1 4F-1 Bookkeeping

- **0 pts** Correct

4F-2. VCGen for let.

It has a bug because of the ignorance of the original value of of x in the let rule. Assume there is a fresh variable called tmp storing the original value of x. Then, if we use the tmp variable to express. VC(let $x=e$ in C, B), we have.

$$
\begin{aligned}
&VC(\text{let } x=e \text{ in } C, B)\\
&= VC(tmp=x;\ x=e;\ C;\ x=tmp;\ B)\\
&= VC(tmp=x;\ VC(x=e;C;\ x=tmp;\ B))\\
&= [x/tmp]\, VC(x=e;\ C;\ x=tmp,\ B)\\
&= [x/tmp]\, VC(x=e,\ VC(C;x=tmp;B))\\
&= [x/tmp]\,[e/x]\, VC(C;x=tmp,B)\\
&= [x/tmp]\,[e/x]\, VC(C, VC(x=tmp, B))\\
&= [x/tmp]\,[e/x]\, VC(C, [tmp/x]B)
\end{aligned}
$$

Therefore, the correct VCGen for let should be

$$[x/tmp]\,[e/x]\, VC(C, [tmp/x]B)$$

2 4F-2 VCGen for Let

- **0 pts** Correct

4F-3 VCGen Mistakes.

We could show the buggy of let rules by doing the following.

1. Set the command c as Let x=j in skip, where j is an arbitrary number.
2. Set the precondition B as x=j.
3. Set a state σ as $\sigma(x)=i$, where i is an arbitrary integer not equal to j.
4. $\sigma \vDash VC(c, B)$ because $VC(c, B)=j$.
5. $\langle c, \sigma\rangle \Downarrow \sigma'$, but $\sigma'(x)=i$ because we need to reassign back x's original value.
6. Therefore, $\sigma, \nvDash B$ because $\sigma'(x)=i$ but the post-condition B is x=j.

We could also provide another detailed example.

Let x=y x=x+1.

$\sigma(x)=i$, $\sigma(y)=1$.

B: x==2

$VC(c,B) = VC(x=y+1, x=2) = \{y=1\}$.

$\sigma \vDash VC(c,B)$

while $\langle \sigma, c\rangle \Downarrow \sigma'$ $\sigma'[x]=1$ $\sigma'[y]=1$ $\sigma' \nvDash B$.

Because it doesn't recover x with the original value after the let rule, it is problematic.

3 4F-3 VCGen Mistakes

- **0 pts** Correct

4F-4.

Because c is executed at least once, before b is tested as in the 4F-4, we have

do c while b = c; while b do c (And we could fold into the $\{B\}$.)

Then, we have

$$\frac{\vdash \{A\}\, c\, \{B\} \quad \vdash \{B \wedge b\}\, c\, \{B\}}{\vdash \{B\}\ \text{do } c \text{ while } b\ \{B \wedge \neg b\}}$$

A is arbitrary precondition, and B represents loop invariant.

~~Hoare p and q and~~

~~Hoare~~

4 4F-4 Axiomatic Do-While

- **0 pts** Correct