



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

{ P } . . . { Q }

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

QQ: 749389476

Gerwin Klein, June Andronick, Miki Tanaka, Johannes Åman Pohjola

<https://tutorcs.com>

T3/2022

# 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<sup>a</sup>]
- 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<sup>b</sup>]
- Proof automation (part 2) [8]
- Hoare logic, proofs about programs, invariants [8,9]
- C verification [9,10]
- Practice, questions, exam prep [10<sup>c</sup>]

WeChat: cstutorcs

Assignment Project Exam Help

Email: tut@163.com

QQ: 749389476

<https://tutorcs.com>

---

<sup>a</sup>a1 due; <sup>b</sup>a2 due; <sup>c</sup>a3 due

程序代写代做 CS编程辅导



# A Crash Course in Semantics

WeChat: estutores

Assignment Project Exam Help

Email: [tutores@163.com](mailto:tutores@163.com)

QQ: 749389476

<https://tutores.com>

程序代写代做 CS编程辅导



(For more,  
see Concrete Semantics)

WeChat: cstutores

Assignment Project Exam Help

Email: [tutores@163.com](mailto:tutores@163.com)

QQ: 749389476

<https://tutores.com>

# IMP - a small Imperative Language

程序代写代做 CS编程辅导

**Commands:**  
**datatype** com



WeChat: cstutors

KIP

assign vname aexp

emi com com

Cond bexp com com

While bexp com

( \_ := \_ )

( \_ ; \_ )

( IF \_ THEN \_ ELSE \_ )

( WHILE \_ DO \_ OD )

**type\_synonym** vname = string

**type\_synonym** state = vname  $\Rightarrow$  nat

**type\_synonym** aexp = state  $\Rightarrow$  nat

**type\_synonym** bexp = state  $\Rightarrow$  bool

Assignment Project Exam Help

Email: [tutorcs@163.com](mailto:tutorcs@163.com)


QQ: 749389476

<https://tutorcs.com>

## Example Program

程序代写代做 CS编程辅导

Usual syntax:



```
;  
= A  $\neq$  0 DO  
= B * A;  
= A - 1  
OD
```

WeChat: cstutorcs

Expressions are functions from state to bool or nat:

Assignment Project Exam Help  
Email: [tutorcs@163.com](mailto:tutorcs@163.com)  
QQ: 749389476  
<https://tutorcs.com>

```
B := ( $\lambda\sigma. 1$ );  
WHILE ( $\lambda\sigma. \sigma \ A \neq 0$ ) DO  
  B := ( $\lambda\sigma. \sigma \ B * \sigma \ A$ );  
  A := ( $\lambda\sigma. \sigma \ A - 1$ );  
OD
```

# What does it do?

程序代写代做 CS编程辅导

So far we have defined:

- **Syntax** of commands and expressions
- **State** of program (from variables to values)



**Now we need:** the meaning (semantics) of programs

WeChat: [cstutorcs](#)

**How to define execution of a program?**

Assignment Project Exam Help

- A wide field of its own
- Some choices:
  - Operational (inductive relations, big step, small step)
  - Denotational (programs as functions on states, state transformers)
  - Axiomatic (pre/post conditions, Hoare logic)

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

QQ: [749389476](#)

<https://tutorcs.com>

# Structural Operational Semantics

程序代写代做 CS编程辅导



$$\frac{\langle P, \sigma \rangle \rightarrow \sigma}{\langle x := e, \sigma \rangle \rightarrow \sigma[x \mapsto v]}$$

WeChat: cstutorcs

$$\frac{\langle c_1, \sigma \rangle \rightarrow \sigma' \quad \langle c_2, \sigma' \rangle \rightarrow \sigma''}{\langle c_1, c_2, \sigma \rangle \rightarrow \sigma''}$$

Assignment Project Exam Help

Email: tutorcs@163.com

$$\frac{b \ \sigma = \text{True} \quad \langle c_1, \sigma \rangle \rightarrow \sigma'}{\langle \text{IF } b \text{ THEN } c_1 \text{ ELSE } c_2, \sigma \rangle \rightarrow \sigma'}$$

QQ: 749389476

<https://tutorcs.com>

$$\frac{b \ \sigma = \text{False} \quad \langle c_2, \sigma \rangle \rightarrow \sigma'}{\langle \text{IF } b \text{ THEN } c_1 \text{ ELSE } c_2, \sigma \rangle \rightarrow \sigma'}$$



# Structural Operational Semantics

程序代写代做 CS编程辅导



$$\frac{\sigma = \text{False}}{\langle \text{V DO } c \text{ OD}, \sigma \rangle \rightarrow \sigma}$$

WeChat: cstutorcs

$$\frac{b \ \sigma = \text{True} \quad \langle c, \sigma \rangle \rightarrow \sigma' \quad \langle \text{WHILE } b \text{ DO } c \text{ OD}, \sigma' \rangle \rightarrow \sigma''}{\langle \text{WHILE } b \text{ DO } c \text{ OD}, \sigma \rangle \rightarrow \sigma''}$$

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



# Demo: The Definitions in Isabelle

WeChat: estutores

Assignment Project Exam Help

Email: [tutores@163.com](mailto:tutores@163.com)

QQ: 749389476

<https://tutores.com>

# Proofs about Programs

程序代写代做 CS编程辅导

Now we know:

- What programs a
- On what they wo
- How they work: S



So we can prove properties about programs

WeChat: cstutorcs

Example:

Assignment Project Exam Help

Show that example program from slide 6 implements the factorial.

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

**lemma**  $\langle \text{factorial}, \sigma \rangle \Rightarrow \sigma' B = \text{fac}(\sigma A)$   
(where  $\text{fac } 0 = 1$ ,  $\text{fac}(\text{Suc } n) = (\text{Suc } n) * \text{fac } n$ )

QQ: 749389476  
<https://tutorcs.com>

程序代写代做 CS编程辅导



Demo: WeChat: cstutores  
Example Proof  
Assignment Project Exam Help

Email: [tutores@163.com](mailto:tutores@163.com)

QQ: 749389476

<https://tutores.com>

Too tedious

程序代写代做 CS编程辅导



Induction needed for each loop

WeChat: cstutorcs

Is there something easier?  
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Floyd/Hoare

程序代写代做 CS编程辅导

**Idea:** describe meaning of program by pre/post conditions



**Examples:**

$\{\text{True}\} \quad x := 2 \quad \{x = 2\}$

$\{y = 2\} \quad x := 21 * y \quad \{x = 42\}$

$\{x = n\} \quad \text{IF } y < 0 \text{ THEN } x := x + y \text{ ELSE } x := x - y \quad \{x = n - |y|\}$

$\{A = n\} \quad \text{factorial} \quad \{B = \text{fac } n\}$

**Proofs:** have rules that directly work on such triples

<https://tutorcs.com>

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

QQ: 749389476

Assignment Project Exam Help

WeChat: tutorcs

# Meaning of a Hoare-Triple

程序代写代做 CS编程辅导



What are the assertions  $P$  and  $Q$ ?

- Here: again function  $c$  maps state to bool (shallow embedding of assertions)
- Other choice: syntax and semantics for assertions (deep embedding)

What does  $\{P\} c \{Q\}$  mean?

**Partial Correctness:**

$$\models \{P\} c \{Q\} \equiv \forall \sigma \sigma'. P \sigma \wedge \langle c, \sigma \rangle \rightarrow \sigma' \longrightarrow Q \sigma'$$

**Total Correctness:**

$$\models \{P\} c \{Q\} \equiv (\forall \sigma \sigma'. P \sigma \wedge \langle c, \sigma \rangle \rightarrow \sigma' \longrightarrow Q \sigma') \wedge (\forall \sigma. P \sigma \longrightarrow \exists \sigma'. \langle c, \sigma \rangle \rightarrow \sigma')$$

This lecture: partial correctness only (easier)

# Hoare Rules

程序代写代做 CS编程辅导

$$\frac{\{P\} \text{ SKIP } \{P[x \mapsto e]\} \quad x := e \quad \{P\}}{\{R\} \quad c_2 \quad \{Q\}} \quad \frac{\{R\} \quad c_2 \quad \{Q\}}{c_1; c_2 \quad \{Q\}}$$

$$\frac{\{P \wedge b\} \quad c_1 \quad \{Q\} \quad \{P \wedge \neg b\} \quad c_2 \quad \{Q\}}{\{P\} \text{ IF } b \text{ THEN } c_1 \text{ ELSE } c_2 \text{ FI } \{Q\}}$$

$$\frac{\{P \wedge b\} \quad c \quad \{P\} \quad P \wedge \neg b \implies Q}{\{P\} \text{ WHILE } b \text{ DO } c \text{ OD } \{Q\}}$$

$$\frac{P \implies P' \quad \{P'\} \quad c \quad \{Q'\} \quad Q' \implies Q}{\{P\} \quad c \quad \{Q\}}$$



# Hoare Rules

程序代写代做 CS编程辅导

$$\frac{}{\vdash \{P\} \text{ SKIP } \{P\}} \quad \frac{}{\vdash \{P\} \text{ } \sigma. P (\sigma(x := e \sigma)) \} \quad x := e \quad \{P\}}$$

$$\frac{\vdash \{R\} c_2 \{Q\}}{\vdash \{P\} c_1; c_2 \{Q\}}$$

$$\frac{\vdash \{\lambda\sigma. P \sigma \wedge b \sigma\} c_1 \{Q\} \quad \vdash \{\lambda\sigma. P \sigma \wedge \neg b \sigma\} c_2 \{Q\}}{\vdash \{P\} \text{ IF } b \text{ THEN } c_1 \text{ ELSE } c_2 \{Q\}}$$

$$\frac{\vdash \{\lambda\sigma. P \sigma \wedge b \sigma\} c \{P\} \quad \vdash \{\lambda\sigma. P \sigma \wedge \neg b \sigma \Rightarrow Q \sigma\}}{\vdash \{P\} \text{ WHILE } b \text{ DO } c \text{ OD } \{Q\}}$$

$$\frac{\vdash \{\lambda\sigma. P \sigma \Rightarrow P' \sigma\} c \{Q'\} \quad \vdash \{\lambda\sigma. Q' \sigma \Rightarrow Q \sigma\}}{\vdash \{P\} c \{Q\}}$$

## Are the Rules Correct?

程序代写代做 CS编程辅导

**Soundness:**  $\vdash \{P\} \langle \text{code} \rangle \models \{P\} \sqsubset \{Q\}$

**Proof:** by rule induction



WeChat: cstutorcs

Demo: Hoare Logic in Isabelle

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>