

Verificação Formal (2021/22)

Coq (1)

O objectivo desta aula é a familiarização com o sistema Coq, um sistema de gestão e desenvolvimento de prova formal.

Deverá ter instalado o `CoqIde`. Como alternativa, se for um utilizador do Emacs, poderá instalar o Coq e o modo `Proof-General` para o Emacs (e opcionalmente o `Company-Coq`). Poderá ainda correr o sistema Coq num browser com o `jsCoq`.

O website do Coq disponibiliza toda a documentação, assim como diversos livros e tutoriais que fornecem uma introdução rápida ao sistema Coq.

1 Aspectos básicos do desenvolvimento de provas em Coq

Vamos ilustrar os aspectos mais básicos do Coq como ferramenta de prova, tendo por base exemplos da lógica proposicional e da lógica de 1ª ordem que se demonstram com base num raciocínio deductivo.

Comece por invocar o sistema Coq e carregue o ficheiro `lesson1.v`.

Neste ficheiro pretendemos ir apresentado a sintaxe do Coq e ilustrar o funcionamento de vários comandos e táticas de prova através de vários exemplos comentados.

Execute, passo a passo, as instruções deste ficheiro e analize o seu efeito. Atente nos comentários lá colocados e no efeito da aplicação de cada tática de prova na evolução do estado da prova.

Recorde algumas das **táticas básicas** de prova:

Proposition (P)	Introduction	Elimination (H of type P)
\perp		<code>elim H</code> , contradiction
$\neg A$	<code>intro</code>	<code>apply H</code>
$A \wedge B$	<code>split</code>	<code>elim H</code> , <code>destruct H as [H1 H2]</code>
$A \Rightarrow B$	<code>intro</code>	<code>apply H</code>
$A \vee B$	<code>left</code> , <code>right</code>	<code>elim H</code> , <code>destruct H as [H1 H2]</code>
$\forall x:A. Q$	<code>intro</code>	<code>apply H</code>
$\exists x:A. Q$	<code>exists witness</code>	<code>elim H</code> , <code>destruct H as [x H1]</code>

`intro`, `intros` – regra de introdução para Π (várias vezes).

`apply` – regra de eliminação para Π .

`assumption` – regra da assunção (quando a conclusão aparece também como uma hipótese).

`exact` – apresenta diretamente um termo de prova (exacto) para a conclusão.

Recorde algumas **táticas automáticas** de prova:

`trivial` – tenta as táticas que podem resolver aprova num passo.

`auto` – tenta uma combinação de táticas `intro`, `apply` e `assumption` usando os teoremas armazenados numa base de dados como dicas para essa tática.

`tauto`, `intuition` – útil para provar tautologias da lógica proposicional intuicionista.

`firstorder` – útil para provar fatos que são tautologias da lógica de 1^a ordem intuicionista.

Complete as provas em falta, substituindo o comando `Admitted` e por uma script de prova apropriada.

Crie agora um novo ficheiro Coq para desenvolver as provas das propriedades que se seguem.

2 Lógica proposicional

Prove as seguintes tautologias da lógica proposicional:

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

3 Lógica de primeira ordem

Prove os seguintes teoremas da lógica de primeira ordem:

1. $(\exists x.P(x) \wedge Q(x)) \rightarrow (\exists x.P(x)) \wedge (\exists x.Q(x))$
2. $(\exists x.\forall y.P(x, y)) \rightarrow \forall y.\exists x.P(x, y)$
3. $(\exists x.P(x)) \rightarrow (\forall x.\forall y.P(x) \rightarrow Q(y)) \rightarrow \forall y.Q(y)$
4. $(\forall x.Q(x) \rightarrow R(x)) \rightarrow (\exists x.P(x) \wedge Q(x)) \rightarrow \exists x.P(x) \wedge R(x)$
5. $(\forall x.P(x) \rightarrow Q(x)) \rightarrow (\exists x.P(x)) \rightarrow \exists y.Q(y)$
6. $(\exists x.P(x)) \vee (\exists x.Q(x)) \leftrightarrow (\exists x.P(x) \vee Q(x))$

4 Lógica clássica

Assumindo o *princípio do meio excluído* como axioma, prove que:

1. $((A \rightarrow B) \rightarrow A) \rightarrow A$ (*lema de Pierce*)
2. $\neg\neg A \rightarrow A$
3. $\neg\forall x.P(x) \rightarrow \exists x.\neg P(x)$