Owicki-Gries:

S. Owicki and D. Gries:
An Axiomatic Proof Technique for Parallel Programs.
Acta Informatica 6, 319-340 (1976).

Goal: Reason about parallel programs with shared variables.

Problem: Execution of thread 1 may change shared variables, and hence influence execution of thread 2.

Idea: (1) Limit influences by atomic(-)

(2) Ensure that influences do not invalidate a proof

$\Rightarrow$ Focus on proofs instead of fine-grained executions.

Approach:
- Prove each thread in isolation
- Show that commands of partner thread do not interfere with the proof
- Conclude correctness of the parallel program.

Technically: Extend Hoare's proof system by a rule for non-interference.

Recall:

The rules of Hoare's proof system are

$$\text{(SKIP)}\ \frac{}{\{P\}\ \mathtt{skip}\ \{P\}} \qquad \frac{}{\{P[E/x]\}\ x := E\ \{P\}}\ \text{(ASSIGN)}$$

$$\text{(ASSUME)}\ \frac{}{\{P\}\ \mathtt{assume}(B)\ \{P \wedge B\}} \qquad \frac{\{P\}C_1\{Q\},\ \{Q\}C_2\{R\}}{\{P\}C_1;C_2\{R\}}\ \text{(SEQ)}$$

$$\text{(CHOICE)}\ \frac{\{P\}C_1\{Q\},\ \{P\}C_2\{Q\}}{\{P\}C_1 \oplus C_2\{Q\}} \qquad \frac{\{P\}C\{P\}}{\{P\}C^*\{P\}}\ \text{(LOOP)}$$

$$\text{(CONSEQ)}\ \frac{P' \Rightarrow P,\ \{P\}C\{Q\},\ Q \Rightarrow Q'}{\{P'\}C\{Q'\}}$$

By HOARE we refer to these 7 rules.

Interference Freedom

(also proof sketch, proof outline in the literature)

- We assume threads are annotated by a full proof*, which means commands are interleaved with assertions

  $\{P_1\}\ C_1\ \{P_2\}\ C_2 \ldots$

  Intuitively, P_i holds when the execution reaches C_i.

- In the presence of other threads, this intuition breaks:

  $\{x=0\}$ $x := x+2;$ $\{x=2\}$ and $\{x=0\}$ $x := 0;$ $\{true\}$ hold

  but $\{x=0\}$ $x := x+2; \parallel x := 0;$ $\{x=2\}$ no longer holds.

- The proofs do not take into account the interferences of the other thread.

<u>Definition:</u>

- An assertion P is <u>invariant under command T</u> with precondition $pre(T)$, if

  $\{P \wedge pre(T)\}\ T\ \{P\}$.

- Let C be a thread with full proof $\{P\}C\{Q\}$.
  Let T be a command from another thread with precondition $pre(T)$ in the corresponding full proof.
  Then <u>T does not interfere with proof $\{P\}C\{Q\}$</u>, if

  $\{R \wedge pre(T)\}\ T\ \{R\}$ holds

  for all assertions R in $\{P\}C\{Q\}$ that are outside atomic blocks.

- Two full proofs $\{P_1\} C_1 \{Q_1\}$ and $\{P_2\} C_2 \{Q_2\}$ are <u>interference-free</u>, if
  every command/atomic block of the second thread does not interfere with the first proof
  and vice versa.

New proof rules

$$\text{(OG-INF)}\ \frac{\{P_1\} C_1 \{Q_1\} \quad \{P_2\} C_2 \{Q_2\} \quad \text{the proofs are interference-free}}{\{P_1 \wedge P_2\}\ C_1 \| C_2\ \{Q_1 \wedge Q_2\}} \qquad \frac{\{P\} C \{Q\}}{\{P\}\ \text{atomic}(C)\ \{Q\}}\ \text{(OG-ATOMIC)}$$

<u>Theorem:</u>
(OG-INF) and (OG-ATOMIC) are sound.

<u>Observation:</u>
- Assume the length of thread C_i is $l_i \in \mathbb{N}$.
  Then there are $2 \cdot l_1 \cdot l_2$ interference-freedom proofs to do.
- Most of these proofs are trivial as
  the command or atomic block
  uses variables disjoint from the ones in the assertion.

<u>Example:</u>
- Consider program $x := x+2 \parallel x := 0$.
- The full proofs

$$\begin{array}{lll} \{x=0\} & \quad\text{and}\quad & \{true\} \\ \quad x := x+2 & & \quad x := 0 \\ \{x=2\} & & \{x=0\} \end{array}$$

are correct, but <u>not interference-free.</u>
↳ As an example, consider
assertion $\{x=0\}$ in thread 2.
Then
$\{x=0 \wedge x=0\}\ x := x+2\ \{x=0\}$ does not hold.

↳ Similarly
$$\{x=2 \wedge true\}\ x:=0\ \{x=2\}$$ does not hold.

- We weaken the postconditions and consider the full proofs

$\{x=0\}$ $\quad x:=x+2$ $\quad \{x=0 \vee x=2\}$ and $\{true\}$ $\quad x:=0$ $\quad \{x=0 \vee x=2\}$.

These proofs <u>are</u> <u>interference-free</u>.

↳ For example does $x:=x+2$ not interfere with assertion $\{x=0 \vee x=2\}$ in the second proof, since
$$\{(x=0 \vee x=2) \wedge x=0\}\ \ x:=x+2\ \ \{x=0 \vee x=2\}$$
holds.

↳ There are 4 such interference-freedom proofs to check.

- With rule (OG-INF), we conclude
$$\{x=0\}\ x:=x+2 \parallel x:=0\ \{x=0 \vee x=2\}.$$
□

Auxiliary Variables

Rule (OG-INF) alone is too weak to prove some programs.

<u>Lemma (Incompleteness)</u>
$$\{true\}\ x:=x+2 \parallel x:=0\ \{x=0 \vee x:=2\}$$
cannot be derived in the proof system HOARE + (OG-INF) (+ (OG-ATOMIC)).

<u>Proof:</u>
Towards a contradiction, assume the triple could be proven in HOARE + (OG-INF).

- Then there are full proofs

$\{P_1\}\ x := x+2\ \{Q_1\}$

$\{P_2\}\ x := 0\ \{Q_2\}$

that are interference-free and that satisfy

$$true \Rightarrow P_1 \wedge P_2 \qquad (1)$$

$$Q_1 \wedge Q_2 \Rightarrow x=0 \vee x=2 \qquad (2)$$

- From (1) we conclude

$P_1 \Leftrightarrow true$ and $P_2 \Leftrightarrow true$.

Hence, by (CONSEQ)

$\{true\}\ x := x+2\ \{Q_1\}$.

This means

$Q_1[x+2/x]$

is valid (holds for all $x \in \mathbb{Z}$).

Hence, also

$$Q_1 \qquad (3)$$

is valid.

- We similarly derive

$\{true\}\ x := 0\ \{Q_2\}$

from which we conclude validity of

$$Q_2[0/x]. \qquad (4)$$

- Using interference freedom, we conclude for $T =\ x := x+2$ with $pre(T) = P_1 \Leftrightarrow true$ that

$\{Q_2 \wedge true\}\ x := x+2\ \{Q_2\}$ holds.

This means

$Q_2 \rightarrow Q_2[x+2/x]$ is valid. (5)

- Using induction, (4) and (5) show validity of

$$\forall x : (x \geq 0 \wedge even(x) \Rightarrow Q_2) \qquad (6)$$

- Since Q_1 is valid by (3)
  and $Q_1 \wedge Q_2$ implies $x = 0 \vee x = 2$ by (2),
  (6) yields validity of

$$\forall x : (x \geq 0 \wedge even(x) \Rightarrow x = 0 \vee x = 2). \quad ↯ \qquad \square$$

- The problem is that we cannot conclude from x
  whether $x := x + 2$ has been executed.
- Trick: Introduce auxiliary variables.

Definition:

Consider a program C and a set of variables A.
Then A is called a set of auxiliary variables for C,
if every $x \in A$ only occurs in assignments $z := t$ with $z \in A$.

This means auxiliary variables

↳ cannot influence the control flow of C,
because they do not occur in conditions (assert).

↳ cannot influence the data flow of C,
because they cannot occur in assignments to variables outside A.

Example:

Consider program

$z := x; (x := x + 1 \parallel y := y + 1).$

Then the following are sets of auxiliary variables:

$\emptyset, \{y\}, \{z\}, \{x, z\}, \{y, z\}, \{x, y, z\}.$

Note that $\{x\}$ is not a set of auxiliary variables
as

$z := x.$

We add a proof rule for introducing/eliminating auxiliary variables

$$\text{(OG-AUX)}\ \frac{\{P\}\,C\,\{Q\} \qquad \begin{array}{l} A \text{ a set of auxiliary variables for } C \\ \text{with } A \cap (fv(P) \cup fv(Q)) = \emptyset \end{array}}{\{P\}\ \mathrm{remove}(C, A)\ \{Q\}}$$

Here, remove(C, A) is a program obtained from C by replacing all assignments $x := t$ with $x \in A$ by skip.

Theorem:

(OG-AUX) is sound.

Example:

- Show

$$\{true\}\ x := x+2 \parallel x := 0\ \{x = 0 \lor x = 2\}$$

using auxiliary variable done.
Variable done indicates whether $x := x+2$ has been executed:
  ↳ initially false
  ↳ set to true atomically with $x := x+2$

- Show

$$\begin{array}{l} \{true\} \\ \quad done := false; \\ \quad \left(\begin{array}{l} atomic(\\ \quad x := x+2; \\ \quad done := true; \\ ) \end{array} \middle\| \ x := 0 \right) \\ \{x = 0 \lor x = 2\} \end{array} \tag{7}$$

With (7), we conclude

$$\{true\}\ \underbrace{skip;\ (atomic(x := x+2;\ skip) \parallel x := 0)}_{\approx\ x := x+2 \parallel x := 0}\ \{x = 0 \lor x = 2\}$$

with the observation that {done} is a set of auxiliary variables and (OG-AUX).

- For the proof (7), note that

$\{\neg done\}$ atomic($x := x+2$; $done :=$ true;) $\{true\}$ and $\{true\}$ $x := 0$ $\{(x=0 \vee x=2) \wedge (\neg done \Rightarrow x=0)\}$ hold.

These are full proofs (atomic sections do not need assertions) that are interference-free.
To show the latter, we check 4 conditions.
For example

$\{(x=0 \vee x=2) \wedge (\neg done \Rightarrow x=0) \wedge \neg done\}$
$\{x=0\}$
atomic(
  $x := x+2$;
  $done :=$ true;
)
$\{x=2 \wedge done\}$
$\{(x=0 \vee x=2) \wedge (\neg done \Rightarrow x=0)\}$.

- For the initialization, we have

$\{true\}$ $done :=$ false $\{\neg done\}$.

- We conclude with (SEQ). □