A Lógica Proposicional

Iniciaremos nosso estudo com a *lógica proposicional* (LP), que é uma lógica baseada na noção de **proposição**. Uma proposição, por sua vez, é simplesmente uma sentença que pode ser qualificada como verdadeira ou falsa. São exemplos de proposição:

- 2+2 = 4.
- 1+3 < 0.
- 2 é um número primo.
- João tem 20 anos e Maria tem 22 anos.

Mas nem toda sentença é uma proposição. De fato, a sentença "Feche a porta!"não pode ser qualificada como verdadeira ou falsa, e portanto não é uma proposição. 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 $p \wedge q$. Vejamos então a gramática utilizada na construção das fórmulas da LP, que serão representadas por letras gregas minúsculas:

$$\varphi ::= p \mid \bot \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \to \varphi) \tag{1}$$

A gramática (1) define como as fórmulas da LP são construídas. Ela 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 ($\bot$), 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 denota a negação.

4. O quarto construtor denota a conjunção.

5. O quinto construtor denota a disjunção.

6. O sexto construtor é a implicação.

Uma gramática como (1) nos fornece as regras sintáticas para a construção das fórmulas da LP. 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.

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 LP. 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 $\varphi_1, \varphi_2, \ldots, \varphi_n$ são as hipóteses de um dado problema, e ψ é a conclusão, escrevemos $\varphi_1, \varphi_2, \ldots, \varphi_n \vdash \psi$ para representar o sequente que nos permite provar ψ a partir das hipóteses $\varphi_1, \varphi_2, \ldots, \varphi_n$. O conjunto $\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$, isto é, a primeira componente do sequente $\varphi_1, \varphi_2, \ldots, \varphi_n \vdash \psi$ 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, 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 de um sequente da forma $\Gamma \vdash \psi$ consiste em uma árvore cujos nós são anotados com sequentes, a raiz da árvore está anotada com o sequente que queremos provar, isto é, $\Gamma \vdash \psi$, e as folhas da árvore estão anotadas com axiomas. Um axioma é uma regra da forma:

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

Ou seja, um axioma é um sequente que tem como conclusão uma fórmula que está no contexto.

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 as regras que definem o sistema de dedução natural. As regras são divididas em dois tipos: *regras de introdução* e *regras de eliminação* dos conectivos. As regras de introdução são bastante intuitivas e, em certo sentido, podem ser vistas como uma definição do conectivo que estão introduzindo. Por exemplo, a primeira regra que veremos consiste na *regra de introdução da conjunção*, denotada por $(\wedge_i)$. Esta regra nos diz o que precisamos fazer para construir uma prova de um sequente que possui uma conjunção na conclusão, isto é, um sequente da forma $\Gamma \vdash \varphi_1 \wedge \varphi_2$, onde Γ é um conjunto finito de fórmulas da LP, e φ_1 e φ_2 são fórmulas da LP. A regra $(\wedge_i)$ é dada pela seguinte regra de inferência:

$$\frac{\Gamma \vdash \varphi_1 \qquad \Gamma \vdash \varphi_2}{\Gamma \vdash \varphi_1 \wedge \varphi_2}\ (\wedge_i) \tag{2}$$

Esta regra nos diz que 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 $\Gamma \vdash \varphi_2$.

Como exemplo de utilização desta regra, construiremos uma prova para o sequente $\varphi, \psi \vdash \varphi \wedge \psi$. Podemos aplicar a regra acima instanciando Γ, φ_1 e φ_2, respectivamente com o conjunto $\{\varphi, \psi\}$, e com

as fórmulas φ e ψ. Como resultado temos a árvore abaixo onde os dois ramos gerados por $(\wedge_i)$ são axiomas:

$$\dfrac{(\text{Ax})\ \dfrac{}{\varphi, \psi \vdash \varphi} \qquad \dfrac{}{\varphi, \psi \vdash \psi}\ (\text{Ax})}{\varphi, \psi \vdash \varphi \wedge \psi}\ (\wedge_i)$$

Apresentaremos as regras do sistema de dedução natural em paralelo com o assistente de provas Coq. Esta é uma forma de aprendermos a utilizar o sistema de uma forma progressiva e suave. O Coq implementa uma lógica de ordem superior baseado em dedução natural. Isto quer dizer que será possível fazer uma analogia entre o sistema de dedução natural que apresentamos aqui e o Coq, mas como veremos, esta analogia não é feita via uma correspondência direta entre as regras em dedução natural e as *regras* do Coq que são chamadas de *táticas*. De fato, as táticas são desenvolvidas para realizarem vários passos de prova de uma vez porque isto facilita o processo de construção de provas em sistemas mais complexos. Iniciaremos com a prova do exemplo anterior, que nos permitirá construir a prova de uma conjunção em Coq. Para simularmos a regra de introdução da conjunção vamos declarar duas variáveis `phi` e `psi`, e em seguida, criaremos uma seção que vai delimitar o escopo da prova. Chamaremos esta seção de `landi`, e então declaramos as hipóteses e o lema propriamente dito dentro da seção:

```
Variables phi psi: Prop.

Section landi.
 Hypothesis H1: phi.
 Hypothesis H2: psi.
 Lemma landi: phi /\ psi.

End landi.
```

Esta é uma forma de declarar o sequente `phi, psi` $\vdash$ `phi` $\wedge$ `psi` no Coq. De fato, na janela de prova temos o sequente como esperado:

```
H1 : phi
H2 : psi
============================
phi /\ psi
```

Portanto uma prova deste sequente é o que vai corresponder a uma aplicação da regra $(\wedge_i)$. A prova é construída entre as palavras reservadas `Proof` (que denota o início da prova) e `Qed` (que indica que a prova foi finalizada). Utilizamos a tática `split` para dividirmos a prova da conjunção em subprovas das suas componentes (já que a construção em Coq é sempre feita de baixo para cima, isto é da raiz para as folhas da árvore) que por sua vez são hipóteses, e a prova de algo que já faz parte do conjunto de hipóteses pode ser concluída com a tática `assumption`:

```
Variables phi psi: Prop.

Section landi.
 Hypothesis H1: phi.
 Hypothesis H2: psi.
 Lemma landi: phi /\ psi.
 Proof.
   split.
   - assumption.
```

```
    - assumption.
  Qed.
 End landi.
```

Logo, a regra $(\wedge_i)$ está relacionada com a tática `split` e o axioma com a tática `assumption`. Uma maneira de ver isto de forma mais explícita consiste em comparar a árvore de prova do sequente `phi, psi` $\vdash$ `phi` $\wedge$ `psi`:

$$\dfrac{(\text{Ax})\ \dfrac{}{\texttt{phi, psi} \vdash \texttt{phi}} \qquad \dfrac{}{\texttt{phi, psi} \vdash \texttt{psi}}\ (\text{Ax})}{\texttt{phi, psi} \vdash \texttt{phi} \wedge \texttt{psi}}\ (\wedge_i) \qquad \dfrac{\texttt{assumption}\ \dfrac{}{\texttt{phi, psi} \vdash \texttt{phi}} \qquad \dfrac{}{\texttt{phi, psi} \vdash \texttt{psi}}\ \texttt{assumption}}{\texttt{phi, psi} \vdash \texttt{phi} \wedge \texttt{psi}}\ \texttt{split}$$

Neste caso, temos uma correspondência direta entre uma regra do sistema de dedução natural e o Coq, mas este não é sempre o caso. .

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

$$\dfrac{\Gamma \vdash \varphi_1 \wedge \varphi_2}{\Gamma \vdash \varphi_1}\ (\wedge_{e_1}) \qquad\qquad \dfrac{\Gamma \vdash \varphi_1 \wedge \varphi_2}{\Gamma \vdash \varphi_2}\ (\wedge_{e_2})$$

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

$$\dfrac{\Gamma \vdash \varphi_1 \wedge \varphi_2}{\Gamma \vdash \varphi_{i \in \{1,2\}}}\ (\wedge_e) \tag{3}$$

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. Como vimos, as provas em Coq são construídas de baixo para cima, isto é, partimos da raiz da árvore de dedução que é o sequente que queremos provar, e subimos até as folhas que são os axiomas. Em provas simples conseguimos seguir este caminho sem dificuldades, mas em provas mais complexas precisamos ter mais flexibilidade. Em papel e lápis, as provas podem ser construídas tanto de baixo para cima (da raiz para as folhas) quanto de cima para baixo, e na prática usamos as duas estratégias porque dependendo do contexto, pode ser melhor uma estratégia ou outra. Veremos diversos exemplos para facilitar a compreensão desta ideia, e em Coq o mesmo acontece: a cada instante da construção de uma prova podemos tanto dar um passo de baixo para cima aplicando uma tática que altera o objetivo (raiz da árvore que estamos construindo) quanto uma tática que altera uma hipótese que corresponde a um passo de cima para baixo na construção da prova. Vejamos um exemplo de tática do Coq que altera uma hipótese. Para isto, considere o sequente que tem uma conjunção como hipótese, e uma das componentes desta conjunção como conclusão: `phi` $\wedge$ `psi` $\vdash$ `phi`:

```
Variables phi psi: Prop.

Section landel.
 Hypothesis H: phi /\ psi.
 Lemma landel: phi.
 Proof.
```

Neste momento a janela de prova tem a seguinte forma:

```
1 subgoal (ID 1)

 H : phi /\ psi
 ============================
 phi
```

Podemos usar tática `inversion H` para decompor a hipótese deste sequente, e obtemos:

```
1 subgoal (ID 1)

 H : phi /\ psi
 H0 : phi
 H1 : psi
 ============================
 phi
```

E a prova pode ser concluída com a tática `assumption`. Não entraremos neste momento nos detalhes técnicos da tática `inversion`, mas grosso modo, ela gera as condições necessárias para a construção da hipótese onde ela está sendo aplicada. Para mais detalhes recomendamos que o leitor consulte o manual do usuário do Coq[1]. Existem outras táticas que podemos usar no lugar de `inversion` no exemplo anterior, como `destruct` e `elim`, mas elas não serão abordadas agora. No entanto, táticas diferentes podem ser usadas para construir provas diferentes de um mesmo sequente, assim como ocorre em papel e lápis.

Com as regras da conjunção já podemos fazer um exercício interessante: provar a comutatividade da conjunção, isto é, queremos construir uma prova para o sequente $\varphi \wedge \psi \vdash \psi \wedge \varphi$, onde φ e ψ são fórmulas quaisquer da LP. Vamos construir esta prova passo a passo. A construção da prova é feita inicialmente de baixo para cima, isto é, da raiz para as folhas. Iniciamos observando que a raiz da nossa árvore de prova possui o sequente $\varphi \wedge \psi \vdash \psi \wedge \varphi$:

$$\frac{?}{\varphi \wedge \psi \vdash \psi \wedge \varphi}$$

Considerando as únicas regras que temos até o momento, a saber (2) e (3), é natural imaginarmos que a conclusão será obtida via a regra $(\wedge_i)$:

$$\frac{\dfrac{?}{\varphi \wedge \psi \vdash \psi} \qquad \dfrac{?}{\varphi \wedge \psi \vdash \varphi}}{\varphi \wedge \psi \vdash \psi \wedge \varphi}\ (\wedge_i)$$

Agora podemos concluir com a regra de eliminação da conjunção e o axioma:

$$\frac{\dfrac{\dfrac{}{\varphi \wedge \psi \vdash \varphi \wedge \psi}\ (\text{Ax})}{\varphi \wedge \psi \vdash \psi}\ (\wedge_e) \qquad \dfrac{\dfrac{}{\varphi \wedge \psi \vdash \varphi \wedge \psi}\ (\text{Ax})}{\varphi \wedge \psi \vdash \varphi}\ (\wedge_e)}{\varphi \wedge \psi \vdash \psi \wedge \varphi}\ (\wedge_i)$$

[1] `https://coq.inria.fr/distrib/current/refman/`

Exercício 1. *Utilize os seus conhecimentos de Coq para provar que a conjunção é comutativa.*

Exercício 2. *Em papel e lápis, prove que a conjunção é associativa, isto é, prove o sequente* $(\varphi \wedge \psi) \wedge \rho \vdash \varphi \wedge (\psi \wedge \rho)$. *Agora prove a associatividade da conjunção no Coq.*

Observe que no exercício anterior, solicitamos primeiro uma solução em papel e lápis para, somente depois, solicitar a prova no Coq. Este é um detalhe importante porque os assistentes de prova não são ferramentas para nos ajudar a construir provas, mas sim para verificar provas. A ideia é utilizar os assistentes de prova para mecanizarmos uma prova que já tenha sido feito em papel e lápis, ou uma prova que temos na cabeça (mesmo que apenas um esboço). Iniciar uma prova em um assistente de provas sem saber inicialmente que caminho seguir, tentando a sorte, em geral não é uma boa ideia.

Vejamos 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 alguma das suas componentes:

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

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

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

As regras $(\vee_{i_1})$ e $(\vee_{i_2})$ são implementadas pelas táticas `left` e `right` do Coq, respectivamente. As árvores de prova abaixo mostram esta correspondência considerando o sequente `phi1` $\vdash$ `phi1` $\vee$ `phi2` para o caso $(\vee_{i_1})$, e `phi2` $\vdash$ `phi1` $\vee$ `phi2` para o caso $(\vee_{i_2})$:

$$\dfrac{\dfrac{}{\texttt{phi1} \vdash \texttt{phi1}}\ (\text{Ax})}{\texttt{phi1} \vdash \texttt{phi1} \vee \texttt{phi2}}\ (\vee_{i_1}) \qquad\qquad \dfrac{\dfrac{}{\texttt{phi1} \vdash \texttt{phi1}}\ \texttt{assumption}}{\texttt{phi1} \vdash \texttt{phi1} \vee \texttt{phi2}}\ \texttt{left}$$

$$\dfrac{\dfrac{}{\texttt{phi2} \vdash \texttt{phi2}}\ (\text{Ax})}{\texttt{phi2} \vdash \texttt{phi1} \vee \texttt{phi2}}\ (\vee_{i_2}) \qquad\qquad \dfrac{\dfrac{}{\texttt{phi2} \vdash \texttt{phi2}}\ \texttt{assumption}}{\texttt{phi2} \vdash \texttt{phi1} \vee \texttt{phi2}}\ \texttt{right}$$

A regra de eliminação da disjunção é um pouco mais sofisticada do que as que vimos até aqui. A ideia é que para provarmos algo, digamos γ, a partir de uma disjunção, precisamos provar γ a partir de cada uma das componentes da disjunção: :

$$\frac{\Gamma \vdash \varphi_1 \vee \varphi_2 \qquad \Gamma, \varphi_1 \vdash \gamma \qquad \Gamma, \varphi_2 \vdash \gamma}{\Gamma \vdash \gamma}\ (\vee_{\text{e}})$$

Assim, para que tenhamos uma prova de γ a partir das fórmulas em Γ (sequente $\Gamma \vdash \gamma$) precisamos de 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 $\Gamma, \varphi_2 \vdash \gamma$). Observe como os contextos mudam em cada um dos sequentes que compõem esta regra. Este é um detalhe importante que não ocorreu nas regras anteriores.

Exemplo 3. *Vamos mostrar 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* $(\vee_{\mathrm{e}})$*. Para isto podemos instanciar* Γ *com o conjunto unitário contendo a fórmula* $\varphi \vee \psi$*. Em função da estrutura da regra* $(\vee_{\mathrm{e}})$*, 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* $(\vee_i)$*:*

$$\dfrac{(\mathrm{Ax})\ \dfrac{}{\varphi \vee \psi \vdash \varphi \vee \psi} \qquad \dfrac{\dfrac{\dfrac{}{\varphi \vdash \varphi}\ (\mathrm{Ax})}{\varphi \vdash \psi \vee \varphi}\ (\vee_i) \quad \dfrac{\dfrac{}{\psi \vdash \psi}\ (\mathrm{Ax})}{\psi \vdash \psi \vee \varphi}\ (\vee_i)}{\varphi \vee \psi \vdash \psi \vee \varphi}\ (\vee_{\mathrm{e}})}$$

Em Coq, o sequente deste exemplo pode ser escrito declarando duas variáveis `phi` e `psi`, e a hipótese `H: phi \/ psi.`:

```
Variables phi psi: Prop.

Section or_comm.
Hypothesis H: phi \/ psi.
Lemma or_comm: psi \/ phi.
Proof.

End or_comm.
```

Temos então o seguinte sequente para ser provado:

```
1 subgoal (ID 1)

 H : phi \/ psi
 ============================
 psi \/ phi
```

A tática `destruct H` vai dividir a prova em duas subprovas. A primeira nos pede para provar `psi \/ phi` a partir de `phi`:

```
 H : phi \/ psi
 H0 : phi
 ============================
 psi \/ phi
```

---

que consiste em usar a tática `right` seguida de `assumption`, enquanto que na segunda subprova precisamos provar `psi \/ phi` a partir de `psi`:

```
 H : phi \/ psi
 H0 : psi
 ============================
 psi \/ phi
```

---

que consiste em uma aplicação da tática `left` seguida de `assumption`.

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 conseguir 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 $\Gamma \vdash \varphi \rightarrow \psi$ ao novo problema (possivelmente mais simples) de provar o sequente $\Gamma, \varphi \vdash \psi$:

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

Também podemos observar esta regra de cima para baixo. Neste caso, ela nos permite passar uma fórmula do conjunto de hipóteses para o consequente 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 consequente. Posteriormente, esta regra será explorada em mais detalhes. Em Coq, esta regra é simulada por meio da tática `intro`.

```
 ============================
  phi -> psi
```

A tática `intro` vai mover o antecedente `phi` da implicação para as hipóteses:

```
H : phi
  ============================
  psi
```

Portanto a tática `intro` corresponde a uma aplicação da regra de introdução da implicação. A regra de *eliminação da implicação* é a mais famosa das regras que veremos, a ponto de possuir um nome próprio, a saber *modus ponens*. Esta regra nos diz como podemos usar a prova de uma implicação juntamente com uma prova do antecedente desta implicação:

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

Quando lida de baixo para cima, esta regra corresponde a uma aplicação da regra do corte, que em Coq corresponde à tática `cut`. Assim, aplicando a tática `cut phi`, a prova se divide em dois ramos, ou em dois subobjetivos.

```
2 subgoals (ID 2)

 ============================
  phi -> psi

subgoal 2 (ID 3) is:
 phi
```

Exemplo 4. *Considere o sequente* $\Gamma, \varphi \to \psi, \varphi \vdash \psi$. *Podemos prová-lo usando a regra* $(\to_e)$ *da seguinte forma:*

$$\dfrac{\dfrac{}{\Gamma, \varphi \to \psi, \varphi \vdash \varphi \to \psi}\,(\text{Ax}) \quad \dfrac{}{\Gamma, \varphi \to \psi, \varphi \vdash \varphi}\,(\text{Ax})}{\Gamma, \varphi \to \psi, \varphi \vdash \psi}\,(\to_e)$$

Agora faremos a prova acima em Coq considerando o conjunto Γ *como sendo o conjunto vazio. A declaração do sequente é feita como a seguir:*

```
Variables phi psi: Prop.

Section imp_e.
 Hypothesis H1: phi -> psi.
 Hypothesis H2: phi.
 Lemma imp_e: psi.
 Proof.
```

e o ambiente de prova corresponde ao sequente abaixo:

```
H1 : phi -> psi
H2 : phi
============================
psi
```

Agora podemos aplicar a tática **`cut phi`** *como explicado acima, e concluir a prova com* **`assumption`**.

```
Section imp_e.
 Hypothesis H1: phi -> psi.
 Hypothesis H2: phi.
 Lemma imp_e: psi.
 Proof.
   cut phi.
   - assumption.
   - assumption.
 Qed.
End imp_e.
```

Um outro caminho possível para provar este sequente é por meio da tática **`apply H1`** *seguida de* **`assumption`**. *Neste caso, o consequente da hipótese* **`H1`** *é confrontado com o objetivo a ser provado. Como eles coincidem, o novo objetivo gerado passa a ser* **`phi`**, *ou seja, o antecedente da hipótese* **`H1`**.

```
Section imp_e.
 Hypothesis H1: phi -> psi.
 Hypothesis H2: phi.
 Lemma imp_e2: psi.
 Proof.
  apply H1.
  assumption.
Qed.
End imp_e.
```

	Dedução Natural	Tática Coq
1	$\dfrac{\Gamma \vdash \varphi_1 \qquad \Gamma \vdash \varphi_2}{\Gamma \vdash \varphi_1 \wedge \varphi_2}\ (\wedge_i)$	`split`
2	$\dfrac{\Gamma \vdash \varphi_1 \wedge \varphi_2}{\Gamma \vdash \varphi_{i\in\{1,2\}}}\ (\wedge_e)$	`destruct`
3	$\dfrac{\Gamma \vdash \varphi_{i\in\{1,2\}}}{\Gamma \vdash \varphi_1 \vee \varphi_2}\ (\vee_i)$	`left`/`right`
4	$\dfrac{\Gamma \vdash \varphi_1 \vee \varphi_2 \qquad \Gamma, \varphi_1 \vdash \gamma \qquad \Gamma, \varphi_2 \vdash \gamma}{\Gamma, \varphi_1 \vee \varphi_2 \vdash \gamma}\ (\vee_e)$	`destruct`
5	$\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}\ (\to_i)$	`intro`
6	$\dfrac{\Gamma \vdash \varphi \to \psi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \psi}\ (\to_e)$	`cut/apply`
7	$\dfrac{\Gamma, \varphi \vdash \bot}{\Gamma \vdash \neg\varphi}\ (\neg_i)$	`intro`
8	$\dfrac{\Gamma \vdash \neg\varphi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \bot}\ (\neg_e)$	`cut/apply`

Tabela 1: Regras da Lógica Minimal

Por fim, podemos também usar a tática ***apply*** *nas hipóteses. Neste caso, o antecedente da hipótese* ***H1*** *é confrontado com a fórmula* ***phi*** *da hipótese* ***H2****. Como estas fórmulas coincidem, a hipótese* ***H2*** *é convertida na fórmula* ***psi*** *e podemos concluir a prova com* ***assumption****.*

```
Section imp_e.
 Hypothesis H1: phi -> psi.
 Hypothesis H2: phi.
 Lemma imp_e3: psi.
 Proof.
  apply H1 in H2.
  assumption.
 Qed.
End imp_e.
```

Exercício 5. *Prove que a disjunção é associativa, isto é, prove o sequente* $(\varphi \vee \psi) \vee \rho \vdash \varphi \vee (\psi \vee \rho)$*. Em seguida, prove a associatividade da disjunção no Coq.*

As regras para a negação são muito similares às regras da implicação, e isto não ocorre por acaso. De fato, uma negação é uma implicação particular porque $\neg\varphi$ é definida como $\varphi \to \bot$. Considerando este fato, a analogia com as regras da implicação é direta.

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

A Tabela 1 apresenta as regras vistas até aqui, assim como algumas táticas do Coq associadas a estas regras. Estas regras formam a chamada *lógica proposicional minimal*.
Agora vamos resolver alguns exercícios na lógica minimal.

Exemplo 6. *Considere o sequente* $\varphi \to \psi, \neg\psi \vdash \neg\varphi$*. Como a fórmula do consequente é uma negação, vamos aplicar a regra de introdução da negação na construção de uma prova de baixo para cima, isto é,*

da raiz para as folhas da árvore:

$$\dfrac{\dfrac{?}{\varphi \to \psi, \neg\psi, \varphi \vdash \bot}}{\varphi \to \psi, \neg\psi \vdash \neg\varphi}\ (\neg_{\mathrm{i}})$$

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 de* φ *da regra 8 da Tabela 1. A princípio temos três opções:* $\varphi \to \psi$*,* $\neg\psi$ *e* φ*. A boa escolha neste caso é* $\neg\psi$ *porque podemos facilmente provar* ψ *a partir deste contexto utilizando a ideia do Exemplo 4:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi \to \psi, \neg\psi, \varphi \vdash \varphi \to \psi}\ (\mathrm{Ax}) \quad \dfrac{}{\varphi \to \psi, \neg\psi, \varphi \vdash \varphi}\ (\mathrm{Ax})}{\varphi \to \psi, \neg\psi, \varphi \vdash \psi}\ (\to_e) \quad \dfrac{}{\varphi \to \psi, \neg\psi, \varphi \vdash \neg\psi}\ (\mathrm{Ax})}{\varphi \to \psi, \neg\psi, \varphi \vdash \bot}\ (\neg_e)}{\varphi \to \psi, \neg\psi \vdash \neg\varphi}\ (\neg_{\mathrm{i}})$$

Depois de concluída a prova é fácil entender o que queríamos dizer com boa escolha *acima: Uma boa escolha é um caminho que vai nos permitir concluir uma prova. Mas como fazer uma boa escolha? Isto depende do problema a ser resolvido. Em alguns casos pode ser simples, mas em outros, bastante complicado. O ponto importante a compreender é que existem caminhos possíveis distintos na construção de provas da lógica proposicional, e muito deste processo depende da nossa criatividade.*

Exercício 7. *O símbolo da negação em Coq é* $\sim$*. Sabendo disto, refaça a prova acima no Coq.*

O sequente que acabamos de provar ocorre com certa frequência em outras provas, de forma que ele é comumente utilizado em outras provas assim como o Exemplo 4 foi utilizado aqui. As regras que são obtidas a partir das regras da Tabela 1 são chamadas de *regras derivadas*. Este é o caso da regra conhecida como *modus tollens* (MT):

$$\dfrac{\Gamma \vdash \varphi \to \psi \qquad \Gamma \vdash \neg\psi}{\Gamma \vdash \neg\varphi}\ (\mathrm{MT})$$

Exemplo 8. *Considere o sequente* $\varphi \to \psi \vdash \neg\psi \to \neg\varphi$*. Inicialmente, devemos observar que a fórmula que queremos provar é uma implicação, e portanto, o mais natural é tentar aplicar a regra* $(\to_{\mathrm{i}})$*, e em seguida aplicar* (MT) *(na construção de baixo para cima) para poder completar a prova:*

$$\dfrac{\dfrac{\dfrac{}{\varphi \to \psi, \neg\psi \vdash \varphi \to \psi}\ (\mathrm{Ax}) \qquad \dfrac{}{\varphi \to \psi, \neg\psi \vdash \neg\psi}\ (\mathrm{Ax})}{\varphi \to \psi, \neg\psi \vdash \neg\varphi}\ (\mathrm{MT})}{\varphi \to \psi \vdash \neg\psi \to \neg\varphi}\ (\to_{\mathrm{i}})$$

Exercício 9. *Reproduza a prova do exemplo anterior no Coq.*

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

$$\dfrac{\Gamma \vdash \varphi \to \psi}{\Gamma \vdash \neg\psi \to \neg\varphi}\ (\mathrm{CP})$$

Este é um bom momento para simplificarmos a notação que estamos usando, e tentaremos deixar clara a vantagem de nossa abordagem com a mudança de notação neste momento. Vamos retomar o Exercício 2 que consiste em provar 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:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{(\phi\wedge\psi)\wedge\varphi\vdash(\phi\wedge\psi)\wedge\varphi}(\mathrm{Ax})}{(\phi\wedge\psi)\wedge\varphi\vdash\phi\wedge\psi}(\wedge_e)}{(\phi\wedge\psi)\wedge\varphi\vdash\phi}(\wedge_e) \qquad \dfrac{\dfrac{\dfrac{}{(\phi\wedge\psi)\wedge\varphi\vdash(\phi\wedge\psi)\wedge\varphi}(\mathrm{Ax})}{(\phi\wedge\psi)\wedge\varphi\vdash\phi\wedge\psi}(\wedge_e)}{(\phi\wedge\psi)\wedge\varphi\vdash\psi} \qquad \dfrac{}{(\phi\wedge\psi)\wedge\varphi\vdash(\phi\wedge\psi)\wedge\varphi}(\mathrm{Ax})}{(\phi\wedge\psi)\wedge\varphi\vdash\varphi}(\wedge_e)}{(\phi\wedge\psi)\wedge\varphi\vdash(\psi\wedge\varphi)}(\wedge_i)}{(\phi\wedge\psi)\wedge\varphi\vdash\phi\wedge(\psi\wedge\varphi)}(\wedge_i)$$

Observe que o contexto, isto é, o antecedente de cada um dos sequentes desta prova é o mesmo. De fato, o contexto em cada nó da árvore acima é o conjunto unitário contendo a fórmula $(\phi\wedge\psi)\wedge\varphi$. Como o que muda ao longo da prova é o consequente dos sequentes, é natural considerar que o foco, ou que a parte principal, desta prova é o consequente de cada sequente. Sabendo com qual contexto estamos trabalhando, podemos removê-lo da prova deixando-a mais limpa e compacta. Veja como fica a prova sem os contextos:

$$\dfrac{\dfrac{\dfrac{(\phi\wedge\psi)\wedge\varphi}{(\phi\wedge\psi)}(\wedge_e)}{\phi}(\wedge_e) \qquad \dfrac{\dfrac{\dfrac{(\phi\wedge\psi)\wedge\varphi}{\phi\wedge\psi}(\wedge_e)}{\psi} \qquad \dfrac{(\phi\wedge\psi)\wedge\varphi}{\varphi}(\wedge_e)}{(\psi\wedge\varphi)}(\wedge_i)}{\phi\wedge(\psi\wedge\varphi)}(\wedge_i)$$

Será que é possível sempre remover os contextos das provas? Sim, e neste exemplo em particular, a situação é simples porque o contexto é o mesmo em toda a prova, mas este não será o caso em geral. Ainda considerando o exemplo anterior, se não soubéssemos qual o contexto que foi apagado, seria possível descobrí-lo? Sim, esta informação vem das folhas da árvore, que são as hipóteses do problema. Neste caso, a única hipótese é a fórmula $(\phi\wedge\psi)\wedge\varphi$ já que todas as folhas são iguais. Agora podemos consultar a Tabela 1 e observar que os contextos da regra $(\wedge_i)$ são os mesmos antes e depois de aplicar a regra. Portanto, o contexto das regras da segunda linha (de baixo para cima) também é o conjunto unitário contendo a fórmula $(\phi\wedge\psi)\wedge\varphi$. A mesma observação vale para a regra $(\wedge_e)$, e portanto o mesmo contexto é utilizado em toda a prova. Outro detalhe importante é que as folhas desta nova árvore não correspondem mais à regra (Ax), e portanto as folhas têm que ser fórmulas pertencentes ao contexto. As árvores onde os contexto ficam implícitos são as árvores normalmente apresentadas na bibliografia que trata de dedução natural.

No Exemplo 6 construímos uma prova do sequente $\varphi\to\psi,\neg\psi\vdash\neg\varphi$, e removendo os contextos obtemos a seguinte árvore de derivação:

$$\dfrac{\dfrac{\dfrac{\varphi\to\psi \qquad \varphi}{\psi} \qquad \neg\psi}{\bot}(\neg_e)}{\neg\varphi}(\neg_i)$$

Agora observe que as fórmulas $\varphi\to\psi$ e $\neg\psi$, que estão nas folhas da árvore, também são fórmulas do contexto. No entanto, a fórmula φ está em uma folha da árvore, mas não é uma fórmula do contexto, e portanto as coisas ficam um pouco mais interessantes aqui. Vamos novamente tentar reconstruir os contextos da raiz para as folhas da árvore. Esta árvore de derivação se refere ao sequente $\varphi\to\psi,\neg\psi\vdash\neg\varphi$, e portanto o contexto da fórmula que está na raiz da árvore é o conjunto $\{\varphi\to\psi,\neg\psi\}$. De acordo com a Tabela 1, a regra $(\neg_i)$ adiciona uma fórmula ao contexto, que neste caso corresponde à fórmula φ,

e portanto o contexto da fórmula $\bot$ (segunda linha de baixo para cima) é o conjunto $\{\varphi \to \psi, \neg\psi, \varphi\}$. Como as regras $(\neg_e)$ e $(\to_e)$ não alteram o contexto, o conjunto $\{\varphi \to \psi, \neg\psi, \varphi\}$ também é o contexto das fórmulas que estão nas folhas desta árvore, e isto é o que permite a fórmula φ, que não faz parte do contexto original do problema, ser uma folha desta árvore. De fato, se o contexto fosse o mesmo em toda a árvore, a folha marcada com φ corresponderia ao sequente $\varphi \to \psi, \neg\psi \vdash \varphi$ que não corresponde a um axioma. Para diferenciarmos as fórmulas que fazem parte do contexto inicial, colocaremos as outras fórmulas entre colchetes para nos lembrarmos deste fato:

$$\dfrac{\dfrac{\dfrac{\varphi \to \psi \qquad [\varphi]}{\psi} (\to_e) \qquad \neg\psi}{\bot} (\neg_e)}{\neg\varphi} (\neg_{\mathrm{i}})$$

e agora fica claro que a fórmula φ não faz parte do contexto inicial. 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 $(\neg_{\mathrm{i}})$. Então, utilizaremos uma letra para registrar este fato:

$$\dfrac{\dfrac{\dfrac{\varphi \to \psi \qquad [\varphi]^u}{\psi} (\to_e) \qquad \neg\psi}{\bot} (\neg_e)}{\neg\varphi} (\neg_i)\ u$$

Agora sabemos em que momento a fórmula φ foi introduzida, e em que momento foi descartada na árvore de derivação. Observe como o Coq faz este trabalho de modificar o contexto da mesma forma que acabamos de descrever:

```
Variables phi psi: Prop.

Section mt.
Hypothesis H1: phi -> psi.
Hypothesis H2: ~psi.
Lemma mt: ~phi.
Proof.
```

Ao iniciarmos a prova do lema `mt`, temos a seguinte configuração:

```
H1 : phi -> psi
H2 : ~ psi
============================
~ phi
```

---

e aplicando a tática `intro`, a fórmula `phi` é introduzida no contexto com a marca `H`:

```
H1 : phi -> psi
H2 : ~ psi
H : phi
============================
False
```

Note que a marca `H` foi criada automaticamente pelo Coq, mas você pode colocar outra marca informando-a como parâmetro da tática `intro`, como por exemplo, `intro u`:

```
H1 : phi -> psi
H2 : ~ psi
u : phi
============================
False
```

Esta prova corresponde ao Exercício 7, cuja solução é dada a seguir:

```
Variables phi psi: Prop.

Section mt.
 Hypothesis H1: phi -> psi.
 Hypothesis H2: ~psi.
 Lemma mt: ~phi.
 Proof.
   intro u.
   apply H2.
   apply H1.
   assumption.
 Qed.

End mt.
```

A seguir, veremos exemplos mais complexos onde fórmulas idênticas podem exigir marcas distintas, mas antes disto comparece as as regras de dedução natural para a lógica proposicional minimal com o contexto explícito e com o contexto implícito na Tabela 2.

Exemplo 10. *Neste exemplo, veremos que é possível fazer a introdução de uma implicação sem precisar descartar uma hipótese, se tivermos uma prova do consequente da implicação que queremos construir. Ou seja, se temos uma prova de ψ então podemos construir uma prova de $\varphi \to \psi$, qualquer que seja a fórmula φ. Em outras palavras, queremos construir uma prova para o sequente $\psi \vdash \varphi \to \psi$. A ideia da prova neste caso é simples. Vamos assumir uma prova de φ, e transformá-la em uma prova de ψ que já temos como hipótese. Para isto basta introduzirmos e em seguida eliminarmos uma conjunção contendo ψ:*

$$\dfrac{\dfrac{\dfrac{[\varphi]^u \qquad \psi}{\varphi \wedge \psi}\,(\wedge_i)}{\psi}\,(\wedge_e)}{\varphi \to \psi}\,(\to_i)\,u$$

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

$$\dfrac{\psi}{\varphi \to \psi}\,(\to_i)\,\emptyset$$

Exercício 11. *Refaça a prova do exemplo anterior no Coq.*

	Contexto explícito	Contexto implícito
1	$\dfrac{\Gamma \vdash \varphi_1 \qquad \Gamma \vdash \varphi_2}{\Gamma \vdash \varphi_1 \wedge \varphi_2}\ (\wedge_i)$	$\dfrac{\varphi_1 \qquad \varphi_2}{\varphi_1 \wedge \varphi_2}\ (\wedge_i)$
2	$\dfrac{\Gamma \vdash \varphi_1 \wedge \varphi_2}{\Gamma \vdash \varphi_{i\in\{1,2\}}}\ (\wedge_e)$	$\dfrac{\varphi_1 \wedge \varphi_2}{\varphi_{i\in\{1,2\}}}\ (\wedge_e)$
3	$\dfrac{\Gamma \vdash \varphi_{i\in\{1,2\}}}{\Gamma \vdash \varphi_1 \vee \varphi_2}\ (\vee_i)$	$\dfrac{\varphi_{i\in\{1,2\}}}{\varphi_1 \vee \varphi_2}\ (\vee_i)$
4	$\dfrac{\Gamma \vdash \varphi_1 \vee \varphi_2 \qquad \Gamma, \varphi_1 \vdash \gamma \qquad \Gamma, \varphi_2 \vdash \gamma}{\Gamma, \varphi_1 \vee \varphi_2 \vdash \gamma}\ (\vee_e)$	$\dfrac{\varphi_1 \vee \varphi_2 \qquad \begin{matrix}[\varphi_1]^u \\ \vdots \\ \gamma\end{matrix} \qquad \begin{matrix}[\varphi_2]^v \\ \vdots \\ \gamma\end{matrix}}{\gamma}\ (\vee_e)\ u, v$
5	$\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}\ (\to_i)$	$\dfrac{\begin{matrix}[\varphi]^u \\ \vdots \\ \psi\end{matrix}}{\varphi \to \psi}\ (\to_i)\ u$
6	$\dfrac{\Gamma \vdash \varphi \to \psi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \psi}\ (\to_e)$	$\dfrac{\varphi \to \psi \qquad \varphi}{\psi}\ (\to_e)$
7	$\dfrac{\Gamma, \varphi \vdash \bot}{\Gamma \vdash \neg\varphi}\ (\neg_i)$	$\dfrac{\begin{matrix}[\varphi]^u \\ \vdots \\ \bot\end{matrix}}{\neg\varphi}\ (\neg_i)\ u$
8	$\dfrac{\Gamma \vdash \neg\varphi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \bot}\ (\neg_e)$	$\dfrac{\neg\varphi \qquad \varphi}{\bot}\ (\neg_e)$

Tabela 2: Regras da Lógica Minimal

Exercício 12. *Sejam φ e γ fórmulas da lógica proposicional. Construa uma prova para o sequente $\varphi, \neg\varphi \vdash \neg\gamma$ na lógica proposicional minimal. Lembre que a negação é o mesmo que uma implicação no absurdo, ou seja, $\neg\varphi$ é o mesmo que $\varphi \to \bot$, qualquer que seja a fórmula φ.*

Também podemos introduzir a dupla negação de uma fórmula qualquer, como solicitado no exercício a seguir:

Exercício 13. *Seja φ uma fórmula da lógica proposicional. Prove o sequente $\varphi \vdash \neg\neg\varphi$.*

O exercício anterior nos dá mais uma regra derivada na lógica proposicional minimal:

$$\frac{\varphi}{\neg\neg\varphi}\ (\neg\neg_i)$$

Como veremos posteriormente, a eliminação da dupla negação de uma fórmula qualquer não pode ser provada na lógica proposicional minimal, mas a dupla eliminação de uma fórmula negada, sim:

Exercício 14. *Seja φ uma fórmula da lógica proposicional. Prove o sequente $\neg\neg\neg\varphi \vdash \neg\varphi$ na lógica proposicional minimal.*

Exercício 15. *Sejam φ e γ fórmulas da lógica proposicional. Construa uma prova para os sequentes $\neg(\varphi \vee \gamma) \vdash (\neg\varphi) \wedge (\neg\gamma)$ e $(\neg\varphi) \wedge (\neg\gamma) \vdash \neg(\varphi \vee \gamma)$ na lógica proposicional minimal.*

Exercício 16. *Sejam φ e γ fórmulas da lógica proposicional. Construa uma prova para o sequente $(\neg\varphi) \vee (\neg\gamma) \vdash \neg(\varphi \wedge \gamma)$ na lógica proposicional minimal.*

Exercício 17. *Sejam φ e γ fórmulas da lógica proposicional. Construa uma prova para o sequente $\neg\neg(\varphi \wedge \gamma) \vdash (\neg\neg\varphi) \wedge (\neg\neg\gamma)$ na lógica proposicional minimal.*

Exercício 18. *Sejam φ e γ fórmulas da lógica proposicional. Construa uma prova para o sequente $(\neg\neg\varphi) \wedge (\neg\neg\gamma) \vdash \neg\neg(\varphi \wedge \gamma)$ na lógica proposicional minimal.*

Exercício 19. *Sejam φ e γ fórmulas da lógica proposicional. Construa uma prova para o sequente $\neg\neg(\varphi \to \gamma) \vdash (\neg\neg\varphi) \to (\neg\neg\gamma)$ na lógica proposicional minimal.*

Exercício 20. *Seja φ uma fórmula da lógica proposicional. Prove o sequente $\vdash \neg\neg(\varphi \vee \neg\varphi)$ na lógica proposicional minimal.*

Os exercícios anteriores incluem diversos resultados importantes que podem ser provados na lógica proposicional minimal, e por isto, é importante que você resolva todos eles em papel e lápis e posteriormente no Coq para verificar se sua solução está correta.

Exercício 21. *Refaça os exercícios anteriores no Coq.*

Referências Bibliográficas

[1] Emilio Jesús Gallego Arias, Benoît Pin, and Pierre Jouvelot. jsCoq: Towards Hybrid Theorem Proving Interfaces. *Electronic Proceedings in Theoretical Computer Science*, 239:15–27, January 2017.

[2] Jeremy Avigad, Kevin Donnelly, David Gray, and Paul Raff. A formally verified proof of the prime number theorem. *ACM Transactions on Computational Logic*, 9(1):2–es, December 2007.

[3] Jeremy Avigad and John Harrison. Formally verified mathematics. *Communications of the ACM*, 57(4):66–75, April 2014.

[4] M. Ayala-Rincón and F. L. C. de Moura. *Applied Logic for Computer Scientists - Computational Deduction and Formal Proofs*. UTCS. Springer, 2017.

[5] G. Gonthier. A computer-checked proof of the Four Colour Theorem. Technical report, Microsoft Research Cambridge, 2008.

[6] T. Hales, M. Adams, G. Bauer, D. Tat Dang, J. Harrison, T. Le Hoang, C. Kaliszyk, V. Magron, S. McLaughlin, T. Tat Nguyen, T. Quang Nguyen, T. Nipkow, S. Obua, J. Pleso, J. Rute, A. Solovyev, A. Hoai Thi Ta, T. N. Tran, D. Thi Trieu, J. Urban, K. Khac Vu, and R. Zumkeller. A formal proof of the Kepler conjecture. *ArXiv e-prints*, January 2015.

[7] Cezary Kaliszyk. Web Interfaces for Proof Assistants. *Electronic Notes in Theoretical Computer Science*, 174(2):49–61, 2007.

[8] Cezary Kaliszyk, Stephan Schulz, Josef Urban, and Jiří Vyskočil. System Description: E.T. 0.1. In Amy P. Felty and Aart Middeldorp, editors, *Automated Deduction - CADE-25*, volume 9195, pages 389–398. Springer International Publishing, Cham, 2015.

[9] Xavier Leroy. Formal Verification of a Realistic Compiler. *Communications of the ACM*, 52(7):107, 2009.

[10] W. McCune. Prover9 and mace4. http://www.cs.unm.edu/~mccune/prover9/, 2005.

[11] Leonardo de Moura and Sebastian Ullrich. The Lean 4 Theorem Prover and Programming Language. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction – CADE 28*, Lecture Notes in Computer Science, pages 625–635, Cham, 2021. Springer International Publishing.

[12] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lncs*. Springer, 2002.

[13] R. B. Nogueira, A. C. A. Nascimento, F. L. C. de Moura, and M. Ayala-Rincón. Formalization of Security Proofs Using PVS in the Dolev-Yao Model. In *Booklet Proc. Computability in Europe - CiE*, 2010.

[14] S. Owre, J. M. Rushby, and N. Shankar. PVS: A Prototype Verification System. In D. Kapur, editor, *CADE*, volume 607 of *Lnai*, pages 748–752. sv, 1992.

[15] Lawrence C. Paulson. A Mechanised Proof of Gödel's Incompleteness Theorems Using Nominal Isabelle. *J Autom Reasoning*, 55(1):1–37, 2015.

[16] Benjamin C. Pierce, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Catvalin Hriatcu, Vilhelm Sjoberg, and Brent Yorgey. *Software Foundations*. Electronic textbook, 2014.

[17] Alexandre Riazanov and Andrei Voronkov. The design and implementation of VAMPIRE. *AI Commun.*, 15(2-3):91–110, 2002.

[18] Raymond Smullyan. *Logical Labyrinths*. AK Peters, 2009.

[19] Leon Sterling and Ehud Y Shapiro. *The Art of Prolog: Advanced Programming Techniques*. MIT press, 1994.

[20] The Coq Development Team. The Coq Proof Assistant. Zenodo, October 2021.