



程序代写代做 CS编程辅导



UNSW
SYDNEY



MP4161

Advanced Topics in Software Verification

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

λ \rightarrow and HOL

Gerwin Klein, June Andronick, Miki Tanaka, Johannes Åman Pohjola
<https://tutorcs.com>

T3/2022

Last time...

程序代写代做 CS编程辅导

- Simply typed lambda calculus: $\lambda \rightarrow$
- Typing rules for $\lambda \rightarrow$ variables, type contexts
- β -reduction in $\lambda \rightarrow$ subject reduction
- β -reduction in $\lambda \rightarrow$ always terminates
- Types and terms in Isabelle

Website: [cstutorcs](http://cstutorcs.com)

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Content

程序代写代做 CS编程辅导

→ Foundations & Principles

- Intro, Lambda natural deduction [1,2]
- Higher Order (part 1) [2,3^a]
- Term rewriting [3,4]



→ Proof & Specification Techniques

- Inductively defined sets, rule induction [4,5]
- Datatype induction, primitive recursion [5,7]
- General recursive functions, termination proofs [7^b]
- Proof automation (part 2) [8]
- Hoare logic, proofs about programs, invariants [8,9]
- C verification [9,10]
- Practice, questions, exam prep [10^c]

WeChat: cstutorcs

Assignment Project Exam Help

Email: tut@163.com

QQ: 749389476

<https://tutorcs.com>

^aa1 due; ^ba2 due; ^ca3 due

程序代写代做 CS编程辅导



Preview: Proofs in Isabelle

WeChat: tutorescs
Assignment Project Exam Help

Email: tutores@163.com

QQ: 749389476

<https://tutores.com>

Proofs in Isabelle

程序代写代做 CS编程辅导

General schema:

```
lemma name: " <goal>  
  apply <method>  
  apply <method>  
  ...  
done
```



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

→ Sequential application of methods until
all **subgoals** are solved

QQ: 749389476

<https://tutorcs.com>

The Proof State

程序代写代做 CS编程辅导

1. $\bigwedge x_1 \dots x_p. \llbracket A \rrbracket \Rightarrow B$
2. $\bigwedge y_1 \dots y_q. \llbracket C \rrbracket \Rightarrow D$



$x_1 \dots x_p$ Parameters
 $A_1 \dots A_n$ Local assumptions
 B Actual (sub)goal

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Isabelle Theories

程序代写代做 CS编程辅导

Syntax:

```
theory MyTh
imports ImpTh1 .
begin
(declarations, definitions, theorems, proofs, ...)*
end
```



WeChat: [cstutorcs](#)

Assignment Project Exam Help

- *MyTh*: name of theory. Must live in file *MyTh.thy*
- *ImpTh*_{*i*}: name of *imported* theories. Import transitive.

Email: tutorcs@163.com

Unless you need something special:

```
theory MyTh imports Main begin ... end
```

QQ: [749389476](#)

<https://tutorcs.com>

Natural Deduction Rules

程序代写代做 CS编程辅导



$$\frac{A \quad B}{A \wedge B} \text{ conjI}$$

$$\frac{A \wedge B \quad \llbracket A; B \rrbracket \Rightarrow C}{C} \text{ conjE}$$

$$\frac{A}{A \vee B} \quad \frac{B}{A \vee B} \text{ disjI1/2}$$

$$\frac{A \vee B \quad A \Rightarrow C \quad B \Rightarrow C}{C} \text{ disjE}$$

$$\frac{A \Rightarrow B}{A \rightarrow B} \text{ implI}$$

Assignment Project Exam Help

$$\frac{A \rightarrow B \quad A \quad B \Rightarrow C}{C} \text{ impE}$$

Email: tutorcs@163.com

QQ: 749389476

For each connective (\wedge , \vee , etc):
<https://tutorcs.com>
introduction and elimination rules

Proof by assumption

程序代写代做 CS编程辅导



assumption

proves

1. $\llbracket B_1; \dots; B_m \rrbracket \implies$ WeChat: cstutorcs

by unifying C with one of the B_i Assignment Project Exam Help

Email: tutorcs@163.com

There may be more than one matching B_i and multiple unifiers.

QQ: 749389476

Backtracking!

<https://tutorcs.com>

Explicit backtracking command: **back**

Intro rules

程序代写代做 CS编程辅导

Intro rules decompose C to the right of \Rightarrow .

assume $\langle \text{intro-rule} \rangle$



Intro rule $\llbracket A_1; \dots; A_n \rrbracket \Rightarrow A$ means

→ To prove A it suffices to show $A_1 \dots A_n$

Assignment Project Exam Help

Applying rule $\llbracket A_1; \dots; A_n \rrbracket \Rightarrow A$ to subgoal C :

→ unify A and C

→ replace C with n new subgoals $A_1 \dots A_n$

QQ: 749389476

<https://tutores.com>

Intro rules: example

程序代写代做 CS编程辅导

To prove subgoal $A \Rightarrow B$ we can use: $\frac{P \Rightarrow Q}{P \rightarrow Q} \text{impl}$

(in Isabelle: $\text{impl} : (P \Rightarrow Q) \Rightarrow P \rightarrow Q$)



Recall:

Applying rule $\llbracket A_1; \dots; A_n \rrbracket \Rightarrow A$ to subgoal C :

- unify A and C
- replace C with n new subgoals $A_1 \dots A_n$

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

Here:

- unify... $P \rightarrow Q$ with $A \rightarrow A$
- replace subgoal... $A \rightarrow A$ (i.e. $\llbracket \cdot \rrbracket \Rightarrow A \rightarrow A$)
with $\llbracket A \rrbracket \Rightarrow A$ (which can be proved with: **apply** assumption)


QQ: 749389476

<https://tutorcs.com>

Elim rules

程序代写代做 CS编程辅导

Elim rules decompose \square on the left of \Rightarrow .

a  le $\langle \text{elim-rule} \rangle$)

Elim rule $\llbracket A_1; \dots; A_n \rrbracket \Rightarrow A$ means

→ If I know A_1 and want to prove A it suffices to show $A_2 \dots A_n$

Assignment Project Exam Help

Applying rule $\llbracket A_1; \dots; A_n \rrbracket \Rightarrow A$ to subgoal C :

Like **rule** but also

→ unifies first premise of rule with an assumption

→ eliminates that assumption

QQ: 749389476

<https://tutorcs.com>

Elim rules: example

程序代写代做 CS编程辅导

To prove $\llbracket B \wedge A \rrbracket \Longrightarrow A$ we can use:

$$\frac{P \wedge Q \quad \llbracket P; Q \rrbracket \Longrightarrow R}{R}$$



(in Isabelle: `conjE` : $\llbracket ?P; ?Q \rrbracket \Longrightarrow ?R \Longrightarrow ?R$)

Recall:

WeChat: cstutorcs

Applying rule $\llbracket A_1; \dots; A_n \rrbracket \Longrightarrow A$ to subgoal C:

Like **rule** but also [Assignment Project Exam Help](#)

- unifies first premise of rule with an assumption
- eliminates that assumption

Email: tutorcs@163.com

QQ: 749389476

Here:

- unify... $?R$ with A
- and also unify... $?P \wedge ?Q$ with assumption $B \wedge A$
- replace subgoal... $\llbracket B \wedge A \rrbracket \Longrightarrow A$
with $\llbracket B; A \rrbracket \Longrightarrow A$ (which can be proved with: **apply** assumption)

<https://tutorcs.com>

程序代写代做 CS编程辅导



Demo

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutores@163.com

QQ: 749389476

<https://tutores.com>

程序代写代做 CS编程辅导



More Proof Rules

WeChat: cstutores

Assignment Project Exam Help


Email: tutores@163.com

QQ: 749389476

<https://tutores.com>

Iff, Negation, True and False

程序代写代做 CS编程辅导

$$\frac{A \implies B \quad B \implies A}{A = B} \quad \text{iffD1} \quad \text{iffE}$$


$$\frac{A = B}{A \implies B} \text{ iffD1}$$

WeChat: cstutorcs

$$\frac{A = B}{B \implies A} \text{ iffD2}$$

Assignment Project Exam Help

$$\frac{A \implies \text{False}}{\neg A} \text{ notI}$$

Email: tutorcs@163.com

$$\frac{\neg A}{P} \text{ notE}$$

QQ: 749389476

$$\overline{\text{True}} \text{ TrueI}$$

<https://tutorcs.com>

$$\frac{\text{False}}{P} \text{ FalseE}$$

Equality

程序代写代做 CS编程辅导

$$\frac{}{t = t} \text{ refl} \quad \text{ym} \quad \frac{r = s \quad s = t}{r = t} \text{ trans}$$


WeChat: cstutorcs

$$\frac{s = t \quad P \quad s}{P \quad t} \text{ subst}$$

Assignment Project Exam Help


Email: tutorcs@163.com

Rarely needed explicitly → used implicitly by term rewriting

<https://tutorcs.com>

Classical

程序代写代做 CS编程辅导

$$\frac{P = \overline{\text{QR}} = \text{False}}{\text{excluded-middle}} \quad \text{True-or-False}$$


$$\frac{\neg A \Rightarrow \text{False}}{A} \text{ ccontr} \quad \frac{\neg A \Rightarrow A}{A} \text{ classical}$$

Assignment Project Exam Help

→ **excluded-middle**, **ccontr** and **classical**
not derivable from the other rules.

→ if we include True or False, they are derivable

They make the logic “classical”, “non-constructive”

Cases

程序代写代做 CS编程辅导



excluded-middle

is a case distinction on type *bool*

WeChat: cstutorcs

Assignment Project Exam Help

Isabelle can do case distinctions on arbitrary terms:

Email: tutorcs@163.com

apply (*case_tac term*)

QQ: 749389476

<https://tutorcs.com>

Safe and not so safe

程序代写代做 CS编程辅导

Safe rules preserve safety

conjI, iffI, refl, ccontr, classical, conjE,
disjE

$$\frac{A \quad B}{A \wedge B} \text{conjI}$$

Unsafe rules can turn a provable goal into an unprovable one

disjI1, disjI2, impE, iffD1, iffD2, notE

$$\frac{A}{A \vee B} \text{disjI1}$$

<https://tutorcs.com>

Apply safe rules before unsafe ones

程序代写代做 CS编程辅导



Demo

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutores@163.com

QQ: 749389476

<https://tutores.com>

What we have learned so far...

程序代写代做 CS编程辅导

- natural deduction \exists , \forall , \longrightarrow , \neg , iff...
- proof by assumption, intro rule, elim rule
- safe and unsafe reduction
- indent your proofs! (one space per subgoal)
- prefer implicit backtracking (chaining) or *rule_tac*, instead of *back*
- *prefer* and *defer*
- *oops* and *sorry*

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>