Let $a$ and $b$ be binary expressions of the prestate (preconditions), and let $A$, $B$, $C$, $P$, $Q$, $R$, $S$, $T$, and $U$ be implementable specifications such that the refinements

\[
A \iff \text{if } a \text{ then } \text{ok} \text{ else if } b \text{ then } P \text{ else } Q \text{ fi fi}
\]

\[
B \iff \text{if } a \text{ then } \text{ok} \text{ else if } b \text{ then } R \text{ else } S \text{ fi fi}
\]

\[
C \iff \text{if } a \text{ then } \text{ok} \text{ else if } b \text{ then } T \text{ else } U \text{ fi fi}
\]

are all theorems. Then $A$ can be executed as follows (using $\%$ for labeling):

\[
A% \iff \text{if } a \text{ then go to } D \text{ else if } b \text{ then } P \text{ go to } B \text{ else } Q \text{ go to } C \text{ fi fi}
\]

\[
B% \iff \text{if } a \text{ then go to } D \text{ else if } b \text{ then } R \text{ go to } C \text{ else } S \text{ go to } A \text{ fi fi}
\]

\[
C% \iff \text{if } a \text{ then go to } D \text{ else if } b \text{ then } T \text{ go to } A \text{ else } U \text{ go to } B \text{ fi fi}
\]

\[
D% \iff \text{ok}
\]

We have replaced refinement and call with labeling and go tos.

(a) Show that it is not possible to replace refinement and call (in this example) with while loops without introducing any new variables.

(b) Show that it is possible to replace refinement and call (in this example) with while loops if you introduce new variables.