

Lógica Computacional 1 (2023-1)
April 10, 2023
Flávio L. C. de Moura

Chapter 1

A Lógica Proposicional

[aula 01 (0 exercícios)]

Iniciaremos nosso estudo a partir da noção de **proposição**, que é uma afirmação que pode ser qualificada como verdadeira ou falsa. Por exemplo, afirmações como “ $2+2=4$ ”, “2 é um número primo”, “ $1+3i0$ ” ou “João tem 20 anos e Maria tem 22 anos” são proposições. Mas nem toda afirmação é uma proposição. De fato, a ordem “Feche a porta!”, ou ainda, a pergunta “Qual é o seu nome?” não podem ser qualificadas como verdadeira ou falsa, e portanto não são proposições. Algumas proposições podem ser divididas em proposições menores. Por exemplo, a proposição “João tem 20 anos e Maria tem 22 anos” é composta da proposição “João tem 20 anos” e da proposição “Maria tem 22 anos”, que por sua vez não podem mais serem divididas porque os pedaços menores não são mais qualificáveis como verdadeiro ou falso. Uma proposição que não pode ser dividida é um elemento básico utilizado na construção de proposições mais complexas, que chamaremos de fórmula atômica. Utilizaremos letras latinas minúsculas para representar fórmulas atômicas. Por exemplo, podemos utilizar a letra q para representar a proposição “Maria tem 22 anos”, e a letra p para “João tem 20 anos”. A proposição do exemplo acima é construída com a utilização do conectivo “E” (conjunção), que será representado pelo símbolo \wedge . Com esta simbologia, podemos codificar a proposição “João tem 20 anos e Maria tem 22 anos” pela fórmula $p \wedge q$. E o que é uma fórmula? As fórmulas são os elementos sintáticos que vão compor a teoria com a qual iremos trabalhar, a saber a Lógica Proposicional. Além da conjunção, existem outros conectivos que compõem as fórmulas da Lógica Proposicional. Estes conectivos estão na gramática das fórmulas da Lógica Proposicional representada a seguir:

$$\varphi ::= p \mid \perp \mid (\varphi \rightarrow \varphi) \mid (\neg \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \quad (1.1)$$

A gramática (1.1) possui 6 construtores:

1. O primeiro denota uma variável proposicional, e caracteriza uma fórmula atômica, i.e. uma fórmula que não pode ser subdividida em fórmulas menores.
2. O segundo construtor é uma constante que denota o absurdo (\perp), que também é uma fórmula atômica. O absurdo será utilizado quando tivermos informações contraditórias em nossas provas. Isto ficará mais claro nos exemplos.

3. O terceiro construtor é a implicação.
4. O quarto construtor denota a negação.
5. O quinto construtor denota a conjunção.
6. O sexto construtor denota a disjunção.

Uma gramática como (1.1) nos fornece as regras sintáticas para a construção das fórmulas da Lógica Proposicional. São quatro construtores recursivos (negação, conjunção, disjunção e implicação) também chamados de conectivos lógicos, e dois não recursivos. Apesar da gramática apresentada acima não incluir a bi-implicação, este é um conectivo bastante utilizado, e pode ser escrito em função dos outros conectivos: $\varphi \leftrightarrow \psi$ é o mesmo que $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$. Na verdade, a gramática apresentada possui redundâncias, isto é, conectivos que podem ser escritos em função de outros, mas veremos isto posteriormente.

A seguir, mostraremos como utilizar os elementos da gramática acima no assistente de provas Coq. As variáveis proposicionais são declaradas como elementos do tipo `Prop`. A declaração abaixo é feita dentro de uma seção para delimitar o escopo da declaração no arquivo:

```
Section aula1.
Variables p q: Prop.
End aula1.
```

Uma declaração global pode ser feita como a palavra reservada `Parameter`:

```
Parameter p q: Prop.
```

A constante \perp (absurdo) está definida em Coq como `False`. Podemos verificar isto solicitando que o sistema retorne o tipo de `False`:

```
Check False.
```

O retorno do comando acima é:

```
False
  : Prop
```

Ou seja, `False` tem tipo `Prop`. A implicação é um conectivo binário que recebe duas fórmulas como argumento:

```
Check (p -> q).
```

O retorno para o comando acima é:

```
p -> q
  : Prop
```

Observe que a forma de representar a implicação no Coq é o hífen seguido do sinal de maior. A fórmula dada como argumento para o comando `Check` é escrito da mesma forma, mas a saída pdf deste arquivo transforma a versão ascii da implicação.

A negação é um conectivo unário que recebe uma fórmula como argumento. Escrevemos $\neg p$ para representar a negação da variável p declarada anteriormente. No Coq, a negação é representada pelo símbolo `~` antes da fórmula, mas na saída pdf deste arquivo a negação está aparecendo com outro símbolo. Podemos solicitar o tipo da fórmula $\neg p$:

```
Check (~ p).
```

Como esperado, obtemos o tipo `Prop`:

```
~ p
  : Prop
```

A notação apresentada acima para a negação é, na verdade, uma abreviação para uma implicação particular. De fato, $\neg phi$ é o mesmo que $phi \rightarrow False$ como veremos na próxima seção. A conjunção é um conectivo binário escrito como a seguir:

```
Check (p & q).
```

```
p & q
  : Prop
```

E por fim, a disjunção é também um conectivo binário escrito como a seguir:

```
Check (p | q).
```

```
p | q
  : Prop
```

O sistema conhecido como dedução natural será utilizado para a construção das provas. Este sistema foi criado pelo lógico alemão Gerhard Gentzen (1909-1945), e consiste em um sistema lógico composto por um conjunto de regras de inferência que tenta capturar o raciocínio matemático da forma mais natural possível. Como veremos, estas regras nos permitem derivar novos fatos a partir das premissas. Os fatos a serem provados são representados por meio de fórmulas da Lógica Proposicional. Neste contexto, o primeiro conceito

importante que aparece é o de sequente. Formalmente, um sequente é um par cujo primeiro elemento é um conjunto finito de fórmulas (hipóteses), e o segundo elemento é uma fórmula (conclusão). Assim, se $\varphi_1, \varphi_2, \dots, \varphi_n$ são as hipóteses de um dado problema, e se ψ é a sua conclusão, escrevemos $\varphi_1, \varphi_2, \dots, \varphi_n \vdash \psi$ para representar o sequente que simboliza que ψ tem uma prova a partir das hipóteses $\varphi_1, \varphi_2, \dots, \varphi_n$. O conjunto $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$, isto é, a primeira componente do sequente $\varphi_1, \varphi_2, \dots, \varphi_n \vdash \psi$ também será chamado de contexto ao longo do texto, e normalmente será escrito sem as chaves que usualmente são usadas para representar conjuntos. Este é um abuso de linguagem usado para não deixar a notação sobrecarregada. Assim, se Γ denota um conjunto finito de fórmulas, ao invés de $\Gamma \cup \{\varphi\} \vdash \psi$, escreveremos simplesmente $\Gamma, \varphi \vdash \psi$, onde Γ, φ deve então ser lido como o conjunto que contém a fórmula φ e todas as fórmulas de Γ . O conceito de prova agora será definido de forma mais precisa. Concretamente, uma prova (ou uma derivação) de um sequente da forma $\Gamma \vdash \psi$ é uma árvore finita onde cada nó corresponde a uma regra válida. A raiz da árvore é anotada com a conclusão, ou seja, o sequente que queremos provar, e as folhas são anotadas com axiomas. As regras são representadas por regras de inferência, onde cada premissa e a conclusão correspondem a um sequente:

$$\frac{\Gamma_1 \vdash \gamma_1 \quad \Gamma_2 \vdash \gamma_2 \quad \dots \quad \Gamma_k \vdash \gamma_k}{\Gamma \vdash \psi}$$

onde $k \geq 0$. Quando $k = 0$ a regra corresponde a um axioma, se ψ é uma das fórmulas em Γ :

$$\frac{}{\Gamma \vdash \psi} \text{ (Ax), se } \psi \in \Gamma$$

Uma prova (sequência de passos dedutivos) pode ser representada por meio de uma estrutura de árvore, onde os nós são anotados com fórmulas. A raiz da árvore é anotada com a fórmula que queremos provar, isto é, uma prova do sequente $\Gamma \vdash \psi$, será representado por uma árvore que terá a fórmula ψ como raiz, e as folhas da árvore são fórmulas de Γ .

Como veremos, a construção desta árvore deve obedecer alguns critérios que detalharemos ao longo deste capítulo, mas em linhas gerais, o principal critério consiste em obedecer um conjunto de regras que definem o sistema de dedução natural. Estas regras são divididas em dois tipos: regras de introdução e regras de eliminação dos conectivos. Na próxima seção veremos as duas primeiras regras que nos permitirão resolver diversos problemas. Para isto, ficaremos restritos inicialmente a fórmulas que contêm apenas a implicação como conectivo. Esta parte da Lógica proposicional é conhecida como *fragmento implicacional da lógica proposicional*.

1.1 O Fragmento Implicacional da Lógica Proposicional

[aula 02 (5 exercícios)]

O fragmento implicacional da lógica proposicional é formado pelas fórmulas da lógica proposicional que contêm apenas a implicação como conectivo lógico, ou seja, consiste nas fórmulas construídas a partir da seguinte gramática:

$$\varphi ::= p \mid \perp \mid (\varphi \rightarrow \psi) \quad (1.2)$$

Este fragmento possui duas regras de derivação: uma regra de *introdução* e outra de *eliminação* da implicação. A regra de eliminação da implicação é bastante conhecida e também chamada de *modus ponens*:

$$\frac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi} (\rightarrow_e)$$

A regra (\rightarrow_e) nos diz que a partir de uma prova de $\Gamma \vdash \varphi \rightarrow \psi$ e de outra prova de $\Gamma \vdash \varphi$ podemos concluir que $\Gamma \vdash \psi$. As regras de introdução são bastante intuitivas e, em certo sentido, podem ser vistas como uma definição do conectivo que estão introduzindo: a regra de *introdução da implicação*, denotada por (\rightarrow_i) , possui alguns detalhes importantes. Para construirmos uma prova de uma implicação, digamos do sequente $\Gamma \vdash \varphi \rightarrow \psi$, precisamos construir uma prova de ψ tendo φ como hipótese adicional ao contexto Γ . Em outras palavras, na leitura de baixo para cima, reduzimos o problema de provar $\Gamma \vdash \varphi \rightarrow \psi$ ao novo problema (possivelmente mais simples) de provar $\Gamma, \varphi \vdash \psi$:

$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi} (\rightarrow_i)$$

Também podemos observar esta regra de cima para baixo. Neste caso, ela nos permite passar uma fórmula do conjunto de hipóteses para o conseqüente como antecedente de uma implicação. Assim, a fórmula φ que era uma das hipóteses necessárias para provar ψ , deixa de ser hipótese, e passa a ser antecedente de uma implicação no conseqüente. Considerando o subconjunto das fórmulas da lógica proposicional construídas apenas com a implicação (variáveis e a constante \perp) e as regras (\rightarrow_e) e (\rightarrow_i) temos o chamado *fragmento implicacional da lógica proposicional*. O interesse computacional deste fragmento está diretamente relacionado ao algoritmo de inferência de tipos em linguagens funcionais[?]. O fundamento teórico destas linguagens é o cálculo λ [?] desenvolvido por Alonzo Church em 1936[?, ?]. Para mais detalhes veja o Capítulo 1 de [?]. Como primeiro exemplo, vamos considerar o sequente $\vdash (p \rightarrow q) \rightarrow (q \rightarrow r) \rightarrow p \rightarrow r$. A primeira observação a ser feita aqui é que a implicação é associativa à direita, ou seja, $\varphi \rightarrow \psi \rightarrow \gamma$ deve ser lido como $\varphi \rightarrow (\psi \rightarrow \gamma)$, e não como $(\varphi \rightarrow \psi) \rightarrow \gamma$. Portanto, o sequente que queremos provar deve ser lido como $\vdash (p \rightarrow q) \rightarrow ((q \rightarrow r) \rightarrow (p \rightarrow r))$. Utilizando inicialmente a regra (\rightarrow_i) , temos a seguinte situação:

$$\frac{p \rightarrow q \vdash (q \rightarrow r) \rightarrow (p \rightarrow r)}{\vdash (p \rightarrow q) \rightarrow ((q \rightarrow r) \rightarrow (p \rightarrow r))} (\rightarrow_i)$$

Agora podemos aplicar novamente a regra (\rightarrow_i) :

$$\frac{\frac{p \rightarrow q, q \rightarrow r \vdash p \rightarrow r}{p \rightarrow q \vdash (q \rightarrow r) \rightarrow (p \rightarrow r)} (\rightarrow_i)}{\vdash (p \rightarrow q) \rightarrow (q \rightarrow r) \rightarrow (p \rightarrow r)} (\rightarrow_i)$$

E mais uma vez, já que a conclusão do sequente é ainda uma implicação:

$$\frac{\frac{\frac{p \rightarrow q, q \rightarrow r, p \vdash r}{p \rightarrow q, q \rightarrow r \vdash p \rightarrow r} (\rightarrow_i)}{p \rightarrow q \vdash (q \rightarrow r) \rightarrow (p \rightarrow r)} (\rightarrow_i)}{\vdash (p \rightarrow q) \rightarrow (q \rightarrow r) \rightarrow (p \rightarrow r)} (\rightarrow_i)$$

Agora não é mais possível utilizar a regra (\rightarrow_i) porque a conclusão r não é uma implicação, mas podemos utilizar a hipótese $q \rightarrow r$ para obter r , desde que tenhamos uma prova de q para utilizarmos (\rightarrow_e) . Neste ponto, a árvore é bifurcada em dois ramos e precisamos dividir o contexto de forma adequada em cada um dos ramos.

$$\frac{\frac{\frac{p \rightarrow q, q \rightarrow r, p \vdash q}{p \rightarrow q, q \rightarrow r, p \vdash q \rightarrow r} (\text{Ax})}{p \rightarrow q, q \rightarrow r, p \vdash r} (\rightarrow_e)}{\frac{\frac{p \rightarrow q, q \rightarrow r, p \vdash r}{p \rightarrow q, q \rightarrow r \vdash p \rightarrow r} (\rightarrow_i)}{p \rightarrow q \vdash (q \rightarrow r) \rightarrow (p \rightarrow r)} (\rightarrow_i)} (\rightarrow_i)$$

Observe que o ramo da direita consiste em um axioma já que a fórmula $q \rightarrow r$ pertence ao conjunto de hipóteses. No ramo da esquerda podemos obter q por meio da regra (\rightarrow_e) com as hipóteses $p \rightarrow q$ e p . A prova completa é dada a seguir:

$$\frac{\frac{\frac{p \rightarrow q, q \rightarrow r, p \vdash p}{p \rightarrow q, q \rightarrow r, p \vdash p \rightarrow q} (\text{Ax})}{p \rightarrow q, q \rightarrow r, p \vdash q} (\rightarrow_e)}{\frac{\frac{p \rightarrow q, q \rightarrow r, p \vdash q}{p \rightarrow q, q \rightarrow r, p \vdash q \rightarrow r} (\text{Ax})}{p \rightarrow q, q \rightarrow r, p \vdash r} (\rightarrow_e)} (\rightarrow_e)$$

$$\frac{\frac{\frac{p \rightarrow q, q \rightarrow r, p \vdash r}{p \rightarrow q, q \rightarrow r \vdash p \rightarrow r} (\rightarrow_i)}{p \rightarrow q \vdash (q \rightarrow r) \rightarrow (p \rightarrow r)} (\rightarrow_i)}{\vdash (p \rightarrow q) \rightarrow (q \rightarrow r) \rightarrow (p \rightarrow r)} (\rightarrow_i)$$

Agora vamos refazer este exercício no assistente de provas Coq. Para isto vamos criar uma seção que chamaremos de *exercicio1*:

Section *exemplo1*.

Variables $p\ q\ r$: Prop.

Lemma *ex1*: $(p \rightarrow q) \rightarrow ((q \rightarrow r) \rightarrow p \rightarrow r)$.

Depois de aberta a seção, declaramos as variáveis p , q e r , e enunciamos o exercício a ser resolvido como um lema chamado *ex1*. Neste momento, a janela de provas nos apresenta a seguinte configuração:

```
1 goal (ID 1)
```

```
p, q, r : Prop
=====
(p -> q) -> (q -> r) -> p -> r
```

A linha dupla pontilhada separa a conclusão das hipóteses (ou contexto). Neste caso, temos a fórmula $(p \rightarrow q) \rightarrow (q \rightarrow r) \rightarrow p \rightarrow r$ como conclusão, e o contexto contém apenas a declaração das variáveis p , q e r . Por uma questão de organização, as provas serão escritas entre as palavras reservadas **Proof** e **Qed**. Seguindo os passos da prova acima, iniciaremos com uma aplicação da regra (\rightarrow_i) que vai colocar o antecedente $p \rightarrow q$ da implicação principal no contexto. A regra (\rightarrow_i) em Coq é chamada de **intro**:

A janela de provas apresenta agora a seguinte configuração:

```
1 goal (ID 2)
```

```
p, q, r : Prop
H : p -> q
=====
(q -> r) -> p -> r
```

Agora a fórmula $p \rightarrow q$ aparece como hipótese, como esperado. Observe que a hipótese $p \rightarrow q$ recebe o nome H . Isto é importante porque sempre que precisarmos utilizar a fórmula $p \rightarrow q$ utilizaremos o nome H para nos referirmos a ela. Note que o nome H foi criado automaticamente pelo Coq, mas podemos escolher outros nomes. Por exemplo, utilizando o comando **intro H1**, o nome utilizado para a fórmula $p \rightarrow q$ seria $H1$, ao invés de H . O passo seguinte da prova consiste em outra aplicação da regra de introdução da implicação para colocar a fórmula $q \rightarrow r$ no contexto:

O comando acima, corresponde a uma nova aplicação da regra (\rightarrow_i) , mas agora a hipótese introduzida deve se chamar H' . A janela de provas da configuração atual é dada a seguir:

1 goal (ID 3)

```
p, q, r : Prop
H : p -> q
H' : q -> r
=====
p -> r
```

Por fim, podemos introduzir a fórmula p no contexto:
A hipótese correspondente a fórmula p será chamada de H_p :

1 goal (ID 4)

```
p, q, r : Prop
H : p -> q
H' : q -> r
Hp : p
=====
r
```

Segundo a prova feita manualmente, o próximo passo corresponde a uma aplicação da regra de eliminação da implicação (\rightarrow_e), que em Coq se chama `apply`. Neste caso, faremos a aplicação da hipótese H' porque ela tem como conclusão a fórmula r que queremos provar:
A janela de prova tem a seguinte forma:

1 goal (ID 5)

```
p, q, r : Prop
H : p -> q
H' : q -> r
Hp : p
=====
q
```

Ou seja, precisamos provar q , que é o antecedente da implicação que acabamos de utilizar. Se considerarmos novamente a estrutura da regra (\rightarrow_e), fica fácil compreender o que está acontecendo antes da aplicação da hipótese H' :

$$\frac{p \rightarrow q, q \rightarrow r, p \vdash q \quad p \rightarrow q, q \rightarrow r, p \vdash q \rightarrow r}{p \rightarrow q, q \rightarrow r, p \vdash r} (\rightarrow_e)$$

Assim, para provarmos r utilizando a regra (\rightarrow_e) precisamos de duas hipóteses, uma implicação que tenha r como conclusão, e o antecedente desta implicação. A implicação que tem r como conclusão é a hipótese H' . Nos resta, então provar o antecedente da implicação que acabamos de usar, ou seja, precisamos provar a fórmula q . Agora podemos repetir o mesmo raciocínio utilizando a hipótese H :

A janela de provas atual é como a seguir:

1 goal (ID 6)

```
p, q, r : Prop
H : p -> q
H' : q -> r
Hp : p
```

```
=====
```

```
p
```

Agora precisamos provar a fórmula p , mas ela faz parte do contexto. De fato, a fórmula p corresponde à hipótese Hp . Ou seja, estamos na situação em que podemos aplicar a regra (Ax) que, em Coq corresponde ao comando `assumption`, e assim, concluímos a prova e fechamos a seção *exemplo1*.

End exemplo1.

Agora é a sua vez de construir provas tanto em papel e lápis, quanto em Coq. Antes de apresentarmos os exercícios, vale um comentário sobre os assistentes de provas em geral. Estas ferramentas não foram projetadas com foco no ensino, e portanto veremos nas próximas seções que não teremos uma correspondência direta entre as regras de dedução natural e os comandos em Coq. Uma observação mais importante ainda é que estas ferramentas (e existem diversos assistentes de provas disponíveis no mercado) não foram projetadas para nos ajudar a construir provas, mas sim para que possamos verificar se uma prova feita em papel e lápis está correta. No exemplo apresentado anteriormente foi exatamente isto que fizemos: primeiro construímos uma prova em papel e lápis para, em seguida, reproduzirmos esta prova no Coq. Se a prova manual possuísse algum erro, não conseguiríamos reproduzi-la no Coq. Nos exercícios a seguir, construa inicialmente uma prova em papel e lápis, e posteriormente, reproduza esta prova no Coq.

Exercícios

1. $\vdash (p \rightarrow p \rightarrow q) \rightarrow p \rightarrow q$

$$\begin{array}{c} \frac{}{\vdash \exists q \rightarrow r \rightarrow t, \underline{p \rightarrow q}, \underline{p} \vdash} \\ \hline \frac{\frac{\frac{}{\vdash p \rightarrow q} \text{ (Ax)}}{\vdash p \rightarrow q} \text{ (Ax)}}{\vdash p \rightarrow q} \text{ (Ax)} \quad \frac{}{\vdash p} \text{ (Ax)}}{\vdash p \rightarrow q} \text{ (Ax)} \\ \hline \frac{q \rightarrow r \rightarrow t, p \rightarrow q, p \vdash q \rightarrow r \rightarrow t \quad q \rightarrow r \rightarrow t, p \rightarrow q, p \vdash q}{q \rightarrow r \rightarrow t, p \rightarrow q, p \vdash q \rightarrow r \rightarrow t} \text{ (Ax)} \\ \hline \frac{q \rightarrow (r \rightarrow t), p \rightarrow q, p \vdash r \rightarrow t}{q \rightarrow r \rightarrow t, p \rightarrow q, p \vdash r \rightarrow t} \text{ (Ax)} \\ \hline \frac{q \rightarrow r \rightarrow t, p \rightarrow q \vdash p \rightarrow (r \rightarrow t)}{q \rightarrow r \rightarrow t, p \rightarrow q \vdash p \rightarrow (r \rightarrow t)} \text{ (Ax)} \\ \hline \frac{q \rightarrow r \rightarrow t \vdash (p \rightarrow q) \rightarrow (p \rightarrow (r \rightarrow t))}{q \rightarrow r \rightarrow t \vdash (p \rightarrow q) \rightarrow (p \rightarrow (r \rightarrow t))} \text{ (Ax)} \\ \hline \frac{q \rightarrow r \rightarrow t \vdash (p \rightarrow q) \rightarrow (p \rightarrow (r \rightarrow t))}{q \rightarrow r \rightarrow t \vdash (p \rightarrow q) \rightarrow (p \rightarrow (r \rightarrow t))} \text{ (Ax)} \end{array}$$

2. $\vdash (p \rightarrow q) \rightarrow (p \rightarrow p \rightarrow q)$
3. $\vdash (q \rightarrow r \rightarrow t) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r \rightarrow t$
4. $\vdash (p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r$
5. $\vdash (p \rightarrow q \rightarrow r) \rightarrow (q \rightarrow p \rightarrow r)$

Uma vez finalizadas as provas em papel e lápis, é hora de verificar se as provas estão corretas. Abaixo apresentamos os enunciados dos exercícios, mas sem as provas. O comando *Admitted* nos permite enunciar um exercício/problema e deixar a prova pendente. Nos exercícios a seguir você deve remover a linha com *Admitted* e substituí-la pela sua prova finalizando com *Qed* como fizemos no exemplo.

Lemma *exercicio1*: $(p \rightarrow p \rightarrow q) \rightarrow p \rightarrow q$.

Lemma *exercicio2*: $(p \rightarrow q) \rightarrow (p \rightarrow p \rightarrow q)$.

Parameter r t : Prop.

Lemma *exercicio3*: $(q \rightarrow r \rightarrow t) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r \rightarrow t$.

Lemma *exercicio3'*: $(q \rightarrow r \rightarrow t) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r \rightarrow t$.

Lemma *exercicio4*: $(p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r$.

Lemma *exercicio5*: $(p \rightarrow q \rightarrow r) \rightarrow (q \rightarrow p \rightarrow r)$.

Na próxima seção, vamos considerar os outros conectivos apresentados na gramática (1.1), assim como as regras de introdução e eliminação de cada um deles.

1.2 A Lógica Proposicional Minimal

[aula 03 (0 exercícios)]

A negação pode ser definida como uma implicação particular: $\neg\varphi = \varphi \rightarrow \perp$. Neste caso, as regras de introdução e eliminação são adaptações diretas das regras (\rightarrow_i) e (\rightarrow_e) :

$$\frac{\Gamma, \varphi \vdash \perp}{\Gamma \vdash \neg\varphi} (\neg_i) \qquad \frac{\Gamma \vdash \neg\varphi \quad \Gamma \vdash \varphi}{\Gamma \vdash \perp} (\neg_e)$$

Como exemplo, mostraremos como provar o sequente $\varphi \vdash \neg\neg\varphi$, qualquer que seja a fórmula φ .

$$\frac{\frac{\frac{}{\varphi, \neg\varphi \vdash \neg\varphi} (Ax) \quad \frac{}{\varphi, \neg\varphi \vdash \varphi} (Ax)}{\varphi, \neg\varphi \vdash \perp} (\neg_e)}{\varphi \vdash \neg\neg\varphi} (\neg_i)$$

Agora vamos reproduzir esta prova no Coq. Note que este sequente tem uma fórmula como hipótese, a saber φ . Todos os sequentes anteriores não tinham hipóteses. Como, então declarar uma hipótese no Coq? Simples, a hipótese é declarada dentro de uma seção:

Section *exemplo2*.

Hypothesis H : p .

Lemma *nni*: $\sim\sim p$.

A janela de prova é como a seguir:

1 goal (ID 6)

```

H : p
=====
~ ~ p

```

Ou seja, temos a hipótese H que corresponde à fórmula p , e queremos concluir a fórmula $\sim\sim p$. O Coq não possui comandos específicos para as regras de introdução e eliminação da negação. De fato, considerando que a negação é uma implicação particular, podemos utilizar os comandos da implicação.

A janela de prova é:

1 goal (ID 8)

```
H : p
HO : ~ p
=====
False
```

E considerando que $\neg p$ é o mesmo que $p \rightarrow False$, podemos aplicar a hipótese HO , e fechá-la com `assumption`.

End *exemplo2*.

E quanto à eliminação da dupla negação? Podemos prová-la agora? Infelizmente não, porque o poder de expressividade que temos até agora com as regras (\rightarrow_e) , (\rightarrow_i) , (\neg_i) e (\neg_e) não é suficiente para provarmos a eliminação da dupla negação. Mas para deixá-lo um pouco intrigado sobre esta questão provaremos o seguinte sequente: $\neg\neg\neg\varphi \vdash \neg\varphi$.

$$\frac{\frac{\frac{}{\neg\neg\neg\varphi, \varphi \vdash \varphi} (Ax)}{\neg\neg\neg\varphi, \varphi \vdash \neg\varphi} (\neg_i)}{\neg\neg\neg\varphi, \varphi \vdash \perp} (\neg_e)}{\neg\neg\neg\varphi \vdash \neg\varphi} (\neg_i)$$

O sequente anterior é a prova da eliminação da dupla negação de uma fórmula negada, e não de uma fórmula qualquer, e isto faz toda a diferença. Voltaremos a falar da eliminação da dupla negação na seção sobre lógica proposicional clássica.

A seguir, vamos refazer esta prova no Coq.

Section *exemplo3*.

Hypothesis H : $\sim\sim\sim p$.

Lemma *nne_neg*: $\neg p$.

End *exemplo3*.

Exercícios

[aula 04 (8 exercícios)]

1. Prove $\varphi \rightarrow \psi \vdash (\neg\neg\varphi) \rightarrow (\neg\neg\psi)$, onde φ e ψ são fórmulas quaisquer;
2. Prove $\neg\neg(\varphi \rightarrow \psi) \vdash (\neg\neg\varphi) \rightarrow (\neg\neg\psi)$, onde φ e ψ são fórmulas quaisquer.
3. Prove $\vdash (((((\varphi \rightarrow \psi) \rightarrow \varphi) \rightarrow \varphi) \rightarrow \psi) \rightarrow \psi)$, onde φ e ψ são fórmulas quaisquer.
4. Prove $\varphi, \neg\varphi \vdash \neg\psi$, onde φ e ψ são fórmulas quaisquer.

Para cada um dos exercícios, reproduza a sua prova no Coq:

```
Section exercicio_nne_imp.
```

```
Hypothesis H: p → q.
```

```
Lemma nne_imp: (~~p) → (~~q).
```

```
End exercicio_nne_imp.
```

```
Section exercicio_nne_to_imp.
```

```
Hypothesis H: ~~(p → q).
```

```
Lemma nne_to_imp: (~~p) → (~~q).
```

```
End exercicio_nne_to_imp.
```

```
Section exercicio_weak_explosion.
```

```
Hypothesis H1: p.
```

```
Hypothesis H2: ¬p.
```

```
Lemma weak_explosion: ¬q.
```

```
End exercicio_weak_explosion.
```

```
Lemma weak_peirce: (((((p → q) → p) → p) → q) → q).
```

A conjunção é um conectivo binário cuja *regra de introdução*, denotada por (\wedge_i) , tem a seguinte forma:

$$\frac{\Gamma \vdash \varphi_1 \quad \Gamma \vdash \varphi_2}{\Gamma \vdash \varphi_1 \wedge \varphi_2} (\wedge_i) \quad (1.3)$$

ou seja, uma prova de $\Gamma \vdash \varphi_1 \wedge \varphi_2$ é construída a partir de uma prova de $\Gamma \vdash \varphi_1$ e de uma prova de $\Gamma \vdash \varphi_2$.

Existem duas regras de eliminação para a conjunção já que podemos extrair qualquer uma das componentes de uma conjunção:

$$\frac{\Gamma \vdash \varphi_1 \wedge \varphi_2}{\Gamma \vdash \varphi_1} (\wedge_{e1}) \qquad \frac{\Gamma \vdash \varphi_1 \wedge \varphi_2}{\Gamma \vdash \varphi_2} (\wedge_{e2})$$

Estas duas regras podem ser representadas de forma mais concisa da seguinte forma:

$$\frac{\Gamma \vdash \varphi_1 \wedge \varphi_2}{\Gamma \vdash \varphi_{i \in \{1,2\}}} (\wedge_e) \quad (1.4)$$

Usaremos o nome (\wedge_e) para designar a utilização da regra de eliminação da conjunção quando não quisermos especificar qual das regras (\wedge_{e_1}) ou (\wedge_{e_2}) foi utilizada.

Exercícios

1. $\varphi \wedge \psi \vdash \psi \wedge \varphi$.
2. $(\varphi \wedge \psi) \wedge \rho \vdash \varphi \wedge (\psi \wedge \rho)$.

Section *exercicio_comm_and*.

Hypothesis H : $p \wedge q$.

Lemma *comm_and*: $q \wedge p$.

End *exercicio_comm_and*.

Section *exercicio_comm_assoc*.

Hypothesis H : $(p \wedge q) \wedge r$.

Lemma *comm_assoc*: $p \wedge (q \wedge r)$.

End *exercicio_comm_assoc*.

Veamos agora as regras para a disjunção. A *regra de introdução da disjunção* nos permite construir a prova de uma disjunção a partir da prova de qualquer uma das suas componentes:

$$\frac{\Gamma \vdash \varphi_1}{\Gamma \vdash \varphi_1 \vee \varphi_2} (\vee_{i_1}) \qquad \frac{\Gamma \vdash \varphi_2}{\Gamma \vdash \varphi_1 \vee \varphi_2} (\vee_{i_2})$$

Como no caso da regra de eliminação da conjunção podemos representar estas duas regras de forma mais compacta:

$$\frac{\Gamma \vdash \varphi_{i \in \{1,2\}}}{\Gamma \vdash \varphi_1 \vee \varphi_2} (\vee_i)$$

A *regra de eliminação da disjunção* nos permite construir a prova de uma fórmula, digamos γ , a partir da prova de uma disjunção da seguinte forma:

$$\frac{\Gamma \vdash \varphi_1 \vee \varphi_2 \quad \Gamma, \varphi_1 \vdash \gamma \quad \Gamma, \varphi_2 \vdash \gamma}{\Gamma \vdash \gamma} (\vee_e)$$

Assim, a construção de uma prova de γ a partir das fórmulas em Γ (sequente $\Gamma \vdash \gamma$) depende de três coisas: uma prova da disjunção (sequente $\Gamma \vdash \varphi_1 \vee \varphi_2$) uma prova de γ a

partir de φ_1 e das fórmulas de Γ (sequente $\Gamma, \varphi_1 \vdash \gamma$) e de outra prova de γ a partir de φ_2 e das fórmulas de Γ (sequente $\Gamma, \varphi_2 \vdash \gamma$). Como exemplo, mostraremos que a disjunção é comutativa, ou seja, queremos construir uma prova para o sequente $\varphi \vee \psi \vdash \psi \vee \varphi$. A ideia aqui é utilizarmos a regra (\vee_e) . Para isto podemos instanciar Γ com o conjunto unitário contendo a fórmula $\varphi \vee \psi$. Em função da estrutura da regra (\vee_e) , precisamos construir duas provas distintas de $\psi \vee \varphi$: uma a partir de φ , e outra a partir de ψ . Podemos fazer isto com a ajuda da regra (\vee_i) :

$$\frac{\text{(Ax)} \frac{\overline{\varphi \vee \psi \vdash \varphi \vee \psi}}{\varphi \vee \psi \vdash \varphi \vee \psi} \quad \frac{\overline{\varphi \vee \psi, \varphi \vdash \varphi} \text{(Ax)} \quad \frac{\overline{\varphi \vee \psi, \psi \vdash \psi} \text{(Ax)}}{\varphi \vee \psi, \varphi \vdash \psi \vee \varphi} \text{(}\vee_i\text{)} \quad \frac{\overline{\varphi \vee \psi, \psi \vdash \psi \vee \varphi} \text{(}\vee_i\text{)}}{\varphi \vee \psi, \psi \vdash \psi \vee \varphi} \text{(}\vee_e\text{)}}{\varphi \vee \psi \vdash \psi \vee \varphi} \text{(}\vee_e\text{)}$$

Agora vamos fazer uma prova análoga no Coq.

Section *exemplo_or_comm*.

Hypothesis *H*: $p \vee q$.

Lemma *or_comm*: $q \vee p$.

1 goal (ID 7)

H : $p \vee q$
=====

q \vee p

A regra de eliminação da disjunção pode ser simulada pelo comando `destruct H` que vai bifurcar a prova em dois ramos.

2 goals (ID 14)

H : $p \vee q$
H0 : p
=====

q \vee p

goal 2 (ID 15) is:

q \vee p

	Regras de introdução	Regras de eliminação
1	$\frac{\Gamma \vdash \varphi_1 \quad \Gamma \vdash \varphi_2}{\Gamma \vdash \varphi_1 \wedge \varphi_2} (\wedge_i)$	$\frac{\Gamma \vdash \varphi_1 \wedge \varphi_2}{\Gamma \vdash \varphi_{i \in \{1,2\}}} (\wedge_e)$
2	$\frac{\Gamma \vdash \varphi_{i \in \{1,2\}}}{\Gamma \vdash \varphi_1 \vee \varphi_2} (\vee_i)$	$\frac{\Gamma \vdash \varphi_1 \vee \varphi_2 \quad \Gamma, \varphi_1 \vdash \gamma \quad \Gamma, \varphi_2 \vdash \gamma}{\Gamma \vdash \gamma} (\vee_e)$
3	$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi} (\rightarrow_i)$	$\frac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi} (\rightarrow_e)$
4	$\frac{\Gamma, \varphi \vdash \perp}{\Gamma \vdash \neg \varphi} (\neg_i)$	$\frac{\Gamma \vdash \neg \varphi \quad \Gamma \vdash \varphi}{\Gamma \vdash \perp} (\neg_e)$

Table 1.1: Regras da Lógica Minimal

No primeiro ramo precisamos provar $q \vee p$ tendo $p \vee q$ e p como hipóteses, e no segundo caso, precisamos provar $q \vee p$ tendo $p \vee q$ e q como hipóteses. Podemos organizar as provas que têm bifurcação colocando um hífen para cada caso. O hífen, apesar de não ser obrigatório, deixa a prova mais legível. No primeiro caso, temos as fórmulas $p \vee q$ e p como hipóteses, e portanto podemos concluir este ramo com uma aplicação da regra de introdução da disjunção. Como a componente da disjunção $q \vee p$ que temos como hipótese está à direita, o comando Coq a ser utilizado é **right**. O raciocínio para o outro caso é análogo. *End exemplo_or_comm.*

Exercício

Sejam φ , ψ e ρ fórmulas quaisquer da lógica proposicional. Prove que a disjunção é associativa, isto é, prove o seguinte $(\varphi \vee \psi) \vee \rho \vdash \varphi \vee (\psi \vee \rho)$.

Section *exercicio_or_assoc.*

Hypothesis H : $(p \vee q) \vee r$.

Lemma *or_assoc*: $p \vee (q \vee r)$.

End *exercicio_or_assoc.*

O conjunto de regras estudado até aqui forma a chamada *Lógica Proposicional Minimal*. A Tabela (1.1) contém todas estas regras:

Agora vamos resolver mais alguns exercícios na lógica minimal. Considere o seguinte $\varphi \rightarrow \psi$, $\neg \psi \vdash \neg \varphi$. Como a fórmula do conseqüente é uma negação, vamos aplicar a regra de introdução da negação na construção de uma prova (raciocinando de baixo para cima, isto é, da raiz para as folhas da árvore):

$$\frac{\frac{?}{\varphi \rightarrow \psi, \neg \psi, \varphi \vdash \perp}}{\varphi \rightarrow \psi, \neg \psi \vdash \neg \varphi} (\neg_i)$$

Agora, precisamos construir uma prova do absurdo, e portanto podemos tentar utilizar a regra (\neg_e) . Para isto precisamos escolher uma fórmula do contexto para fazer o papel

O sequente que acabamos de provar é outro caso que aparece com frequência em provas, e corresponde a uma regra conhecida como *contrapositiva*:

$$\frac{\Gamma \vdash \varphi \rightarrow \psi}{\Gamma \vdash \neg\psi \rightarrow \neg\varphi} \text{ (CP)}$$

Section *exemplo_cp*.

Hypothesis *H*: $p \rightarrow q$.

Lemma *cp*: $\neg q \rightarrow \neg p$.

End *exemplo_cp*.

Exercício

Sejam φ e ψ fórmulas da lógica proposicional. Prove o sequente $\varphi \rightarrow \neg\psi \vdash \psi \rightarrow \neg\varphi$. Em seguida, reproduza a sua prova no Coq.

Section *exercicio_cp1*.

Hypothesis *H*: $p \rightarrow \neg q$.

Lemma *cp1*: $q \rightarrow \neg p$.

End *exercicio_cp1*.

[**aula 05 (16 exercícios)**]

Este é um bom momento para simplificarmos a notação que estamos usando para fazer as provas em papel e lápis. Na notação atual, cada nó na árvore de derivação é anotado com um sequente. Na nova notação, anotaremos cada nó apenas com a fórmula correspondente ao conseqüente do sequente. Isto vai deixar a estrutura das provas mais simples, e como veremos, nenhuma informação da prova será perdida porque temos como inferir o contexto correspondente em cada nó da árvore de derivação. Vejamos um exemplo onde o contexto é o mesmo ao longo de toda a prova. Vamos retomar o exercício em que provamos que a conjunção é um conectivo que satisfaz a propriedade associativa. Neste ponto acreditamos que você já resolveu este exercício. Em caso negativo, resolva o exercício antes de prosseguir. Em seguida, compare sua solução com a que apresentamos a seguir, ok? Tentar resolver os exercícios antes de olhar qualquer solução é um passo muito importante para a sua evolução nos estudos de lógica.

Considere a prova a seguir:

$$\frac{\frac{\frac{\text{(Ax)}}{(\phi \wedge \psi) \wedge \varphi \vdash (\phi \wedge \psi) \wedge \varphi}}{(\wedge_e)} \quad \frac{\frac{\text{(Ax)}}{(\phi \wedge \psi) \wedge \varphi \vdash (\phi \wedge \psi) \wedge \varphi}}{(\wedge_e)} \quad \frac{\frac{\text{(Ax)}}{(\phi \wedge \psi) \wedge \varphi \vdash (\phi \wedge \psi) \wedge \varphi}}{(\wedge_e)}}{(\phi \wedge \psi) \wedge \varphi \vdash \phi} \quad \frac{\frac{\frac{\text{(Ax)}}{(\phi \wedge \psi) \wedge \varphi \vdash (\phi \wedge \psi) \wedge \varphi}}{(\wedge_e)} \quad \frac{\frac{\text{(Ax)}}{(\phi \wedge \psi) \wedge \varphi \vdash (\phi \wedge \psi) \wedge \varphi}}{(\wedge_e)}}{(\phi \wedge \psi) \wedge \varphi \vdash \psi} \quad \frac{\frac{\text{(Ax)}}{(\phi \wedge \psi) \wedge \varphi \vdash (\phi \wedge \psi) \wedge \varphi}}{(\wedge_e)}}{(\phi \wedge \psi) \wedge \varphi \vdash (\psi \wedge \varphi)}}{(\phi \wedge \psi) \wedge \varphi \vdash \phi \wedge (\psi \wedge \varphi)} \text{ (}\wedge_i\text{)}$$

Ao remover os contextos, obtemos a seguinte árvore:

$$\begin{array}{c}
 (\wedge_e) \frac{(\phi \wedge \psi) \wedge \varphi}{(\phi \wedge \psi)} \quad (\wedge_e) \frac{(\phi \wedge \psi) \wedge \varphi}{\phi \wedge \psi} \quad (\wedge_e) \frac{(\phi \wedge \psi) \wedge \varphi}{\varphi} \\
 (\wedge_e) \frac{\quad}{\phi} \quad \frac{\quad}{\psi} \quad \frac{\quad}{\varphi} \\
 \hline
 \frac{\quad}{\phi \wedge (\psi \wedge \varphi)} (\wedge_i)
 \end{array}$$

A árvore ficou muito mais compacta e limpa, e este é o objetivo da mudança de notação. Como o contexto é o mesmo em toda a árvore neste exemplo, então podemos facilmente recuperá-lo a partir do enunciado do exercício que é o sequente $(\phi \wedge \psi) \wedge \varphi \vdash \phi \wedge (\psi \wedge \varphi)$. Do ponto de vista prático, esta mudança facilita a escrita de provas maiores, uma vez que uma árvore de derivação pode crescer rapidamente. Como a janela de prova do Coq mostra o sequente correspondente ao nó da árvore de prova no momento da prova, esta mudança de notação não terá nenhum impacto em relação ao Coq. Você verá também que a maioria das bibliografias sobre dedução natural utilizam a nova notação $[?, ?, ?, ?]$. O que ocorre então com as regras que promovem uma mudança de contexto? Será que é possível sempre remover os contextos das provas de uma forma sistemática? Sim! Ainda considerando o exemplo anterior, se não soubéssemos qual o contexto que foi apagado, seria possível descobri-lo? Sim, esta informação vem das folhas da árvore, que são as hipóteses do problema. Na notação com o contexto explícito, as folhas da árvore têm que ser axiomas, e portanto o contexto de todas as folhas é a fórmula $(\phi \wedge \psi) \wedge \varphi$ já que todas as folhas são iguais. Podemos também consultar a Tabela 1.1 e observar que o contexto não muda nas regras (\wedge_e) e (\wedge_i) . Portanto, o contexto em cada nó da árvore é formado pelo conjunto unitário contendo a fórmula $(\phi \wedge \psi) \wedge \varphi$. Assim, as folhas de uma árvore de derivação têm que ser fórmulas pertencentes ao contexto.

Em um exemplo anterior construímos uma prova do sequente $\varphi \rightarrow \psi, \neg\psi \vdash \neg\varphi$ com contextos explícitos, a versão com contextos implícitos é dada pela seguinte árvore de derivação:

$$\begin{array}{c}
 (\rightarrow_e) \frac{\varphi \rightarrow \psi \quad \varphi}{\psi} \quad \neg\psi \\
 \hline
 \perp \quad (\neg_e) \\
 \hline
 \neg\varphi \quad (\neg_i)
 \end{array}$$

Note que esta árvore possui três folhas, cada uma contendo uma fórmula distinta. O contexto inicial é o conjunto contendo todas as fórmulas que aparecem nas folhas, ou seja, nosso contexto inicial é o conjunto $\{\varphi \rightarrow \psi, \neg\psi, \varphi\}$. As regras (\rightarrow_e) e (\neg_e) preservam o contexto, e portanto o contexto da linha 3 é, ainda, o conjunto $\{\varphi \rightarrow \psi, \neg\psi, \varphi\}$. De acordo com a Tabela 1.1, a regra (\neg_i) adiciona uma fórmula ao contexto (leitura de baixo para cima), ou remove uma fórmula do contexto, se a leitura for feita de cima para baixo. Neste caso, a fórmula removida é φ , e portanto o contexto da fórmula $\neg\varphi$ (raiz da árvore) é o conjunto $\{\varphi \rightarrow \psi, \neg\psi\}$ como esperado pelo enunciado do exemplo. Na notação sem

contexto, as fórmulas que são removidas do contexto ao longo da prova, como a fórmula φ deste exemplo são colocadas entre colchetes para enfatizar que são hipóteses temporárias que, em determinado momento serão removidas do contexto:

$$\begin{array}{c}
 (\rightarrow_e) \frac{\varphi \rightarrow \psi \quad [\varphi]}{\psi} \quad \neg\psi \\
 \hline
 \perp \quad (\neg_e) \\
 \hline
 \neg\varphi \quad (\neg_i)
 \end{array}$$

e agora fica claro que a fórmula φ não faz parte do contexto original do sequente a ser provado. Note que os colchetes são colocados *apenas nas folhas* que contêm fórmulas que não fazem parte do contexto dado pelo problema. Adicionalmente, como o contexto da raiz tem que ser o contexto dado pelo problema, caso contrário a prova não é uma prova do problema proposto, precisamos de um mecanismo para nos informar quando as fórmulas marcadas com os colchetes são *removidas* (ou *descartadas*) do contexto. No exemplo acima, isto ocorre ao aplicarmos a regra (\neg_i) . Então, utilizaremos uma letra para registrar este fato:

$$\begin{array}{c}
 (\rightarrow_e) \frac{\varphi \rightarrow \psi \quad [\varphi]^u}{\psi} \quad \neg\psi \\
 \hline
 \perp \quad (\neg_e) \\
 \hline
 \neg\varphi \quad (\neg_i) u
 \end{array}$$

Agora sabemos em que momento a fórmula φ foi introduzida (folha $[\varphi]^u$), e em que momento foi descartada (regra $(\neg_i) u$) na árvore de derivação.

A seguir, veremos exemplos mais complexos onde, por exemplo, fórmulas idênticas podem exigir marcas distintas, mas antes disto compare as regras de dedução natural para a lógica proposicional minimal com o contexto explícito e com o contexto implícito na Tabela 1.2, e veja como o mecanismo de descarte simula a mudança de contexto antes e depois de uma aplicação das regras (\vee_e) , (\rightarrow_i) e (\neg_i) . Como a notação com contexto implícito é mais compacta, a partir deste momento não utilizaremos mais contextos explícitos. A intenção de iniciar esta apresentação utilizando contextos explícitos foi de permitir uma explicação mais fácil e natural para o descarte de hipóteses, que sempre gerou muitas dúvidas entre os alunos. Por exemplo, quando a razão do descarte não está clara, é comum aparecerem soluções (erradas!) com árvores de derivação onde o descarte de hipóteses é feito em regras como (\wedge_i) , (\wedge_e) , (\vee_i) , (\rightarrow_e) e (\neg_e) . Se você acha que está tudo bem uma árvore de derivação conter descarte de hipóteses nas regras citadas na frase anterior, volte para o início deste capítulo e reinicie o estudo do sistema de dedução natural antes de prosseguir :-)

No exemplo a seguir, veremos que é possível fazer a introdução de uma implicação sem o descarte de uma hipótese, se tivermos uma prova do conseqüente da implicação que queremos provar. Ou seja, se temos uma prova de ψ então podemos construir uma prova de $\varphi \rightarrow \psi$, qualquer que seja a fórmula φ . Em outras palavras, queremos construir uma prova para o

	Contexto explícito	Contexto implícito
1	$\frac{\Gamma \vdash \varphi_1 \quad \Gamma \vdash \varphi_2}{\Gamma \vdash \varphi_1 \wedge \varphi_2} (\wedge_i)$	$\frac{\varphi_1 \quad \varphi_2}{\varphi_1 \wedge \varphi_2} (\wedge_i)$
2	$\frac{\Gamma \vdash \varphi_1 \wedge \varphi_2}{\Gamma \vdash \varphi_{i \in \{1,2\}}} (\wedge_e)$	$\frac{\varphi_1 \wedge \varphi_2}{\varphi_{i \in \{1,2\}}} (\wedge_e)$
3	$\frac{\Gamma \vdash \varphi_{i \in \{1,2\}}}{\Gamma \vdash \varphi_1 \vee \varphi_2} (\vee_i)$	$\frac{\varphi_{i \in \{1,2\}}}{\varphi_1 \vee \varphi_2} (\vee_i)$
4	$\frac{\Gamma \vdash \varphi_1 \vee \varphi_2 \quad \Gamma, \varphi_1 \vdash \gamma \quad \Gamma, \varphi_2 \vdash \gamma}{\Gamma \vdash \gamma} (\vee_e)$	$\frac{\varphi_1 \vee \varphi_2 \quad \begin{array}{c} [\varphi_1]^u \\ \vdots \\ \gamma \end{array} \quad \begin{array}{c} [\varphi_2]^v \\ \vdots \\ \gamma \end{array}}{\gamma} (\vee_e) u, v$
5	$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi} (\rightarrow_i)$	$\frac{\begin{array}{c} [\varphi]^u \\ \vdots \\ \psi \end{array}}{\varphi \rightarrow \psi} (\rightarrow_i) u$
6	$\frac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi} (\rightarrow_e)$	$\frac{\varphi \rightarrow \psi \quad \varphi}{\psi} (\rightarrow_e)$
7	$\frac{\Gamma, \varphi \vdash \perp}{\Gamma \vdash \neg \varphi} (\neg_i)$	$\frac{\begin{array}{c} [\varphi]^u \\ \vdots \\ \perp \end{array}}{\neg \varphi} (\neg_i) u$
8	$\frac{\Gamma \vdash \neg \varphi \quad \Gamma \vdash \varphi}{\Gamma \vdash \perp} (\neg_e)$	$\frac{\neg \varphi \quad \varphi}{\perp} (\neg_e)$

Table 1.2: Regras da Lógica Proposicional Minimal

$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \neg\neg\varphi} (\neg\neg_i)$	$\frac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \neg\psi}{\Gamma \vdash \neg\varphi} (\text{MT})$	$\frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi} (\rightarrow_i) \emptyset$	$\frac{\Gamma \vdash \varphi \rightarrow \psi}{\Gamma \vdash \neg\psi \rightarrow \neg\varphi} (\text{CP})$
--	---	---	---

Table 1.3: Regras derivadas da Lógica Proposicional Minimal

sequente $\psi \vdash \varphi \rightarrow \psi$. Raciocinando de baixo para cima, temos que a aplicação da regra (\rightarrow_i) reduzirá o problema de provar $\varphi \rightarrow \psi$ ao problema de provar ψ tendo φ como hipótese adicional:

$$\frac{[\varphi]^u \quad \psi}{\varphi \rightarrow \psi} (\rightarrow_i) u$$

Note que temos como hipótese a fórmula ψ que precisamos provar, então o trabalho a ser feita consiste unicamente em organizar a árvore de forma que $[\varphi]^u$ e ψ sejam suas folhas. Uma maneira de fazer isto é introduzindo e eliminando uma conjunção contendo ψ :

$$\frac{\frac{[\varphi]^u \quad \psi}{\varphi \wedge \psi} (\wedge_i)}{\psi} (\wedge_e) (\rightarrow_i) u$$

Em Coq não precisamos lidar com a introdução e eliminação da conjunção:

Section *exemplo_imp_empty*.

Hypothesis *H*: q .

Lemma *imp_empty*: $p \rightarrow q$.

End *exemplo_imp_empty*.

Como este raciocínio aparece com frequência nas provas, vamos colocá-lo como uma regra derivada:

$$\frac{\psi}{\varphi \rightarrow \psi} (\rightarrow_i) \emptyset$$

Exercícios

Prove os seguintes a seguir:

1. $\neg(\varphi \vee \psi) \vdash (\neg\varphi) \wedge (\neg\psi)$

2. $(\neg\varphi) \wedge (\neg\psi) \vdash \neg(\varphi \vee \psi)$
3. $\varphi \rightarrow \psi \vdash (\delta \vee \varphi) \rightarrow (\delta \vee \psi)$
4. $\varphi \rightarrow \psi \vdash \neg(\varphi \wedge \neg\psi)$
5. $\varphi \wedge \psi \vdash \neg(\neg\varphi \vee \neg\psi)$
6. $\neg(\varphi \vee \gamma) \vdash (\neg\varphi) \wedge (\neg\gamma)$
7. $(\neg\varphi) \wedge (\neg\gamma) \vdash \neg(\varphi \vee \gamma)$
8. $(\neg\varphi) \vee (\neg\gamma) \vdash \neg(\varphi \wedge \gamma)$
9. $\neg\neg(\varphi \wedge \gamma) \vdash (\neg\neg\varphi) \wedge (\neg\neg\gamma)$
10. $(\neg\neg\varphi) \wedge (\neg\neg\gamma) \vdash \neg\neg(\varphi \wedge \gamma)$
11. $\varphi \vee (\psi \wedge \gamma) \vdash (\varphi \vee \psi) \wedge (\varphi \vee \gamma)$
12. $(\varphi \vee \psi) \wedge (\varphi \vee \gamma) \vdash \varphi \vee (\psi \wedge \gamma)$
13. $\varphi \wedge (\psi \vee \gamma) \vdash (\varphi \wedge \psi) \wedge (\varphi \wedge \gamma)$
14. $(\varphi \wedge \psi) \wedge (\varphi \wedge \gamma) \vdash \varphi \wedge (\psi \vee \gamma)$
15. $\vdash \neg\neg(\varphi \vee \neg\varphi)$
16. $\vdash \neg(\varphi \wedge \neg\varphi)$

Section *ejercicio_min1*.

Hypothesis *H*: $\sim(p \vee q)$.

Lemma *min1*: $(\sim p) \wedge (\sim q)$.

End *ejercicio_min1*.

Section *ejercicio_min2*.

Hypothesis *H*: $(\sim p) \wedge (\sim q)$.

Lemma *min2*: $\sim(p \vee q)$.

End *ejercicio_min2*.

Section *ejercicio_min3*.

Hypothesis *H*: $p \rightarrow q$.

Lemma *min3*: $(r \vee p) \rightarrow (r \vee q)$.

End *ejercicio_min3*.

Section *ejercicio_min4*.

Hypothesis *H*: $p \rightarrow q$.

Lemma *min4*: $\sim(p \wedge \neg q)$.

End *ejercicio_min4*.

Section *ejercicio_min5*.

Hypothesis *H*: $p \wedge q$.

Lemma *min5*: $\sim(\sim p \vee \neg q)$.

End *ejercicio_min5*.

Section *ejercicio_min6*.

Hypothesis *H*: $\sim(p \vee q)$.

Lemma *min6*: $(\sim p) \wedge (\sim q)$.

End *ejercicio_min6*.

Section *ejercicio_min7*.

Hypothesis *H*: $(\sim p) \wedge (\sim q)$.

Lemma *min7*: $\sim(p \vee q)$.

End *ejercicio_min7*.

Section *ejercicio_min8*.

Hypothesis *H*: $(\sim p) \vee (\sim q)$.

Lemma *min8*: $\sim(p \wedge q)$.

End *ejercicio_min8*.

Section *ejercicio_min9*.

Hypothesis *H*: $\sim\sim(p \wedge q)$.

Lemma *min9*: $(\sim\sim p) \wedge (\sim\sim q)$.

End *ejercicio_min9*.

Section *ejercicio_min10*.

Hypothesis *H*: $(\sim\sim p) \wedge (\sim\sim q)$.

Lemma *min10*: $\sim\sim(p \wedge q)$.

End *ejercicio_min10*.

Section *ejercicio_min11*.

Hypothesis *H*: $p \vee (q \wedge r)$.

Lemma *min11*: $(p \vee q) \wedge (p \vee r)$.

End *ejercicio_min11*.

Section *ejercicio_min12*.

Hypothesis *H*: $(p \vee q) \wedge (p \vee r)$.

Lemma *min12*: $p \vee (q \wedge r)$.

End *ejercicio_min12*.

Section *ejercicio_min13*.

Hypothesis *H*: $p \wedge (q \vee r)$.

Lemma *min13*: $(p \wedge q) \vee (p \wedge r)$.

End *ejercicio_min13*.

Section *ejercicio_min14*.

Hypothesis *H*: $(p \wedge q) \vee (p \wedge r)$.

Lemma *min14*: $p \wedge (q \vee r)$.

End *ejercicio_min14*.

Lemma *min15*: $\sim\sim(p \vee \neg p)$.

Lemma *min16*: $\sim(p \wedge \neg p)$.

1.3 A Lógica Proposicional Intuicionista

[aula 06 (3 exercícios)]

Agora vamos estender a lógica proposicional minimal com uma nova regra chamada de *regra da explosão* ou *regra da eliminação do absurdo intuicionista*. Esta regra nos permite concluir qualquer fórmula a partir do absurdo:

$$\frac{\perp}{\varphi} (\perp_e)$$

A lógica obtida adicionando-se a regra da explosão à lógica proposicional minimal é denominada *lógica proposicional intuicionista*. Observe que a lógica proposicional minimal possui uma versão mais fraca de regra de explosão. De fato, podemos na lógica proposicional minimal concluir qualquer fórmula negada a partir do absurdo. A lógica proposicional intuicionista é conhecida por corresponder à noção de lógica construtiva que é particularmente interessante para a Computação. De forma simplificada, a lógica proposicional intuicionista pode ser vista como a lógica que rejeita a lei do terceiro excluído, ou seja, nesta lógica o sequente $\vdash \varphi \vee \neg\varphi$ não é um axioma e nem uma regra derivada, quando φ é uma fórmula arbitrária. Vejamos um exemplo. Considere o seguinte sequente $\neg\varphi \vee \gamma \vdash \varphi \rightarrow \gamma$. Iniciando esta prova de baixo para cima, isto é, partindo do consequente, podemos aplicar a regra de introdução da implicação:

$$\frac{\neg\varphi \vee \gamma \quad [\varphi]^u}{\varphi \rightarrow \gamma} (\rightarrow_i) u$$

Agora precisamos construir uma prova de γ tendo as fórmulas $\neg\varphi \vee \gamma$ e $[\varphi]^u$ como contexto. Uma ideia possível é usar a regra de eliminação da disjunção porque com o lado esquerdo, isto é, com $\neg\varphi$ e com $[\varphi]^u$ temos o absurdo, e com a regra da explosão podemos concluir γ como queríamos. O lado direito da disjunção já é igual a γ , e assim concluimos a prova:

$$\frac{\frac{\frac{\neg\varphi \vee \gamma}{\neg\varphi \vee \gamma} \quad \frac{\frac{[\neg\varphi]^v \quad [\varphi]^u}{\perp} (\neg_e)}{\gamma} (\perp_e)}{\gamma} (\vee_e) v, w}{\varphi \rightarrow \gamma} (\rightarrow_i) u$$

Segue a mesma prova em Coq:

Section *exemplo_or_to_imp*.

Hypothesis *H*: $(\sim p) \vee q$.

Lemma *or_to_imp*: $p \rightarrow q$.

End *exemplo_or_to_imp*.

Exercícios

Prove os seguintes a seguir:

1. $(\neg\neg\varphi) \rightarrow (\neg\neg\psi) \vdash \neg\neg(\varphi \rightarrow \psi)$
2. $\vdash \neg\neg(\neg\neg\varphi \rightarrow \varphi)$
3. $\vdash \neg\neg(((\varphi \rightarrow \psi) \rightarrow \varphi) \rightarrow \varphi)$

A seguir, reproduza as suas soluções em Coq.

Section *exercicio_nnto*.

Hypothesis *H*: $(\sim\sim p) \rightarrow (\sim\sim q)$.

Lemma *nnto*: $\sim\sim(p \rightarrow q)$.

End *exercicio_nnto*.

Lemma *nne_to*: $\sim\sim(\sim\sim p \rightarrow p)$.

1.4 A Lógica Proposicional Clássica

[aula 07 (10 exercícios)]

Vamos caminhar na direção de mais uma extensão, agora da lógica intuicionista para a lógica clássica. Iniciamos com a lógica proposicional minimal, depois a estendemos para a lógica proposicional intuicionista, e agora vamos estendê-la com a lei do terceiro excluído, obtendo assim a lógica proposicional clássica. Na Tabela ?? apresentamos também as regras com contexto explícito para que tenhamos sempre em mente como os contextos mudam de acordo com a aplicação das regras.

Como exemplo, vamos construir uma prova de uma regra conhecida como *prova por contradição* (PBC). A ideia desta regra consiste em assumir a negação do que se quer provar, e a partir daí derivar uma contradição. O seguinte a ser provado é o seguinte $(\neg\varphi) \rightarrow \perp \vdash \varphi$. Veja que queremos provar φ , e para isto estamos assumindo que a negação de φ nos leva a uma contradição. Vamos então tomar uma instância da (LEM), e provar φ via a eliminação da disjunção:

$$(LEM) \frac{\frac{\frac{\varphi \vee \neg\varphi}{\varphi \vee \neg\varphi} \quad [\varphi]^u}{\varphi} \quad \frac{\frac{\frac{(\neg\varphi) \rightarrow \perp \quad [\neg\varphi]^v}{\perp} (\rightarrow_e)}{\varphi} (\perp_e)}{\varphi} (\vee_e) \quad u, v}{\varphi}$$

	Contexto explícito	Contexto implícito
1	$\frac{\Gamma \vdash \varphi_1 \quad \Gamma \vdash \varphi_2}{\Gamma \vdash \varphi_1 \wedge \varphi_2} (\wedge_i)$	$\frac{\varphi_1 \quad \varphi_2}{\varphi_1 \wedge \varphi_2} (\wedge_i)$
2	$\frac{\Gamma \vdash \varphi_1 \wedge \varphi_2}{\Gamma \vdash \varphi_{i \in \{1,2\}}} (\wedge_e)$	$\frac{\varphi_1 \wedge \varphi_2}{\varphi_{i \in \{1,2\}}} (\wedge_e)$
3	$\frac{\Gamma \vdash \varphi_{i \in \{1,2\}}}{\Gamma \vdash \varphi_1 \vee \varphi_2} (\vee_i)$	$\frac{\varphi_{i \in \{1,2\}}}{\varphi_1 \vee \varphi_2} (\vee_i)$
4	$\frac{\Gamma \vdash \varphi_1 \vee \varphi_2 \quad \Gamma', \varphi_1 \vdash \gamma \quad \Gamma'', \varphi_2 \vdash \gamma}{\Gamma, \Gamma', \Gamma'' \vdash \gamma} (\vee_e)$	$\frac{\varphi_1 \vee \varphi_2 \quad \begin{array}{c} [\varphi_1]^u \\ \vdots \\ \gamma \end{array} \quad \begin{array}{c} [\varphi_2]^v \\ \vdots \\ \gamma \end{array}}{\gamma} (\vee_e) u, v$
5	$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi} (\rightarrow_i)$	$\frac{\begin{array}{c} [\varphi]^u \\ \vdots \\ \psi \end{array}}{\varphi \rightarrow \psi} (\rightarrow_i) u$
6	$\frac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi} (\rightarrow_e)$	$\frac{\varphi \rightarrow \psi \quad \varphi}{\psi} (\rightarrow_e)$
7	$\frac{\Gamma, \varphi \vdash \perp}{\Gamma \vdash \neg \varphi} (\neg_i)$	$\frac{\begin{array}{c} [\varphi]^u \\ \vdots \\ \perp \end{array}}{\neg \varphi} (\neg_i) u$
8	$\frac{\Gamma \vdash \neg \varphi \quad \Gamma \vdash \varphi}{\Gamma \vdash \perp} (\neg_e)$	$\frac{\neg \varphi \quad \varphi}{\perp} (\neg_e)$
9	$\frac{\Gamma \vdash \perp}{\Gamma \vdash \varphi} (\perp_e)$	$\frac{\perp}{\varphi} (\perp_e)$
10	$\frac{}{\vdash \varphi \vee \neg \varphi} (\text{LEM})$	$\frac{}{\varphi \vee \neg \varphi} (\text{LEM})$

Table 1.4: Regras da Lógica Clássica

A regra de prova por contradição é dada a seguir. Observe como o contexto muda por conta do descarte de hipóteses:

Contexto explícito	Contexto implícito
$\frac{\Gamma, \neg\varphi \vdash \perp}{\Gamma \vdash \varphi} \text{ (PBC)}$	$\frac{[\neg\varphi]^u \quad \vdots \quad \perp}{\varphi} \text{ (PBC) } u$

Section *exemplo_pbc*.

Hypothesis *lem*: $p \vee \neg p$.

Hypothesis *H*: $\neg p \rightarrow \text{False}$.

Lemma *pb*: p .

End *exemplo_pbc*.

Acabamos de caracterizar a lógica proposicional clássica como sendo a lógica proposicional intuicionista juntamente com a lei do terceiro excluído (ver Tabela 1.4), mas será que poderíamos ter, por exemplo, caracterizado a lógica proposicional clássica como sendo a lógica proposicional intuicionista juntamente com a regra (PBC), e a partir daí construir uma prova para a lei do terceiro excluídos? Sim! Veja a seguir uma prova de (LEM) utilizando as regras da lógica proposicional intuicionista e (PBC):

$$\begin{array}{c}
 \frac{[\varphi \vee \neg\varphi]^u \quad \frac{[\varphi]^v}{\varphi \vee \neg\varphi} \text{ (V}_i\text{)}}{\perp} \text{ (}\neg_e\text{)} \\
 \frac{\perp}{\neg\varphi} \text{ (}\neg_i\text{) } v \\
 \frac{\neg\varphi}{\varphi \vee \neg\varphi} \text{ (V}_i\text{)} \quad [\varphi \vee \neg\varphi]^u \\
 \frac{\varphi \vee \neg\varphi \quad [\varphi \vee \neg\varphi]^u}{\perp} \text{ (}\neg_e\text{)} \\
 \frac{\perp}{\varphi \vee \neg\varphi} \text{ (PBC) } u
 \end{array}$$

As duas últimas provas nos mostram que as regras (LEM) e (PBC) são equivalentes módulo a lógica proposicional intuicionista. Em outras palavras, podemos provar (LEM) assumindo (PBC) e vice-versa, dadas as regras da lógica proposicional intuicionista. Como exercício, refaça a prova anterior no Coq.

Exercícios

Section *exercicio_pbc_to_lem*.

Hypothesis *pb*: $(\sim(p \vee \neg p) \rightarrow \text{False}) \rightarrow p \vee \neg p$.

Lemma *lem*: $p \vee \neg p$.

End *exercicio_pbc_to_lem*.

Outras caracterizações para a lógica proposicional clássica também são possíveis. Por exemplo, a regra de eliminação da dupla negação também é equivalente a (LEM) e (PBC). O mesmo vale para a lei de Peirce (LP) conforme sugerido nos exercícios a seguir:

1. $\neg\neg\varphi \vdash_{i+(PBC)} \varphi$, i.e. prove $(\neg\neg_e)$ utilizando a regra (PBC) juntamente com as regras da lógica proposicional intuicionista;
2. $\neg\varphi \rightarrow \perp \vdash_{i+(\neg\neg_e)} \varphi$, i.e. prove (PBC) utilizando a regra $(\neg\neg_e)$ juntamente com as regras da lógica proposicional intuicionista;
3. $\vdash_{i+(\neg\neg_e)} ((\varphi \rightarrow \psi) \rightarrow \varphi) \rightarrow \varphi$, i.e. prove (LP) utilizando a regra $(\neg\neg_e)$ juntamente com as regras da lógica proposicional intuicionista;
4. $\vdash_{i+(PBC)} ((\varphi \rightarrow \psi) \rightarrow \varphi) \rightarrow \varphi$, i.e. prove (LP) utilizando a regra (PBC) juntamente com as regras da lógica proposicional intuicionista;
5. $\vdash_{i+(LEM)} ((\varphi \rightarrow \psi) \rightarrow \varphi) \rightarrow \varphi$, i.e. prove (LP) utilizando a regra (LEM) juntamente com as regras da lógica proposicional intuicionista;
6. $\vdash_{i+(LP)} \varphi \vee \neg\varphi$, i.e. prove (LEM) utilizando a regra (LP) juntamente com as regras da lógica proposicional intuicionista;
7. $(\neg\varphi) \rightarrow \perp \vdash_{i+(LP)} \varphi$, i.e. prove (PBC) utilizando a regra (LP) juntamente com as regras da lógica proposicional intuicionista;
8. $\neg\neg\varphi \vdash_{i+(LP)} \varphi$, i.e. prove $(\neg\neg_e)$ utilizando a regra (LP) juntamente com as regras da lógica proposicional intuicionista;
9. Os exercícios propostos no itens anteriores são interessantes porque cada um estabelece um desafio diferente a ser provado, no entanto, não precisamos de tantas provas para estabelecer a equivalência entre as regras (LEM), (PBC), $(\neg\neg_e)$ e (LP). Qual é o número mínimo de provas necessárias para estabelecer esta equivalência? Justifique sua resposta e use o Coq para certificar sua resposta.
10. Prove que a regra da explosão (\perp_e) pode ser derivada a partir das regras da lógica proposicional intuicionista juntamente com (PBC).

1.4.1 A semântica da lógica proposicional clássica

[aula 08 (0 exercícios)]

Iniciamos nosso estudo dizendo que a lógica proposicional é baseada na noção de proposição, e que uma proposição é uma afirmação que pode ser qualificada como verdadeira ou falsa. Nesta seção veremos como atribuir valor de verdade a uma fórmula da lógica proposicional. Temos dois valores de verdade possíveis: verdadeiro (V) ou falso (F). Considerando novamente a gramática (1.1) da lógica proposicional

$$\varphi ::= p \mid \perp \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \quad (1.5)$$

“noindent vamos detalhar a seguir como atribuir valor de verdade para as fórmulas considerando cada um dos construtores possíveis:

1. O primeiro construtor denota uma variável proposicional, que pode assumir qualquer um dos valores de verdade possível. Assim, para atribuir um valor de verdade específico para uma variável precisaremos de alguma informação adicional que nos permita concluir o seu valor de verdade;
2. O segundo construtor é uma constante que denota o absurdo, que é utilizado para representar uma fórmula que tem valor de verdade falso (F);
3. O terceiro construtor denota a negação, cuja semântica consiste em inverter o valor de verdade da fórmula recebida como argumento. Assim, se uma fórmula φ é verdadeira (T) então sua negação é falsa (F), e vice-versa. Normalmente, representamos este fato via a seguinte tabela:

φ	$\neg\varphi$
V	F
F	V

4. O quarto construtor denota a conjunção, cuja semântica é apresentada na seguinte tabela:

φ	ψ	$\varphi \wedge \psi$
V	V	V
V	F	F
F	V	F
F	F	F

5. O quinto construtor denota a disjunção, cuja semântica é apresentada na seguinte tabela:

φ	ψ	$\varphi \vee \psi$
V	V	V
V	F	V
F	V	V
F	F	F

6. Por fim, o sexto construtor denota a implicação, cuja semântica é apresentada na seguinte tabela:

φ	ψ	$\varphi \rightarrow \psi$
V	V	V
V	F	F
F	V	V
F	F	V

O sentido usual (ou popular) da implicação assume implicitamente uma relação de causa e efeito, ou causa e consequência no sentido de que o antecedente φ é o que gera o conseqüente ψ como em "Se eu não beber água então ficarei desidratado". No entanto, o sentido da implicação na lógica proposicional é diferente pois tem como fundamento a *preservação da verdade*, que não necessariamente possui uma relação de causa e efeito. Por exemplo, a proposição *Se $2+2=4$ então o dia tem 24 horas* é verdadeira, mas não existe relação causal entre a igualdade $2+2=4$ e o fato de o dia ter 24 horas de duração.

Apesar da gramática (1.1) não incluir a bi-implicação, este é um conectivo bastante utilizado. No entanto, podemos facilmente expressar a bi-implicação usando a implicação e conjunção. Ao construirmos a tabela verdade de uma fórmula qualquer, teremos três situações possíveis que originam a seguinte nomenclatura:

1. **Tautologia:** Fórmula que é sempre verdadeira, independentemente dos valores de verdade associados às suas variáveis;
2. **Contradição:** Fórmula que é sempre falsa, independentemente dos valores de verdade associados às suas variáveis;
3. **Contingência:** Fórmula que pode ser tanto verdadeira quanto falsa dependendo dos valores de verdade associados às suas variáveis.

As tautologias e as contradições são particularmente importantes, e possuem símbolos especiais para representá-las. Na gramática (1.1) já vimos que a constante \perp é o representante das contradições. As tautologias, por sua vez, podem ser representadas pelo símbolo \top .

Agora chegamos em um momento chave do nosso estudo. Considere um seqüente arbitrário, digamos $\Gamma \vdash \varphi$, onde Γ é um conjunto finito de fórmulas da lógica proposicional. Podemos então perguntar: é possível provar este seqüente? Ou em outras palavras, qualquer seqüente possui uma prova? A resposta certamente é não. Se tudo pudesse ser provado então não teríamos razão para estudar a lógica proposicional. Como então é possível separar os seqüentes que têm prova dos que não têm prova? Para responder a esta pergunta precisamos inicialmente compreender a noção de {"it consequência lógica}. Dizemos que uma fórmula φ é consequência lógica da fórmula ψ , notação $\psi \models \varphi$, se φ for verdadeira sempre que ψ for verdadeira. Este conceito pode ser facilmente estendido para um conjunto Γ de fórmulas, de forma que $\Gamma \models \varphi$, isto é, φ é consequência lógica do conjunto Γ se φ for verdadeira sempre que as fórmulas em Γ forem verdadeiras. Agora podemos enunciar dois teoremas importantes que nos permitirão responder à questão anterior:

Teorema 1 (Correção da LP) *Sejam Γ um conjunto, e φ uma fórmula da lógica proposicional. Se $\Gamma \vdash \varphi$ então $\Gamma \models \varphi$.*

A prova deste teorema é por indução em $\Gamma \vdash \varphi$. Não detalharemos aqui esta prova, que pode ser encontrada por exemplo em [?].

Teorema 2 (Completeness da LP) *Sejam Γ um conjunto, e φ uma fórmula da lógica proposicional. Se $\Gamma \models \varphi$ então $\Gamma \vdash \varphi$.*

A prova do teorema de completude da LP é um pouco mais complexa do que a prova de correção, e também pode ser encontrada em [?]. Note que este lema responde a nossa pergunta anterior: um sequente tem prova exatamente quando seu conseqüente for consequência lógica do seu antecedente.

A lógica proposicional nos permite resolver diversos problemas interessantes, e constitui a base de tudo o que faremos nos próximos capítulos. Apesar de muito importante como ponto de partida no estudo que estamos fazendo, a lógica proposicional possui limitações importantes, como a impossibilidade de quantificar de forma explícita sobre elementos de um conjunto. Por exemplo, podemos representar a sentença *Todo mundo gosta de Matemática* na lógica proposicional via uma variável, mas esta representação não expressa a quantificação universal *Todo mundo* de forma explícita. O mesmo vale para uma sentença da forma *Existe um número natural que não é primo*. O próprio princípio da indução, tão importante em Matemática e Computação, precisa de uma linguagem mais expressiva do que a proposicional. A lógica que nos permitirá expressar este tipo de quantificação (tanto existencial quanto universal) é conhecida como *Lógica de Primeira Ordem* (LPO), ou *Lógica de Predicados* que estudaremos no próximo capítulo.

[aula 09]

Aula de revisão

[aula 10]

Primeira avaliação escrita (individual e sem consulta) (20 pontos).

Referências

- [ARdM17] M. Ayala-Rincón and F.L.C. de Moura. *Applied Logic for Computer Scientists - computational deduction and formal proofs*. UTiCS, Springer, 2017.
- [CLRS09] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Electrical Engineering and Computer Science Series. MIT press, third edition, 2009.