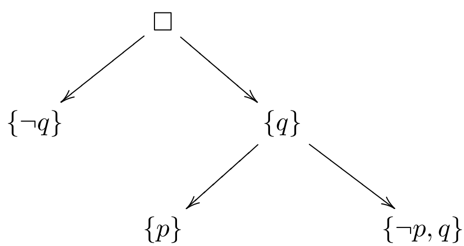
- Prove $p \rightarrow q, \neg q \vdash \neg p$.
- Equivalente: $p \rightarrow q \wedge \neg q \rightarrow \neg p$ é um teorema.
- Equivalente: $(\neg p \vee q) \wedge \neg q \wedge p$ (em FNC) é insatisfatível.
- Representação: $\{\{\neg p, q\}, \{\neg q\}, \{p\}\}$.
- Resolução:

$$\{\neg p, q\}, \{p\} \Rightarrow \{q\}$$
$$\{q\}, \{\neg q\} \Rightarrow \Box$$

Provas com resolução

Apresentamos resoluções a partir de uma fórmula S

- como árvores binárias e folhas em S



- ou linearmente

1	$\{\neg p, q\}$	C_1
2	$\{\neg q\}$	C_2
3	$\{p\}$	C_2
4	$\{q\}$	Res 1,3 em p
5	$\square$	Res 2,4 em q

Prova com resolução

- Formalmente, uma *prova com resolução* de C a partir da fórmula S é uma seqüencia

$$C_1, C_2, \ldots, C_n$$

  tal que $C_n = C$ e para cada C_i

  - $C_i \in S$ ou
  - C_i é o resolvente de C_j e C_k com $j, k < i$.

- Escrevemos $S \vdash_R C$ se existe uma prova de C com resolução a partir de S.

Consistência

- A consistência garante que uma prova é semanticamente válida.

- Ela é uma conseqüência do

  Lema 1 (Consistência da regra de resolução)
  Se $\{C_1, C_2\}$ é satisfatível e C é um resolvente de C_1 e C_2, C também é satisfatível. Mais preciso, cada modelo de $\{C_1, C_2\}$ é um modelo de C.

- Uma prova com indução mostre

  Teorema 3 (Consistência da resolução)
  $S \vdash_R C \Rightarrow S \models C$.

- Em particular, se temos $S \vdash_R \square$, S é insatisfatível.

Completude

- A completude garante que cada seqüente válido tem uma prova com resolução.
- Nosso interesse é

  Teorema 4 (Completude da resolução)
  Se uma fórmula S é insatisfatível, existe uma refutação com resolução de S ($S \vdash_R \Box$).

Exemplo

$(p \vee q) \wedge r \vdash p \vee (q \wedge r)$?

- Teorema: $(p \vee q) \wedge r \rightarrow p \vee (q \wedge r)$
- Insatisfatível: $\neg((p \vee q) \wedge r \rightarrow p \vee (q \wedge r))$
- FNC: $(p \vee q) \wedge r \wedge \neg p \wedge (\neg q \vee \neg r)$
- ou: $\{\{p, q\}, \{r\}, \{\neg p\}, \{\neg q, \neg r\}\}$

1	$\{p, q\}$	Cláusula 1
2	$\{r\}$	Cláusula 2
3	$\{\neg p\}$	Cláusula 3
4	$\{\neg q, \neg r\}$	Cláusula 4
5	$\{q\}$	Res 1,3 com p
6	$\{\neg r\}$	Res 4,5 com q
7	$\Box$	Res 2,6 com r

Discussão

- SAT é NP-completo.
- Por isso, nenhum método pode evitar um trabalho super-linear no caso pior.
- A resolução não faz exceção, que mostra

  Teorema 5 (Haken e Urquhart, 1987)
  Para qualquer n, existe um conjunto de cláusulas de tamanho $O(n)$ tal que o número de cláusulas em uma refutação com resolução é maior que 2^{cn} para um $c > 0$.

Resolução SLD

Motivação

- A resolução é uma forma mais eficiente de buscar uma prova.
- Mas o espaço da busca ainda é grande demais na prática.
- Por isso queremos restrições da resolução que
  - ainda são relevantes na prática,
  - são consistentes e completos e
  - reduzem o espaço da busca.

Motivação

- Uma restrição possível (base de Prolog) é
  - considerar somente cláusulas de Horn (ou cláusulas definidas),
  - resolução linear com
  - uma regra de seleção.
- Essas três componentes levaram ao nome *resolução SLD*.
- Vamos estudar um versão simples na lógica proposicional.

Cláusulas de Horn

- Uma cláusula é *Horn* se ela contém nenhum ou um literal positivo (não negado).
- Exemplos: $\neg p \vee \neg q \vee \neg r$, $p \vee \neg q \vee \neg r$.
- Contra-exemplos: $\neg p \vee q \vee r$, $\neg(p \vee q) \vee \neg r \vee s$.
- Usando a implicação, cláusulas de Horn podem ser escritos

$$p_1 \wedge \cdots \wedge p_n \to q$$

- Cláusulas de Horn são a base da representação de conhecimento em Prolog.
- Prolog representa elas na forma `q :- p1,p2,...,pn.`

Cláusula de Horn em Prolog

- Cláusulas com *exatamente* um literal positivo são *cláusulas de programação* ou *cláusulas definidas* que podem ser
  - *Regras*, se eles contém um ou mais literais negativos `q :- p,r.`
  - *Fatos*, se eles não contém literais negativos `q.`
- Uma cláusula sem literal positivo é uma *cláusula-objetivo* `?- p`

Exemplo
(1) Se Inter ganha e tem sol estou feliz. (2) Se Grêmio ganha estou feliz. (3) Inter ganha. (4) Se Grêmio ganha Inter ganha também. (5) Tem sol. (6) Em dias ímpares Grêmio ganha. (7) Grêmio ganha.

Exemplo

1. `p :- q,r.`
2. `p :- s.`
3. `q.`
4. `q :- s.`
5. `r.`
6. `s :- t.`
7. `s.`

Resolução linear

- Uma prova com *resolução linear* de C a partir de uma fórmula S é uma seqüencia
$$\langle C_0, B_0\rangle, \ldots \langle C_n, B_n\rangle$$
tal que
  - C_0 e cada B_i pertence a S ou é um C_j com $j < i$;
  - Cada C_{i+1} é o resolvente de C_i e B_i e
  - $C = C_{n+1}$

Resolução linear

$$
\begin{tikzcd}
C_0 \arrow[d] & B_0 \arrow[dl] \\
C_1 \arrow[d] & B_1 \arrow[dl] \\
C_2 \arrow[d] & B_2 \arrow[dl] \\
\vdots & \\
C_n \arrow[d] & B_n \arrow[dl] \\
C &
\end{tikzcd}
$$

Resolução SLD

Vantagens da resolução linear:

- Em cada passo temos que somente (i) uma cláusula e (ii) um literal para resolver.
- O método ainda é completo.

Descobrimento fundamental em Prolog: Tem um método ainda mais eficiente para um conjunto de cláusulas de programação P é uma cláusula-objetiva G.

- Define qualquer *regra de seleção*, que escolha um literal para resolver de uma cláusula-objetivo, p.ex. escolhe sempre o primeiro literal.
- Começa a resolução com $C_0 = G$.
- Escolhe cada $B_i \in P$.

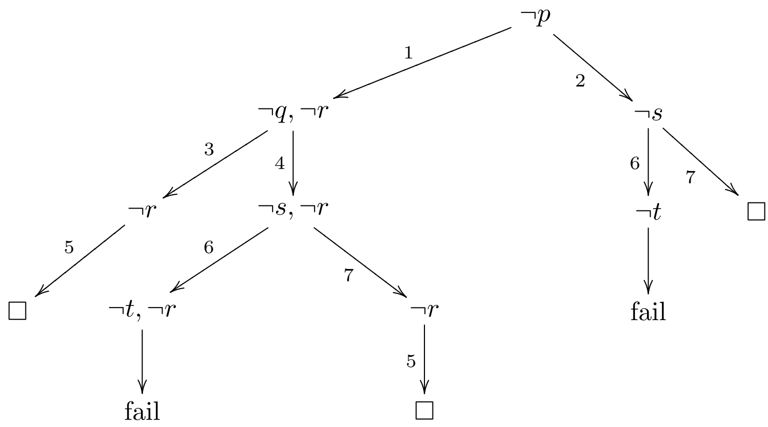
Resolução SLD

- A resolução SLD reduz a busca para uma cláusula para resolver.
- Prolog busca as cláusula na ordem que eles aparecem no programa.

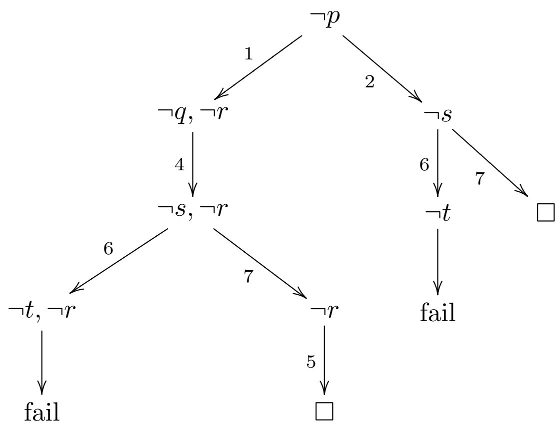
- A completude da resolução SLD garante

  Teorema 6 (Completude da resolução SLD)
  Seja P um programa, G um goal é R alguma regra de seleção. Se $P \cup \{G\}$ é insatisfatível, existe uma refutação com resolução SLD ($P \cup \{G\} \models_{\text{SLD}} \square$).

Exemplo



Exemplo: Sem regra 3



Prolog

- Nosso desenvolvimento foi no contexto lógica proposicional.

- Prolog contém muito mais que isso:
  - Lógica da primeira ordem.
  - Estrutures de dados como listas e strings.
  - Bibliotecas com predicados predefinidos, p.ex. para entrada/saída.
- Da outro lado Prolog não pode representar uma negação.

Exemplo: Prolog

```
professor(ritt, inf05508).
professor(ritt, inf05516).
conteudo(inf05508, logica_proposicional).
conteudo(inf05508, logica_de_predicados).
conteudo(inf05516, semantica_operacional).
conteudo(inf05516, semantica_axiomatica).
conteudo(inf05516, semantica_denotational).
ensina(P,C) :- professor(P,D), conteudo(D,C).

?- ensina(ritt,logica_proposicional).

Yes
?- ensina(ritt,xurumbambo).

No
```

Resolução SLD

- A resolução SLD desenvolve o seu poder só no contexto da lógica da primeira ordem.
- Para resolver programas em "Prolog proposicional" existem algoritmos mais eficientes ($O(N+n)$ com N tamanho das cláusulas e n proposições).
- Idéia: Assume os fatos verdadeiras, elimina eles da premissas busca (iterativamente) nova proposições verdadeiras.

Resolução na lógica da primeira ordem

A resolução em lógica da primeira ordem involve

- um universo concreto chamada *Universo de Herbrand*

- um procedimento chamada *unificação*.
- Exemplo: Para provar `ensina(ritt,logica_proposicional)` o sistema tem que "saber" que a escolha `P=ritt` e `C=logica_proposicional`

2.7. Notas históricas

Gerhard Gentzen inventou o sistema de dedução natural em 1933 [2, 3][1] com o objetivo de modelar raciocínio humano mais intuitivamente[2]. A decibilidade da lógica de predicados usando tabelas de verdade é um resultado de Emil Post [7].

2.8. Exercícios

(Soluções a partir da página 129.)

Exercício 2.1 (Fórmula ou não?)
Quais expressões são fórmulas?

$$pq, p \to q, p\vee \to q, \to q$$
$$\neg p \to \neg, \neg\neg\neg\neg p, q\neg \wedge \neg q$$

Exercício 2.2 (Sub-fórmulas)
Quais são as subfórmulas da árvore de parse da página 11?

Exercício 2.3 (Fórmula ou não?)
Quais expressões são fórmulas?

(a) $p \vee q \wedge r$ (b) $p \wedge \wedge q$
(c) $p\neg\neg q$ (d) $p \to \neg q \wedge r \vee \neg\neg s$

Exercício 2.4 (Fórmulas da lógica proposicional)
Escreve os seguintes proposições (não-atômicos) com fórmulas de lógica proposicional. Cuida de escolher proposições atômicas.

1. Não está chovendo.

[1] Veja também a discussão em [1, começo 3.3].

[2] Gentzen: "Die Formalisierung des logischen Schließens, wie sie insbesondere durch Frege, Russell und Hilbert entwickelt worden ist, entfernt sich ziemlich weit von der Art des Schließens, wie sie in Wirklichkeit bei mathematischen Beweisen geübt wird. Dafür werden beträchtliche formale Vorteile erzielt. Ich wollte nun zunächst einmal einen Formalismus aufstellen, der dem wirklichen Schließen möglichst nahe kommt." [2]

2. Se a luz alcançe um corpo, o corpo projeto sombra.

3. Se tem interferência de aparelhos rádio-transmissores ou de diatermia, aparecem linhas diagonais ou entrelaçadas.

4. Se o sinal está fraco, a imagem está com granulação chuviscos e som ruidoso.

5. Se o meu consumo do cafézinho implica que eu vou pagar para ele, eu vou sair.

Exercício 2.5 (Desenhe as árvores de parse das seguintes fórmulas)

(a) $\neg p \wedge q \rightarrow r$ (b) $(p \rightarrow q) \wedge \neg(r \vee p \rightarrow q)$
(c) $(p \rightarrow q) \rightarrow (r \rightarrow s \vee t)$ (d) $p \vee (\neg q \rightarrow (p \wedge q))$
(e) $p \vee q \rightarrow \neg p \wedge r$ (f) $p \vee p \rightarrow \neg q$

Exercício 2.6 (Seqüentes)

Modela as situações seguintes com fórmulas de lógica proposicional. Escreve o raciocínio usando seqüentes. Quais seqüentes são válidos?
Dicas: Cuida de escolher proposições atômicas. Também observe, que um seqüente é valido se podemos chegar na conclusão usando as premissas: As premissas não necessariamente são verdadeiras na realidade.

1. Se o tanque é vazio, o carro não anda. O tanque é vazio. Logo, o carro não anda.

2. Se o tanque é vazio, o carro não anda. O carro não anda. Logo, o tanque é vazio.

3. Se o carro não anda, o tanque é vazio ou o motor não funciona. O carro não anda. O tanque não é vazio. Logo, o motor não funciona.

4. Se o mundo é redondo, as pessoas do outro lado vai cair. As pessoas do outro lado não caiam. Logo, o mundo não é redondo.

Exercício 2.7 (Provas)

Prove que os seguintes seqüentes são válidos:

(a) $(p \wedge q) \wedge r, s \wedge t \vdash q \wedge s$ (b) $p \vdash (p \rightarrow q) \rightarrow q$
(c) $(p \vee (q \rightarrow p)) \wedge q \vdash p$ (d) $p \rightarrow (q \vee r), q \rightarrow s, r \rightarrow s \vdash p \rightarrow s$
(e) $\neg p \rightarrow p \vdash p$ (f) $r, p \rightarrow (r \rightarrow q) \vdash p \rightarrow (q \wedge r)$
(f) $(p \wedge q) \wedge r \vdash p \wedge (q \wedge r)$

Exercício 2.8 (Alfabeto grego)
Pesquisa as letras do alfabeto grego. Transcreva os seguintes palavras usando letras gregas: sofía (sabedoria), fílos (amigo), filosofia (filosofia), andropos (homem), lógos (palvara, razão), phóbos (medo), logikí (lógica), logikó (razão).

Exercício 2.9 (Provas)
Prove os seguintes seqüentes:

1. O lei da contraposição (estendida)

$$(p \wedge q) \to r \dashv\vdash (p \wedge \neg r) \to \neg q$$

2. O seqüente

$$(p \to q) \to (r \to s) \vdash (p \to s) \to (r \to s)$$

Exercício 2.10 (Sim-Não do Sul)
Finalmente, chegei em Sim-Não do Sul. "Eles acharam ouro?", queria saber. Mas cuidado: Pessoas sempre honestas aqui se misturam com mentirosos sistemáticos. Saiu do cavalo e me aproximou a dois habitantes, Pedro e Paulo. "Senhores", perguntei, "tem ouro em Sim-Não do Sul?" "Sim", respondeu Pedro, "tem ouro e Paulo é bem honesto." "É verdade, tem ouro, mas Pedro é um mentiroso", falou Paulo.

Então: Vale a pena de buscar ouro ou não?

1. Modela essa situação na lógica proposicional: Acha proposições atômicos. Cuida que as afirmações de Pedro e Paulo não são premissas, porque a validade deles depende do que a pessoa falando é honesto ou mentiroso!

2. Tem ouro ou não? Prove um desses casos usando a dedução natural.

Exercício 2.11 (Leis)
Usando dedução natural prove os seguintes leis.

$$p \vee (q \wedge r) \dashv\vdash (p \vee q) \wedge (p \vee r) \tag{2.30}$$

$$p \vee p \dashv\vdash p \tag{2.31}$$

$$p \vee (q \vee r) \dashv\vdash (p \vee q) \vee r \tag{2.32}$$

$$\neg(p \vee q) \dashv\vdash \neg p \wedge \neg q \tag{2.33}$$

Exercício 2.12 (Regras derivadas)
Prove os seguintes leis.

$$
\begin{aligned}
p \lor q, \neg p &\vdash q && \text{(modus tollendo ponens)}\\
p \to q, p &\vdash b && \text{(modus ponendo ponens)}\\
p \to q, \neg q &\vdash \neg p && \text{(modus tollendo ponens)}\\
\neg p \lor \neg q, p &\vdash \neg q && \text{(modus ponendo tollens)}
\end{aligned}
$$

(Os nomes são em Latin, do "modus" (modo), "ponere" (por) e "tolere" (tirar, abolir)).

Exercício 2.13 (Implicação e seqüentes)

1. Prove o seguinte fato: $p \vdash q$ é válido se e somente se $p \to q$ é um teorema ($\vdash p \to q$).

2. Use a argumentação de (a) para provar um fato mais geral: $p_1, p_2, \ldots p_n \vdash q$ sse $\vdash p_1 \to (p_2 \to \cdots (p_n \to q) \cdots)$. (Por exemplo $p_1, p_2 \vdash q$ sse $\vdash p_1 \to (p_2 \to q)$ e $p_1, p_2, p_3 \vdash q$ sse $\vdash p_1 \to (p_2 \to (p_3 \to q)))$.

Exercício 2.14 ($\land$ e $\lor$)
Prove as equivalências $A \land B \dashv\vdash \neg(A \to \neg B)$ e $A \lor B \dashv\vdash \neg A \to B$ com dedução natural.

Exercício 2.15 (O sistema $\mathcal{H}$)
Prove as seguintes regras no sistema $\mathcal{H}$ de Hilbert:

1. A regra da introdução da negação dupla: $\vdash A \to \neg\neg A$. (Dica: Usa $\neg\neg A \to A$ da aula!)

2. A regra da introdução da disjunção: $\vdash A \to (\neg A \to B)$. (Usa as abreviações da 65 por exemplo $A \lor B \equiv \neg A \to B$, e as provas conhecidas!).

Exercício 2.16 (Tabelas de verdade)
Construa a tabela de verdade das seguintes fórmulas:

(a) $p \land \neg p$
(b) $p \land \neg q$
(c) $p \land q \to p \lor q$
(d) $p \to (q \to p)$
(e) $(p \to q) \to (\neg p \to \neg q)$
(f) $\neg(\neg p \land \neg q)$
(g) $((p \land q) \oplus (r \land s)) \oplus ((p \lor q) \land (r \oplus s))$
(h) $(\neg p \land \neg q \land \neg r) \lor (\neg p \land \neg r \land \neg s) \lor (q \land r \land s) \lor (p \land q)$
(i) $((p \land q) \lor (r \land s)) \oplus ((p \lor q) \land (r \lor s))$

Exercício 2.17 (Relação de conseqüência semântica)
Usando tabelas de verdade, pesquisa se as seguintes relações de conseqüência semântica são corretas:

1. $\neg(\neg p \vee q) \models p$
2. $\neg p \rightarrow p \models p$
3. $p \rightarrow q \models \neg(p \wedge \neg q)$
4. $p \oplus p \models p$
5. $(p \oplus q) \oplus p \models q$
6. $p \oplus 1 \models \neg p$
7. $(p \oplus q) \wedge r \models (p \wedge r) \oplus (q \wedge r)$
8. $p \oplus q \models \neg(p \equiv q)$

Faz a mesma análise com a relação contrária (por exemplo $p \models \neg(\neg p \vee q)$ no caso 1): Quais fórmulas são semanticamente equivalentes?

Exercício 2.18 (Retorno a Sim-Não do Sul)
Lembre-se de Sim-Não do Sul (exercício 2.10)? Desta vez, usa tabelas de verdade para analisar a situação.

Exercício 2.19 (Consistência)
Na aula vimos a prova da consistência da dedução natural em relação à semântica. Usando indução completa, ela está baseado numa análise de última regra aplicada em uma prova de k linhas. Na aula foram analisados os casos de $\wedge_{\mathrm{i}}$ e $\rightarrow_{\mathrm{i}}$. Faz a análise se a última regra aplicada foi: (a) $\vee_{\mathrm{e}}$. (b) $\rightarrow_{\mathrm{e}}$.

Exercício 2.20 (Associatividade)
Pesquisa a associatividade misturada de $\vee$ e $\wedge$, ou seja, decide os seqüentes

$$p \vee (q \wedge r) \vdash (p \vee q) \wedge r?$$
$$p \vee (q \wedge r) \dashv (p \vee q) \wedge r?$$

Dica: A pergunta é sobre $\vdash$, é assim sobre a existência de provas na dedução desses seqüentes. Usando a consistência e completude da dedução natural e de árvores de refutação é provavelmente mais simples de fazer a desvia usando uma dessas técnicas. Justifique a sua resposta.

Exercício 2.21 (Avaliação semântica)
Em qualquer linguagem de programação, implementa um programa, que avalia seqüentes. A entrada será uma representação de um seqüente

$$\Phi_1, \Phi_2, \ldots, \Phi_n \models \Psi$$

e a saída será "Verdadeira", se o seqüente é valido, ou "Falso" senão.
Dica: Concentra na avaliação de fórmulas compostas usando tabelas de verdade. Não é necessário de implementar um parser para a entrada: e permitido de codificar uma entrada diretamente na linguagem. Assim, o tarefa consiste em (a) Achar uma estrutura de dados para representar fórmulas de lógica proposicional. (b) Achar um algoritmo que produz uma tabela de verdade para todas as premissas e a conclusão e decide se o seqüente em consideração está correto.

Exercício 2.22 (Somadores)

1. Um *meio somador* com entradas a, b produz saídas s (a soma), c (o carry) conforme a tabela de verdade

a	b	s	c
f	f	f	f
f	v	v	f
v	f	v	f
v	v	f	v

   Acha fórmulas da lógica proposicional para s e c.

2. Um *somador inteiro* com entradas a, b e c (carry) produz saídas s(soma), c' (novo carry) conforme a tabela de verdade

a	b	c	s	c'
f	f	f	f	f
f	f	v	v	f
f	v	f	v	f
f	v	v	f	v
v	f	f	v	f
v	f	v	f	v
v	v	f	f	v
v	v	v	v	v

   Acha fórmulas da lógica proposicional para s e c.

Exercício 2.23 (Subtratores)
Analogamente ao exercício 2.22 podemos definir um *meio subtrator* e um *subtrator inteiro* que, para entradas a, b e "underflow" (bit emprestado) u produzem a diferença d e um "underflow" atualizado u' da forma

t

a	b	d	u
f	f	f	f
f	v	v	v
v	f	v	f
v	v	f	f

a	b	u	d	u'
f	f	f	f	f
f	f	v	v	v
f	v	f	v	v
f	v	v	f	v
v	f	f	v	f
v	f	v	f	f
v	v	f	f	f
v	v	v	v	v

Acha fórmulas da lógica proposicional para d e u' nos dois casos.

Exercício 2.24 (Indução)
Seja S algum alfabeto (um conjunto de letras) é S^* o conjunto de cadeias ("strings") sobre esse alfabeto (por exemplo, com $S = \{x, y\}$,

$$S^* = \{\epsilon, x, y, xx, xy, yx, yy, xxx, \ldots\}$$

, com a cadeia vazia ϵ.) Prova que não tem uma cadeia $s \in S^*$ e símbolos $a, b \in S$ tal que $as = sb$, se $a \neq b$ usando indução natural.

Exercício 2.25 (Árvores de refutação)
Prove com árvores de refutação ou mostre um contra-exemplo.

1. $\vdash (p \to q) \to ((p \to r) \to (q \to r))$
2. $p \to q \dashv\vdash \neg p \lor q$
3. $p \to (q \to r) \vdash p \to (r \to q)$
4. $q \lor p, q \to \neg r \vdash q \lor ((\neg r \to p) \land (p \to \neg r))$
5. $(s \to t) \land (t \to s), t \to (s \to q), \neg r \to \neg p \vdash (p \lor s) \to (r \lor q)$
6. $\neg p \lor (q \to p) \vdash \neg p \lor q$
7. $\vdash ((p \land q) \to r) \to (p \to (q \to r))$
8. $\neg r \to (p \lor q), r \land \neg q \vdash r \to q$
9. $p, q, \neg r \vdash \neg(((q \to r) \to (q \to q)) \land ((r \land q) \land (p \land q)))$

10. $\vdash ((p \to r \land q \to r) \land (p \lor q)) \to r$

Exercício 2.26 (Dedução natural)
Prove com dedução natural ou mostre um contra-exemplo.

1. $p \to (q \to r) \dashv\vdash q \to (p \to r)$
2. $p \to (q \to r) \dashv\vdash (p \land q) \to r$
3. $p, \neg q, r \vdash \neg(p \to ((q \lor r) \to (r \to \neg p)))$

Exercício 2.27 (Formas normais)
Acha a forma normal conjuntiva.

1. $(p \to q) \to (q \to r)$
2. $(((p \to p) \lor (r \lor r)) \to (\neg p \land \neg r))$
3. $(((p \lor p) \to \neg r) \to ((r \land p) \to (r \to q)))$

Exercício 2.28 (Extensões da lógica proposicional)
Queremos estender a lógica proposicional com os seguintes conectivos

$$p \otimes q =_{def} (\neg p \land q) \lor (p \land \neg q)$$
$$p \leftrightarrow q =_{def} (p \to q) \land (q \to p).$$

1. Define a semântica dos novos conectivos.
2. Estende o sistema de dedução natural com regras adequadas.
3. Estende as árvores de refutação com regras adequadas.
4. Justifique corretude e completude dos novos sistemas de prova.

3. Lógica de predicados

3.1. Introdução

Introdução

- Considere as declarações

  Francesco é filho de Laura. Laura é filha de Ana.

  Todos humanos são mortais. Diego é um ser humano.

  Alguns passaros não voam.

- A lógica proposicional somente permite de modelar os exemplos com proposições atômicas.

  p: Francesco é o filho de Laura. q: Laura é a filha de Ana.

  r: Todos humanos são mortais. s: Diego é um humano.

  t : Alguns aves não voam.

Introdução

- Usando só proposições atômicas, não podemos concluir

  Francesco é o neto de Ana. Diego é mortal.

- Essas sentenças na linguagem natural contém mais detalhes que a lógica proposicional permite de modelar.
- O que precisamos?
- Temos objetos como "Francesco", "Ana", "Diego".
- Temos características deles como "ser mortal".
- Temos relações entre eles como "ser o neto de ".
- Temos afirmações como "Todos" ou "Alguns".

Predicados

- *Predicados* melhoram essa situação.
- Intuitivamente, um predicado afirma uma característica de um objeto ou uma relação entre objetos.
- Por exemplo, escrevemos

  mortal(Diego): Diego é mortal. filho(Francesco,Laura): Francesco é o filho de Laura.

- Mais breve, escrevemos
  - Predicados com letras maiúsculas: M(Diego), F(Francesco,Laura)
  - Constantes (ou objetos) com letras minúsculas $M(d)$, $F(f, l)$.

Predicados: Exemplos

- O que seriam os predicados para

  Laura é filha de Ana. Ricardo é mais novo que Diego. Laura e Rita são filhas de Ana. Rita não é professora.

- Podemos usar os conectivos da lógica de predicados!
- Um predicado pode ter algum número de argumentos.
- O número de argumentos se chama *aridade*. (Qual é a aridade de "mortal" e "filho de"? Exemplo de um predicado com três argumentos?)

Exemplo 6
Predicados com três argumentos

$$\begin{array}{ll} S(x, y, z) & z \text{ é a soma de } x \text{ e } y \\ F(x, y, z) & x \text{ tem pai } y \text{ e mão } z \end{array}$$

$\Diamond$

Variáveis

- Em algumas situações não queremos falar sobre objetos concretos.
- Por exemplo, um predicado pode ser definido mais claro como

  $M(x)$: x é mortal. $F(x, y)$: x é filho de y.
- $x, y, z, \ldots$ são variáveis para objetos.
- Também permitimos alterações como $x_1, x', x'_2, y_1, \ldots$.

Quantificadores

- Como modelar "Todos os humanos são mortais"?
- Temos a "lista"

Humano	Mortal
H(Diego)	M(Diego)
H(Ricardo)	M(Ricardo)
...	
¬H(Zeus)	¬M(Zeus)
¬H(Passaro)	M(Passaro)
...	

- Se x é humano, logo, x é mortal: $H(x) \to M(x)$
- Para enfatizar que essa frase se aplica *para cada* x, escrevemos

$$\forall x H(x) \to M(x)$$

  (lê: para cada x: humano(x) implica mortal(x))
- $\forall$ é o *quantificador universal*.
- Ele afirma que uma fórmula é correta para qualquer objeto.

Quantificador universal: exemplos

Todos estudantes gostam café e churrasco. Todos professores gostam café ou churrasco. Qualquer objeto é uma rã verde. Não é o caso que todos humanos são estudantes. Cada empregado ganha um salário. Qualquer coisa não tem valor. Qualquer objeto é um gato ou um cachorro.

Quantificadores

- Como modelar "Alguns passaros não voam." ?
- Temos a "lista"

Passaro	Voa
P(andorinha africana)	V(andorinha africana)
P(andorinha européia)	V(andorinha européia)
P(águia)	V(águia)
...	
P(pinguim)	¬V(pinguim)

- Logo, existem objetos x tal que $P(x) \land \neg V(x)$.
- Escrevemos
$$\exists x P(x) \land \neg V(x)$$
(lê: existe x tal que humano(x) implica não mortal(x)).
- $\exists$ é o *quantificador existencial.*
- Ele afirma que uma fórmula é correto para ao menos um objeto.

Quantificador existencial: exemplos

Alguns estudantes gostam de tomar café. Alguns professores não gostam de tomar café. Existem rãs verdes. Tem pessoas que gostam dormir ou comer. Não existem cachorros brancos. Tem pessoas que não gostam dormir nem comer. Se nada é verde, então não existem rãs verdes.

Observações

- Usamos as convenções da prioridade da lógica proposicional.
- Os quantificadores novos $\forall$ e $\exists$ tem a mesma prioridade como $\neg$.

Mais Predicados: Identidade

- Considera de formalizar "Só pode haver um Highlander".
- Em outras palavras: Se dois objetos são um Highlander, se trata do mesmo objeto.

- Com os predicados $H(x)$: x é um Highlander. $I(x, y)$: x é y são idêntico (igual). temos
$$\forall x \forall y : H(x) \wedge H(y) \rightarrow I(x, y)$$

- O predicado I da identidade é tão comum, que ele faz parte da lógica de predicados
  - O seu nome é $=$
  - A notação é $x = y$ (infix) ao invés de $=(x, y)$ (prefix).
  - $=$ é uma relação de equivalência (reflexiva, comutativa, transitiva).

Funções

- Considere "A mãe do meu irmã é meu mãe".
- Com

  $M(x, y)$: x é a mãe de y. $I(x, y)$: x é a irmã do y. e: Eu.

  uma formalização possível é
$$\forall x \forall y I(x, e) \wedge M(y, x) \rightarrow M(y, e)$$

- Considere "Para qualquer x,y a soma de x e y é igual a soma de y e x."
- Com "$S(x, y, z)$: z é a soma de x e y." uma formalização possível é
$$\forall x \forall y \forall s_1 \forall s_2 S(x, y, s_1) \wedge S(x, y, s_2) \rightarrow s_1 = s_2$$

Funções

- Usando funções podemos formalizar essas sentenças mais fácil.
- Por exemplo, com uma função "mãe" m e uma função soma(x, y) (ou $+(x, y)$) temos
$$\forall x I(x, e) \rightarrow m(x) = m(e)$$
$$\forall x \forall y : \text{soma}(x, y) = \text{soma}(y, x)$$

- Observe: Usar uma função é possível, se um objeto sempre tem uma relação com um único outro objeto.

Previsão

Em analogia com a lógica proposicional, nosso interesse é

- Definir a linguagem formal da lógica de predicados.
- Modelar situações e provar seqüentes como

$$\Phi_1, \dots, \Phi_n \vdash \Psi$$
$$\Phi_1, \dots, \Phi_n \models \Psi.$$

- Para isso, os sistemas de prova (dedução, árvores de refutação) precisam novas regras.
- Saber o que uma fórmula da lógica de predicados significa (definir a semântica).
- Depois, é necessário de responder à questão da consistência e completude novamente.
- Aplicar a lógica de predicados!

3.2. Sintaxe

Linguagem formal

- Diferente da lógica proposicional temos dois tipos de frases:
  - Constantes, funções e variáveis (e a suas combinações) denotam objetos.
  - Predicados, conectivos e quantificadores (e os seus combinações) denotam fórmulas.
- Logo, a definição da linguagem consiste em
  - Uma gramática para *termos.*
  - Uma gramática para *fórmulas.*

Termos

Os termos consistem em

- um conjunto de variáveis $\mathcal{V}$,
- e um conjunto de funções $\mathcal{F}$.

- E os constantes?
  - Podemos tratar constantes como funções sem argumentos (funções de aridade 0).
  - Com $c \in \mathcal{F}$ de aridade 0, escrevemos c ao invés de $c()$.
  - (O objetivo é de simplificar a modelagem matemática.)
- Exemplos de termos são

$$m(x), f(g(x)), f(c, m(x))$$

Termos

- Com variáveis $\mathcal{V}$ e funções $\mathcal{F}$ os termos são definidos (indutivamente) como
  1. Uma variável $v \in \mathcal{V}$ é um termo.
  2. Se $c \in \mathcal{F}$ é uma função de aridade 0, c é um termo.
  3. Se $f \in \mathcal{F}$ é uma função de aridade n, é $t_1, \ldots, t_n$ são termos, $f(t_1, \ldots, t_n)$ é um termo.
- A gramática correspondente (em forma de Backus Naur) é

$$t ::= v \mid c \mid f(t_1, t_2, \ldots, t_n)$$

com $v \in \mathcal{V}$.

Fórmulas

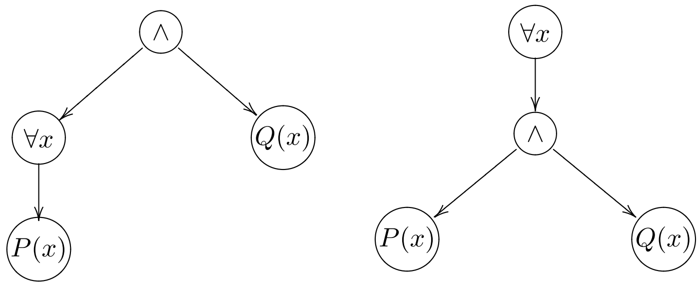
- Para definir as fórmulas precisamos um conjunto de predicados $\mathcal{P}$.
- Os fórmulas são definidos (indutivamente) como
  1. Se $p \in P$ é um predicado de aridade $n \geq 1$, e $t_1, \ldots, t_n$ são termos, $p(t_1, \ldots, t_n)$ é uma fórmula.
  2. $\top$ e $\bot$ são fórmulas.
  3. Se Φ é uma fórmula, $\neg\Phi$ também.
  4. Se Φ e Ψ são fórmulas, $\Phi \wedge \Psi$, $\Phi \vee \Psi$ e $\Phi \to \Psi$ também.
  5. Se Φ é uma fórmula e $v \in \mathcal{V}$, $\forall v\Phi$ e $\exists v\Phi$ também são fórmulas.
- A gramática correspondente é

$$\Phi ::= p(t_1, \ldots, t_n) | (\neg\Phi) | (\Phi \vee \Psi) | (\Phi \wedge \Psi) | (\Phi \to \Psi) | (\forall v\Phi) | (\exists v\Phi) | \top | \bot$$

com $v \in \mathcal{V}$.

Observações

- Na fórmulas $\forall x \Phi$ e $\exists x \Phi$, Φ se chama o *escopo* da variável x.
- Fórmulas como $\forall x P(x) \land Q(x)$ são ambíguas; as árvores de parse



  são possíveis.
- Em caso de dúvidas, escrevemos $(\forall x P(x)) \land Q(x)$ ou $\forall x (P(x) \land Q(x))$.
- No primeiro caso, com a convenção que os quantificadores tem as mesma prioridade que $\neg$, podemos retirar as parênteses.

Observações

- A ordem dos quantificadores é importante. Compare

$$\forall x \exists y P(x, y) \qquad \exists y \forall x P(x, y).$$

- Com $P(x, y)$: "x ama y" (e só considerando pessoas) temos

  > Cada pessoa ama alguma outra. Tem uma pessoa que todo mundo ama.

- Um quantificador pode ocorrer numa sub-fórmula. Compare

$$\forall x P(x) \to \exists x Q(y) \qquad \forall x \exists y P(x) \to Q(y).$$

- A segunda fórmula tem *a forma normal prenexa.*

Tipos de variáveis

- Uma variável x num escopo de um $\forall x$ ou $\exists x$ é *ligada*.
- Uma variável não ligada é *livre*.
- Podemos definir (indutivamente) o conjunto de variáveis livres L

Variáveis livres

- Para um termo t
    1. Se t tem a forma x, $L(t) = \{x\}$.
    2. Se t tem a forma c, $L(t) = \emptyset$.
    3. Se t tem a forma $f(t_1, \ldots, t_n)$, $L(t) = L(t_1) \cup \cdots \cup L(t_n)$.
- Para fórmulas Φ
    1. Se Φ tem a forma $P(t_1, \ldots, t_n)$, $L(\Phi) = L(t_1) \cup \cdots L(t_n)$.
    2. Se Φ tem a forma $\neg\Psi$, $L(\Phi) = L(\Psi)$.
    3. Se Φ tem a forma $\Psi \vee \chi$, $\Psi \wedge \chi$, $\Psi \to \chi$, $L(\Phi) = L(\Psi) \cup L(\chi)$.
    4. Se Φ tem a forma $\forall x : \Psi$, $\exists x : \Psi$, $L(\Phi) = L(\Psi) \setminus \{x\}$.

Observações

- Numa árvore de parse as variáveis sempre ocorrem numa folha. Uma variável ligada encontra um quantificador com a mesma variável num caminho para a raiz.
- Se $L(\Phi) = \{x_1, \ldots, x_n\}$, o *fecho universal* de Φ é

$$\forall x_1 \cdots \forall x_n \Phi.$$

- Se $L(\Phi) = \{x_1, \ldots, x_n\}$, o *fecho existencial* de Φ é

$$\exists x_1 \cdots \exists x_n \Phi.$$

Substituições

- A definição das regras da dedução precisa a substituição.
- Exemplo: Considere a relação entre
$$\forall x P(x) \qquad \text{e} \qquad P(a).$$
- Escrevemos $\Phi[t/x]$ para a substituição de um termo t para a variável x numa fórmula Φ.
- Observe, que "$\Phi[t/x]$" não é uma fórmula, mas faz parte da meta-linguagem; somente o resultado da substituição.
- Exemplo: Com $\Phi \equiv P(x)$, temos $\Phi[a/x] \equiv P(a)$.
- Substitua com cautela
  - Não substitua variáveis ligadas: $\forall x : P(x)[y/x] \equiv ?$
  - Não liga variáveis livres: $\forall x : P(y)[x/y] \equiv ?$

3.3. Teoria de modelos

Introdução

- Em analogia com a lógica proposicional, queremos atribuir valores de verdade a uma fórmula da lógica de predicados.
- Na lógica proposicional a abordagem foi
  - Atribuir um valor de verdade a cada proposição elementar.
  - Definir os valores de verdade de fórmulas compostas.

A abordagem

- Intuitivamente, um predicado denota um valor de verdade.
- A partir disso, é possível de usar as definições da lógica proposicional.
- Em aberto: Como obter valores de verdade dos predicados e dos quantificadores?
  - No nível dos termos, temos variáveis e funções (que intuitivamente denotam objetos). Mas a nossa definição foi somente sintático: temos nada mais que nomes!
  - No nível das fórmulas, temos, além dos predicados, os quantificadores.

Universo

- O *universo (de discurso)* é o conjunto de todos objetos que queremos "discutir" ou "interpretar" a lógica de predicados.
- Com um universo de discurso concreto, é possível de definir o resto da semântica:
  - Cada variável e constante denota um elemento do universo.
  - Uma função (nome) denota uma função (real) entre elementos do universo.
  - Um predicado denota uma função do universo para os valores de verdade (=um subconjunto de tuplas do universo).

Estruturas

Uma *estrutura* $\mathcal{M}$ de uma lógica de predicados com *vocabulário* $(\mathcal{F}, \mathcal{P})$ (símbolos de funções e predicados) consiste em

- um universo $U \neq \emptyset$ de elementos (objetos,valores) concretos.
- uma mapa a que associa
  - com cada símbolo de função de aridade 0 (constante) $c \in \mathcal{F}$, um elemento $c^{\mathcal{M}} \in U$.
  - com cada símbolo de função de aridade n, $f \in \mathcal{F}$, uma função $f^{\mathcal{M}} : U^n \to U$.
  - com cada símbolo de predicado de aridade n, $P \in \mathcal{P}$, uma relação $P^{\mathcal{M}} \subseteq U^n$.
- Notação: $\mathcal{M} = (U, m)$ ou $\mathcal{M} = (U, P_1^{\mathcal{M}}, \ldots, f_1^{\mathcal{M}}, \ldots, c_1^{\mathcal{M}}, \ldots)$.

Exemplo 7
A aritmética pode ser formalizada com uma lógica de primeira ordem com símbolos de funções $+, \cdot$ e símbolos de constantes $0, 1$. A estrutura correspondente $\mathcal{N}$ consiste em

- o universo $\mathbb{N}$ do números naturais e
- uma função m que associa a função de adição $+^N$ com $+$, a função de multiplicação $\cdot^{\mathcal{N}}$ com $\cdot$ e os números zero e um com 0 e 1, i.e.

$$\mathcal{N} = (\mathbb{N}, m) = (\mathbb{N}, +, \cdot, 0, 1).$$

Obtemos aritmética com comparação adicionando o símbolo de predicado $<$ que é associado na estrutura padrão com a relação de comparação $<^{\mathcal{M}}$

$$\mathcal{N}^{<} = (\mathbb{N}, <, +, \cdot, 0, 1).$$

◊

A semântica da identidade

- A identidade faz parte da (nossa) lógica de predicados independente do universo.
- Portanto, ela tem um significado pré-definido para qualquer universo:

$$=^{\mathcal{M}} = \{(u, u) | u \in U\}$$

Atribuições

- O que aconteceu com as variáveis?
- A estrutura não atribuiu um elemento do universo nelas.
- Uma *atribuição* serve para definir os significado das variáveis.
- Uma atribuição é uma função $a : \mathcal{V} \to U$.
- Separamos a atribuição da estrutura, porque freqüentemente
  - o significado das variáveis depende da aplicação ou
  - uma fórmula não tem variáveis livres (é uma *sentença*), e logo, o significado das variáveis não importa.
- Notação: $a[x \mapsto u](v) = \begin{cases} u & v = a \\ a(v) & \text{senão} \end{cases}$
- Chamamos um par de uma estrutura e uma atribuição uma *interpretação* $I = (\mathcal{M}, a)$.

A relação de satisfação

- Uma interpretação $I = (\mathcal{M}, a)$ é suficiente para definir o valor de verdade de uma fórmula Φ:

  $$\mathcal{M} \models_a \Phi$$

  (Lê: Φ é correto (é verdadeiro) na estrutura $\mathcal{M}$ e atribuição a ou eles são um *modelo* de Φ)

- (Na lógica de predicados a situação foi mais simples e não usamos esse notação. Ela corresponde com uma linha da tabela de verdade!)

- Para a afirmação negada, escrevemos $\mathcal{M} \not\models_a \Phi$.

- Se a corretude não depende da atribuição a (em fórmulas sem variáveis livres), escrevemos

  $$\mathcal{M} \models \Phi.$$

Definição da relação de satisfação

- Se $\Phi \equiv P(t_1, \ldots, t_n)$: A interpretação fornece um elemento $u_i \in U$ para cada t_i. Neste caso $\mathcal{M} \models_a \Phi$ sse $(u_1, \ldots, u_n) \in P^{\mathcal{M}}$.

- Se $\Phi \equiv \neg\Psi$: $\mathcal{M} \models_a \Phi$ sse não $\mathcal{M} \models_a \Psi$.

- Se $\Phi \equiv \Psi_1 \land \Psi_2$: $\mathcal{M} \models_a \Phi$ sse $\mathcal{M} \models_a \Psi_1$ e $\mathcal{M} \models_a \Psi_2$.

- Se $\Phi \equiv \Psi_1 \lor \Psi_2$: $\mathcal{M} \models_a \Phi$ sse $\mathcal{M} \models_a \Psi_1$ ou $\mathcal{M} \models_a \Psi_2$ (ou ambas).

- Se $\Phi \equiv \Psi_1 \to \Psi_1$: $\mathcal{M} \models_a \Phi$ sse $\mathcal{M} \not\models_a \Psi_1$ ou $\mathcal{M} \models_a \Psi_2$.

- Se $\Phi \equiv \forall x\Psi$: $\mathcal{M} \models_a \Phi$ sse para todos $u \in U$ temos $\mathcal{M} \models_{a[x \mapsto u]} \Psi$.

- Se $\Phi \equiv \exists x\Psi$: $\mathcal{M} \models_a \Phi$ sse existe um $u \in U$ tal que $\mathcal{M} \models_{a[x \mapsto u]} \Psi$.

Exemplos

- $\mathcal{M} \models (x > 0 \land y > 0 \to xy > 0)$ (com a estrutura intuitiva)?

- $\mathcal{M} \models (sgn(x) = sgn(y) \to xy > 0)$?

- Qual seria uma estrutura tal que

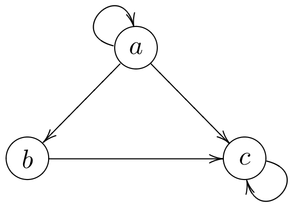
  - $\mathcal{M} \models \forall x P(x)$
  - $\mathcal{M} \models \exists x P(x)$

  é correto? Qual uma tal que eles não são corretos?

Exemplos

- Com $\mathcal{F} = \{i\}$ e $\mathcal{P} = \{S, F\}$ seja U o conjunto dos estados (de uma máquina ou programa).
- Por exemplo seja $U = \{a, b, c\}$, $i^{\mathcal{M}} = a$, $R^{\mathcal{M}} = \{(a, a), (a, b), (a, c), (b, c), (c, c)\}$ e $F^{\mathcal{M}} = \{b, c\}$.
- Considere

$$\mathcal{M} \models \exists y R(i, y)$$
$$\mathcal{M} \models \neg F(i)$$
$$\mathcal{M} \models \forall x \forall y \forall z (R(x, y) \land R(x, z) \to y = z)$$
$$\mathcal{M} \models \forall x \exists y R(x, y)$$



Características de fórmulas

- As noções *satisfatível*, *válido*, *falsificável* e *insatisfatível* se aplicam na lógica de predicados também

Uma fórmula Φ é	se
satisfatível	existem $\mathcal{M}$ e a, tal que $\mathcal{M} \models_a \Phi$
válida	para todos $\mathcal{M}$ e a temos $\mathcal{M} \models_a \Phi$
falsificável	existem $\mathcal{M}$ e a, tal que e $\mathcal{M} \not\models_a \Phi$
insatisfatível	para todos $\mathcal{M}$ e a temos $\mathcal{M} \not\models_a \Phi$

Características de conjuntos de fórmulas

- Qual seria uma definição adequada da relação de conseqüência semântica?
- Escrevemos

$$\Gamma \models \Phi$$

  com $\Gamma = \{\Phi_1, \Phi_2, \ldots\}$
- e definimos

  - Γ é *válido* ou *consistente*, se existe $\mathcal{M}$ e a tal que $\mathcal{M} \models_a \Phi$ para qualquer $\Phi \in \Gamma$.
  - $\Gamma \models \Phi$ sse para todos $\mathcal{M}$ e a, sempre quando $\mathcal{M} \models_a \Phi_i$ para todos $\Phi_i \in \Gamma$, também temos $\mathcal{M} \models_a \Phi$.

Decidir as características

- Definir noções é simples, mas como decidir se eles se aplicam?
- Na lógica proposicional, com n proposições o trabalho em geral foi $O(2^n)$ para decidir a validade de uma fórmula, uma relação de conseqüência, etc.: Teste todas as estruturas possíveis.
- Qual seria a abordagem correspondente na lógica de predicados?
- Suponha um universo tal que $|U| = n$.
- Para testar todas as estruturas, temos
  - $n^{|V|}$ possibilidades de escolher elementos para variáveis,
  - $n^{(n^k)}$ possibilidades para *cada* função com aridade k,
  - $2^{(n^k)}$ possibilidades para *cada* predicado com aridade k
- Certamente essa busca precisa um trabalho exorbitante.
- Ainda pior, suponha um universo infinito (por exemplo, $\mathbb{Z}$).

Model checking

- Mais relevante na prática e o problema de *verificação de modelos* (model checking problem):

  Instância Uma estrutura $\mathcal{M}$ e uma fórmula Φ.

  Problema Decide se existe uma atribuição a tal que $\mathcal{M} \models_a \Phi$.

- Esse problema permite uma solução direta em $O(|\Phi| \cdot |U|^w \cdot w)$ com w o número máximo de variáveis livres em uma subfórmula do Φ.

Decidir as características...

- Mas então, como decidir características?
- Como na lógica proposicional, a semântica permite resultados negativas: Um contra-exemplo mostra que uma fórmula é falsificável ou uma relação de conseqüência não é correta.
- Para resultados positivos (validade, relação de conseqüência correra) somos obrigados de usar argumentos gerais.
- Uma outra possibilidade é de usar a dedução natural (ou um outro sistema de prova).

Exemplo

$$\forall x(P(x) \to Q(x)) \models (\forall x P(x)) \to (\forall x Q(x))$$

Prova. Seja $\mathcal{M}$ uma alguma estrutura e a alguma atribuição, tal que

$$\mathcal{M} \models_a \forall x(P(x) \to Q(x))$$

Então: Ou $\mathcal{M} \models_a \forall x P(x)$ ou $\mathcal{M} \not\models_a \forall x P(x)$.
Caso $\mathcal{M} \models_a \forall x P(x)$: Para qualquer $u \in U$ temos $\mathcal{M} \models_{a[x\mapsto u]} P(x)$ e $\mathcal{M} \models_{a[x\mapsto u]} P(x) \to Q(x)$. Pela definição da implicação $\mathcal{M} \models_{a[x\mapsto u]} Q(x)$, e logo $\mathcal{M} \models_a \forall x Q(x)$ e também $\mathcal{M} \models_a (\forall x P(x)) \to (\forall x Q(x))$.
Caso $\mathcal{M} \not\models (\forall x P(x))$, pela definição da implicação $\mathcal{M} \models (\forall x P(x)) \to (\forall x Q(x))$.
■

Comparação

Lógica proposicional
Proposições $p, q, \ldots$
Fórmula $p \vee q$
Atribuição (=Interpretação): Função Atom $\to \mathbb{B}$
Atribuição t.q. $p \vee q$ é correto (modelo): $\mathcal{M} = \{p \mapsto v, q \mapsto f\}$
Atribuição t.q. $p \vee q$ não é correto: $\mathcal{M} = \{p \mapsto f, q \mapsto f\}$

Lógica de predicados
Predicados $P(x), \ldots$
Fórmula $\exists x P(x)$
Estrutura: $U, f^{\mathcal{M}}, P^{\mathcal{M}}$.
Atribuição: Função $a : \mathcal{V} \to U$
Interpretação: Estrutura + atribuição.
Interpretação t.q. $\exists x P(x)$ é correto (modelo): $U = \{a\}, P^{\mathcal{M}} = U$
Interpretação t.q. $\exists x P(x)$ não é correto: $U = \{a\}, P^{\mathcal{M}} = \emptyset$

3.4. Teoria de provas

As regras da lógica proposicional são consistentes na lógica de predicados também. Por isso, a teoria das provas da lógica de predicados *estende* a lógica proposicional com regras para (i) a identidade e (ii) os quantificadores.

Motivação

- Queremos *estender* o sistema de dedução natural da lógica de predicados.
- O que falta? Ao menos regras para identidade e os quantificadores.
- Precisamos regras para termos também?
- Não, na dedução provamos *conseqüências independente da estrutura.*
- Logo, os constantes, funções e predicados não tem interpretação.

Aplicação das regras da lógica proposicional

A regras da lógica proposicional se aplicam da mesma maneira:

1	$P(c) \to Q(c)$	premissa
2	$P(c) \lor \neg P(c)$	LEM
3	$P(c)$	hipótese
4	$Q(c)$	$\to_{\mathrm{e}}$ 3,1
5	$\neg P(c) \lor Q(c)$	$\lor_{\mathrm{i}_2}$ 4
6	$\neg P(c)$	hipótese
7	$\neg P(c) \lor Q(c)$	$\lor_{\mathrm{i}_1}$ 6
8	$\neg P(c) \lor Q(c)$	$\lor_{\mathrm{e}}$ 2,3–5,6–7

Obviamente todo termo t é igual a si mesmo. Além disso, as regras para a identidade tem que capturar a idéia de "substituir iguais para iguais". Com $t_1 = t_2$ queremos substituir uma ou mais ocorrências de t_1 com t_2. Por exemplo a partir da fórmula $R(t_1, t_2) \land Q(t_1)$ regras para a identidade tem que permitir de provar todas as fórmulas

$$R(t_2, t_2) \land Q(t_1); \quad R(t_1, t_2) \land Q(t_2); \quad R(t_2, t_2) \land Q(t_2).$$

Regras para a identidade

- O que seriam regras certas para identidade?
- Queremos que $t = t$ para qualquer termo. Logo, introduzimos

$$\frac{}{t = t} =_{\mathrm{i}}$$

- É isso? Como deduzir a partir de $x = y$, por exemplo, que $P(x) = P(y)$?
- Precisamos uma outra regra, que permite de *substituir* “igual para igual”.

$$\frac{t_1 = t_2 \qquad \Phi[t1/x]}{\Phi[t_2/x]} =_{\mathrm{e}}$$

A abordagem

$$\frac{t_1 = t_2 \qquad \Phi}{\Phi[t_2/t_1]} =_{\mathrm{e}}$$

não funciona por duas razões: (i) Tecnicamente, não definimos a substituição de termos para termos. (ii) Essa regra permitiria somente a substituição de *todas* instâncias de t_1 para t_2. A regra acima captura melhor nossa intuição. Na fórmula original, as ocorrências de t_1 que nos queremos substituir com t_2 foram substituídos com uma (nova) variável x e a regra permite de substituir essas ocorrências com t_2. (Imagina x como ela “marca” as ocorrências que nos queremos substituir.) Logo, no exemplo acima um Φ adequado nos três casos seria

$$R(x, t_2) \land Q(t_1); \quad R(t_1, t_2) \land Q(x); \quad R(x, t_2) \land Q(x).$$

Exemplo: Comutatividade
$x = y \vdash y = x$?

1	$x = y$	premissa
2	$x = x$	$=_{\mathrm{i}}$
3	$y = x$	$=_{\mathrm{e}}$ 1,2 ($\Phi \equiv s = x$)

Compare com

$$\frac{x = y \qquad (s = x)[x/s]}{(s = x)[y/s]} =_{\mathrm{e}}$$

e observe que $(s = x)[x/s] \equiv x = x$ (linha 2) e $(s = x)[y/s] \equiv y = x$ (linha 3). Receita: Dado uma fórmula Ψ. Se quero substituir algumas ocorrências de um termo t_1 para t_2, uma fórmula Φ adequada é Ψ com a substituição de uma nova variável x^* para esses ocorrências.

Exemplo: Transitividade
$x = y, y = z \vdash x = z$?

1	$x = y$	premissa
2	$y = z$	premissa
3	$x = z$	$=_{\mathrm{e}}$, $(\Phi \equiv x = y)$

Observações

- A aplicação da regra $=_{\mathrm{e}}$ fica mais claro, se notamos a fórmula Φ junto com a justificação.
- Notação: $\Phi \equiv \cdots$.
- Observe: As substituições em $=_{\mathrm{e}}$ são permitidas só se eles não ligam variáveis livres.
- Prova *errada*:

1	$y = x$	premissa
2	$\forall x P(y)$	premissa
3	$\forall x P(x)$	$=_{\mathrm{e}}$ 1,2 $(\Phi \equiv \forall x P(y))$

Quantificação universal: Idéia

- O que seriam regras adequadas para a quantificação universal?
- Comparando com a lógica proposicional a quantificação universal é parecido com uma conjunção:

$$\forall x P(x) \qquad P(a) \wedge P(b) \wedge P(c) \wedge \cdots$$

- Logo, é provável que uma generalização das regras para $\wedge$ serve para $\forall$.

Eliminação

- A eliminação da conjunção permite concluir ambos lados de conjunção.
- Logo, para a eliminação da quantificação universal, permitimos uma *instância* com qualquer termo:

$$\frac{\forall x \Phi}{\Phi[t/x]} \; \forall x\mathrm{e}$$

Exemplo
$P(t), \forall x P(x) \rightarrow \neg Q(x) \vdash \neg Q(t)$?

1	$P(t)$	premissa
2	$\forall x P(x) \rightarrow \neg Q(x)$	premissa
3	$P(t) \rightarrow \neg Q(t)$	$\forall x$e 2 $(t \equiv t)$
4	$\neg Q(t)$	$\rightarrow_{\mathrm{e}}$ 1,3

Introdução: Idéia

- A introdução da conjunção tem duas premissas.
- Para introduzir uma quantificação universal, isso significa problemas:
  1. Temos provavelmente uma conjunto infinito das premissas e
  2. Pior, como a dedução natural se aplica para qualquer estrutura, não temos um universo concreto, e não conhecemos (concretamente) todas premissas.
- Suponhe um objeto qualquer x_0 sem características especiais e prove uma característica $P(x_0)$ de x_0.
- Se essa prova não depende do x_0 particular, podemos concluir que ela se aplica para qualquer x_0.
- Logo, conclui-mos $\forall x P(x)$.

Introdução: Regra

- Esse raciocínio justifique a regra

$$\frac{\boxed{\begin{array}{l} x_0 \\ \quad\vdots \\ \Phi[x_0/x] \end{array}}}{\forall x \Phi}\ \forall x\mathrm{i}$$

Exemplo
$\forall x P(x) \rightarrow Q(x), \forall x P(x) \vdash \forall x Q(x)$?

1	$\forall x P(x) \rightarrow Q(x)$	premissa
2	$\forall x P(x)$	premissa
3	$\mathbf{x_0}$	(qualquer x_0)
4	$P(x_0)$	$\forall x$e 2 ($t \equiv x_0$)
5	$P(x_0) \rightarrow Q(x_0)$	$\forall x$e 1 ($t \equiv x_0$)
6	$Q(x_0)$	$\rightarrow_{\mathrm{e}}$ 4,5
7	$\forall x Q(x)$	$\forall x$i 3–6

Observações

- Para a prova ser válido para qualquer x_0, x_0 *não deve ocorrer em qualquer lugar fora da caixa*!
- Como a conclusão da caixa é $\Phi[x_0/x]$, ela *não contém um x livre*!
- Prova *errada*:

1	$P(x)$	premissa
2	$\mathbf{x_0}$	(qualquer x_0)
3	$P(x)$	cópia 1 (*erro: contém x livre!*)
4	$\forall x P(x)$	$\forall x$i 2–3

- Prova *correta*:

1	$P(x)$	premissa
2	$\mathbf{x_0}$	(qualquer x_0)
3	$P(x)$	cópia 1 (*ok: não contém y livre!*)
4	$\forall y P(x)$	$\forall x$i 2–3

Quantificação existencial: Idéia

- O que seriam regras adequadas para a quantificação existencial?
- Comparando com a lógica proposicional a quantificação existencial é parecido com uma disjunção:

$$\exists x P(x) \qquad P(a) \vee P(b) \vee P(c) \vee \cdots$$

- Com a mesma abordagem da quantificação universal, é provável que uma generalização das regras para $\vee$ serve para $\exists$.

Introdução

- Para introduzir uma disjunção, é suficiente de provar um parte da fórmula.
- Logo, se conseguimos de provar um caso particular, a introdução de um quantificador existencial é justificada.

$$\frac{\Phi[t/x]}{\exists x \Phi}\ \exists x\mathrm{i}$$

Exemplo

$\forall x P(x) \vdash \exists x P(x)$?

1	$\forall x P(x)$	premissa
2	$P(y)$	$\forall x$e 1 $(t \equiv y)$
3	$\exists x P(x)$	$\exists x$i 2 $(t \equiv y)$

Eliminação

- Como eliminar uma quantificação universal $\exists x \Phi$?
- Em comparação com $\vee_\mathrm{e}$, temos que provar uma fórmula χ.
- Para isso, podemos assumir que para algum x_0 a fórmula Φ é correto.
- Se a prova de χ não depende do x_0 particular, podemos concluir que ela se aplica para qualquer x_0 (em particular o x_0 certo).
- Esse raciocínio justifique a regra

$$\dfrac{\exists x\Phi \qquad \boxed{\begin{array}{ll} x_0 & \Phi[x_0/x] \\ & \vdots \\ & \chi \end{array}}}{\chi}\ \exists x\mathrm{e}$$

Exemplo

$\exists x P(x), \exists x P(x) \to Q(x) \vdash \exists x Q(x)$?

Não! (Contra-exemplo?)

Por exemplo $\mathcal{U} = \{a, b\}, P = \{P, Q\}, P^{\mathcal{M}} = \{a\}, Q^{M} = \emptyset$.

Exemplo

$\exists x P(x), \forall x P(x) \to Q(x) \vdash \exists x Q(x)$?

1	$\exists x P(x)$	premissa
2	$\forall x P(x) \to Q(x)$	premissa
3	$\mathbf{x_0}$ $P(x_0)$	hipótese (para $\exists x$e)
4	$P(x_0) \to Q(x_0)$	$\forall x$e 2 $(t \equiv x_0)$
5	$Q(x_0)$	$\to_\mathrm{e}$ 3,4
6	$\exists x Q(x)$	$\exists x$i 5 $(t \equiv x_0)$
7	$\exists x Q(x)$	$\exists x$e

Observações

- As regras $\forall x$e, $\forall x$i, $\exists x$e, $\exists x$i se aplicam para qualquer variável!
- $\forall x$i não tem hipótese no começo! Se a prova precisa uma hipótese, se abre mais uma caixa!
- Em comparação $\exists x$i tem (que ter!) uma hipótese!

3.4.1. Teoremas importantes

Teoremas (1)

- Negação da quantificação

$$\neg\forall x P(x) \dashv\vdash \exists x \neg P(x) \tag{nu}$$
$$\neg\exists x P(x) \dashv\vdash \forall x \neg P(x) \tag{ne}$$

- Comutação da quantificação

$$\forall x \forall y P(x,y) \dashv\vdash \forall y \forall x P(x,y) \tag{cu}$$
$$\exists x \forall y P(x,y) \vdash \forall y \exists x P(x,y) \tag{cue}$$
$$\exists x \exists y P(x,y) \dashv\vdash \exists y \exists x P(x,y) \tag{ce}$$

Teoremas (2)

- Distribuição da quantificação sobre $\wedge$ e $\vee$

$$\exists x(P(x) \vee Q(x)) \dashv\vdash (\exists x P(x)) \vee (\exists x Q(x)) \tag{ded}$$
$$\forall x(P(x) \wedge Q(x)) \dashv\vdash (\forall x P(x)) \wedge (\forall x Q(x)) \tag{duc}$$
$$\exists x(P(x) \wedge Q(x)) \vdash (\exists x P(x)) \wedge (\exists x Q(x)) \tag{dec}$$
$$\forall x(P(x) \vee Q(x)) \dashv (\forall x P(x)) \vee (\forall x Q(x)) \tag{dud}$$

- Distribuição da quantificação sobre $\rightarrow$

$$\exists x(P(x) \rightarrow Q(x)) \dashv\vdash (\forall x P(x)) \rightarrow (\exists x Q(x)) \tag{di1}$$
$$(\exists x P(x)) \rightarrow (\forall x Q(x)) \vdash \forall x(P(x) \rightarrow Q(x)) \tag{di2}$$
$$\forall x(P(x) \rightarrow Q(x)) \vdash (\forall x P(x)) \rightarrow (\forall x Q(x)) \tag{di3}$$
$$\forall x(P(x) \rightarrow Q(x)) \vdash (\forall x P(x)) \rightarrow (\exists x Q(x)) \tag{di4}$$
$$\forall x(P(x) \rightarrow Q(x)) \vdash (\exists x P(x)) \rightarrow (\exists x Q(x)) \tag{di5}$$

Negação

Teorema nu, $\vdash$

1	$\neg\forall x P(x)$	premissa
2	$\neg\exists x \neg P(x)$	hipótese
3	$\mathbf{x_0}$	(qualquer x_0)
4	$\neg P(x_0)$	hipótese
5	$\exists x \neg P(x)$	$\exists x$i $(t \equiv x_0)$
6	$\perp$	$\neg_{\mathrm{e}}$ 5,2
7	$P(x_0)$	PBC 4–6
8	$\forall x P(x)$	$\forall x$i 3–7
9	$\perp$	$\neg_{\mathrm{e}}$ 8,2
10	$\exists x \neg P(x)$	PBC 2–9

Negação

Teorema nu, $\dashv$

1	$\exists x \neg P(x)$	premissa
2	$\forall x P(x)$	hipótese
3	$\mathbf{x_0}$ $\neg P(x_0)$	hipótese
4	$P(x_0)$	$\forall x$e 2
5	$\perp$	$\neg_{\mathrm{e}}$ 4,3
6	$\perp$	$\exists x$e 1,3–5
7	$\neg\forall x P(x)$	$\neg_{\mathrm{i}}$ 2–6

Teorema ne, $\vdash$.

1	$\neg\exists x P(x)$	premissa
2	x_0	(qualquer x_0)
3	$P(x_0)$	hipótese
4	$\exists x P(x)$	$\exists x$i3
5	$\perp$	$\neg_{\mathrm{e}}$4,1
6	$\neg P(x_0)$	$\neg_{\mathrm{i}}$3–5
7	$\forall x \neg P(x)$	$\forall x$i2–6

Teorema ne, $\dashv$.

1	$\forall x \neg P(x)$	premissa
2	$\exists x P(x)$	hipótese
3	$\mathbf{x_0}$ $P(x_0)$	hipótese
4	$\neg P(x_0)$	$\forall x$e1
5	$\perp$	$\neg_{\mathrm{e}}$3,4
6	$\perp$	$\exists x$e2,3–5
7	$\neg\exists x P(x)$	$\neg_{\mathrm{i}}$2–6

Comutatividade

Teorema cu, $\dashv\vdash$

1	$\forall x \forall y P(x,y)$	premissa
2	$\mathbf{y_0}$	(qualquer y_0)
3	$\mathbf{x_0}$	(qualquer y_0)
4	$\forall y P(x_0,y)$	$\forall x$e 1
5	$P(x_0,y_0)$	$\forall y$e 4
6	$\forall x P(x,y_0)$	$\forall x$i 3–5
7	$\forall y \forall x P(x,y)$	$\forall y$i 2–6

Comutatividade

Teorema cue, $\vdash$

1	$\exists x \forall y P(x,y)$	premissa
2	$\mathbf{y_0}$	(qualquer y_0)
3	$\mathbf{x_0}$ $\forall y P(x_0,y)$	hipótese
4	$P(x_0,y_0)$	$\forall y$e 3
5	$\exists x P(x,y_0)$	$\exists x$i 4
6	$\exists x P(x,y_0)$	$\exists x$e 1,3–5
7	$\forall y \exists x P(x,y)$	$\forall y$i 2–6

(Porque a prova do contrário não funciona?)

Comutatividade

Teorema ce, $\dashv\vdash$

1	$\exists x \exists y P(x,y)$	premissa
2	$\mathbf{x_0}$ $\exists y P(x_0,y)$	hipótese
3	$\mathbf{y_0}$ $P(x_0,y_0)$	hipótese
4	$\exists x P(x,y_0)$	$\exists x$i 3
5	$\exists y \exists x P(x,y)$	$\exists y$i 4
6	$\exists y \exists x P(x,y)$	$\exists y$e 2,3–5
7	$\exists y \exists x P(x,y)$	$\exists x$e 1,2–6

Distribuição sobre $\vee$

Teorema ded, $\vdash$

1	$\exists x(P(x) \vee Q(x))$	premissa
2	$\mathbf{x_0}$ $P(x_0) \vee Q(x_0)$	hipótese
3	$P(x_0)$	hipótese
4	$\exists x P(x)$	$\exists x$i 3
5	$(\exists x P(x)) \vee (\exists x Q(x))$	$\vee_{\mathrm{i}_1}$ 4
6	$Q(x_0)$	hipótese
7	$\exists x Q(x)$	$\exists x$i 6
8	$(\exists x P(x)) \vee (\exists x Q(x))$	$\vee_{\mathrm{i}_2}$ 7
9	$(\exists x P(x)) \vee (\exists x Q(x))$	$\vee_{\mathrm{e}}$ 2,3–5,6–8
10	$(\exists x P(x)) \vee (\exists x Q(x))$	$\exists x$e 1,2–9

Distribuição sobre $\vee$

Teorema ded, $\dashv$

1	$(\exists x P(x)) \vee (\exists x Q(x))$	premissa
2	$\exists x P(x)$	hipótese
3	$\mathbf{x_0}$ $P(x_0)$	hipótese
4	$P(x_0) \vee Q(x_0)$	$\vee_{\mathrm{i}_1}$ 3
5	$\exists x(P(x) \vee Q(x))$	$\exists x$i 4
6	$\exists x(P(x) \vee Q(x))$	$\exists x$e 2,3–5
7	$\exists x Q(x)$	hipótese
8	$\mathbf{x_0}$ $Q(x_0)$	hipótese
9	$P(x_0) \vee Q(x_0)$	$\vee_{\mathrm{i}_2}$ 8
10	$\exists x(P(x) \vee Q(x))$	$\exists x$i 9
11	$\exists x(P(x) \vee Q(x))$	$\exists x$e 7,8–10
12	$\exists x(P(x) \vee Q(x))$	$\vee_{\mathrm{e}}$ 1,2–6,7–11

Distribuição sobre $\wedge$

Teorema dec, $\vdash$

1	$\exists x(P(x) \wedge Q(x))$	premissa
2	$\mathbf{x_0}$ $P(x_0) \wedge Q(x_0)$	hipótese
3	$P(x_0)$	$\wedge_{\mathrm{e}_1}$ 2
4	$Q(x_0)$	$\wedge_{\mathrm{e}_2}$ 2
5	$\exists x P(x)$	$\forall x$i 3
6	$\exists x Q(x)$	$\forall x$i 4
7	$(\exists x P(x)) \wedge (\exists x Q(x))$	$\wedge_{\mathrm{i}}$ 5,6
8	$(\exists x P(x)) \wedge (\exists x Q(x))$	$\exists x$e 1,2–7

Distribuição sobre $\rightarrow$

Teorema di3, $\vdash$

1	$\forall x(P(x) \to Q(x))$	premissa
2	$\forall x P(x)$	hipótese
3	$\mathbf{x_0}$	(qualquer x_0)
4	$P(x_0)$	$\forall x$e 2
5	$P(x_0) \to Q(x_0)$	$\forall x$e 1
6	$Q(x_0)$	$\to_{\mathrm{e}}$ 4,5
7	$\forall x Q(x)$	$\forall x$i 3–6
8	$(\forall x P(x)) \to (\forall x Q(x))$	$\to_{\mathrm{i}}$ 2–8

Distribuição sobre $\to$

Teorema di4, $\vdash$

1	$\forall x(P(x) \to Q(x))$	premissa
2	$\forall x P(x)$	hipótese
3	$P(x)$	$\forall x$e 2 $(t \equiv x)$
4	$P(x) \to Q(x)$	$\forall x$e 1 $(t \equiv x)$
5	$Q(x)$	$\to_{\mathrm{e}}$ 3,4
6	$\exists x Q(x)$	$\exists x$i 5 $(t \equiv x)$
7	$(\forall x P(x)) \to (\exists x Q(x))$	$\to_{\mathrm{i}}$ 2–6

Distribuição sobre $\to$

Teorema di5, $\vdash$

1	$\forall x(P(x) \to Q(x))$	premissa
2	$\exists x P(x)$	hipótese
3	$\mathbf{x_0}$ $P(x_0)$	hipótese
4	$P(x_0) \to Q(x_0)$	$\forall x$e 1
5	$Q(x_0)$	$\to_{\mathrm{e}}$ 3,4
6	$\exists x Q(x)$	$\exists x$i 5
7	$\exists x Q(x)$	$\exists x$e 2,3–6
8	$(\exists x P(x)) \to (\exists x Q(x))$	$\to_{\mathrm{i}}$ 2–7

Renomear variáveis

1	$\forall x P(x)$	premissa
2	$\mathbf{x_0}$	(qualquer x_0)
3	$P(x_0)$	$\forall x$e 1
4	$\forall y P(y)$	$\forall y$i 2–3 $(t \equiv x_0)$

1	$\exists x P(x)$	premissa
2	$\mathbf{x_0}$ $P(x_0)$	hipótese
3	$\exists y P(y)$	$\exists x$i 2 $(t \equiv x_0)$
4	$\exists y P(y)$	$\exists x$e 1,2–3

3.4.2. Árvores de refutação

Árvores de refutação: Algoritmo

Para testar um seqüente, procedemos assim:

T1. Init Construi uma árvore inicial, que consiste em um ramo só. Cada premissa e a negação da conclusão é um nó.

T2. Expansão Enquanto existe uma fórmula, que não foi expandida seguinda as regras, expande ela (e marca ela "expandida").

T3. Inválido? Se um ou mais ramos são consistentes: Imprime "O argumento não é válido" e para.

T4. Válido? (Aqui, todos ramos são inconsistentes) Imprime "O argumento é válido" e para.

Extensões

- Como estender as árvores de refutação para a lógica de predicados?
- Ao invés de parar nas literais, vamos parar com predicados.
- Parecido com dedução natural, precisamos novas regras para
  - os quantificadores e
  - a identidade.

Regras para a quantificação universal

$$\begin{tikzcd} \neg\forall x\Phi \arrow[d] & \forall x\Phi \arrow[d] \\ \exists x\neg\Phi & \Phi[t/x] \end{tikzcd}$$

- No caso da negação: Estendemos cada ramo com a quantificação existencial correspondente e marcamos a fórmula.
- No caso da afirmação:
  - Estendemos cada ramo com um caso particular.
  - t pode ser qualquer termo com constantes que *já ocorrem no ramo*.
  - *Não marcamos* $\forall x\Phi$: A fórmula pode ser aplicada várias vezes!

Exemplo: Quantificação universal

$P(a), \forall x P(x) \to \neg Q(x) \vdash \neg Q(a)?$

1		$P(a)$		premissa
2		$\forall x P(x) \to \neg Q(x)$		premissa
3		$\neg\neg Q(a)$		negação da conclusão
4		$Q(a)$		$\neg\neg 3$
5		$P(a) \to \neg Q(a)$		$\forall 2$
6	$\neg P(a)$		$\neg Q(a)$	$\to 5$
6	$\times$		$\times$	

Exemplo: Quantificação universal

$\forall x P(x) \to Q(x), \forall x P(x) \vdash \forall x Q(x)?$

1		$\forall x P(x) \to Q(x)$		premissa
2		$\forall x P(x)$		premissa
3		$\neg \forall x Q(x)$		negação da conclusão
4		$\exists x \neg Q(x)$		$\neg\forall 3$
5		$\neg Q(a)$		$\exists 4$
6		$P(a)$		$\forall 2$
7		$P(a) \to Q(a)$		$\forall 1$
8	$\neg P(a)$		$Q(a)$	$\to 7$
9	$\times$		$\times$	

Exemplo: Quantificação universal

$\forall x P(x) \to Q(x), \forall x P(x) \vdash \forall x Q(x)?$

1	$\forall x P(x) \to Q(x)$		premissa
2	$\forall x P(x)$		premissa
3	$\neg\forall x Q(x)$		negação da conclusão
4	$\exists x \neg Q(x)$		$\neg\forall 3$
5	$\neg Q(a)$		$\exists 4$
6	$P(a)$		$\forall 2$
7	$P(a) \to Q(a)$		$\forall 1$
8	$\neg P(a)$	$Q(a)$	$\to 7$
9	$\times$	$\times$	

Regras para a quantificação existencial

$$
\begin{array}{ccc}
\neg\exists x\Phi & & \exists x\Phi \\
\downarrow & & \downarrow \\
\forall x\neg\Phi & & \Phi[t/x]
\end{array}
$$

- No caso da negação: Estendemos cada ramo com a quantificação universal correspondente e marcamos a fórmula.
- No caso da afirmação:
  - Estendemos cada ramo com um caso particular.
  - t tem que ser um termo com constantes completamente novas!
  - Depois marcamos a fórmula (usa só uma vez!).

Exemplo: Quantificação existencial
$\forall x P(x) \vdash \exists x P(x)$?

1	$\forall x P(x)$	premissa
2	$\neg\exists x P(x)$	negação da conclusão
3	$\forall x \neg P(x)$	$\neg\exists 3$
4	$P(a)$	$\forall 1$
5	$\neg P(a)$	$\forall 3$
6	$\times$	

Exemplo: Quantificação existencial
$\exists x P(x), \forall x P(x) \to Q(x) \vdash \exists x Q(x)$?

1		$\exists x P(x)$		premissa
2		$\forall x P(x) \to Q(x)$		premissa
3		$\neg\exists x Q(x)$		negação da conclusão
4		$\forall x \neg Q(x)$		$\neg\exists 3$
5		$P(a)$		$\exists 1$
6		$P(a) \to Q(a)$		$\forall 2$
7		$\neg Q(a)$		$\forall 4$
8	$\neg P(a)$		$Q(a)$	$\to 6$
9	$\times$		$\times$	

Regras para a identidade

$$
\begin{array}{c} \neg t = t \\ \times \end{array}
\qquad\qquad
\begin{array}{c} t_1 = t_2 \\ \Phi[t_1/x] \\ \Phi[t_2/x] \end{array}
$$

- Se encontramos uma identidade $t = t$ negada, o ramo fecha.
- Com uma identidade, podemos substituir em alguma fórmula (ainda não marcada) um ou mais ocorrências do lado esquerda de uma identidade com o lado direita.

Exemplos: Identidade
$a = b \vdash b = a$?

1	$a = b$	premissa
2	$\neg(b = a)$	negação da conclusão
3	$\neg(b = b)$	$= 1, 2$
4	$\times$	

Exemplos: Identidade

$a = b, b = c \vdash a = c?$

1	$a = b$	premissa
2	$b = c$	premissa
3	$\neg(a = c)$	negação da conclusão
4	$\neg(b = c)$	$= 1, 3$
5	$\neg(c = c)$	$= 2, 4$
6	$\times$	

Distribuição

Teorema di3: $\forall x(P(x) \to Q(x)) \vdash (\forall x P(x)) \to (\forall x Q(x))?$

1		$\forall x P(x) \to Q(x)$		premissa
2		$\neg((\forall x P(x)) \to (\forall x Q(x)))$		negação da conclusão
3		$\forall x P(x)$		$\neg \to 2$
4		$\neg \forall x Q(x)$		$\neg \to 2$
5		$\exists x \neg Q(x)$		$\neg\forall 4$
6		$\neg Q(a)$		$\exists 5$
7		$P(a)$		$\forall 3$
8		$P(a) \to Q(a)$		$\forall 1$
9	$\neg P(a)$		$Q(a)$	$\to 8$
	$\times$		$\times$	

Distribuição

Teorema dud: $(\forall x P(x)) \vee (\forall x Q(x)) \vdash \forall x P(x) \vee Q(x)?$

1		$(\forall x P(x)) \lor (\forall x Q(x))$		premissa
2		$\neg \forall x P(x) \lor Q(x)$		negação da conclusão
3		$\exists x \neg (P(x) \lor Q(x))$		$\neg\forall 2$
4		$\neg (P(a) \lor Q(a))$		$\exists 3$
5		$\neg P(a)$		$\neg \lor 4$
6		$\neg Q(a)$		$\neg \lor 4$
7	$\forall x P(x)$		$\forall x Q(x)$	$\lor 1$
8	$P(a)$		$Q(a)$	$\forall 7$
9	$\times$		$\times$	

Negação da quantificação
Teorema nu, $\dashv$: $\exists x \neg P(x) \vdash \neg \forall x P(x)$?

1	$\exists x \neg P(x)$	premissa
2	$\neg\neg \forall x P(x)$	negação da conclusão
3	$\forall x P(x)$	$\neg\neg 2$
4	$\neg P(a)$	$\exists 1$
5	$P(a)$	$\forall 3$
6	$\times$	

- Observe que $\neg \forall x P(x) \vdash \exists x P(x)$ já faz parte das regras!

Comutação
(cu): $\forall x \forall y P(x,y) \vdash \forall y \forall x P(x,y)$?

1	$\forall x \forall y P(x,y)$	premissa
2	$\neg \forall y \forall x P(x,y)$	negação da conclusão
3	$\exists y \neg \forall x P(x,y)$	$\neg\forall 2$
4	$\neg \forall x P(x,a)$	$\exists 3$
5	$\exists x \neg P(x,a)$	$\neg\forall 4$
6	$\neg P(b,a)$	$\exists 5$
7	$\forall y P(b,y)$	$\forall 1$
8	$P(b,a)$	$\forall 7$
9	$\times$	

Comutação

(cue): $\exists x \forall y P(x,y) \vdash \forall y \exists x P(x,y)$?

1	$\exists x \forall y P(x,y)$	premissa
2	$\neg\forall y \exists x P(x,y)$	negação da comutação
3	$\exists y \neg\exists x P(x,y)$	$\neg\forall 2$
4	$\neg\exists x P(x,a)$	$\exists 3$
5	$\forall x \neg P(x,a)$	$\neg\exists 4$
6	$\forall y P(b,y)$	$\exists 1$
7	$\neg P(b,a)\forall 5$	
8	$P(b,a)$	$\forall 6$
9	$\times$	

Características

- Como na lógica proposicional, as árvores de refutação são consistentes é completos ao respeito da semântica.
- Se uma árvore não fecha, podemos encontrar contra-exemplos.
- Do outro lado, vimos que a lógica de predicados não mais é decidível.
- Esse fato, tem conseqüências para árvores de refutação também:
  - As árvores não produzem mais todos os contra-exemplos.
  - Existem árvores infinitos.

Considere um seqüente $\Phi_1, \ldots, \Phi_n \vdash \Psi$. Se a árvore de refutação correspondente fecha, pela consistência o seqüente é válido. Do outro lado, caso um seqüente é válido, a completude garante que existe um árvore de refutação que fecha. De fato, na lógica proposicional qualquer árvore fechou, uma característica que não na lógica de predicados não mais é verdadeira.

Exemplo 8

Considere o seqüente

$$\forall x P(x), \forall x Q(x) \vdash \forall x P(x).$$

Podemos construir uma árvore de refutação da forma

1	$\forall x P(x)$	premissa
2	$\forall x Q(x)$	premissa
3	$\neg\forall x P(x)$	negação da conclusão
4	$\exists x \neg P(x)$	$\neg\forall 3$
5	$\neg P(a)$	$\exists 4$
6	$Q(a)$	$\forall 2$
7	$Q(a')$	$\forall 2$
8	$Q(a'')$	$\forall 2$
9	$\cdots$	

que não termina. $\Diamond$

O problema nessa caso é simples de detectar: $\forall x P(x)$ (que levaria imediatamente a uma contradição) nunca foi usada. Esse tipo de problemas pode ser evitado construindo árvores sistematicamente:

- Sempre aplica a regra para o nó mais alto não marcado.
- Marca todos nós, inclusive $\forall$.
- Caso a regra $\forall$ foi aplicada, depois de marcar cópia o nó para baixo do todo ramo ainda em aberto.

Contra-exemplo: Distribuição

Teorema dud, contrário: $\forall x P(x) \vee Q(x) \vdash (\forall x P(x)) \vee (\forall x Q(x))$?

1		$\forall x P(x) \lor Q(x)$			premissa
2		$\neg((\forall x P(x)) \lor (\forall x Q(x)))$			negação da conclusão
3		$\neg\forall x P(x)$			$\neg \lor 2$
4		$\neg\forall x Q(x)$			$\neg \lor 2$
5		$\exists x \neg P(x)$			$\neg\forall 3$
6		$\exists x \neg Q(x)$			$\neg\forall 4$
7		$\neg P(a)$			$\exists 5$
8		$\neg Q(b)$			$\exists 6$
9		$P(a) \lor Q(a)$			$\forall 1$
10		$P(b) \lor Q(b)$			$\forall 1$
11	$P(a)$		$Q(a)$		$\lor 9$
12	$\times$		$P(b)$	$Q(b)$	$\lor 10$
13			$\odot$	$\times$	

Contra-exemplo: Comutação
Teorema cue, contrário: $\forall y \exists x P(x,y) \vdash \exists x \forall y P(x,y)$?

1	$\forall y \exists x P(x,y)$	premissa
2	$\neg\exists x \forall y P(x,y)$	negação da conclusão
3	$\forall x \neg\forall y P(x,y)$	$\neg\exists 2$
4	$\neg\forall y P(a,y)$	$\forall 3$
5	$\exists y \neg P(a,y)$	$\neg\forall 4$
6	$\neg P(a,b)$	$\exists 5$
7	$\exists x P(x,b)$	$\forall 1$
8	$P(c,b)$	$\exists 7$
9	$\odot$	

Heurísticas

- Em geral a seguinte ordem de aplicar as regras resulta em menos trabalho:
    1. Regras da lógica proposicional, que não aumentam o número de ramos.

2. Regras para a negação de quantificadores.
3. A regra do quantificador existencial.
4. Regras da lógica proposicional, que aumentam o número de ramos.
5. A regra do quantificador universal.

3.5. Adequação e decibilidade

Na seção 2.5 vimos que a sistema de dedução natural é adequado, i.e. consistente e complete, para a lógica proposicional. Também vimos que a validade de fórmulas é decidível, mesmo os algoritmos de decisão conhecidos sendo não *eficiente*: o problema SAT de decidir a satisfatibilidade é NP-completo (e pela dualidade, a questão da validade também).
Essas perguntas são ainda mais do interesse para a lógica de predicados, porque ela é suficientemente poderoso para formalizar a matemática. A lógica de predicados essencialmente adiciona objetos, predicados (funções lógicas) e quantificação à lógica proposicional. Um passo intermediário é considerar somente quantificações: queremos decidir as fórmulas verdadeiras da linguagem

$$\{Q_1x_1 \ldots Q_nx_n\Phi(x_1,\ldots,x_n) | Q_i \in \{\forall,\exists\}, \Phi \in \mathcal{L}\}.$$

Esse problema se chama *fórmulas booleanas quantificadas* (ingl. quantified Boolean formulas, QBF). Observe que QBF contém SAT como sub-problema: para uma fórmula Φ temos que decidir o seu fecho existencial. Um algoritmo é possível: podemos testar recursivamente todas atribuições:

```
ISTRUE(Q₁x₁ ... Qₙxₙ Φ(x₁,...,xₙ)) ≡
 if (n = 0) then
   return Φ ≡ 1
 else if (Q₁ = ∀) then
   return
     ISTRUE(Q₂x₂ ... Qₙx_N Φ(0,x₂,...,xₙ)) and
     ISTRUE(Q₂x₂ ... Qₙx_N Φ(1,x₂,...,xₙ))
 else
     ISTRUE(Q₂x₂ ... Qₙx_N Φ(0,x₂,...,xₙ)) or
     ISTRUE(Q₂x₂ ... Qₙx_N Φ(1,x₂,...,xₙ))
 endif
```

Em comparação com o problema NP-completo SAT, nesse caso descobrimos que o problema é PSPACE-completo, que não são boas perspectivas para a lógica de predicados, que começamos de estudar agora.

Teorema 7 (Gödel, Completude da lógica de predicados)
Existe um sistema de prova completa da lógica de predicados [4, 5] (a saber, o sistema de Hilbert). Nesse sistema, para um conjunto Φ de fórmulas (que pode ser contável) é uma fórmula ϕ, temos

$$\Phi \vdash \phi \Longleftrightarrow \Phi \models \phi$$

Gödel provou que teorema 7 se aplica ao sistema de Hilbert, mas ele se aplica a outras formalizações equivalentes, em particular, ao sistema de Gentzen. Ambos sistemas também são consistentes. É um fato importante, que o teorema fala explicitamente da lógica da primeira ordem. A lógica da segunda ordem não tem um teorema da completude.
Observe, que o teorema não implica a decibilidade da lógica da primeira ordem: Church mostrou em 1936

Teorema 8 (Church, Indecibilidade da lógica de predicados)
A validade de sentenças na lógica de predicados não é decidível.

Do outro lado, como provas são objetos finitas, podemos "listar" todas as provas possíveis uma depois da outra. Se o seqüente é válido, a completude garante que uma prova existe. Assim, testando cada prova possível, depois um número finito de passos vamos achar a prova desejada. Mas caso não existe uma prova, esse algoritmo não termina. Em termos da teoria da computação, a validade de fórmulas da lógica de primeira ordem é *semi-decidível*, ou seja as sentenças válidas são *computávelmente enumeráveis*.
O problema de decidir a validade de sentenças fica indecidível mesmo quando elas (i) contém somente predicados binários, uma constante é uma função unária ou (ii) são puras (não contém funções ou constantes). Esses resultados não implicam, que qualquer característica da lógica da primeira ordem é indecidível.

Exemplo 9 (Problemas decidíveis na lógica de predicados)

1. A validade da lógica monádica (somente predicados unários) da primeira ordem é decidível. Mesmo assim, o problema é realmente complicado: ele é NEXPTIME-completo, que temos que considerar como intratável[1].

2. A validade de fórmulas puras em forma normal prenex conjuntiva com prefixo da forma $\forall^*\exists^*$, $\forall^*\exists\forall^*$ ou $\forall^*\exists\exists\forall^*$ é decidível.

$\Diamond$

[1] A classe de problemas NEXPTIME é demonstrávelmente maior que NP.

3.6. Tópicos

3.6.1. Formalização

Notação de conjuntos

- Semanticamente, um predicado unário é equivalente a um subconjunto do universo.
- Isso justifica a *abreviação*

$$x \in P =_{\text{def}} P(x)$$

- Exemplos:

$$\forall x\, x \in P = \forall x P(x)$$
$$\exists x\, x \in P = \exists x P(x)$$

Abreviação de critérios

- Temos os seguintes formlizações comums
  - Todos objetos com característica P satisfazem Φ

$$\forall x P(x) \to \Phi$$

  - Existe um objeto com característica P que satisfaz Φ

$$\exists x P(x) \land \Phi$$

- Isso motiva a *abreviação*

$$\forall x \in P : \Phi =_{\text{def}} \forall x P(x) \to \Phi$$
$$\exists x \in P : \Phi =_{\text{def}} \exists x P(x) \land \Phi$$

Exemplos

- A abreviação pode ser aplicada mais que uma vez

$$\begin{aligned}
\forall x \in P : \forall y \in Q : \Phi &= \forall x P(x) \to (\forall y \in Q : \Phi) \\
&= \forall x P(x) \to (\forall y Q(x) \to \Phi) \\
&= \forall x \forall y P(x) \to Q(x) \to \Phi \\
&= \forall x \forall y (P(x) \land Q(x)) \to \Phi
\end{aligned}$$

- Exercício: Traduz

$$\forall x \in P : \exists y \in Q : \Phi$$
$$\exists x \in P : \exists y \in Q : \Phi$$
$$\exists x \in P : \forall y \in Q : \Phi$$

3.7. Exercícios

Exercício 3.1 (Formalização)
Considere os seguintes sentenças na linguagem natural. Para cada um

1. formalize a sentença: Escolhe constantes, variáveis e predicados adequados e acha uma fórmula correspondente na lógica de predicados.
2. desenhe a árvore de parse da fórmula resultante.

(a) Para cada número inteiro existe um número inteiro maior.
(b) Se todo mundo tivesse um Porsche, eu não queria um.
(c) Ou existe um herói que nos salve ou todo mundo vai morrer.
(d) A mãe da mãe da mãe do meu pai é humano.
(e) Uma pessoa ganhou todos prêmios.
(f) Todos os prêmios foram ganhando de alguém.

Exercício 3.2 (Significado de fórmulas)
Intuitivamente, o que significam os seguintes fórmulas da lógica de predicados? Escolhe um significado dos predicados e explica a fórmula na linguagem natural.
(a) $\forall a \forall b((b \geq a) \rightarrow \exists c(a < c \land c < b))$ falando sobre números inteiros.
(b) $\forall x \exists x(\neg P(x))$
(c) $\forall x \forall y(P(x, y) \rightarrow P(y, x))$
(d) $(\forall z\ R(z)) \rightarrow (\exists x m(x) = x)$
(e) $\exists z((\forall x P(x, z)) \lor (\forall x R(x, z)))$
(f) $\exists z R(z, z)$

Exercício 3.3 (Variáveis livres e ligadas)
Quais são os conjuntos de variáveis livres e ligadas das seguintes fórmulas (usando a convenção que x, y, z denotam variáveis, a, b, c constantes, P, Q, R predicados e f, g, h funções).

1. $\exists x \neg Q(c)$

2. $\forall z \neg R(z)$
3. $(P(c) \lor P(a)) \land Q(c, z, b, x)$
4. $\exists x(P(x) \to P(c))$
5. $\forall y \exists x Q(z)$
6. $\forall z Q(x, z, x, z)$

Exercício 3.4 (Fecho)
Qual é o fecho universal das fórmulas do exercício 3.3?

Exercício 3.5 (Substituições)
Qual é o resultado das seguintes substituições?

1. $\Phi[g(c)/x]$ com $\Phi \equiv \forall z \exists y \neg P(y)$
2. $\Phi[g(c)/y]$ com $\Phi \equiv \forall z \exists y \neg P(y)$
3. $\Phi[g(c)/z]$ com $\Phi \equiv \forall z \exists y \neg P(y)$
4. $\Phi[f(z)/y]$ com $\Phi \equiv \forall z(R(b) \lor R(z) \lor Q(y, y, z))$
5. $\Phi[f(x)/y]$ com $\Phi \equiv \forall z(R(b) \lor R(z) \lor Q(y, y, z))$
6. $\Phi[f(x)/z]$ com $\Phi \equiv \forall z(R(b) \lor R(z) \lor Q(y, y, z))$

Exercício 3.6 (Estruturas)

1. Cria uma estrutura da sua família: Escolhe os constantes (pessoas) adequadas, usa as funções "mãe" e "pai" e os predicados "filho/a", "irmão/ã".
2. Da uma estrutura (finito) que defina a semântica das constantes, funções é predicados (usando o entendimento intuitivo dessas relações).

Exercício 3.7 (Estruturas)
Seja P um predicado com dois argumentos. Acha um modelo de

$$\forall x \neg P(x, x)$$

(estrutura tal que a fórmula está correta) e uma estrutura tal que a fórmula não é correta.

Exercício 3.8 (Estruturas)
Para os seguintes seqüentes, acha contra-exemplos (estruturas, tal que o seqüente não é válido).

1. $\forall y \exists x P(x, y) \models \exists x \forall y P(x, y)$
2. $(\exists x P(x)) \wedge (\exists x Q(x)) \models \exists x(P(x) \wedge Q(x))$
3. $\forall x(P(x) \vee Q(x)) \models (\forall x P(x)) \vee (\forall x Q(x))$
4. $\forall x(P(x) \rightarrow Q(x)) \models (\exists x P(x)) \rightarrow (\forall x Q(x))$

Exercício 3.9 (Provas com dedução natural)
Usando dedução natural prove os seguintes seqüentes da lógica de predicados:

1. $P(x) \vdash \exists x P(x)$
2. $\forall x P(x) \vdash \forall x \forall x P(x)$
3. $\exists x P(x), \forall x P(x) \rightarrow Q(x) \vdash \exists x Q(x)$
4. $\exists x P(x) \vdash \neg \forall x \neg P(x)$
5. $\neg \forall x \neg P(x) \vdash \exists x P(x)$
6. $\vdash \forall x P(x) \rightarrow P(x)$
7. $\forall x P(x) \wedge Q(x) \vdash (\forall x P(x)) \wedge (\forall x Q(x))$
8. $(\forall x P(x) \rightarrow Q(x)) \wedge (\forall x Q(x) \rightarrow R(x)) \vdash (\forall x P(x) \rightarrow R(x))$

Exercício 3.10 (Dedução natural)
Prove os seguintes seqüentes usando a dedução natural:

1. $\exists x(S \rightarrow Q(x)) \vdash S \rightarrow \exists x Q(x)$
2. $(\forall x P(x)) \rightarrow S \vdash \exists x(P(x) \rightarrow S)$
3. $\exists x f(x) = x \vdash \exists x f(f(x)) = x$
4. $P(b) \vdash \forall x(x = b \rightarrow P(x))$
5. $\exists x \exists y(H(x, y) \wedge H(y, x)), \neg \exists x H(x, x) \vdash \exists x \exists y \neg(x = y)$
6. $\forall x(P(x) \leftrightarrow x = b) \vdash P(b) \wedge \forall x \forall y(P(x) \wedge P(y) \rightarrow x = y)$
   (Observe que $p \leftrightarrow q$ é uma abreviação para $p \rightarrow q \wedge q \rightarrow p$).

Exercício 3.11 (Árvores de refutação)
Prove os seqüentes do exercício 3.10 usando árvores de refutação.

Exercício 3.12 (Formalização com lógica de predicados)
Formalize os seguintes afirmações usando a lógica de predicados:

1. Todos os retângulos são quadrilaterais.
2. Alguns retângulos são quadrados.
3. Alguns quadrilaterais são quadrados.

Prove usando dedução natural, que os itens 1,2 implicam item 3.

Exercício 3.13 (Dedução natural e árvores de refutação)
Seja $p \otimes q$ uma abreviação para $(p \land \neg q) \lor (\neg p \land q)$. Prove ou mostre um contra-exemplo para os seguintes seqüentes usando o método preferido:

1. $(p \land \neg q) \lor (\neg p \land q) \vdash (p \lor q) \land \neg(p \land q)$
2. $\forall y \exists x P(x, y) \vdash \exists x \forall y P(x, y)$
3. $\forall x(P(x) \otimes Q(x)) \vdash (\forall x P(x)) \otimes (\forall x Q(x))$
4. $(\forall x P(x)) \otimes (\forall x Q(x)) \vdash \forall x(P(x) \otimes Q(x))$
5. $\exists x(P(x) \otimes Q(x)) \vdash (\exists x P(x)) \otimes (\exists x Q(x))$
6. $(\exists x P(x)) \otimes (\exists x Q(x)) \vdash \exists x(P(x) \otimes Q(x))$
7. $\exists x(P(x) \otimes Q(x)) \vdash (\exists x P(x)) \otimes (\forall x Q(x))$
8. $(\exists x P(x)) \otimes (\forall x Q(x)) \vdash \exists x(P(x) \otimes Q(x))$
9. $\forall x(P(x) \otimes Q(x)) \vdash (\exists x P(x)) \otimes (\forall x Q(x))$
10. $(\exists x P(x)) \otimes (\forall x Q(x)) \vdash \forall x(P(x) \otimes Q(x))$

A. Todas regras

A.1. Lógica proposicional

Gramática

A linguagem $\mathcal{L}$ da lógica proposicional é definido em base de um conjunto de proposições atômicas Atom. Com $\Phi, \Psi \in \mathcal{L}$ e $p \in$ Atom a sua gramática é

$$\Phi ::= p|(\neg\Phi)|(\Phi \vee \Psi)|(\Phi \wedge \Psi)|(\Phi \rightarrow \Psi)|\top|\bot$$

Dedução natural

Introdução da conjunção

$$\frac{\Phi \quad \Psi}{\Phi \wedge \Psi}\wedge_{\mathrm{i}}$$

Eliminação da conjunção

$$\frac{\Phi \wedge \Psi}{\Phi}\wedge_{\mathrm{e}_1} \qquad \frac{\Phi \wedge \Psi}{\Psi}\wedge_{\mathrm{e}_2}$$

Introdução da implicação

$$\frac{\boxed{\begin{array}{c}\Phi\\ \vdots\\ \Psi\end{array}}}{\Phi \rightarrow \Psi}\rightarrow_{\mathrm{i}}$$

Eliminação da implicação

$$\frac{\Phi \quad \Phi \rightarrow \Psi}{\Psi}\rightarrow_{\mathrm{e}}$$

Introdução da disjunção

$$\frac{\Phi}{\Phi \vee \Psi}\vee_{\mathrm{i}_1} \quad \frac{\Psi}{\Phi \vee \Psi}\vee_{\mathrm{i}_2}$$

Eliminação da disjunção

$$\frac{\Phi \vee \Psi \quad \boxed{\begin{array}{c}\Phi\\ \vdots\\ \chi\end{array}} \quad \boxed{\begin{array}{c}\Psi\\ \vdots\\ \chi\end{array}}}{\chi}\vee_{\mathrm{e}}$$

Introdução da negação

$$\frac{\boxed{\begin{array}{c}\Phi\\ \vdots\\ \bot\end{array}}}{\neg\Phi}\neg_{\mathrm{i}}$$

Eliminação da negação

$$\frac{\Psi \quad \neg\Psi}{\bot}\neg_{\mathrm{e}}$$

Prova por contradição[12]

$$\frac{\boxed{\begin{array}{c}\neg\Phi\\ \vdots\\ \bot\end{array}}}{\Phi}\mathrm{PBC}$$

Eliminação da contradição

$$\frac{\bot}{\Phi}\bot_{\mathrm{e}}$$

Introdução da negação dupla[1]

$$\frac{\Phi}{\neg\neg\Phi}\ \neg\neg_{\mathrm{i}}$$

Eliminação da negação dupla[2]

$$\frac{\neg\neg\Phi}{\Phi}\ \neg\neg_{\mathrm{e}}$$

Modus tollens[1]

$$\frac{\Phi \to \Psi \quad \neg\Psi}{\neg\Phi}\ \mathrm{MT}$$

Lei do terceiro excluído[1,2]

$$\frac{}{\Phi \vee \neg\Phi}\ \mathrm{LEM}$$

Semântica: Tabelas de verdade

Conjunção

Φ	Ψ	$\Phi \wedge \Psi$
f	f	f
f	v	f
v	f	f
v	v	v

Disjunção

Φ	Ψ	$\Phi \vee \Psi$
f	f	f
f	v	v
v	f	v
v	v	v

Implicação

Φ	Ψ	$\Phi \to \Psi$
f	f	v
f	v	v
v	f	f
v	v	v

Negação

Φ	$\neg\Phi$
f	v
v	f

Falsidade

$\bot$
f

Verdade

$\top$
v

Árvores de refutação

Conjunção

$$\begin{tikzcd}
a \wedge b \arrow[d] \\
a \arrow[d] \\
b
\end{tikzcd}$$

Conjunção negada

$$\begin{tikzcd}
& \neg(a \wedge b) \arrow[dl] \arrow[dr] & \\
\neg a & & \neg b
\end{tikzcd}$$

Negação dupla

$$\begin{tikzcd}
\neg\neg a \arrow[d] \\
a
\end{tikzcd}$$

[1] Regra derivada.
[2] Regra clássica só: não faz parte da lógica intuicionista.

Disjunção

$$\begin{tikzcd}
 & a \lor b \arrow[dl] \arrow[dr] & \\
a & & b
\end{tikzcd}$$

Disjunção negada

$$\begin{tikzcd}
\neg(a \lor b) \arrow[d] \\
\neg a \arrow[d] \\
\neg b
\end{tikzcd}$$

Negação

$$\begin{tikzcd}
\neg a \arrow[d, dotted] \\
a \\
\times
\end{tikzcd}$$

Implicação

$$\begin{tikzcd}
 & a \to b \arrow[dl] \arrow[dr] & \\
\neg a & & b
\end{tikzcd}$$

Implicação negada

$$\begin{tikzcd}
\neg(a \to b) \arrow[d] \\
a \arrow[d] \\
\neg b
\end{tikzcd}$$

A.2. Lógica de predicados

Gramática

$$t ::= v|c|f(t_1, t_2, \ldots, t_n)$$

$$\Phi ::= p(t_1, \ldots, t_n)|(\neg\Phi)|(\Phi \lor \Psi)|(\Phi \land \Psi)|(\Phi \to \Psi)|(\forall v \Phi)|(\exists v \Phi)|\top|\bot$$

Dedução natural

Axioma de identidade

$$\frac{}{t = t} =_{\mathrm{i}}$$

Substituição

$$\frac{t_1 = t_2 \qquad \Phi[t1/x]}{\Phi[t_2/x]} =_{\mathrm{e}}$$

Eliminação da quantificação universal

$$\frac{\forall x \Phi}{\Phi[t/x]} \forall x\mathrm{e}$$

Introdução da quantificação universal

1

$$\frac{\boxed{\begin{array}{l} x_0 \\ \quad\vdots \\ \Phi[x_0/x] \end{array}}}{\forall x \Phi} \forall x\mathrm{i}$$

Eliminação da quantificação existencial[1]

$$\dfrac{\exists x\Phi \qquad \boxed{\begin{matrix} x_0 & \Phi[x_0/x] \\ & \vdots \\ & \chi \end{matrix}}}{\chi}\ \exists x\mathrm{e}$$

Introdução da quantificação existencial

$$\dfrac{\Phi[t/x]}{\exists x\Phi}\ \exists x\mathrm{i}$$

Árvores de refutação

Quantificação universal

$$\begin{array}{c} \forall x\Phi \\ \downarrow \\ \Phi[t/x] \end{array}$$

Se aplica *várias* vezes.
t preferencialmente com constantes existentes.

Quantificação universal negada

$$\begin{array}{c} \neg\forall x\Phi \\ \downarrow \\ \exists x\neg\Phi \end{array}$$

Quantificação existencial

$$\begin{array}{c} \exists x\Phi \\ \downarrow \\ \Phi[t/x] \end{array}$$

Novas constantes em t.

Quantificação existencial negada

$$\begin{array}{c} \neg\exists x\Phi \\ \downarrow \\ \forall x\neg\Phi \end{array}$$

Identidade

$$\dfrac{\begin{matrix} t_1 = t_2 \\ \Phi[t_1/x] \end{matrix}}{\Phi[t_2/x]}$$

Identidade negada

$$\begin{array}{c} \neg t = t \\ \times \end{array}$$

[1] x_0 é uma variável que *ainda não ocorreu* na prova.

B. Soluções dos exercícios

Solução do exercício 2.3.
(b) não é bem formada, porque não é possível de juntar dois $\wedge$. (c) não é bem formada, porque não é possível de construir uma fórmula com uma proposição seguido uma negação. (a) e (d) são bem formadas, mas ambiguas em notação linear: (a) pode significar $p \vee (q \wedge r)$ ou $(p \vee q) \wedge r$, (d) pode significar $(p \rightarrow \neg q \wedge r) \vee \neg\neg s$ ou $(p \rightarrow \neg q) \wedge (r \vee \neg\neg s)$.

Solução do exercício 2.4.

1. $\neg p$, com p "Está chovendo".

2. $p \rightarrow q$, com p "A luz alcançe o corpo" e q "O corpo projeto sombra".

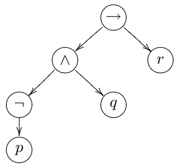
3. p: "Tem interferência de aparelhos rádio-transmissores", q: "Tem diatermia", r: "Aparecem linhas diagonais", s: "Aparecem linhas entrecaladas". $p \vee q \rightarrow r \vee s$.

4. p: "o sinal está fraco", q: "a imagem está com granulação chuviscos", r: "a imagem está com som ruidoso". $p \rightarrow q \wedge r$.

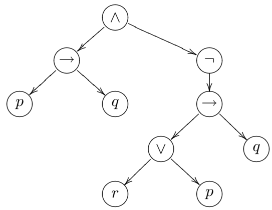
5. p: "Eu consumo o cafézinho", q: "Eu pago o cafézinho", r: "Eu saio.". $(p \rightarrow q) \rightarrow r$.

Solução do exercício 2.5.

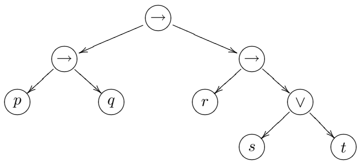
(a)



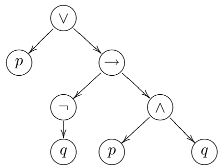
(b)



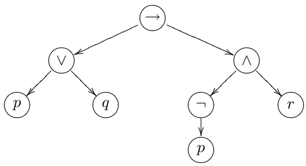
(c)



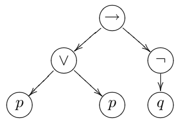
(d)



(e)



(f)



Solução do exercício 2.6.

1. p: O tanque é vazio. q: O carro anda. $p \to \neg q, p \vdash q$. Válido.
2. p: O tanque é vazio. q: O carro anda. $p \to \neg q, q \vdash p$. Não valido.
3. r: O motor funciona. $q \to p \vee r, \neg q, \neg p \vdash \neg r$. Válido.
4. s: O mundo é redondo. t: As pessoas da outro lado cai. $s \to t, \neg t \vdash \neg s$. Válido.

Solução do exercício 2.7.

1.

1	$(p \wedge q) \wedge r$	premissa
2	$s \wedge t$	premissa
3	$p \wedge q$	$\wedge_{e_1}$ 1
4	q	$\wedge_{e_2}$ 3
5	s	$\wedge_{e_1}$ 2
6	$q \wedge s$	$\wedge_{i}$ 4,5

2.

1	p	premissa
2	$p \to q$	hipótese
3	q	$\to_{\mathrm{e}}$ 1,2
4	$(p \to q) \to q$	$\to_{\mathrm{i}}$ 2–3

3.

1	$(p \vee (q \to p)) \wedge q$	premissa
2	$p \vee (q \to p)$	$\wedge_{\mathrm{e}_1}$ 1
3	q	$\wedge_{\mathrm{e}_2}$ 1
4	p	hipótese
5	$q \to p$	hipótese
6	q	$\to_{\mathrm{e}}$ 3,5
7	p	$\vee_{\mathrm{e}}$ 2,4,5–6

4.

1	$p \to (q \vee r)$	premissa
2	$q \to s$	premissa
3	$r \to s$	premissa
4	p	hipótese
5	$q \vee r$	$\to_{\mathrm{e}}$ 4,1
6	q	hipótese
7	s	$\to_{\mathrm{e}}$ 6,2
8	r	hipótese
9	s	$\to_{\mathrm{e}}$ 8,3
10	s	$\vee_{\mathrm{e}}$ 5,6–7,8–9
11	$p \to s$	$\to_{\mathrm{i}}$ 4–10

5.

1	$\neg p \to p$	premissa
2	$\neg p$	hipótese
3	$\perp$	$\neg_{\mathrm{e}}$ 1,2
4	p	PBC 2–3

6.

1	r	premissa
2	$p \to (r \to q)$	premissa
3	p	hipótese
4	$r \to q$	$\to_{\mathrm{e}}$ 3,2
5	q	$\to_{\mathrm{e}}$ 1,4
6	$q \wedge r$	$\wedge_{\mathrm{i}}$ 5,1
7	$p \to (q \wedge r)$	$\to_{\mathrm{i}}$ 3–6

7.

1	$(p \wedge q) \wedge r$	premissa
2	$p \wedge q$	$\wedge_{e_1}$ 1
3	r	$\wedge_{e_2}$ 1
4	p	$\wedge_{e_1}$ 2
5	q	$\wedge_{e_2}$ 2
6	$q \wedge r$	$\wedge_i$ 5,3
7	$p \wedge (q \wedge r)$	$\wedge_i$ 4,6

Solução do exercício 2.8.
σοφα, φιλος, φιλοσοφια, ανθρωπος, λογος, φοβος, λογικη, λογικο.

Solução do exercício 2.9.

1. O lei da contraposição (extendida): $(p \wedge q) \to r \dashv\vdash (p \wedge \neg r) \to \neg q$

1	$(p \wedge q) \to r$	premissa
2	$p \wedge \neg r$	hipótese
3	p	$\wedge_{e_1}$ 3
4	$\neg r$	$\wedge_{e_2}$ 3
5	q	hipótese
6	$p \wedge q$	$\wedge_i$ 3,5
7	r	$\to_e$ 6,1
8	$\bot$	$\neg_e$ 7,4
9	$\neg q$	$\neg_i$ 5–8
10	$(p \wedge \neg r) \to \neg q$	$\to_i$ 2–9

2. $(p \to q) \to (r \to s) \vdash (p \to s) \to (r \to s)$
A prova usa duas lemas $r_1 : \neg(p \to q) \vdash p$ e $r_2 : \neg(p \to q) \vdash \neg q$

1	$(p \to q) \to (r \to s)$	premissa
2	$p \to s$	hipótese
3	$\neg(r \to s)$	hipótese
4	$\neg(p \to q)$	Modus tollens 3,1
5	p	lema r_1 4
6	s	$\to_e$ 5,2
7	$\neg s$	lema r_2 3
8	$\bot$	$\neg_e$ 6,7
9	$r \to s$	PBC 3–8
10	$(p \to s) \to (r \to s)$	$\to_i$ 2–9

Prova da lema r_1:

1	$\neg(p \to q)$	premissa
2	$\neg p$	hipótese
3	p	hipótese
4	$\bot$	$\neg_e$ 3,2
5	q	$\bot_e$ 4
6	$p \to q$	$\to_i$ 2–5
7	$\bot$	$\neg_e$ 6,1
8	p	PBC 2–7

Prova da lema r_2:

1	$\neg(p \to q)$	premissa
2	q	hipótese
3	p	hipótese
4	q	cópia 2
5	$p \to q$	$\to_i$ 3–4
6	$\bot$	$\neg_e$ 5,1
7	$\neg q$	$\neg_i$ 2–6

Solução do exercício 2.10.

1. Temos os seguintes proposições atômicas: p_1: "Pedro é honesto". p_2: "Paulo é honesto.". o: "Tem ouro na cidade". Usando eles, Pedro afirma: $o \wedge p_2$, e Paulo afirma: $o \wedge \neg p_1$. Como os dois podem mentir não temos premissas ainda! Mas se Pedro é honesto, a sua afirmação é válido: $p_1 \to o \wedge p_2$ é uma premissa. Do mesmo jeito, se $o \wedge p_2$, Pedro é honesto: $o \wedge p_2 \to p_1$ é uma premissa (em breve $p_1 \leftrightarrow o \wedge p_2$ é uma premissa). Com o mesmo raciocínio chegamos na segunda premissa $p_2 \leftrightarrow o \wedge \neg p_1$.

2. Suponhamos que tem ouro:

1	$p_1 \to o \land p_2$	premissa
2	$o \land p_2 \to p_1$	premissa
3	$p_2 \to o \land \neg p_1$	premissa
4	$o \land \neg p_1 \to p_2$	premissa
5	o	hipótese
6	$\neg p_1$	hipótese
7	$o \land \neg p_1$	$\land_{\mathrm{i}}$ 5,6
8	p_2	$\to_{\mathrm{e}}$ 7,4
9	$o \land p_2$	$\land_{\mathrm{i}}$ 5,8
10	p_1	$\to_{\mathrm{e}}$ 9,2
11	$\bot$	$\neg_{\mathrm{e}}$ 6,10
12	p_1	PBC 6–11
13	$o \land p_2$	$\to_{\mathrm{e}}$ 12,1
14	p_2	$\land_{\mathrm{e_1}}$ 13
15	$o \land \neg p_1$	$\to_{\mathrm{e}}$ 14,3
16	$\neg p_1$	$\land_{\mathrm{e_2}}$ 15
17	$\bot$	$\neg_{\mathrm{e}}$ 12,16
18	$\neg o$	PBC 6–17

Então, não tem ouro e também podemos provar, que Pedro e Paulo mentem:

19	p_1	hipótese
20	$o \land p_2$	$\to_{\mathrm{e}}$ 19,1
21	o	$\land_{\mathrm{e_1}}$ 20
22	$\bot$	$\neg_{\mathrm{e}}$ 21,18
23	$\neg p_1$	$\neg_{\mathrm{i}}$ 19–22
24	p_2	hipótese
25	$o \land \neg p_1$	$\to_{\mathrm{e}}$ 24,3
26	o	$\land_{\mathrm{e_1}}$ 25
27	$\bot$	$\neg_{\mathrm{e}}$ 26,18
28	$\neg p_2$	$\neg_{\mathrm{i}}$ 24–27

Solução do exercício 2.11.

(1) $p \lor (q \land r) \vdash (p \lor q) \land (p \lor r)$

1	$p \vee (q \wedge r)$	premissa
2	p	hipótese
3	$p \vee q$	$\vee_{i_1} 2$
4	$p \vee r$	$\vee_{i_1} 2$
5	$(p \vee q) \wedge (p \vee r)$	$\wedge_i 3, 4$
6	$q \wedge r$	hipótese
7	q	$\wedge_{e_1} 6$
8	r	$\wedge_{e_2} 6$
9	$p \vee q$	$\vee_{i_2} 7$
10	$p \vee r$	$\vee_{i_2} 8$
11	$(p \vee q) \wedge (p \vee r)$	$\wedge_i 9, 10$
12	$(p \vee q) \wedge (p \vee r)$	$\vee_e 1, 2-5, 6-11$

(1) $p \vee (q \wedge r) \dashv (p \vee q) \wedge (p \vee r)$

1	$(p \vee q) \wedge (p \vee r)$	hipótese
2	$p \vee q$	$\wedge_{e_1}$
3	$p \vee r$	$\wedge_{e_2}$
4	p	hipótese
5	$p \vee (q \wedge r)$	$\vee_{i_1} 4$
6	q	hipótese
7	$p \vee r$	cópia 3
8	p	hipótese
9	$p \vee (q \wedge r)$	$\vee_{i_1} 8$
10	r	hipótese
11	$q \wedge r$	$\wedge_i 6, 10$
12	$p \vee (q \wedge r)$	$\vee_{i_2} 11$
13	$p \vee (q \wedge r)$	$\vee_e 7, 8-9, 10-12$
14	$p \vee (q \wedge r)$	$\vee_e 2, 4-5, 6-12$

(2) $p \vee p \vdash p$

1	$p \vee p$	premissa
2	p	hipótese
3	p	cópia 1
4	p	$\vee_e$ 1,2–3,2–3

(2) $p \vee p \dashv p$

1	p	premissa
2	$p \vee p$	$\vee_{i_1}$ 1

(3) $p \vee (q \vee r) \vdash (p \vee q) \vee r$

1	$p \lor (q \lor r)$	premissa
2	p	hipótese
3	$p \lor q$	$\lor_{i_1}$ 2
4	$(p \lor q) \lor r$	$\lor_{i_1}$ 3
5	$q \lor r$	hipótese
6	q	hipótese
7	$p \lor q$	$\lor_{i_2}$ 6
8	$(p \lor q) \lor r$	$\lor_{i_1}$ 7
9	r	hipótese
10	$(p \lor q) \lor r$	$\lor_{i_2}$ 9
11	$(p \lor q) \lor r$	$\lor_e$ 5,6–8,9–10
12	$(p \lor q) \lor r$	$\lor_e$ 1,2–4,5–11

(3) $p \lor (q \lor r) \dashv (p \lor q) \lor r$ (o mesmo princípio da (3) $\vdash$).

(4) $\neg(p \lor q) \vdash \neg p \land \neg q$

1	$\neg(p \lor q)$	premissa
2	p	hipótese
3	$p \lor q$	$\lor_{i_1}$ 2
4	$\bot$	$\neg_e$ 3,1
5	$\neg p$	PBC 2–4
6	q	hipótese
7	$p \lor q$	$\lor_{i_2}$ 6
8	$\bot$	$\neg_e$ 7,1
9	$\neg q$	PBC 6–8
10	$\neg p \land \neg q$	$\land_i$ 5,9

(4) $\neg(p \lor q) \dashv \neg p \land \neg q$

1	$\neg p \land \neg q$	premissa
2	$\neg p$	$\land_{e_1}$ 1
3	$\neg q$	$\land_{e_2}$ 2
4	$p \lor q$	hipótese
5	p	hipótese
6	$\bot$	$\neg_e$ 5,2
7	q	hipótese
8	$\bot$	$\neg_e$ 7,3
9	$\bot$	$\lor_e$ 4,5–6,7–8
10	$\neg(p \lor q)$	PBC 4–9

Solução do exercício 2.12.

O segundo lei é a regra que chamamos "modus ponens" é não precisa ser provada. O terceiro lei é a regra que chamamos "modus tollens" é não precisa ser provada.

1	$p \lor q$	premissa
2	$\neg p$	premissa
3	p	hipótese
4	$\bot$	$\neg_e 3,2$
5	q	$\bot_e 4$
6	q	hipótese
7	q	$\lor_e 1,3–5,6–6$

1	$\neg p \lor \neg q$	premissa
2	p	premissa
3	$\neg p$	hipótese
4	$\bot$	$\neg_e 2,3$
5	$\neg q$	$\bot_e 4$
6	$\neq q$	hipótese
7	$\neg q$	$\lor_e 1,3–5,6–6$

Solução do exercício 2.13.
(a) $p \vdash q$ significa, que temos uma prova com premissa p chegando em q. Com isso, podemos construir uma outra prova:

1	p	hipótese
2	...	usa a prova $p \vdash q$
3	q	
4	$p \to q$	

Ao contrário, se $\vdash p \to q$, com premissa p podemos usar o modus ponens ($\to_e$) para chegar a q, logo $p \vdash q$.
(b) A mesma construção funciona no caso geral. $p_1, \ldots p_n \vdash q$ significa, que temos uma prova com premissa $p_1, \ldots, p_n$ chegando em q. Podemos construir uma prova como

1	p_1	hipótese
2	p_2	hipótese
	...	
n	p_n	hipótese
	...	usa a prova $p_1, \ldots p_n \vdash q$
m	q	
m+1	$p_n \to q$	$\to_i$ n–m
	...	
m+n-1	$p_2 \to (\cdots (p_n \to q) \cdots)$	$\to_i$ 2–(m+n-2)
m+n	$p_1 \to (p_2 \to (\cdots (p_n \to q) \cdots)))$	$\to_i$ 1–(m+n-1)

e ao contrário, com teorema $p_1 \to (p_2 \to (\cdots (p_n \to q) \cdots)))$ e premissas $p_1, \ldots, p_n$ podemos usar n vezes $\to_e$ (modus ponens) para chegar na conclusão q. Logo, $p_1, \ldots, p_n \vdash q$.

Solução do exercício 2.14.

1	$A \wedge B$	premissa
2	A	$\wedge_{e_1}$ 1
3	B	$\wedge_{e_2}$ 1
4	$A \to \neg B$	hipótese
5	$\neg B$	$\to_e$ 2,4
6	$\perp$	$\neg_e$ 3,5
7	$\neg(A \to \neg B)$	PBC

1	$\neg(A \to \neg B)$	premissa
2	$\neg A$	hipótese
3	A	hipótese
4	$\perp$	$\neg_e$ 3,2
5	$\neg B$	$\perp_e$ 4
6	$A \to \neg B$	$\to_i$ 3–5
7	$\perp$	$\neg_e$ 6,1
8	A	PBC 2–7
9	$\neg B$	hipótese
10	A	hipótese
11	$\neg B$	cópia 9
12	$A \to \neg B$	$\to_i$ 10-11
13	$\perp$	$\neg_e$ 12,1
14	B	PBC 9–13
15	$A \wedge B$	$\wedge_i$ 8,14

1	$A \vee B$	premissa
2	A	hipótese
3	$\neg A$	hipótese
4	$\perp$	$\neg_e$ 2,3
5	B	$\perp_e$ 4
6	$\neg A \to B$	$\to_i$ 3–5
7	B	hipótese
8	$\neg A$	hipótese
9	B	cópia 7
10	$\neg A \to B$	$\to_i$ 8–9
11	$\neg A \to B$	$\vee_e$ 1,2–6,7–10

1	$\neg A \to B$	premissa	
2	$\neg(A \vee B)$	hipótese	
3	$\neg A \wedge \neg B$	deMorgan	
4	$\neg A$	$\wedge_{e_1}$ 3	
5	$\neg B$	$\wedge_{e_2}$ 3	
6	B	$\to_e$ 4,1	
7	$\perp$	$\neg_e$ 6,5	
8	$A \vee B$	PBC	2–7

Solução do exercício 2.15.

1	$\neg\neg\neg A \to \neg A$	Regra $\neg\neg_e$, substituindo $A \mapsto \neg A$
2	$(\neg\neg\neg A \to \neg A) \to (A \to \neg\neg A)$	ax_3
3	$A \to \neg\neg A$	$\to_e$ 1,2

1	$A \to \neg\neg A$	Regra $\neg\neg_i$.
2	$\neg\neg A \to (\neg B \to \neg\neg A)$	ax_1, substituindo $A \mapsto \neg\neg A$, $B \mapsto \neg b$.
3	$(\neg B \to \neg\neg A) \to (\neg A \to B)$	ax_3 substituindo $A \mapsto \neg A$.
4	$A \to (\neg B \to \neg\neg A)$	Trans 1,2
5	$A \to (\neg A \to b)$	Trans 4,3

Solução do exercício 2.16.

Seja o nome das funções S_2, P, S_{23} respetivamente (S_2 é a função simétrica de quatro variáveis, que é verdadeira, se exatamente 2 argumentos são verdadeiros; S_{23} é verdadeira se exatamente 2 ou 3 argumentos são verdadeiros).

(a)

p	$p \wedge \neg p$
f	f
v	f

(b)

p	q	$p \wedge \neg q$
f	f	f
f	v	f
v	f	v
v	v	f

(c)

p	q	$p \wedge q \to p \vee q$
f	f	v
f	v	v
v	f	v
v	v	v

(d)

p	q	$p \to (q \to p)$
f	f	v
f	v	v
v	f	v
v	v	v

(e)

p	q	$(p \to q) \to (\neg q \to \neg p)$
f	f	v
f	v	f
v	f	v
v	v	v

(f)

p	q	$\neg(\neg p \wedge \neg q)$
f	f	f
f	v	v
v	f	v
v	v	v

(g)

p	q	r	s	S_{23}
f	f	f	f	f
f	f	f	v	f
f	f	v	f	f
f	f	v	v	v
f	v	f	f	f
f	v	f	v	v
f	v	v	f	v
f	v	v	v	v
v	f	f	f	f
v	f	f	v	v
v	f	v	f	v
v	f	v	v	v
v	v	f	f	v
v	v	f	v	f
v	v	v	f	f
v	v	v	v	f

(h)

p	q	r	s	P
f	f	f	f	v
f	f	f	v	v
f	f	v	f	f
f	f	v	v	f
f	v	f	f	v
f	v	f	v	f
f	v	v	f	f
f	v	v	v	v
v	f	f	f	f
v	f	f	v	f
v	f	v	f	f
v	f	v	v	f
v	v	f	f	v
v	v	f	v	v
v	v	v	f	v
v	v	v	v	v

(i)

a	b	c	d	S_2
f	f	f	f	f
f	f	f	v	f
f	f	v	f	f
f	f	v	v	v
f	v	f	f	f
f	v	f	v	v
f	v	v	f	v
f	v	v	v	f
v	f	f	f	f
v	f	f	v	v
v	f	v	f	v
v	f	v	v	f
v	v	f	f	v
v	v	f	v	f
v	v	v	f	f
v	v	v	v	f

Solução do exercício 2.17.

(a)

p	q	$\neg(\neg p \vee q)$
f	f	f
f	v	f
v	f	v
v	v	f

(b)

p	$\neg p \rightarrow p$
f	f
v	v

(c)

p	q	$p \rightarrow q$	$\neg(p \wedge \neg q)$
f	f	v	v
f	v	v	v
v	f	f	f
v	v	v	v

(d)

p	$p \oplus p$
f	f
v	f

(e)

p	q	$(p \oplus q) \oplus p$
f	f	f
f	v	v
v	f	f
v	v	v

(f)

p	$p \oplus 1$	$\neg p$
f	v	v
v	f	f

(g)

p	q	r	$(p \oplus q) \wedge r$	$(p \wedge r) \oplus (q \wedge r)$
f	f	f	f	f
f	f	v	f	f
f	v	f	f	f
f	v	v	v	v
v	f	f	f	f
v	f	v	v	v
v	v	f	f	f
v	v	v	f	f

(h)

p	q	$p \oplus q$	$\neg(p \equiv q)$
f	f	f	f
f	v	v	v
v	f	v	v
v	v	f	f

Usando as tabelas de verdade resulta que todas relações são corretos.

Solução do exercício 2.18.

p_1	p_2	o	$p_1 \leftrightarrow o \wedge p_2$	$p_2 \leftrightarrow o \wedge \neg p_1$
f	f	f	v	v
f	f	v	v	f
f	v	f	v	f
f	v	v	f	v
v	f	f	f	v
v	f	v	f	v
v	v	f	f	f
v	v	v	v	f

Assim, se todas a premissas são verdadeiras, o único caso e que p_1, p_2 e o são falsos e a relação de conseqüência semântica entre as premissas e a conclusão é correto nesse caso:

$$p_1 \leftrightarrow o \wedge p_2, p_2 \leftrightarrow o \wedge \neg p_1 \models \neg o$$

Solução do exercício 2.19.

1. Se a última regra aplicada (linha k) foi $\vee_e$ com resultado χ a prova refere a três items: uma disjunção $\Phi \vee \Psi$, uma prova $\Phi \cdots \chi$ e uma prova $\Psi \cdots \chi$ (todas em linhas $< k$). Junto com as premissas, obtemos provas $\Phi_1, \ldots, \Phi_n \vdash \Phi \vee \Psi$, $\Phi_1, \ldots, \Phi_n, \Phi \vdash \chi$, $\Phi_1, \ldots, \Phi_n, \Psi \vdash \chi$ em menos que k linhas e usando a hipótese de inducução obtemos também $\Phi_1, \ldots, \Phi_n \models \Phi \vee \Psi$, $\Phi_1, \ldots, \Phi_n, \Phi \models \chi$, $\Phi_1, \ldots, \Phi_n, \Psi \models \chi$. Logo, $\Phi_1, \ldots, \Phi_n \vdash \chi$: Suponha $\Phi_1, \ldots, \Phi_n$ verdadeiras e χ falso. A primereira relação implica que $\Phi \vee \Psi$ é verdadeira, mas se χ é falso, as últimas duas regras implicam que Φ e Ψ são falsos, uma contradição.

2. Se a última regra aplicada (linha k) foi $\rightarrow_e$ a prova refere a uma fórmula Φ e uma implicação $\Phi \rightarrow \Psi$ (em linhas $< k$). Logo, obtemos provas $\Phi_1, \ldots, \Phi_n \vdash \Phi$ e $\Phi_1, \ldots, \Phi_n \vdash \Phi \rightarrow \Psi$ com menos que k linhas e usando a hipótese da indução obtemos também $\Phi_1, \ldots, \Phi_n \models \Phi$ e $\Phi_1, \ldots, \Phi_n \models \Phi \rightarrow \Psi$. Esses dois relações implicam que $\Phi_1, \ldots, \Phi_n \models \Psi$ é correto também.

Solução do exercício 2.20.
Usando uma tabela de verdade, obtemos

p	q	r	$p \vee (q \wedge r)$	$(p \vee q) \wedge r$	
f	f	f	f	f	
f	f	v	f	f	
f	v	f	f	f	
f	v	v	v	v	
v	f	f	v	f	$*$
v	f	v	v	v	
v	v	f	v	f	$*$
v	v	v	v	v	

Analisando as linhas com valores de verdade diferentes (*), a definição da relação de conseqüência da semântica permite concluir que

$$p \vee (q \wedge r) \not\models (p \vee q) \wedge r$$
$$(p \vee q) \wedge r \models p \vee (q \wedge r)$$

Então, a consistência e completude permite concluir que

$$p \vee (q \wedge r) \nvdash (p \vee q) \wedge r$$
$$p \vee (q \wedge r) \dashv (p \vee q) \wedge r.$$

Solução do exercício 2.21.
O rascunho de uma implementação em OCaml. Uma representação simples de fórmulas da lógica proposicional é

```
(* representation of a boolean formula *)
type formula =
    BinaryOperation of formula*char*formula |
    UnaryOperation of char * formula |
    Proposition of char;;
```

com caracteres para os operadores binários e as proposições. Com uma certa atribuição das proposições, a avaliação é

```
(* evaluate a formula given a valuation v

   Input: formula f of type formula
          valuation v
          (association list of propositions with booleans)
   Output: true or false
*)
```

```
let rec evaluateFormula f v =
  match f with
  | BinaryOperation (f1,op,f2) >
      let v1 = evaluateFormula f1 v
      and v2 = evaluateFormula f2 v
      in begin
        match op with
          '+' > v1 || v2
        | '*' > v1 && v2
        | '>' > not v1 || v2
        | op > raise (IllegalOperator op)
      end
  | UnaryOperation (op,f1) >
      let v1 = evaluateFormula f1 v in not v1
  | Proposition p >
      List.assoc p v
;;
```

com a representação $+$, $*$, $>$ para a disjunção, conjunção e implicação, respetivamente. A única operação unária é a negação. Construindo todas atribuições possíveis, podemos aplicar `evaluateFormula` para obter uma tabela de verdade para uma fórmula.

Solução do exercício 2.22.

1. $s \equiv a \oplus b$, $c' \equiv a \wedge b$
2. $s \equiv a \oplus b \oplus c$, $c' \equiv (a \wedge c) \vee (b \wedge c) \vee (a \wedge b)$

Solução do exercício 2.23.

1. $d \equiv a \oplus b$ e $u' \equiv \neg a \wedge b$.
2. $d \equiv a \oplus b \oplus u$ e $u' \equiv (u \wedge \neg a) \vee (b \wedge \neg a) \vee (u \wedge b)$.

Solução do exercício 2.24.
Indução sobre o número dos símbolos $|s \models n$ na cadeia s. Base: Com $n = 0$, temos $s = \epsilon$. Logo $as = sb$ não é válido, porque $a \neq b$. Passo: Suponha que não tem cadeias de tamanho n tal que $as = sb$. Suponha também, que tem uma cadeia s' de tamanho $n + 1$ tal que $as' = s'b$. s' tem que ter a forma $s' = at$ com uma cadeia $|t \models n$ porque o primeiro símbolo é iqual. Temos

$aat = atb$, que implica que $at = tn$, uma contradição. Logo, não tem uma cadeia s' de tamanho $n+1$ tal que $as' = s'b$.

Solução do exercício 2.25.

1. $q \vee p, q \to \neg r \vdash q \vee ((\neg r \to p) \wedge (p \to \neg r))$

1					$q \vee p$			premissa
2					$q \to \neg r$			premissa
3					$\neg(q \vee ((\neg r \to p) \wedge (p \to \neg r)))$			negação da conclus
4					$\neg q$			$\neg \vee 3$
5					$\neg((\neg r \to p) \wedge (p \to \neg r))$			$\neg \vee 3$
6	$\neg(\neg r \to p)$				$\neg(p \to \neg r)$			$\neg \wedge 5$
7	$\neg r$				p			$\neg \to 6$
8	$\neg p$				$\neg\neg r$			$\neg \to 6$
9					r			$\neg\neg 8$
10	$\neg q$		$\neg r$		$\neg q$		$\neg r$	$\to 2$
11	q	p	q	p	q	p	$\times$	$\vee 1$
12	$\times$	$\times$	$\times$	$\times$	$\times$	$\odot$		

$p = v$, $q = f$ e $r = v$ é um contra-examplo.

2. $(s \to t) \wedge (t \to s), t \to (s \to q), \neg r \to \neg p \vdash (p \vee s) \to (r \vee q)$

1	$s \to t \wedge t \to s$		premissa
2	$t \to (s \to q)$		premissa
3	$\neg r \to \neg p$		premissa
4	$\neg((p \vee s) \to (r \vee q))$		premissa
5	$s \to t$		$\wedge 1$
6	$t \to s$		$\wedge 1$
7	$p \vee s$		$\neg \to 4$
8	$\neg(r \vee q)$		$\neg \to 4$
9	$\neg r$		$\neg \vee 8$
10	$\neg q$		$\neg \vee 9$
11	$\neg p$	$\neg\neg r$	$\to 3$
12		r	$\neg\neg 11$
13		$\times$	
14	s	p	$\vee 7$
15		$\times$	
16	t	$\neg s$	$\to 5$
17		$\times$	
18	s	$\neg t$	$\to 6$
19		$\times$	
20	$s \to q$	$\neg t$	$\to 2$
21		$\times$	
22	$\neg s$	q	$\to 20$
23	$\times$	$\times$	

3. $p, q, \neg r \vdash \neg(((q \to r) \to (q \to q)) \wedge ((r \wedge q) \wedge (p \wedge q)))$

1	p	premissa
2	q	premissa
3	$\neg r$	premissa
4	$\neg\neg(((q \to r) \to (q \to q)) \wedge ((r \wedge q) \wedge (p \wedge q)))$	negação da conclusão
5	$(((q \to r) \to (q \to q)) \wedge ((r \wedge q) \wedge (p \wedge q)))$	$\neg\neg 4$
6	$((q \to r) \to (q \to q))$	$\wedge 5$
7	$(r \wedge q) \wedge (p \wedge q)$	$\wedge 5$
8	$r \wedge q$	$\wedge 7$
9	$p \wedge q$	$\wedge 7$
10	r	$\wedge 8$
11	$\times$	

Solução do exercício 2.26.

1. $p \to (q \to r) \dashv\vdash q \to (p \to r)$

1	$p \to (q \to r)$	premissa
2	q	hipótese
3	p	hipótese
4	$q \to r$	MP 3,1
5	r	MP 2,4
6	$p \to r$	$\to_{\mathrm{i}}$3–5
7	$q \to (p \to r)$	$\to_{\mathrm{i}}$2–6

O prova da direção contrária é a mesma com q e p trocado.

2. $p \to (q \to r) \dashv\vdash (p \land q) \to r$

1	$p \to (q \to r)$	premissa
2	$p \land q$	hipótese
3	p	$\land_{\mathrm{e}_1}$ 2
4	q	$\land_{\mathrm{e}_2}$ 2
5	$q \to r$	MP 3,1
6	r	MP 4,5
7	$(p \land q) \to r$	$\to_{\mathrm{i}}$2–6

1	$(p \land q) \to r$	premissa
2	p	hipótese
3	q	hipótese
4	$p \land q$	$\land_{\mathrm{i}}$2,3
5	r	MP 4,1
6	$q \to r$	$\to_{\mathrm{i}}$3–5
7	$r \to (q \to r)$	$\to_{\mathrm{i}}$2–6

3. $p, \neg q, r \vdash \neg(p \to ((q \lor r) \to (r \to \neg p)))$

1	p	premissa
2	$\neg q$	premissa
3	r	premissa
4	$p \to ((q \lor r) \to (r \to \neg p))$	hipótese
5	$(q \lor r) \to (r \to \neg p)$	MP 1,4
6	$q \lor r$	$\lor_{\mathrm{i}_1}$ 3
7	$r \to \neg p$	MP 6,5
8	$\neg p$	MP 3,7
9	$\bot$	$\neg_{\mathrm{e}}$1,8
10	$\neg(p \to ((q \lor r) \to (r \to \neg p)))$	$\neg_{\mathrm{i}}$4–9

Solução do exercício 2.27.

1. $(p \to q) \to (q \to r)$

$$((p \lor (\neg q \lor r)) \land (\neg q \lor (\neg q \lor r)))$$

2. $(((p \to p) \lor (r \lor r)) \to (\neg p \land \neg r))$

$$\begin{aligned}((((p \lor \neg p) \land (p \lor \neg r)) \land ((\neg p \lor \neg p) \land (\neg p \lor \neg r)))\\ \land (((\neg r \lor \neg p) \land (\neg r \lor \neg r)) \land ((\neg r \lor \neg p) \land (\neg r \lor \neg r))))\end{aligned}$$

3. $(((p \lor p) \to \neg r) \to ((r \land p) \to (r \to q)))$

$$(((p \lor p) \lor ((\neg r \lor \neg p) \lor (\neg r \lor q))) \land (r \lor ((\neg r \lor \neg p) \lor (\neg r \lor q))))$$

Solução do exercício 2.28.

1. Tabelas de verdade

p	q	$p \otimes q$	$p \leftrightarrow q$
f	f	f	v
f	v	v	f
v	f	v	f
v	v	f	v

2. Regras dedutivas

$$\frac{p \quad p \leftrightarrow q}{q}\leftrightarrow_{e1} \qquad \frac{q \quad p \leftrightarrow q}{p}\leftrightarrow_{e2} \qquad \frac{p \to q \quad q \to p}{p \leftrightarrow q}\leftrightarrow_i n$$

$$\frac{p \quad p \otimes q}{\neg q}\otimes_{e1} \qquad \frac{q \quad p \otimes q}{\neg p}\otimes_{e2} \qquad \frac{p \to \neg q \quad \neg p \to q}{p \otimes q}\otimes_i$$

3. Regras para árvores

$$\begin{array}{ccc} & p \leftrightarrow q & \\ \swarrow & & \searrow \\ p & & \neg p \\ q & & \neg q \end{array} \qquad \begin{array}{ccc} & \neg(p \leftrightarrow q) & \\ \swarrow & & \searrow \\ p & & \neg p \\ \neg q & & q \end{array}$$

com as regras para $\neg(p \otimes q)$ e $p \otimes q$ as mesma que para $p \leftrightarrow q$ e $\neg(p \leftrightarrow q)$, respectivamente.

Solução do exercício 3.1.

1. $\forall n \exists m(m > n)$, com $\mathcal{F} = Z$ e $\mathcal{P} = \{>, =\}$ e o significado informal

   $x > y$: x é maior que y.

2. $(\forall p P(p)) \to \neg Q(e)$, com $\mathcal{F} = \{e\}$, $\mathcal{P} = \{P, Q\}$ e significado informal

   P(x): x tem um Porsche.
   Q(x): x quer um Porsche.
   e: Eu.

   Alternativa: Ao invés de falar só sobre pessoas, podemos permitir objetos como um Porsche. Uma possibilidade é $(\forall h H(h) \to T(h,p)) \to \neg Q(e,p)$, com $\mathcal{F} = \{e, p\}$, $\mathcal{P} = \{H, T, Q\}$ e o significado informal

   H(x): x é humano.
   T(x,y): x tem y.
   Q(x,y): x quer y.
   e: Eu.
   p: Porsche.

3. $(\exists p H(p)) \vee (\forall p M(p))$, com $\mathcal{F} = \emptyset$, $\mathcal{P} = \{H, P\}$ e significado informal

   H(x): x é um heroi que nos salve.
   M(x): x vai morrer.

   Dependente do contexto, queremos uma fórmula que garante um dos dois eventos só, por exemplo $((\exists p H(p)) \vee (\forall p M(p))) \wedge \neg((\exists p H(p)) \wedge (\forall p M(p)))$.

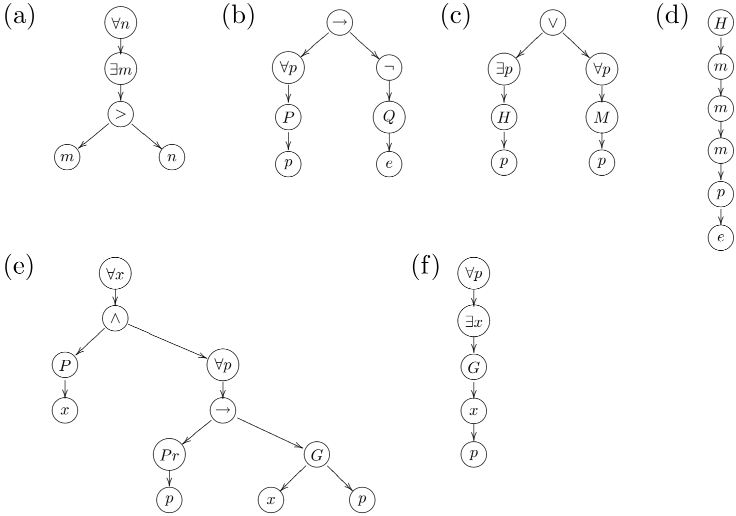
4. $H(m(m(m(p(e)))))$ com $\mathcal{F} = \{e, m, p\}$ e $\mathcal{P} = \{H\}$ e significado

   m(x): Mãe de x.
   p(x): Pai de x.
   H(x): x é humano.
   e: Eu.

5. $(\exists x P(x) \wedge \forall p : Pr(p) \to G(x,p))$ com $\mathcal{F} = \emptyset$ e $\mathcal{P} = \{P, Pr, G\}$ e significado informal

   P(x): x é uma pessoa.
   Pr(x): x é um premio.
   G(x,p): x ganhou premio p.

6. $\forall p \exists x G(x,p)$ com a mesma interpretação do item anterior.



Solução do exercício 3.2.

1. Usando uma interpretação sobre números inteiros, a fórmula afirme que "Para todos pares de números, tal que um é maior ou igual que o outro, existe um terceiro diferente e entre dos dois" (que não é correto nessa interpretação).

2. Como x so ocorre no escopo do x, a fórmula intuitivamente é equivalente com $\exists x \neg P(x)$ e afirme que existe um objeto que não tem a característica P. Por exemplo, com
$P(x)$: x é um número primo.
$\exists x \neg P(x)$, sobre números inteiros significa "Tem números que não são primos".

3. A fórmula afirme que o predicado $P(x, y)$ é simétrico. Um exemplo é
P(x,y): x está casado com y.
com a interpretação "Se alguem está casado com alguma pessao, essa pessoa também está casada com a primeira pessao".

4. A fórmula afirme que se todos objetos tem a característica R, existe um objeto que é identíco com o resultado da aplicação da função m para si. Por exemplo se escolhemos $R(x)$: x mora em Porto Alegre.
$m(x)$: O prefeito da cidade que x mora.
o significado seria "Se todo mundo mora em Porto Alegre, existe uma pessoa que é o prefeito da cidade em que ela mora".

5. A fórmula afirma a existencia de um objeto z que tem uma relação P ou uma relação R com todos objetos. Por exemplo, com
$P(x, y)$: x é menor ou igual que y.
$R(x, y)$: x é maior ou igual que y.
considerando números $\in \mathbb{N}$, temos "Existe um número que é menor ou igual ou maior o igual que todos números".

6. A fórmula afirme que a relação R é reflexivo. Por exemplo com
$R(x, y)$: x é menor ou igual que y.
a interpretação é "Todos números são menor ou igual a si mesmo".

Solução do exercício 3.3.

1. $L(\exists x \neg Q(c)) = \emptyset$. Ligadas são $\{x\}$.
2. $L(\forall z \neg R(z)) = \emptyset$. Ligadas são $\{z\}$.
3. $L((P(c) \vee P(a)) \wedge Q(c, z, b, x)) = \emptyset$. Ligadas são $\emptyset$.
4. $L(\exists x(P(x) \rightarrow P(c))) = \emptyset$. Ligadas são $\{x\}$.
5. $L(\forall y \exists x Q(z)) = \{z\}$. Ligadas são $\{x, y\}$.
6. $L(\forall z Q(x, z, x, z)) = \{x\}$. Ligadas são $\{z\}$.

Solução do exercício 3.4.
Como so os items 5 e 6 tem variáveis livres, o resto das fórmulas não muda.

1. $\exists x \neg Q(c)$
2. $\forall z \neg R(z)$
3. $(P(c) \vee P(a)) \wedge Q(c, z, b, x)$
4. $\exists x(P(x) \rightarrow P(c))$
5. $\forall z \forall y \exists x Q(z)$
6. $\forall x \forall z Q(x, z, x, z)$

Solução do exercício 3.5.

1. Com $\Phi \equiv \forall z \exists y \neg P(y)$ temos$\Phi[g(c)/x] = \Phi$

2. Com $\Phi \equiv \forall z \exists y \neg P(y)$ temos $\Phi[g(c)/y] = \Phi$

3. com $\Phi \equiv \forall z \exists y \neg P(y)$ temos $\Phi[g(c)/z] = \Phi$

4. Com $\Phi \equiv \forall z(R(b) \lor R(z) \lor Q(y,y,z))$ temos $\Phi[f(z)/y] = \forall x(R(b) \lor R(x) \lor Q(f(z), f(z), x))$

5. Com $\Phi \equiv \forall z(R(b) \lor R(z) \lor Q(y,y,z))$ temos $\Phi[f(x)/y] = \forall z(R(b) \lor R(z) \lor Q(f(x), f(x), z))$

6. Com $\Phi \equiv \forall z(R(b) \lor R(z) \lor Q(y,y,z))$ temos $\Phi[f(x)/z] = \Phi$

Solução do exercício 3.6.

1. Sejam $\mathcal{F} = \{f, n, e, m_1, k, m_2, h, w, \text{mae}, \text{pai}\}$ (tal que os primeiros 8 elementos são constantes e os ultimos dois funções com um argumento) e $\mathcal{P} = \{\text{filho}, \text{irmao}\}$ (dois predicados de aridade dois).

2. O universo seja

   $$U = \{\text{francesco}, \text{nina}, \text{edelweis}, \text{marcus}, \text{klaus}, \text{marianne}, \text{hilton}, \text{wilma}, \bot\}.$$

   Como as funções mae e pai tem que ser total, usamos um elemento $\bot$ para o valor “não definido” nos casos que o universo não contém uma pessoa adequada.

   O significado dos constantes seja $f^{\mathcal{M}} = \text{francesco}$, $n^{\mathcal{M}} = \text{nina}$, $e^{\mathcal{M}} = \text{edelweis}$, $m_1^{\mathcal{M}} = \text{marcus}$, $k^{\mathcal{M}} = \text{klaus}$, $m_2^{\mathcal{M}} = \text{marianne}$, $h^{\mathcal{M}} = \text{hilton}$, $w^{\mathcal{M}} = \text{wilma}$. O significado das funções seja

   $$\text{mae}^{\mathcal{M}} = \{(\text{francesco}, \text{edelweis}), (\text{nina}, \text{edelweis}), (\text{edelweis}, \text{wilma}), (\text{marcus}, \text{marianne}), (\text{klaus}, \bot), (\text{marianne}, \bot), (\text{hilton}, \bot), (\text{wilma}, \bot)\}$$

   e

   $$\text{pai}^{\mathcal{M}} = \{(\text{francesco}, \text{marcus}), (\text{nina}, \text{marcus}), (\text{edelweis}, \text{hilton}), (\text{marcus}, \text{klaus}), (\text{klaus}, \bot), (\text{marianne}, \bot), (\text{hilton}, \bot), (\text{wilma}, \bot)\}.$$

   O significado dos predicados seja

   $$\begin{aligned} \text{filho}^{\mathcal{M}} = \{&(\text{edelweis}, \text{francesco}), (\text{edelweis}, \text{nina}), (\text{marcus}, \text{francesco}), \\ &(\text{marcus}, \text{nina}), (\text{klaus}, \text{marcus}), (\text{marianne}, \text{marcus}), \\ &(\text{hilton}, \text{edelweis}), (\text{wilma}, \text{edelweis})\} \end{aligned}$$

   e $\text{irmao}^{\mathcal{M}} = \{(\text{francesco}, \text{nina}), (\text{nina}, \text{francesco})\}$.

Solução do exercício 3.7.

1. Com qualquer universo U tal que $P^{\mathcal{M}} = \emptyset \subseteq U^2$, a fórmula está correta.
2. Com qualquer universo U tal que $P^{\mathcal{M}} = U^2$ a fórmula não está correta.

Solução do exercício 3.8.

1. Escolhe $U = \{a, b\}$ e $P^{\mathcal{M}} = \{(a, b), (b, a)\}$.
2. Escolhe $U = \{a, b\}$, $P^{\mathcal{M}} = \{a\}$ e $Q^{\mathcal{M}} = \{b\}$.
3. Escolhe a mesma estrutura do item anterior.
4. Escolhe $U = \{a, b\}$, $P^{\mathcal{M}} = \{a\}$ e $Q^{\mathcal{M}} = \{a, b\}$.

Solução do exercício 3.9.

1. $P(x) \vdash \exists x P(x)$

1	$P(x)$	premissa
2	$\exists x P(x)$	$\exists x$i, $(t \equiv x)$

2. $\forall x P(x) \vdash \forall x \forall x P(x)$

1	$\forall x P(x)$	premissa
2	$\mathbf{x_0}$	(qualquer x_0)
3	$\forall x P(x)$	cópia 1
4	$\forall x \forall x P(x)$	$\forall x$i 3

3. $\exists x P(x), \forall x P(x) \to Q(x) \vdash \exists x Q(x)$

1	$\exists x P(x)$	premissa
2	$\forall x P(x) \to Q(x)$	premissa
3	$\mathbf{x_0}$ $P(x_0)$	hipótese
4	$P(x_0) \to Q(x_0)$	$\forall x$e 2 $(t \equiv x_0)$
5	$Q(x_0)$	$\to_e$ 3,4
6	$\exists x Q(x)$	$\exists x$i 5, $(t \equiv x_0)$
7	$\exists x Q(x)$	$\exists x$e 2,3–6

4. $\exists x P(x) \vdash \neg \forall x \neg P(x)$

1	$\exists x P(x)$	premissa
2	$\forall x \neg P(x)$	hipótese
3	$\mathbf{x_0}$ $\quad P(x_0)$	hipótese (para $\exists x$i 1)
4	$\neg P(x_0)$	$\forall x$e 2
5	$\perp$	$\neg_e$ 3,4
6	$\perp$	$\exists x$e 1,3–5
7	$\neg \forall x \neg P(x)$	PBC 2–6

5. $\neg \forall x \neg P(x) \vdash \exists x P(x)$

1	$\neg \forall x \neg P(x)$	premissa
2	$\neg \exists x P(x)$	hipótese
3	$\mathbf{x_0}$	(qualquer x_0)
4	$P(x_0)$	hipótese
5	$\exists x P(x)$	$\exists x$i 4 $(t \equiv x_0)$
6	$\perp$	$\neg_e$ 5,2
7	$\neg P(x_0)$	PBC 4–6
8	$\forall x \neg P(x)$	$\forall x$i 3–7
9	$\perp$	$\neg_e$ 8,1
10	$\exists x P(x)$	PBC 2–9

6. $\vdash \forall x P(x) \rightarrow P(x)$

1	$\mathbf{x_0}$	(qualquer x_0)
2	$P(x_0)$	hipótese
3	$P(x_0) \rightarrow P(x_0)$	$\rightarrow_i$ 2–2
4	$\forall x P(x) \rightarrow P(x)$	$\forall x$i 1–3

7. $\forall x P(x) \wedge Q(x) \vdash (\forall x P(x)) \wedge (\forall x Q(x))$

1	$\forall x P(x) \wedge Q(x)$	premissa
2	$\mathbf{x_0}$	(qualquer x_0)
3	$P(x_0) \wedge Q(x_0)$	$\forall x$e 1 $(t \equiv x_0)$
4	$P(x_0)$	$\wedge_{e_1}$ 3
5	$\forall x P(x)$	$\forall x$i 2–4
6	$\mathbf{x_0}$	(qualquer x_0)
7	$P(x_0) \wedge Q(x_0)$	$\forall x$e 1 $(t \equiv x_0)$
8	$Q(x_0)$	$\wedge_{e_2}$ 7
9	$\forall x Q(x)$	$\forall x$i 6–8
10	$(\forall x P(x)) \wedge (\forall x Q(x))$	$\wedge_i$ 5,8

8. $(\forall x P(x) \rightarrow Q(x)) \wedge (\forall x Q(x) \rightarrow R(x)) \vdash (\forall x P(x) \rightarrow R(x))$

1	$\forall x P(x) \to Q(x)$	premissa
2	$\forall x Q(x) \to R(x)$	premissa
3	$\mathbf{x_0}$	(qualquer x_0)
4	$P(x_0) \to Q(x_0)$	$\forall x$e 1
5	$Q(x_0) \to R(x_0)$	$\forall x$e 1
6	$P(x_0)$	hipótese
7	$Q(x_0)$	$\to_{\mathrm{e}}$ 6,4
8	$R(x_0)$	$\to_{\mathrm{e}}$ 7,5
9	$P(x_0) \to R(x_0)$	$\to_{\mathrm{i}}$ 6–8
10	$\forall x P(x) \to R(x)$	$\forall x$i 3–9

Solução do exercício 3.10.
Prova os seguintes seqüentes usando a dedução natural:

1. $\exists x(S \to Q(x)) \vdash S \to \exists x Q(x)$

1	$\exists x S \to Q(x)$	premissa
2	S	hipótese
3	$\mathbf{x_0} \quad S \to Q(x_0)$	hipótese
4	$Q(x_0)$	MP 2,3
5	$\exists x Q(x)$	$=_{\mathrm{i}}$ 4
6	$\exists x Q(x)$	$=_{\mathrm{e}}$ 1,3–5
7	$S \to \exists x Q(x)$	$\to_{\mathrm{i}}$ 2–6

2. $(\forall x P(x) \to S \vdash \exists x(P(x) \to S)$

1	$(\forall x P(x)) \to S$	premissa
2	$(\forall x P(x)) \vee \neg(\forall x P(x))$	LEM
3	$\forall x P(x)$	hipótese
4	$P(y)$	hipótese
5	S	MP 3,1
6	$P(y) \to S$	$\to_{\mathrm{i}}$ 4–5
7	$\exists x P(x) \to S$	$\exists x$i 6
8	$\neg(\forall x P(x))$	hipótese
9	$\exists x \neg P(x)$	Teorema (nu)
10	$\mathbf{x_0} \neg P(x_0)$	hipótese
11	$P(x_0)$	hipótese
12	$\perp$	$\neg_{\mathrm{e}}$ 11,10
13	S	$\perp_{\mathrm{e}}$ 12
14	$P(x_0) \to S$	$\to_{\mathrm{i}}$ 11-13
15	$\exists x P(x) \to S$	$\exists x$i 14
16	$\exists x P(x) \to S$	$\exists x$e 9,10-15
17	$\exists x P(x) \to S$	$\vee_{\mathrm{e}}$ 2,3–7,8–16

3. $\exists x f(x) = x \vdash \exists x f(f(x)) = x$

1	$\exists x f(x) = x$	premissa
2	$\mathbf{x_0}$ $f(x_0) = x_0$	hipótese
3	$f(f(x_0)) = f(f(x_0))$	$=_\mathrm{i}$ $(t \equiv f(f(x_0)))$
4	$f(f(x_0)) = f(x_0)$	$=_\mathrm{e}$ $(\Phi \equiv f(f(x_0)) = f(x))$
5	$f(f(x_0)) = x_0$	$=_\mathrm{e}$ $(\Phi \equiv f(f(x_0)) = x)$
6	$\exists x f(f(x)) = x$	$\exists x$i 5
7	$\exists x f(f(x)) = x$	$\exists x$e 1,2–6

4. $P(b) \vdash \forall x(x = b \to P(x))$

1	$P(b)$	premissa
2	$\mathbf{x_0}$	(qualquer x_0)
3	$x = b$	hipótese
4	$b = x$	(simetria da identidade)
5	$P(x)$	$=_\mathrm{e}$ $(\Phi \equiv P(x))$
6	$x = b \to P(x)$	$\to_\mathrm{i}$ 3–5
7	$\forall x\,(x = b \to P(x))$	$\forall x$i 2–6

5. $\exists x \exists y (H(x,y) \land H(y,x)), \neg\exists x H(x,x) \vdash \exists x \exists y \neg(x = y)$

1	$\exists x \exists y H(x,y) \land H(y,x)$	premissa
2	$\neg\exists x H(x,x)$	premissa
3	$\neg\exists x \exists y \neg(x = y)$	hipótese
4	$\forall x \neg(\exists y \neg(x = y))$	(generalização teorema ne)
5	$\forall x \forall y \neg\neg(x = y)$	(generalização teorema ne)
6	$\mathbf{x_0} \exists y H(x_0, y) \land H(y, x_0)$	hipótese
7	$\forall y \neg\neg(x_0 = y)$	$\forall x$e 5
8	$\mathbf{y_0}$ $H(x_0, y_0) \land H(y_0, x_0)$	hipótese
9	$\neg\neg x_0 = y_0$	$\forall x$e 7
10	$x_0 = y_0$	$\neg\neg_\mathrm{e}$ 9
11	$H(x_0, y_0)$	$\land_{\mathrm{e}_1}$ 8
12	$H(y_0, y_0)$	$=_\mathrm{e}$ 11 $(\Phi \equiv H(x, y_0))$
13	$\exists x H(x,x)$	$\exists x$i 12
14	$\bot$	$\neg_\mathrm{e}$ 13,2
15	$\bot$	$\exists x$e 6,8–14
16	$\bot$	$\exists x$e 1,6–15
17	$\exists x \exists y \neg(x = y)$	PBC 3–16

com a seguinte generalização do teorema (ne)

$$\neg\exists x \Phi \dashv\vdash \forall x \neg\Phi$$

para qualquer fórmula Φ (a prova é igual, só usando Φ ao invés de $P(x)$).

6. $\forall x(P(x) \leftrightarrow x = b) \vdash P(b) \land \forall x \forall y (P(x) \land P(y) \to x = y)$

1	$\forall x((P(x) \to x = b) \wedge (x = b \to P(x)))$	premissa
2	$(\forall x P(x) \to x = b) \wedge (\forall x(x = b \to P(x)))$	(teorema duc)
3	$\forall x P(x) \to x = b$	$\wedge_{e_1}$ 2
4	$\forall x(x = b \to P(x))$	$\wedge_{e_2}$ 2
5	$b = b$	$=_\mathrm{i}$
6	$b = b \to P(x)$	$\forall x$e 4
7	$P(b)$	MP 5,6
8	$\mathbf{x_0}$	(qualquer x_0)
9	$\mathbf{y_0}$	(qualquer y_0)
10	$P(x_0) \wedge P(y_0)$	hipótese
11	$P(x_0)$	$\wedge_{e_1}$ 10
12	$P(x_0) \to x_0 = b$	$\forall x$e 3
13	$x_0 = b$	MP 11,12
14	$P(y_0)$	$\wedge_{e_2}$ 10
15	$P(y_0) \to y_0 = b$	$\forall x$e 3
16	$y_0 = b$	MP 14,15
17	$b = y_0$	(simetria da identidade) 16
18	$x_0 = y_0$	$=_\mathrm{e}$ 17,13
19	$P(x_0) \wedge P(y_0) \to x_0 = y_0$	$\to_\mathrm{i}$ 10-18
20	$\forall y P(x_0) \wedge P(y) \to x_0 = y$	$\forall x$i 9–19
21	$\forall x \forall P(x) \wedge P(y) \to x = y$	$\forall x$i 8–20
22	$P(b) \wedge \forall x \forall P(x) \wedge P(y) \to x = y$	$\wedge_\mathrm{i}$ 7,21

Solução do exercício 3.11.

1. $\exists x(S \to Q(x)) \vdash S \to \exists x Q(x)$

1		$\exists x S \to Q(x)$		premissa
2		$\neg(S \to \exists x Q(x))$		negação da conclusão
3		S		regra $\neg \to$ 2
4		$\neg \exists x Q(x)$		regra $\neg \to$ 2
5		$\forall x \neg Q(x)$		regra $\neg\exists$4
6		$S \to Q(a)$		regra $\exists$ 1
7		$\neg Q(a)$		regra $\forall$ 5
8	$\neg S$		$Q(a)$	regra $\to$ 7
	$\times$		$\times$	

2. $(\forall x P(x)) \to S \vdash \exists x(P(x) \to S)$

1	$(\forall x P(x)) \to S$	premissa
2	$\neg \exists x (P(x) \to S)$	negação da conclusão
3	$\forall x \neg (P(x) \to S)$	regra $\neg\exists$ 2

4	$\neg \forall x P(x)$	regra $\to$ 1	11	S	regra $\to$ 1
5	$\exists x \neg P(x)$	regra $\neg\forall$ 4	12	$\neg(P(a) \to S)$	regra $\forall$ 3
6	$\neg P(a)$	regra $\exists$ 5	13	$P(a)$	regra $\neg \to$ 12
7	$\neg(P(a) \to S)$	regra $\forall$ 3	14	$\neg S$	regra $\neg \to$ 12
8	$P(a)$	regra $\neg \to$ 7		$\times$	
9	$\neg S$	regra $\neg \to$ 7			
10	$\times$				

3. $\exists x f(x) = x \vdash \exists x f(f(x)) = x$

1	$\exists x f(x) = x$	premissa
2	$\neg \exists x f(f(x)) = x$	negação da conclusão
3	$\forall x \neg (f(f(x)) = x)$	regra $\neg\exists$ 2
4	$f(a) = a$	regra $\exists$ 1
5	$\neg(f(f(a)) = a)$	regra $\forall$ 3
6	$\neg(f(a) = a)$	regra $=$ 4,5
7	$\neg(a = a)$	regra $=$ 4,6
7	$\times$	regra $\neg =$

4. $P(b) \vdash \forall x (x = b \to P(x))$

1	$P(b)$	premissa
2	$\neg \forall x (x = b \to P(x))$	negação da conclusão
3	$\exists x \neg (x = b \to P(x))$	regra $\neg\forall$ 2
4	$\neg(a = b \to P(a))$	regra $\exists$ 3
5	$a = b$	regra $\neg \to$ 4
6	$\neg P(a)$	regra $\neg \to$ 4
7	$\neg P(b)$	regra $=$ 5,6
8	$\times$	

5. $\exists x \exists y (H(x,y) \wedge H(y,x)), \neg \exists x H(x,x) \vdash \exists x \exists y \neg (x = y)$

1	$\exists x \exists y H(x,y) \wedge H(y,x)$	premissa
2	$\neg\exists x H(x,x)$	premissa
3	$\neg\exists x\exists y\neg(x=y)$	negação da conclusão
4	$\forall\neg H(x,x)$	regra $\neg\exists$ 2
5	$\forall x\neg\exists y\neg(x=y)$	regra $\neg\exists$ 3
6	$\exists y H(a,y) \wedge H(y,a)$	regra $\exists$ 1
7	$H(a,b) \wedge H(b,a)$	regra $\exists$ 6
8	$H(a,b)$	regra $\wedge$ 7
9	$H(b,a)$	regra $\wedge$ 7
10	$\neg\exists y\neg(a=y)$	regra $\forall$ 5
11	$\forall y\neg\neg(a=y)$	regra $\neg\exists$ 10
12	$\neg\neg(a=b)$	regra $\forall$ 11
13	$a=b$	regra $\neg\neg$ 12
14	$H(b,b)$	regra $=$ 13,8
15	$\neg H(b,b)$	regra $\forall$ 4
16	$\times$	

6. $\forall x(P(x) \leftrightarrow x=b) \vdash P(b) \wedge \forall x\forall y(P(x) \wedge P(y) \rightarrow x=y)$

1			$\forall x P(x) \rightarrow x = b \wedge x = b \rightarrow P(x)$	premissa
2			$\neg(P(b) \wedge \forall x \forall y (P(x) \wedge P(y) \rightarrow x = y))$	neg. da conclu
3	$\neg P(b)$			regra $\neg\wedge$ 2
4	$P(b) \rightarrow b = b \wedge b = b \rightarrow P(b)$			regra $\forall$ 1
5	$P(b) \rightarrow b = b$			regra $\wedge$ 4
6	$b = b \underset{\vee}{\rightarrow} P(b)$			regra $\wedge$ 4
7	$\neg(b = b)$	$P(b)$		
8	$\times$	$\times$		
9			$\neg\forall x \forall y (P(x) \wedge P(y) \rightarrow x = y)$	regra $\neg\wedge$ 2
10			$\exists x \neg\forall y (P(x) \wedge P(y) \rightarrow x = y)$	regra $\neg\forall$ 9
11			$\neg\forall y P(a) \wedge P(y) \rightarrow a = y$	regra $\exists$ 10
12			$\exists y \neg(P(a) \wedge P(y) \rightarrow a = y)$	regra $\neg\forall$ 11
13			$\neg(P(a) \wedge P(c) \rightarrow a = c)$	regra $\exists$ 12
14			$P(a) \wedge P(c)$	regra $\neg \rightarrow$13
15			$\neg(a = c)$	regra $\neg \rightarrow$13
16			$P(a)$	regra $\wedge$ 14
17			$P(c)$	regra $\wedge$ 14
18			$P(a) \rightarrow a = b \wedge a = b \rightarrow P(a)$	regra $\forall$ 1
19			$P(a) \rightarrow a = b$	regra $\wedge$ 18
20			$a = b \rightarrow P(a)$	regra $\wedge$ 18
21			$P(c) \rightarrow c = b \wedge c = b \rightarrow P(c)$	regra $\forall$ 1
22			$P(c) \rightarrow c = b$	regra $\wedge$ 21
23			$c = b \rightarrow P(c)$	regra $\wedge$ 21
24, 25	$\neg P(a)$		$a = b$	regra $\rightarrow$ 19
	$\times$			
26, 27	$\neg P(c)$		$c = b$	regra $\rightarrow$ 22
28	$\times$		$\neg(b = c)$	regra = 25,1
29			$\neg(b = b)$	regra = 27,2
			$\times$	

Solução do exercício 3.12.
Com predicados $R(x)$: "x é um retângulo", $L(x)$: "x é um quadrilateral" e $Q(x)$: "x é um quadrado" temos

1. Todos os retângulos são quadrilaterais: $\forall x R(x) \rightarrow L(x)$.
2. Algumas retângulos são quadrados: $\exists x R(x) \wedge Q(x)$.

3. Algumas quadrilaterias são quadrados: $\exists x L(x) \wedge Q(x)$.

Uma prova de $\forall x R(x) \to L(x), \exists x R(x) \wedge Q(x) \vdash \exists x L(x) \wedge Q(x)$ é

1	$\forall x R(x) \to L(x)$	premissa
2	$\exists x R(x) \wedge Q(x)$	premissa
3	$R(x_0) \wedge Q(x_0)$	hipótese
4	$R(x_0)$	$\wedge_{\mathrm{e}_1}$ 3
5	$R(x_0) \to L(x_0)$	$\forall x$e 1
6	$L(x_0)$	MP 4,5
7	$Q(x_0)$	$\wedge_{\mathrm{e}_2}$ 3
8	$L(x_0) \wedge Q(x_0)$	$\wedge_{\mathrm{i}}$ 6,7
9	$\exists x L(x) \wedge Q(x)$	$\exists x$i 8
10	$\exists x L(x) \wedge Q(x)$	$\exists x$e 2,3–9

Solução do exercício 3.13.
Para $\otimes$ podemos introduzir as seguintes regras novas para árvores de refutação (elas são uma conseqüência das regras existentes):

$$\begin{array}{ccc} & p \otimes q & \\ p \swarrow & & \searrow \neg p \\ \neg q & & q \end{array} \qquad \begin{array}{ccc} & \neg(p \otimes q) & \\ p \swarrow & & \searrow \neg p \\ q & & \neg q \end{array}$$

1. $(p \wedge \neg q) \vee (\neg p \wedge q) \vdash (p \vee q) \wedge \neg(p \wedge q)$ é valido:

1	$(p \wedge \neg q) \vee (\neg p \wedge q)$	premissa
2	$p \wedge \neg q$	hipótese
3	p	$\wedge_{\mathrm{e}_1}$ 2
4	$\neg q$	$\wedge_{\mathrm{e}_2}$ 2
5	$p \vee q$	$\vee_{\mathrm{i}_1}$ 3
6	$\neg p \vee \neg q$	$\vee_{\mathrm{i}_2}$ 5
7	$\neg(p \wedge q)$	de Morgan 6
8	$(p \vee q) \wedge \neg(p \wedge q)$	$\wedge_{\mathrm{i}}$ 7
9	$\neg p \wedge q$	hipótese
10	$\neg p$	$\wedge_{\mathrm{e}_1}$ 9
11	q	$\wedge_{\mathrm{e}_2}$ 9
12	$\neg p \vee \neg q$	$\vee_{\mathrm{i}_1}$ 10
13	$p \vee q$	$\vee_{\mathrm{e}}2$ 11
14	$\neg(p \wedge q)$	de Morgan 12
15	$(p \vee q) \wedge \neg(p \wedge q)$	$\wedge_{\mathrm{i}}$ 13,14
16	$(p \vee q) \wedge \neg(p \wedge q)$	$\vee_{\mathrm{e}}$ 1,2–8,9–15

2. $\forall y \exists x P(x, y) \vdash \exists x \forall y P(x, y)$

3. $\forall x(P(x) \otimes Q(x)) \vdash (\forall x P(x)) \otimes (\forall x Q(x))$
   A árvore não fecha. Um contra-exemplo é $U = \{a, b\}$ com $P = \{a\}$ e $Q = \{b\}$.

4. $(\forall x P(x)) \otimes (\forall x Q(x)) \vdash \forall x(P(x) \otimes Q(x))$
   A árvore não fecha. Um contra-exemplo é $U = \{a, b\}$ com $P = U$ e $Q = \{a\}$.

5. $\exists x(P(x) \otimes Q(x)) \vdash (\exists x P(x)) \otimes (\exists x Q(x))$
   A árvore não fecha. Um contra-exemplo é $U = \{a, b\}$ com $P = \{a\}$ e $Q = \{b\}$.

6. $(\exists x P(x)) \otimes (\exists x Q(x)) \vdash \exists x(P(x) \otimes Q(x))$ é valido

$$
\begin{array}{c}
(\exists x P(x)) \otimes (\exists x Q(x)) \\
\neg\exists x(P(x) \otimes Q(x)) \\
\forall x \neg(P(x) \otimes Q(x))
\end{array}
$$

$$
\begin{array}{ccc}
\begin{array}{c}
\exists x P(x) \\
\neg\exists x Q(x) \\
\forall \neg Q(x) \\
P(a) \\
\neg Q(a) \\
\neg(P(a) \otimes Q(a)) \\
\begin{array}{cc}
P(a) & \neg P(a) \\
Q(a) & \neg Q(a) \\
\times & \times
\end{array}
\end{array}
& \qquad &
\begin{array}{c}
\neg\exists x P(x) \\
\exists x Q(x) \\
\forall x \neg P(x) \\
Q(a) \\
\neg P(a) \\
\neg(P(a) \otimes Q(a)) \\
\begin{array}{cc}
P(a) & \neg P(a) \\
Q(a) & \neg Q(a) \\
\times & \times
\end{array}
\end{array}
\end{array}
$$

7. $\exists x(P(x) \otimes Q(x)) \vdash (\exists x P(x)) \otimes (\forall x Q(x))$

1			$\exists x P(x) \otimes Q(x)$ $\surd$			premissa
2			$\neg((\exists x P(x)) \otimes (\forall x Q(x)))$ $\surd$			negação da co
3			$P(a) \otimes Q(x)$			$\exists 1$
4	$\exists x P(x)$			$\neg\exists x P(x)$		$\neg \otimes 2$
5	$\forall x Q(x)$			$\neg\forall x Q(x)$		$\neg \otimes 2$
6				$\forall x \neg P(x)$		$\neg\exists 4$
7				$\exists x \neg Q(x)$		$\neg\forall 5$
8	$P(a)$	$\neg P(a)$		$P(a)$	$\neg P(a)$	$\otimes 3$
9	$\neg Q(a)$	$Q(a)$		$\neg Q(a)$	$Q(a)$	$\otimes 3$
10	$Q(x)$	$P(b)$		$\neg P(a)$	$\neg Q(b)$	
11	$\times$	$Q(b)$		$\times$	$\neg P(b)$	
12		$\odot$			$\odot$	

A árvore não fecha. Um contra-exemplo é $U = \{a, b\}$ com $P = \{b\}$ e $Q = U$.

8. $(\exists x P(x)) \otimes (\forall x Q(x)) \vdash \exists x(P(x) \otimes Q(x))$
   A árvore não fecha. Um contra-exemplo é $U = \{a, b\}$ com $P = Q = \{a\}$.

9. $\forall x(P(x) \otimes Q(x)) \vdash (\exists x P(x)) \otimes (\forall x Q(x))$ é valido.

		$\forall x P(x) \otimes Q(x)$		
		$\neg((\forall x P(x)) \otimes (\exists x Q(x)))$		
$\forall x P(x)$			$\neg\forall x P(x)$	
$\exists x Q(x)$			$\neg\exists x Q(x)$	
			$\exists x \neg P(x)$	
			$\forall x \neg Q(x)$	
$Q(a)$			$\neg P(a)$	
$P(a)$			$\neg Q(a)$	
$P(a) \otimes Q(a)$			$P(a) \otimes Q(a)$	
$P(a)$	$\neg P(a)$		$P(a)$	$\neg P(a)$
$\neg Q(a)$	$Q(a)$		$\neg Q(a)$	$Q(a)$
$\times$	$\times$		$\times$	$\times$

10. $(\exists x P(x)) \otimes (\forall x Q(x)) \vdash \forall x(P(x) \otimes Q(x))$
    A árvore não fecha. Um contra-exemplo é $U = \{a, b\}$ com $P = \{a\}$, $Q = U$.

Como resultado desses exercícios, o seguinte tabela contém a regras da distribuição da quantificação sobre $\otimes$: Com

$$A \equiv Q_1 x P(x) \otimes Q(x), \qquad B \equiv (Q_2 x P(x)) \otimes (Q_3 x Q(x))$$

temos

	Q_1	Q_2	Q_3	$A \vdash B$	$B \vdash A$	Justificação
1	$\exists$	$\exists$	$\exists$	f	v	(e), (f)
2	$\exists$	$\exists$	$\forall$	f	v	(g), (h)
3	$\exists$	$\forall$	$\exists$	f	v	Simetría linha 2
4	$\exists$	$\forall$	$\forall$	f	v	$P = \emptyset, Q = \{a\}$ // Árvore fecha
5	$\forall$	$\exists$	$\exists$	f	f	$P = \{a\}, Q = \{b\}$ // $P = \{b\}, Q = \emptyset$
6	$\forall$	$\exists$	$\forall$	v	f	(i), (j)
7	$\forall$	$\forall$	$\exists$	v	f	Simetría linha 6
8	$\forall$	$\forall$	$\forall$	f	f	(c), (d)

C. Breve história da lógica

Visão geral

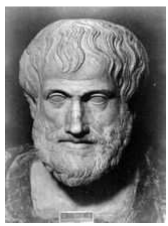
- Lógica: em grego $\lambda o \gamma \iota \kappa \eta$, "a arte ou método pensativa".
- A lógica que nos aprendemos parece compacto, razoável e compreensível.
- Mas nossos sistema de lógica são o resultado de 2000 anos de pesquisa.
- Na história da lógica ocidental, tem quatro fases importantes
  - Período clássico (350–200): Grécia
  - Boethius (480-524/525)
  - Idade média (1150–1450): Abaelardus, Ockham, Buridan, Burley.
  - Leibniz (1646–1716)
  - Período moderno (1850–): Boole, Peano, de Morgan, Pierce, Schröder, Frege, Bernays, Hilbert, Gentzen, Gödel, Löwenheim,...

> Toda a história da lógica consiste em definir um conceito aceitável de estupidez. Umberto Eco, Pêndulo de Foucault

Período clássico

Aristóteles

> Não é possível que a mesma qualidade contém e não contém na mesma objeto [...] Isso é o mais certa de todos princípios [...] Por isso, os que demonstram se referem ao isso como uma opinião ultimativa. Por que é per natura o fonte de todos os outros axiomas.
>
> Metaphysica, 3, III.



Aristóteles (*384, +322)

- Aristóteles é considerado como fundador da lógica.
- A lógica foi um ferramenta ("Organon") para ele.

Silogismos

- Silogismo é o palavra grego para "conclusão" ou "interferência".
- Os silogismos de Aristóteles foram o primeiro sistema de lógica.
- Eles ficavam o sistema lógico mais importante até a idade média.

Silogismos

Um silogismo é composto de

- Três proposições (dois premissas e uma conclusão) da forma
  - A: Cada S é P. (**A**firmo.)
  - E: Nenhum S é P ou Cada S não é P. (N**e**go.)
  - I: Algum S é P. (Af**i**rmo.)
  - O: Algum S não é P. (Neg**o**.)
- Das $4^3 = 256$ possibilidades, 24 deles são "válidas".

Exemplo

Camestres:

Cada S é P. Nenhum T é P. Logo, nenhum T é S.

(Coloca S:Humano, P: Mortal. T: Deus.)

Barbara:

Cada S é P. Cada T é S. Logo, cada T é P.

(Coloca S:Humano, P: Mortal, T: Homem.)

Boethius

- Varias traduções de Aristóteles.
- Comentários das trabalhos de Aristóteles.
- Trabalhos sobre silogismos.

Boethius (*480, +524/525)

Período moderna

Leibniz

- Idéia de uma "linguagem universal" como fundamento da matemática.
- Noções básicas da lógica ("do verdadeiro só pode deduzir verdadeiro")
- Princípio da identidade (extensional): Dois objetos são idênticos se eles tem o mesmo "comportamento" em qualquer contexto.

Gottfried Wilhelm Leibniz (*1646, +1716)

George Boole

Por que uma teoria da lógica?

> In the more complex examples of logical deduction [...], the aid of a directive method, such as a Calculus alone can supply, is indispensible. A investigation of the Laws of Thought - I.

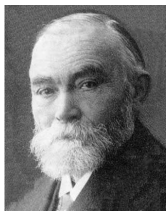
- George Boole (1815-1864).
- Boole mostrou os limites da lógica de Aristóteles.
- Ele inventou uma lógica matemática (álgebra booleana).

Exemplo: "Lei da dualidade"

$$x(1 - x) = 0$$

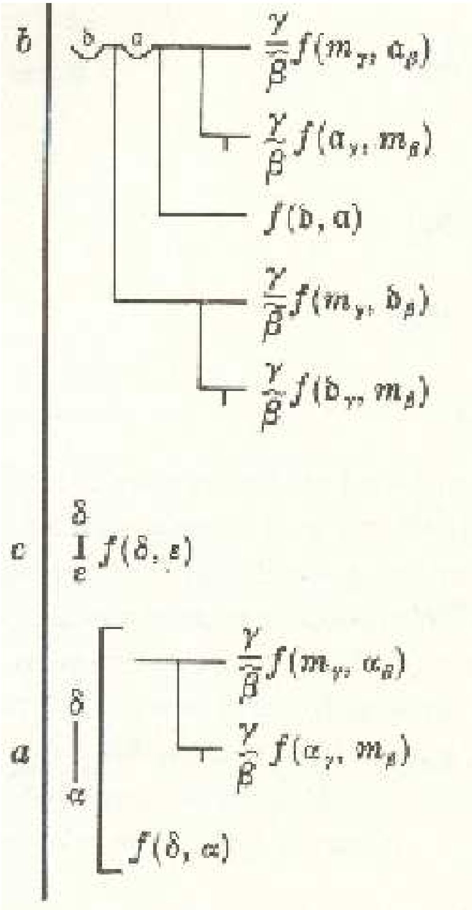
Frege

- 1879: Begriffsschrift ("conceitografia").
- Ele criou os primeiros sistemas axiomáticas de lógica (lógica de predicados e da primeira ordem).



Gottlob Frege (*1848, +1925)

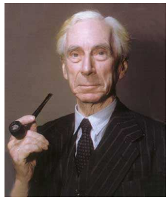
Begriffsschrift



Whitehead e Russell

- O conjunto de todos os conjuntos que não contém si mesmo:
$$C = \{C' | C' \notin C'\}.$$
- $C \in C$?
- O descobrimento de contradições na teoria de conjuntos...
- "Principia matemática": Tentativa de um fundamento lógico para a matemática.
- Idéia: Sistemas de tipos...

Alfred North Whitehead (*1861, +1947)



Bertrand Russell (*1872, +1970)

David Hilbert

- 1904: O programa de Hilbert.
  - Hipótese do Contínuo (1).
  - Consistência da aritmética (2).
- 1928: O "Entscheidungsproblem".

David Hilbert (*1862, +1943)

Hilbert e Gödel

- Teorema da completude da lógica de predicados (1929).
- Teorema da incompletude da lógica de predicados (1931).
- Independência da Hipótese do Contínuo da lógica de predicados.

Kurt Gödel
(*1906, +1978)

Observe que a noções da completude e incompletude fala sobre diferentes noções de "completude". O primeiro teorema mostra a completude da lógica de predicados no sentido da definição 2.3. No segundo teorema, "completo" é a característica de um sistema de axiomas de ser capaz de provar para cada fórmula Φ ou ela mesma ou a sua negação. O resultado do Gödel foi, que qualquer sistema da axiomas da lógica de predicados que é capaz de formalizar a aritmética ou e inconsistente (i.e. pode ser provado uma fórmula e a sua negação) ou incompleto.

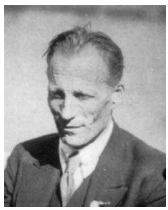
Lógicas não-clássicas

Lógicas não-clássicas

- Lógicas de ordem superior.
- Lógica intuicionista.
- Lógica fuzzy.
- Lógica modal, para-consistente, relevante, ...

Lógica intuicionista

- Idéia: Provas construtivas.
- Regras problemáticas: PBC, LEM, $\neg\neg_e$.
- Construtividade: é uma idéia que corresponde bem com a computação!



Luitzen Egbertus
Jan Brouwer
(*1881, +1966)

Exemplo 10
Existem a, b irracionais, tal que a^b é racional.
Prova. Seja $b = \sqrt{2}$. (a) Seja b^b racional. Escolhe $a = \sqrt{2}$. (b) Seja b^b irracional. Escolhe $a = \sqrt{2}^{\sqrt{2}}$ (que é irracional). ■
◊

Aplicações da lógica

- Linguagens de programação.
- Sistemas de tipos.
- Banco de dados.
- Complexidade de algoritmos.
- Inteligência artificial.
- Verificação de sistemas.
- Segurança (p.ex. proof-carrying code).

Bibliografia

[1] M. Ben-Ari. *Mathematical logic for computer science.* Springer, second edition, 2001. INF:510.6 B456m.

[2] G. Gentzen. Untersuchungen über das logische Schließen I. *Mathematische Zeitschrift*, 39:176–210, 1934.

[3] G. Gentzen. Untersuchungen über das logische Schließen II. *Mathematische Zeitschrift*, 39:405–431, 1934.

[4] K. Gödel. *Über die Vollständigkeit des Logikkalküls.* PhD thesis, Universität Wien, 1929.

[5] K. Gödel. Über die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik*, 37:349–360, 1930.

[6] D. E. Knuth. The Art of Computer Programming, pre-fascicle 0b: Boolean basics, 2006.

[7] E. Post. Introduction to a general theory of elementary propositions. *Amer. Journ. Math.*, 43:163–185, 1921.

Índice