SHARED-VARIABLE CONCURRENCY

proof outls.
/
outlines

~~a~~ proof outlines ~~are~~ interference free if none of their assignm. actions interfere with critical assertions in other outlines

NONINTERFERENCE or INTERFERENCE FREE :

- an assignment action is an assignm. or an await statement containing one or more assign. statems
- a critical assertion : a pre- or post-condition not contained within an ~~assignm~~ await statement
- noninterference : let **a** be an assignment action in one process, with **pre(a)** its precondition
  let **c** be a critical assertion in another process, possibly renaming its local vars s.t. $var(a) \cap localvars(other\ pr.) = \emptyset$

THEN **a** does not interfere with **c** if

$$\models \{ c \wedge pre(a) \} \; a \; \{ c \}$$

~~2.11.a~~
2.13.a

Await Statement Rule:

$$\frac{\{P \wedge B\}\ S\ \{Q\}}{\{P\}\ \langle \texttt{await}\ (B)\ S;\rangle\ \{Q\}}$$

Co Statement Rule:

$$\frac{\{P_i\}\ S_i\ \{Q_i\}\ \text{are interference free}}{\begin{array}{l}\{P_1 \wedge \ldots \wedge P_n\} \\ \texttt{co}\ S_1;\ //\ \ldots\ //\ S_n;\ \texttt{oc} \\ \{Q_1 \wedge \ldots \wedge Q_n\}\end{array}}$$ proof outlines

Inference Rules for Await and Co Statements

2.ма

Observe that, by applying the rules for assignment,

$$\{x = 0 \vee x = 2\} x := x + 1 \{x = 1 \vee x = 3\}$$

and

$$\{x = 0 \vee x = 1\} x := x + 2 \{x = 2 \vee x = 3\}$$

are proof outlines. To prove interference freedom of these, we have to prove the verification conditions generated by the following four assignments:

- $\{(x = 0 \vee x = 2) \wedge (x = 0 \vee x = 1\}\ x := x + 1\ \{x = 0 \vee x = 1\}$,
- $\{(x = 0 \vee x = 2) \wedge (x = 2 \vee x = 3\}\ x := x + 1\ \{x = 2 \vee x = 3\}$,
- $\{(x = 0 \vee x = 1) \wedge (x = 0 \vee x = 2\}\ x := x + 2\ \{x = 0 \vee x = 2\}$, and
- $\{(x = 0 \vee x = 1) \wedge (x = 1 \vee x = 3\}\ x := x + 2\ \{x = 1 \vee x = 3\}$.

These formulae follow from the guarded-assignment rule. By applying the parallel composition rule we obtain the following proof outline:

$$\begin{array}{l} \quad \{(x = 0 \vee x = 2) \wedge (x = 0 \vee x = 1)\} \\ [\ \ \{x = 0 \vee x = 2\}\ x := x + 1\ \{x = 1 \vee x = 3\} \\ \|\ \{x = 0 \vee x = 1\}\ x := x + 2\ \{x = 2 \vee x = 3\} \\ ]\ \{(x = 1 \vee x = 3) \wedge (x = 2 \vee x = 3)\}. \end{array}$$

Since

$$\models x = 0 \rightarrow (x = 0 \vee x = 2) \wedge (x = 0 \vee x = 1)$$

and

$$\models (x = 1 \vee x = 3) \wedge (x = 2 \vee x = 3) \rightarrow x = 3,$$

we can extend this to a proof outline of the form:

$$\begin{array}{l} \{x = 0\} \\ \{(x = 0 \vee x = 2) \wedge (x = 0 \vee x = 1)\} \\ [\cdots \| \cdots] \\ \{(x = 1 \vee x = 3) \wedge (x = 2 \vee x = 3)\} \\ \{x = 3\}. \end{array}$$

By applying the consequence rule, we can transform this into the desired Hoare formula: $\vdash \{x = 0\}\ \langle\ [x := x + 1\ \|\ x := x + 2]\ \rangle\ \{x = 3\}$. □

EX.:

WEAKENED ASSERTIONS

$p_1 \wedge p_2 \; \{x=0\} \; \underline{co} \; \{x=0 \vee x=2\} - p_1$

$\langle x:=x+1 \rangle_1$

$\{x=1 \vee x=3\} - q_1$

$\|$

$\{x=0 \vee x=1\} - p_2$

$\langle x:=x+2 \rangle_2$

$\{x=2 \vee x=3\} - q_2$

$q_1 \wedge q_2 \quad \underline{oc} \; \{x=3\}$

check IFFreed.!

DISJOINT VARS:

$\{x=0 \wedge y=0\} \; \underline{co} \; \{x=0\} \; x:=x+1 \; \{x=1\}$

$\| \; \{y=0\} \; y:=y+1 \; \{y=1\}$

$\underline{oc} \; \{x=1 \wedge y=1\}$

Example 10.15 As the next example consider $\langle\, [x := x+1 \parallel x := x+1]\, \rangle$. The aim is to prove $\{x = 0\}\langle\, [x := x+1 \parallel x := x+1]\, \rangle\{x = 2\}$. Analogous to the previous example, we first have to try using the proof outlines

$$\{x = 0 \lor x = 1\} x := x+1 \{x = 1 \lor x = 2\}$$

and

$$\{x = 0 \lor x = 1\} x := x+1 \{x = 1 \lor x = 2\}.$$

These proof outlines, however, are not interference free. For instance, $\{(x = 0 \lor x = 1) \land (x = 0 \lor x = 1)\}\ x := x+1\ \{x = 0 \lor x = 1\}$ is not valid. A second problem is that the conjunction of the postassertions $(x = 1 \lor x = 2) \land (x = 1 \lor x = 2)$ does not imply the desired postassertion $x = 2$. As proved in Example 3.12 within the context of the inductive assertion method, it is even impossible to prove $\{x = 0\} x := x+1 \parallel x := x+1 \{x = 2\}$ by making use of assertions whose only free variable is x. This proof carries over to the present framework.

226

Definition 10.13 (Auxiliary variables)
Consider a program S_0. Let $A \subseteq var(S_0)$, where $var(S_0)$ denotes the set of variables that occur (within assignments and boolean tests) in S_0. We call A a *set of auxiliary variables of* S_0 if the following conditions are satisfied:

- Each variable from A occurs in S_0 only within assignments, that is, it may *not* occur within the boolean tests b of guarded assignments and guarded commands.
- When it occurs in an assignment $x_1, \ldots, x_n := e_1, \ldots, e_n$ it does so only within its components (x_i, e_i) when $x_i \in A$. In words: a variable from A cannot be used in assignments to variables outside A. □

Next we present a version of the auxiliary-variables rule. Note that the premise of the rule has the form of a proof outline, whereas its conclusion is a Hoare formula.

A PROOF OUTLINE FOR S_0

Rule 10.6 (Auxiliary variables)

$$\frac{\{p\}\ A(S_0)\ \{q\}}{\{p\}\ \langle S\rangle\ \{q\}},$$

where, for some set of auxiliary variables A of S_0 with $A \cap var(q) = \emptyset$, program S results from S_0 by deleting all assignments to the variables in A, and, in case this results in **skip** statements, dropping the latter.

2.9

a solution to this problem is the use of *auxiliary variables*. In our example we can use, for instance, two auxiliary variables $done1$ and $done2$, which record whether the assignment has been performed in, respectively, the first or second process.

Now consider the following proof outlines:

$$\begin{array}{l}\{\neg done1 \wedge (\neg done2 \rightarrow x = 0) \wedge (done2 \rightarrow x = 1)\}\\ x, done1 := x + 1, true\\ \{done1 \wedge (\neg done2 \rightarrow x = 1) \wedge (done2 \rightarrow x = 2)\}\end{array}$$

and

$$\begin{array}{l}\{\neg done2 \wedge (\neg done1 \rightarrow x = 0) \wedge (done1 \rightarrow x = 1)\}\\ x, done2 := x + 1, true\\ \{done2 \wedge (\neg done1 \rightarrow x = 1) \wedge (done1 \rightarrow x = 2)\}.\end{array}$$

These proof outlines are interference free. For instance,

$$\begin{array}{l}\{\neg done1 \wedge (\neg done2 \rightarrow x = 0) \wedge (done2 \rightarrow x = 1) \wedge\\ \neg done2 \wedge (\neg done1 \rightarrow x = 0) \wedge (done1 \rightarrow x = 1)\}\\ x, done1 := x + 1, true\\ \{\neg done2 \wedge (\neg done1 \rightarrow x = 0) \wedge (done1 \rightarrow x = 1)\}\end{array}$$

is valid, since its precondition is equivalent to $\neg done1 \wedge \neg done2 \wedge x = 0$.

Con-
sequently, we can apply the parallel composition rule. We also introduce an initialisation of the auxiliary variables, and obtain the proof outline below, where we have used the following abbreviations:

$$
\begin{aligned}
p_1 &\stackrel{\text{def}}{=} \neg done1 \land (\neg done2 \to x = 0) \land (done2 \to x = 1) \\
p_2 &\stackrel{\text{def}}{=} \neg done2 \land (\neg done1 \to x = 0) \land (done1 \to x = 1) \\
q_1 &\stackrel{\text{def}}{=} done1 \land (\neg done2 \to x = 1) \land (done2 \to x = 2) \\
q_2 &\stackrel{\text{def}}{=} done2 \land (\neg done1 \to x = 1) \land (done1 \to x = 2).
\end{aligned}
$$

The proof outline is given by

$$
\begin{array}{l}
\{x = 0\} \\
done1, done2 := \mathit{false}, \mathit{false}\,; \\
\{p_1 \land p_2\} \\
[\ \ \{p_1\}\ x, done1 := x + 1, \mathit{true}\ \{q_1\} \\
\|\ \ \{p_2\}\ x, done2 := x + 1, \mathit{true}\ \{q_2\} \\
]\ \{q_1 \land q_2\} \\
\{x = 2\}.
\end{array}
$$

By the auxiliary variables rule we obtain

$$\vdash \{x = 0\}\langle [x := x + 1 \ \|\ x := x + 1] \rangle \{x = 2\}. \qquad \square$$