A Proof System for Communicating Sequential Processes

Outline

1. Introduction and preliminaries

2. The proof system

3. Case studies

4. Deadlock freedom

5. Conclusion and comparison with related work

1 . Introduction and preliminaries

Introduction

1. Main Work

This system deals with proofs of partial correctness and of deadlock freedom.

2. **CSP's properties**

- Simultaneity .
- Input and output commands (as a guard in guarded choices and repetitions).
- CSP focuses on terminating concurrent computations.

3. A (meta) rule to establish joint cooperation between isolated proofs for CSP's sequential components.

Preliminaries

1 The basic command of CSP is $[P_1 \| \cdots \| P_n]$ expressing *concurrent* execution of processes $P_1, \ldots, P_n$, $n \geq 2$.

2 Every P_i refers to a statement S_i, as indicated by $P_i :: S_i$. No S_i contains variables subject to change in S_j $(i \neq j)$.

$\boxed{\text{Var}(s_i) \cap \text{change}(s_j) = \varnothing \quad \text{(disjoint)}}$

3 **Communication:**

Input:

$P_i :: S_i \quad S_i = P_j\,?\,x$

Output:

$P_j :: S_j \quad S_j = P_i\,!\,v$

Preliminaries

4.

Guarded *selection*: $[B_1 \to S_1 \square B_2 \to S_2]$ $\boxed{\textit{fails}}$ for $B_1 \vee B_2 = \textbf{false}$, and leads to (possibly nondeterministic) selection of S_i for execution if $B_i = \textbf{true}$.

Guarded *iteration*: $*[B_1 \to S_1 \square B_2 \to S_2]$ $\boxed{\textit{terminates}}$ for $B_1 \vee B_2 = \textbf{false}$, and otherwise executes $[B_1 \to S_1 \square B_2 \to S_2]; *[B_1 \to S_1 \square B_2 \to S_2]$.

Example:

$[P_1 :: [\boxed{P_2?x} \to \text{skip} \square P_2!x \to \text{skip}] \| P_2 :: \text{skip}]$ **failure**

$[P_1 :: *[P_2?x \to \text{skip} \square P_2!x \to \text{skip}] \| P_2 :: \text{skip}]$ **terminate**

$[P_1 :: [\boxed{P_2?x} \to \text{skip} \square P_2!x \to \text{skip}] \| P_2 :: [P_1?y \to \text{skip} \square \boxed{P_1!y} \to \text{skip}]]$ **x := y or y := x**

$[P_1 :: *[P_2?x \to \text{skip}] \| P_2 :: *[P_1!0 \to \text{skip}]]$ **x := 0 *(infinite chattering)***

As an expression Pj?x (respectively, Pi!y) evaluates to false in case Pj (respectively, Pi) has terminated.

2. The Proof System

Chapter Structure

1.The axioms and proof rules for isolated processes

- Isolated CSP process ： {p} P_i {q}, where P_i is a process
- Axioms and proof rules：A1、A2、R1、R2、A3、A4、R3、R4、R5、R6、R7

2. Cooperating

- Concurrent :[P_1 ||… || P_n]
- Method:cooperation test
- Axioms and proof rules: meta rule、A5、A6、R8
- Example 1 → The above axioms and proof rules are used for a complete proof process.
- Example 2 → A2->A2’
- Example 3 → a problem : Semantically unmatched pairs of I/O instructions do not fail the cooperation test → global invariant I and bracket 〈〉
- After introducing global invariant I and square bracket < >, the axioms and proof rules are supplemented：R9、R10、R11、R12
- Example 4 → Solve the problem in Example 3

Axioms and proof rules

A1. *Input*

$$\{p\}\,P_i?x\boxed{\{q\}}.$$

A2. *Output*

$$\{p\}P_i!y\boxed{\{p\}.}$$

R1. *I/O Guarded Selection*

αᵢ stand for I/O commands

$$\frac{\{p \wedge b_i\}\alpha_i\boxed{\{r_i\},}\ \{r_i\}S_i\{q\},\ i = 1, \ldots, m}{\{p\}[\Box(i = 1, \ldots, m)\ b_i;\ \alpha_i \rightarrow S_i]\{q\}}.$$

R2. *I/O Guarded Repetition*

$$\frac{\{p \wedge b_i\}\alpha_i\{r_i\},\ \{r_i\}S_i\{p\},\ i = 1, \ldots, m}{\{p\}\,{*}[\Box(i = 1, \ldots, m)\ b_i;\ \alpha_i \rightarrow S_i]\{p\}}.$$

Axioms and proof rules

A3. *Assignment*

$$\{p[t/x]\}x := t\{p\}.$$

A4. *Skip*

$$\{p\}\text{skip}\{p\}.$$

R3. *Alternative Command*

$$\frac{\{p \wedge b_i\}S_i\{q\},\ i = 1, \ldots, m}{\{p\}\,[\square(i = 1, \ldots, m)\ b_i \rightarrow S_i]\{q\}}.$$

R4. *Repetitive Command*

$$\frac{\{p \wedge b_i\}S_i\{p\},\ i = 1, \ldots, m}{\{p\}\,{*}[\square(i = 1, \ldots, m)\ b_i \rightarrow S_i]\{p \wedge \boxed{\neg(b_1 \vee \cdots \vee b_m)}\}}.$$

Axioms and proof rules

R5. *Composition*

$$\frac{\{p\}S_1\{q\},\ \{q\}S_2\{r\}}{\{p\}S_1;\ S_2\,\{r\}}.$$

R6. *Consequence*

$$\frac{p \rightarrow p_1,\ \{p_1\}S\{q_1\},\ q_1 \rightarrow q}{\{p\}S\{q\}}.$$

R7. *Conjunction*

$$\frac{\{p\}S\{q\},\ \{p\}S\{r\}}{\{p\}S\{q \wedge r\}}.$$

Meta rule and cooperate

$$\frac{\text{proofs of } \{p_i\}P_i\{q_i\},\ i = 1, \ldots, n, \text{ cooperate}}{\{p_1 \wedge \cdots \wedge p_n\}[P_1\|\cdots\|P_n]\{q_1 \wedge \cdots \wedge q_n\}}.$$

Intuitively, proofs cooperate if they help each other to validate the postassertions of the I/O statements mentioned in those proofs.

Example:

P_1:: P_2? x P_2:: P_1! 2

Proof outline: {true} P_2?x {x=2}, {true} P_1!2 {true}

{true ∧ true} P2?x || P1!2 {x=2 ∧ true}

Meta rule and cooperate

$$\frac{\text{proofs of } \{p_i\}P_i\{q_i\},\ i = 1, \ldots, n, \text{ cooperate}}{\{p_1 \wedge \cdots \wedge p_n\}[P_1 \| \cdots \| P_n]\{q_1 \wedge \cdots \wedge q_n\}}.$$

this property is expressed as follows: The proofs of $\{p_i\}P_i\{q_i\}$, $i = 1, \ldots, n$, cooperate if

(i) the assertions used in the proof of $\{p_i\}P_i\{q_i\}$ contain no variables subject to change in P_j for $i \neq j$;

(ii) $\boxed{\{\text{pre}_1 \wedge \text{pre}_2\}P_j?x \| P_i!y\ \{\text{post}_1 \wedge \text{post}_2\}}$ holds whenever $\boxed{\{\text{pre}_1\}P_j?x\{\text{post}_1\}}$ and $\boxed{\{\text{pre}_2\}P_i!y\{\text{post}_2\}}$ are taken from the proofs of $\{p_i\}P_i\{q_i\}$ and $\{p_j\}P_j\{q_j\}$, respectively.[1]

Cooperation Rule

A5. *Communication*

$$\{\mathbf{true}\}P_i?x \parallel P_j!y\{x = y\}$$

provided $P_i?x$ and $P_j!y$ are taken from P_j and P_i, respectively.

A6. *Preservation*

$$\{p\}S\{p\}$$

$\forall x \in A$ or $\exists x \in A$

provided no free variable of p is subject to change in S.

R8. *Substitution*

$$\frac{\{p\}S\{q\}}{\{p[t/z]\}S\{q\}}$$

provided z does not appear free in S and q.

Example 1

Example 1. Using the system above we can prove

$$\{\mathbf{true}\}[P_1 \parallel P_2 \parallel P_3]\{x = u\},$$

where $P_1 :: P_2!x$, $P_2 :: P_1?y; P_3!y$, and $P_3 :: P_2?u$.

1. Proof outlines :

$$\{x = z\}P_2!x\{x = z\},$$
$$\{\mathbf{true}\}P_1?y\{y = z\}; P_3!y\{y = z\},$$
$$\{\mathbf{true}\}P_2?u\{u = z\}.$$

A1. *Input*

$$\{p\}P_i?x\{q\}.$$

A2. *Output*

$$\{p\}P_i!y\{p\}.$$

2. Verify whether the matched I/O pairs in the proof outline are cooperated.

{x=z} P2!x || P1?y {x=z ∧ y=z}

{y=z} P3!y || P2?u {y= z ∧ u=z}

3. Meta rule

Example 1

Example 1. Using the system above we can prove

$$\{\mathbf{true}\}[P_1 \parallel P_2 \parallel P_3]\{x = u\},$$

where $P_1 :: P_2!x$, $P_2 :: P_1?y; P_3!y$, and $P_3 :: P_2?u$.

{x=z} P2!x || P1?y {x=z ∧ y=z}

Prove：

1. {true} P2!x ||P1?y {x=y} (A5)
2. x=z → true (R6) => {x=z} P2!x ||P1?y {x=y}
3. {x=z} P2!x ||P1?y {x=z} (A6)
4. {x=z} P2!x ||P1?y {x=y ∧ x=z} (R7)
5. x=y ∧ x=z → x=z ∧ y=z (R6) => {x=z} P2!x || P1?y {x=z ∧ y=z}
6. {x=z} P2!x || P1?y {x=z ∧ y=z} holds. {y=z} P3!y || P2?u {y= z ∧ u=z} holds.

A5. *Communication*

$$\{\mathbf{true}\}P_i?x \parallel P_j!y\{x = y\}$$

provided $P_i?x$ and $P_j!y$ are taken from P_j and P_i, respectively.

A6. *Preservation*

$$\{p\}S\{p\}$$

provided no free variable of p is subject to change in S.

R6. *Consequence*

$$\frac{p \rightarrow p_1,\ \{p_1\}S\{q_1\},\ q_1 \rightarrow q}{\{p\}S\{q\}}.$$

R7. *Conjunction*

$$\frac{\{p\}S\{q\},\ \{p\}S\{r\}}{\{p\}S\{q \wedge r\}}.$$

Example 1

Example 1. Using the system above we can prove

$$\{\mathbf{true}\}[P_1 \parallel P_2 \parallel P_3]\{x = u\},$$

where $P_1 :: P_2!x$, $P_2 :: P_1?y; P_3!y$, and $P_3 :: P_2?u$.

Meta rule:

$$\frac{\text{proofs of } \{p_i\}P_i\{q_i\},\ i = 1, \ldots, n, \text{ cooperate}}{\{p_1 \wedge \cdots \wedge p_n\}[P_1 \parallel \cdots \parallel P_n]\{q_1 \wedge \cdots \wedge q_n\}}$$

Proof outline:

$\boxed{\{x = z\}}P_2!x\boxed{\{x = z\}}$,
$\boxed{\{\mathbf{true}\}}P_1?y\{y = z\}; P_3!y\boxed{\{y = z\}}$,
$\boxed{\{\mathbf{true}\}}P_2?u\boxed{\{u = z\}}$.

R6. *Consequence*

$$\frac{p \to p_1,\ \{p_1\}S\{q_1\},\ q_1 \to q}{\{p\}S\{q\}}.$$

R8. *Substitution*

$$\frac{\{p\}S\{q\}}{\{p[t/z]\}S\{q\}}$$

provided z does not appear free in S and q.

7. {x=z} [P1 || P2 || P3] {x=z $\wedge\ y = z\ \wedge u = z$}

8. x=z $\wedge$ y=z $\wedge$ u=z $\Rightarrow$ x=u **(R6)** $\Rightarrow$ {x=z} [P1 || P2 || P3] {x=u}

9. {true} [P1 || P2 || P3] {x=u}**(*R8*)**

Example 2

A1. *Input*

$$\{p\} P_i?x\{q\}.$$

A2. *Output*

$$\{p\} P_i!y\{p\}.$$

Example 2. Let

$$P_1 :: P_2!0,$$
$$P_2 :: [P_1?x \rightarrow \text{skip} \;\square\; P_3!y \rightarrow \text{skip} \;\square\; P_3?y \rightarrow \text{skip}],$$
$$P_3 :: \text{skip}.$$

Clearly, {true}[P1 || P2 || P3]{x = 0} holds. However, this cannot be proved in the above system.

Reasons： { true } [P_1 || P_2 || P_3] { x = 0 } holds .

requires { true } P_3!y { x = 0 } {???how to get?} and

{ true } P_3?y { x = 0 } {by using A1}.

By using A2， we can get { true } P_3!y { true }, which does not imply { true } P_3!y { x = 0 }.

A2′. *Output*

$$\{p\} P_i!y\{q\}.$$

Syntactic match and Semantic match

Example 3. Let

$$P_1 :: [\boxed{P_2?x} \rightarrow \text{skip} \,\square\, P_2!0 \rightarrow P_2?x;\ x := x + 1],$$
$$P_2 :: [P_1!2 \rightarrow \text{skip} \,\square\, P_1?z \rightarrow \boxed{P_1!1}].$$

Syntactic match:

The $P_2?X$ and $P_1!1$ matches syntactically.

Semantic match:

This communication will take place.

So we find that $P_2?X$ and $P_1!1$ are syntactic matching.

But they are not semantic matching.

Example 3

A5. *Communication*

$$\{\mathbf{true}\}P_i?x \parallel P_j!y\{x = y\}$$

provided $P_i?x$ and $P_j!y$ are taken from P_j and P_i, respectively.

Example 3. Let

$$P_1 :: [\boxed{P_2?x} \to \text{skip} \,\square\, P_2!0 \to P_2?x;\ x := x + 1],$$
$$P_2 :: [P_1!2 \to \text{skip} \,\square\, P_1?z \to \boxed{P_1!1}].$$

Clearly, { true } [P_1 || P_2] { x = 2 } holds. However, this cannot be proved in the above system.

Reasons：

The $P_2?X$ and $P_1!1$ are **syntactic matching but not semantic matching.**

{ true } [P_1 || P_2] { x = 2 } holds

- **requires** {true} $P_2?X$ {x = 2} and {z = 0} $P_1!1$ {true} can pass **cooperation test** .
- **requires** {z = 0} $P_2?X$ | | $P_1!1$ {x = 2} is true.
- But we obtain { true } $P_2?x$ || $P_1!1$ { x = 1 } (by using **A5**).
- We find that $x = 1 \nRightarrow x = 2$. ({z = 0} $P_2?X$ | | $P_1!1$ {x = 2} is false)
- $P_2?X$ and $P_1!1$ fail the cooperation test.

Conclusion:

This cannot be proved in the above system. So we should guarantee that semantically unmatched pairs of I/O instructions do not fail the cooperation test.

Global Invariant I and <>

In order to take care that semantically unmatched pairs of ***I/O*** instructions do not fail the cooperation test as above, we introduce a global invariant ***I*** .

$$\cdots P_2?x;\ i := i + 1 \cdots \| \cdots P_1!y;\ j := j + 1 \cdots$$

I: i=j is not a global invariant .

Definition. A process P_i is *bracketed* if the brackets "$\langle$" and "$\rangle$" are interspersed in its text, so that for each program section $\langle S \rangle$ (to be called a *bracketed section*), S is of one of the following forms:

$$S_1;\ \alpha;\ S_2 \qquad \text{or} \qquad \alpha \rightarrow S_1,$$

and S_1 and S_2 do not contain any I/O statements. □

Regarding the program sections just considered, the bracketing is

$$\cdots \langle P_2?x;\ i := i + 1 \rangle \cdots \| \cdots \langle P_1!y;\ j := j + 1 \rangle \cdots,$$

so that $i = j$ holds outside the brackets.

New Meta Rule

R9. *Parallel Composition*

$$\frac{\text{proofs of } \{p_i\}P_i\{q_i\},\ i = 1, \ldots, n, \text{ cooperate}}{\{p_1 \wedge \cdots \wedge p_n \wedge \boxed{I}\}[P_1 \| \cdots \| P_n]\{q_1 \wedge \cdots \wedge q_n \wedge \boxed{I}\}}$$

provided no variable free in I is subject to change outside a bracketed section.

Definition. Let $\langle S_1 \rangle$ and $\langle S_2 \rangle$ denote two bracketed sections from P_i and P_j $(i \neq j)$. We say that $\langle S_1 \rangle$ and $\langle S_2 \rangle$ *match* if S_1 and S_2 contain matching I/O commands. □

Definition. The proofs of the $\{p_i\}P_i\{q_i\}$, $i = 1, \ldots, n$, *cooperate* if

(i) the assertions used in the proof of $\{p_i\}P_i\{q_i\}$ have no free variables subject to change in P_j $(i \neq j)$;

(ii) $\{\text{pre}(S_1) \wedge \text{pre}(S_2) \wedge I\}\ S_1 \| S_2\{\text{post}(S_1) \wedge \text{post}(S_2) \wedge I\}$ holds for all matching pairs of bracketed sections $\langle S_1 \rangle$ and $\langle S_2 \rangle$. □

Proof rules concerning parallel composition

R10. *Formation*

$$\frac{\{p\}S_1;\, S_3\{p_1\},\ \{p_1\}\alpha \parallel \bar{\alpha}\{p_2\},\ \{p_2\}S_2;\, S_4\{q\}}{\{p\}(S_1;\, \alpha;\, S_2) \parallel (S_3;\, \bar{\alpha};\, S_4)\{q\}}$$

provided α and $\bar{\alpha}$ match and S_1, S_2, S_3, and S_4 do not contain any I/O commands.

R11. *Arrow*

$$\frac{\{p\}(\alpha;\, S) \parallel S_1\{q\}}{\{p\}(\alpha \rightarrow S) \parallel S_1\{q\}}.$$

R12. *Auxiliary Variables.* Let AV be a set of variables such that $x \in \mathrm{AV} \Rightarrow x$ appears in S' only in assignments $y := t$, where $y \in \mathrm{AV}$. Then if q does not contain free any variables from AV, and S is obtained from S' by deleting all assignments to variables in AV,

$$\frac{\{p\}S'\{q\}}{\{p\}S\{q\}}.$$

Definition 7.5. Let A be a set of simple variables in a program S. We call A a *set of auxiliary variables* of S if each variable from A occurs in S only in assignments of the form $z := t$ with $z \in A$. □

Example 4

I/O pair cannot pass the cooperation test.

Example 4. We now show how to verify the program from Example 3. Two auxiliary variables i and j are needed. We give proof outlines for the already bracketed program S'.

$$P_1 :: [\boxed{P_2?x} \rightarrow \text{skip} \,\square\, P_2!0 \rightarrow P_2?x;\ x := x+1],$$
$$P_2 :: [P_1!2 \rightarrow \text{skip} \,\square\, P_1?z \rightarrow \boxed{P_1!1}].$$

$$I \equiv (i = j)$$

$\{i = 0 \wedge j = 0\}$
$[\{i = 0\}$
$[\langle \boxed{P_2?x}\{x = 2\} \rightarrow i := 1\rangle\{x = 2 \wedge i = 1\};\ \textbf{skip}\{x = 2\}$
$\square$
$\quad \langle P_2!0\{\textbf{true}\} \rightarrow i := 1\rangle\{i = 1\};$
$\quad\quad \langle P_2?x\ \{x = 1\};\ i := 2\rangle\{x = 1 \wedge i = 2\};\ x := x + 1\ \{x = 2\}$
$]\{x = 2\}$
$\|$
$[\{j = 0\}$
$\quad \langle P_1!2\{\textbf{true}\} \rightarrow j := 1\rangle\{j = 1\}\ \textbf{skip}\{\textbf{true}\}$
$\square$
$\quad \langle P_1?z\{z = 0\} \rightarrow j := 1\rangle\{z = 0 \wedge j = 1\};$
$\quad\quad \langle \boxed{P_1!1}\{\textbf{true}\};\ j := 2\rangle\{j = 2\}$
$]\{\textbf{true}\}$
$]$
$\{x = 2\}$

- $p = \{ p_1 \wedge \ldots \wedge p_n \wedge I \}$
  $= \{ (i = 0) \wedge (z = 0 \wedge j = 1) \wedge (i = j) \}$
  $=$ false (by using Theorem 1)
- We obtain { false } S { q } is true in partial correctness.
- We gain $\{ i = 0 \}\ \langle P_2?x \rightarrow I := 1 \rangle\ \{ x = 2 \wedge i = 1 \}$ and $\{ z = 0 \wedge j = 1 \}\ \langle P_1!1 ;\ j := 2 \rangle\ \{ j = 2 \}$ pass the cooperation test trivially.

Brief Summary

- Isolated proof rules.
- The concept of cooperation is introduced and meta rule is given .
- The meta rule is supplemented by the introduction of global invariant I and square brackets < >. (Semantically unmatched pairs of I/O instructions do not fail the cooperation test)

Thank you!