ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D. Basin and P. Müller

Formal Methods and Functional Programming

Solutions of Exercise Sheet 9: IMP States and Expressions

Assignment 1 (Simplifying State Updates)

(i) **Proof:** We need to show that, $\forall y.\sigma[x\mapsto v_1][x\mapsto v_2](y) = \sigma[x\mapsto v_2](y)$. For arbitrary y, we have (using the definition of state update):

$$\begin{array}{rcl}\sigma[x\mapsto v_1][x\mapsto v_2](y) & = & \begin{cases} v_2 & \textit{if } y = x \\ \sigma[x\mapsto v_1](y) & \textit{otherwise} \end{cases} \\ & = & \begin{cases} v_2 & \textit{if } y = x \\ \sigma(y) & \textit{otherwise} \end{cases} \\ & = & \sigma[x\mapsto v_2](y)\end{array}$$

(ii) **Proof:** We need to show that, if $x \neq y$, then $\forall z.\sigma[x\mapsto v_1][y\mapsto v_2](z) = \sigma[y\mapsto v_2][x\mapsto v_1](z)$. For arbitrary z, we have (using the definition of state update):

$$\begin{array}{rcl}\sigma[x\mapsto v_1][y\mapsto v_2](z) & = & \begin{cases} v_2 & \textit{if } z = y \\ \sigma[x\mapsto v_1](z) & \textit{otherwise} \end{cases} \\ & = & \begin{cases} v_2 & \textit{if } z = y \\ v_1 & \textit{if } z \neq y \textit{ and } z = x \\ \sigma(z) & \textit{otherwise} \end{cases} \\ & =^* & \begin{cases} v_1 & \textit{if } z = x \\ v_2 & \textit{if } z \neq x \textit{ and } z = y \\ \sigma(z) & \textit{otherwise} \end{cases} \\ & = & \begin{cases} v_1 & \textit{if } z = x \\ \sigma[y\mapsto v_2](z) & \textit{otherwise} \end{cases} \\ & = & \sigma[y\mapsto v_2][x\mapsto v_1](z)\end{array}$$

$*$Note that the rewriting of the cases in the function definition only works because we assumed $x \neq y$ - this condition is necessary for each of the sets of three cases to be disjoint, and for the two variants to define the same function (and indeed, the overall result isn't true without this condition).

(iii) **Proof:** Let x be an arbitrary variable, and let v_1, v_2 be arbitrary values.
Let $P(m)$ be the statement: "For all sequences $\vec{y}$ and $\vec{v'}$ of length m, and for all states σ', $\sigma'[x\mapsto v_1][\vec{y}\mapsto\vec{v'}][x\mapsto v_2] = \sigma'[\vec{y}\mapsto\vec{v'}][x\mapsto v_2]$"

We prove $\forall m.P(m)$ by induction on m.
Induction base (m = 0):
To show: $P(0)$
Then the sequences $\vec{y}$ and $\vec{v'}$ can only be empty. Let σ' be an arbitrary state. We are left with showing that $\sigma[x{\mapsto}v_1][x{\mapsto}v_2] = \sigma[x{\mapsto}v_2]$. This follows from part (i) of the question.

Induction step:
To show: $P(k+1)$, for some k, i.e., $\forall \vec{y}, \vec{v'}$ of length $k+1$, $\forall \sigma'$. $\sigma'[x{\mapsto}v_1][\vec{y}{\mapsto}\vec{v'}][x{\mapsto}v_2] = \sigma'[\vec{y}{\mapsto}\vec{v'}][x{\mapsto}v_2]$
IH: $P(k)$, i.e., $\forall \vec{y}, \vec{v'}$ of length k, $\forall \sigma'$. $\sigma'[x{\mapsto}v_1][\vec{y}{\mapsto}\vec{v'}][x{\mapsto}v_2] = \sigma'[\vec{y}{\mapsto}\vec{v'}][x{\mapsto}v_2]$

To show $P(k+1)$, let $\vec{y}$ and $\vec{v'}$ be arbitrary sequences of length $k+1$, and let σ be an arbitrary state (we use a different name from the quantified σ', to make the argument clearer later in the proof). Then, we need to show $\sigma[x{\mapsto}v_1][\vec{y}{\mapsto}\vec{v'}][x{\mapsto}v_2] = \sigma[\vec{y}{\mapsto}\vec{v'}][x{\mapsto}v_2]$, and our induction hypothesis tells us that the corresponding property holds for any sequences of length k, and for all states σ'.

Consider the first variable y_1 from the sequence $\vec{y}$. We consider two further (sub-)cases:

$(y_1 = x)$**:** Then, by part 1 of the question, $\sigma[x{\mapsto}v_1][y_1{\mapsto}v'_1] = \sigma[y_1{\mapsto}v'_1]$, and so we have $\sigma[x{\mapsto}v_1][\vec{y}{\mapsto}\vec{v'}][x{\mapsto}v_2] = \sigma[\vec{y}{\mapsto}\vec{v'}][x{\mapsto}v_2]$ as required.

$(y_1 \neq x)$**:** Then, by part 2 of the question, $\sigma[x{\mapsto}v_1][y_1{\mapsto}v'_1] = \sigma[y_1{\mapsto}v'_1][x{\mapsto}v_1]$. Using this fact, what we need to show is equivalent to showing:
$(\sigma[y_1{\mapsto}v'_1])[x{\mapsto}v_1][y_2{\mapsto}v'_2]\ldots[y_{k+1}{\mapsto}v'_{k+1}][x{\mapsto}v_2] =$
$(\sigma[y_1{\mapsto}v'_1])[y_2{\mapsto}v'_2]\ldots[y_{k+1}{\mapsto}v'_{k+1}][x{\mapsto}v_2]$. Since the sequences $y_2, \ldots, y_{k+1}$ and $v'_2, \ldots, v'_{k+1}$ are of length k, this follows from our induction hypothesis, taking the sequences there to be $y_2, \ldots, y_{k+1}$ and $v'_2, \ldots, v'_{k+1}$, and taking σ' to be $(\sigma[y_1{\mapsto}v'_1])$.

Assignment 2 (Substitution Properties)

Recall that $b[x \mapsto e]$ is defined as follows:

$$b[y \mapsto e] = \begin{cases} e_1[x \mapsto e]\ op\ e_2[x \mapsto e] & \text{if } b \text{ is the arithmetic comparison } e_1\ op\ e_2, \\ \texttt{not}\ b'[x \mapsto e] & \text{if } b \text{ is the Boolean expression } \texttt{not}\ b', \text{ and} \\ b_1[x \mapsto e] \oplus b_1[x \mapsto e] & \text{if } b \text{ is the Boolean expression } b_1 \oplus b_2 \\ & \quad \text{with } \oplus \in \{\texttt{and}, \texttt{or}\}. \end{cases}$$

The proof uses the previously-proven substitution lemma for arithmetic expressions (see question); we write "Lemma" below where it is applied.

We need to prove: $\forall b, e, x, \sigma.\ \mathcal{B}[\![b[x \mapsto e]]\!]\sigma = \mathcal{B}[\![b]\!](\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma])$

To do this, let e, x and σ be arbitrary (note that we deal with inner quantifiers first here, but several for-all quantifiers can always be reordered).

Now let $P(b) \equiv \mathcal{B}[\![b[x \mapsto e]]\!]\sigma = \mathcal{B}[\![b]\!](\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma])$
We prove $\forall b.P(b)$ by structural induction on b.

Case $b \equiv e_1\ op\ e_2$:
To Show: $P(e_1\ op\ e_2)$, i.e., that: $\mathcal{B}[\![(e_1\ op\ e_2)[x \mapsto e]]\!]\sigma = \mathcal{B}[\![(e_1\ op\ e_2)]\!](\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma])$

We can show this as follows:

$$\begin{array}{lcll}
\mathcal{B}[\![(e_1\ op\ e_2)[x \mapsto e]]\!]\sigma & = & \mathcal{B}[\![e_1[x \mapsto e]\ op\ e_2[x \mapsto e]]\!]\sigma & (\textit{defn. } [x \mapsto e]) \\
& = & \mathcal{A}[\![e_1[x \mapsto e]]\!]\sigma\ \overline{op}\ \mathcal{A}[\![e_2[x \mapsto e]]\!]\sigma & (\textit{defn. } \mathcal{B}) \\
& = & \mathcal{A}[\![e_1]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]\big)\ \overline{op}\ \mathcal{A}[\![e_2]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]\big) & (\textit{using Lemma}) \\
& = & \mathcal{B}[\![e_1\ op\ e_2]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]\big) & (\textit{defn. } \mathcal{B})
\end{array}$$

Case $b \equiv \texttt{not}\ b'$:

To Show: $P(\texttt{not}\ b')$, i.e., that: $\mathcal{B}[\![(\texttt{not}\ b')[x \mapsto e]]\!]\sigma = \mathcal{B}[\![(\texttt{not}\ b')]\!](\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma])$

IH: $P(b')$, i.e., that: $\mathcal{B}[\![b'[x \mapsto e]]\!]\sigma = \mathcal{B}[\![b']\!](\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma])$

We can show this as follows:

$$\begin{array}{lcll}
\mathcal{B}[\![(\texttt{not}\ b')[x \mapsto e]]\!]\sigma & = & \mathcal{B}[\![\texttt{not}\ b'[x \mapsto e]]\!]\sigma & (\textit{defn. } [x \mapsto e]) \\
& = & \neg\ \mathcal{B}[\![b'[x \mapsto e]]\!]\sigma & (\textit{defn. } \mathcal{B}) \\
& = & \neg\ \mathcal{B}[\![b']\!]\big(\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]\big) & (\textit{IH}) \\
& = & \mathcal{B}[\![\texttt{not}\ b']\!]\big(\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]\big) & (\textit{defn. } \mathcal{B})
\end{array}$$

Case $b \equiv b_1 \oplus b_2$:

To Show: $P(b_1 \oplus b_2)$, with $\oplus \in \{\texttt{and}, \texttt{or}\}$, i.e., that:
$\mathcal{B}[\![(b_1 \oplus b_2)[x \mapsto e]]\!]\sigma = \mathcal{B}[\![(b_1 \oplus b_2)]\!](\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma])$

IH: $P(b_1), P(b_2)$, i.e., that: $\mathcal{B}[\![b_1[x \mapsto e]]\!]\sigma = \mathcal{B}[\![b_1]\!](\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma])$, and
$\mathcal{B}[\![b_2[x \mapsto e]]\!]\sigma = \mathcal{B}[\![b_2]\!](\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma])$

We can show this as follows:

$$\begin{array}{lcll}
\mathcal{B}[\![(b_1 \oplus b_2)[x \mapsto e]]\!]\sigma & = & \mathcal{B}[\![(b_1[x \mapsto e]\ \oplus\ b_2[x \mapsto e])]\!]\sigma & (\textit{defn. } [x \mapsto e]) \\
& = & \mathcal{B}[\![b_1[x \mapsto e]]\!]\sigma\ \overline{\oplus}\ \mathcal{B}[\![b_2[x \mapsto e]]\!]\sigma & (\textit{defn. } \mathcal{B}) \\
& = & \mathcal{B}[\![b_1]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]\big)\ \overline{\oplus}\ \mathcal{B}[\![b_2]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]\big) & (\textit{IH, twice}) \\
& = & \mathcal{B}[\![b_1 \oplus b_2]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]\big) & (\textit{defn. } \mathcal{B})
\end{array}$$

Here, $\overline{\oplus}$ denotes the corresponding Boolean operation.

Assignment 3 (Similar States)

We want to prove $\forall \sigma, \sigma', e.((\forall x.x \in FV(e) \Rightarrow \sigma(x) = \sigma'(x)) \Rightarrow \mathcal{A}[\![e]\!]\sigma = \mathcal{A}[\![e]\!]\sigma')$
Let σ and σ' be arbitrary states.
Let $P(e)\ \equiv\ (\forall x.x \in FV(e) \Rightarrow \sigma(x) = \sigma'(x)) \Rightarrow \mathcal{A}[\![e]\!]\sigma = \mathcal{A}[\![e]\!]\sigma'$.
We prove $\forall e.P(e)$ by structural induction on e.

Case $e \equiv n$ (for some numeral n):

To Show: $P(n)$, i.e., that: $(\forall x.x \in FV(n) \Rightarrow \sigma(x) = \sigma'(x)) \Rightarrow \mathcal{A}[\![n]\!]\sigma = \mathcal{A}[\![n]\!]\sigma'$
Therefore, we assume that $(\forall x.\ x \in FV(n) \Rightarrow \sigma(x) = \sigma'(x))$, and seek to prove that $\mathcal{A}[\![n]\!]\sigma = \mathcal{A}[\![n]\!]\sigma'$. We can show the conclusion directly; using the definition of $\mathcal{A}$, we get $\mathcal{A}[\![n]\!]\sigma = \mathcal{N}[\![n]\!] = \mathcal{A}[\![n]\!]\sigma'$.

Case $e \equiv y$ (for some variable y):

To Show: $P(y)$, i.e., that: $(\forall x.x \in FV(y) \Rightarrow \sigma(x) = \sigma'(x)) \Rightarrow \mathcal{A}[\![y]\!]\sigma = \mathcal{A}[\![y]\!]\sigma'$
Therefore, we assume that $(\forall x.\ x \in FV(y) \Rightarrow \sigma(x) = \sigma'(x))$, and seek to prove that $\mathcal{A}[\![y]\!]\sigma = \mathcal{A}[\![y]\!]\sigma'$.
Note that $FV(y) = \{y\}$. Therefore, from our assumption, taking x to be y, we deduce that $\sigma(y) = \sigma'(y)$. Using this fact, and the definition of $\mathcal{A}$, we have $\mathcal{A}[\![y]\!]\sigma = \sigma(y) = \sigma'(y) = \mathcal{A}[\![y]\!]\sigma'$ as required.

Case $e \equiv (e_1\ \mathit{op}\ e_2)$:

To Show: $P(e_1\ \mathit{op}\ e_2)$, i.e., that:
$(\forall x.x \in FV(e_1\ \mathit{op}\ e_2) \Rightarrow \sigma(x) = \sigma'(x)) \Rightarrow \mathcal{A}[\![e_1\ \mathit{op}\ e_2]\!]\sigma = \mathcal{A}[\![e_1\ \mathit{op}\ e_2]\!]\sigma'$
IH: $P(e_1), P(e_2)$, i.e., that:
$(\forall x.x \in FV(e_1) \Rightarrow \sigma(x) = \sigma'(x)) \Rightarrow \mathcal{A}[\![e_1]\!]\sigma = \mathcal{A}[\![e_1]\!]\sigma'$, and
$(\forall x.x \in FV(e_2) \Rightarrow \sigma(x) = \sigma'(x)) \Rightarrow \mathcal{A}[\![e_2]\!]\sigma = \mathcal{A}[\![e_2]\!]\sigma'$.

We assume $(\forall x.x \in FV(e_1\ \mathit{op}\ e_2) \Rightarrow \sigma(x) = \sigma'(x))$, and seek to prove $\mathcal{A}[\![e_1\ \mathit{op}\ e_2]\!]\sigma = \mathcal{A}[\![e_1\ \mathit{op}\ e_2]\!]\sigma'$.
By definition of $FV()$, we have $FV(e_1\ \mathit{op}\ e_2) = FV(e_1) \cup FV(e_2) \supseteq FV(e_1)$. Therefore, from our assumption, we can deduce that $(\forall x.x \in FV(e_1) \Rightarrow \sigma(x) = \sigma'(x))$ holds. Using this along with our IH, we obtain that $\mathcal{A}[\![e_1]\!]\sigma = \mathcal{A}[\![e_1]\!]\sigma'$. Symmetrically to the above argument, we can derive from our assumption and IH that $\mathcal{A}[\![e_2]\!]\sigma = \mathcal{A}[\![e_2]\!]\sigma'$. Using these two facts, we can obtain our desired conclusion as follows:

$$\begin{array}{rcll} \mathcal{A}[\![e_1\ \mathit{op}\ e_2]\!]\sigma & = & \mathcal{A}[\![e_1]\!]\sigma\ \overline{\mathit{op}}\ \mathcal{A}[\![e_2]\!]\sigma & (\textit{defn. } \mathcal{A}) \\ & = & \mathcal{A}[\![e_1]\!]\sigma'\ \overline{\mathit{op}}\ \mathcal{A}[\![e_2]\!]\sigma' & (\textit{above facts}) \\ & = & \mathcal{A}[\![e_1\ \mathit{op}\ e_2]\!]\sigma' & (\textit{defn. } \mathcal{A}) \end{array}$$

Assignment 4 (Implementing IMP Expressions)

```
data Aexp = Bin Op Aexp Aexp
          | Var String
          | Num Integer

data Op = Add | Sub | Mul

data Bexp = Or Bexp Bexp
          | And Bexp Bexp
          | Not Bexp
          | Rel Rop Aexp Aexp

data Rop =  Eq | Neq | Le
         | Leq | Ge | Geq

data State = VarAssign (String -> Integer)

evalAexp :: Aexp -> State -> Integer
evalAexp (Num n)     _               = n
```

```
evalAexp (Var x)     (VarAssign val) = val x
evalAexp (Bin Add e1 e2) sigma = (evalAexp e1 sigma) + (evalAexp e2 sigma)
evalAexp (Bin Sub e1 e2) sigma = (evalAexp e1 sigma) - (evalAexp e2 sigma)
evalAexp (Bin Mul e1 e2) sigma = (evalAexp e1 sigma) * (evalAexp e2 sigma)

evalBexp :: Bexp -> State -> Bool
evalBexp (Rel op e1 e2) sigma =
    (evalOp op) (evalAexp e1 sigma) (evalAexp e2 sigma)
  where evalOp Eq  = (==)
        evalOp Neq = (/=)
        evalOp Le  = (<)
        evalOp Leq = (<=)
        evalOp Ge  = (>)
        evalOp Geq = (>=)
evalBexp (Not b)     sigma = not (evalBexp b sigma)
evalBexp (Or b1 b2)  sigma = (evalBexp b1 sigma) || (evalBexp b2 sigma)
evalBexp (And b1 b2) sigma = (evalBexp b1 sigma) && (evalBexp b2 sigma)
```

Assignment 5 - Headache of the week

(i) Suppose that $y \in FV(e_1)$ were allowed (but that $x \neq y$ is still true). Then, by choosing e to be x, and e_1 to be y, and e_3 to be 1, we get a counter-example by observing that: $x[x \mapsto y][y \mapsto 1] = 1 \neq y = x[y \mapsto 1][x \mapsto y]$.

(ii) Let x, y, e_1, e_2 be arbitrary. We need to prove that:

$$(x \neq y \wedge y \notin FV(e_1) \wedge x \notin FV(e_2) \Rightarrow \forall e.\ e[x \mapsto e_1][y \mapsto e_2] = e[y \mapsto e_2][x \mapsto e_1])$$

To do this, we assume that $x \neq y$ and $y \notin FV(e_1)$ and $x \notin FV(e_2)$ all hold, and seek to prove: $\forall e.\ e[x \mapsto e_1][y \mapsto e_2] = e[y \mapsto e_2][x \mapsto e_1]$
Let $P(e) \equiv e[x \mapsto e_1][y \mapsto e_2] = e[y \mapsto e_2][x \mapsto e_1]$. We prove $\forall e.P(e)$ by structural induction on e.

Case $e \equiv n$:

To Show: $P(n)$, i.e., that: $n[x \mapsto e_1][y \mapsto e_2] = n[y \mapsto e_2][x \mapsto e_1]$
By definition of substitution, $n[x \mapsto e_1][y \mapsto e_2] = n[y \mapsto e_2] = n$ and, similarly, $n[y \mapsto e_2][x \mapsto e_1] = n$.

Case $e \equiv z$:

To Show: $P(z)$, i.e., that: $z[x \mapsto e_1][y \mapsto e_2] = z[y \mapsto e_2][x \mapsto e_1]$
We consider three cases:

$(z = x)$**:** Then, by our assumption $(x \neq y)$ we have $z \neq y$. Therefore, we know that:

$$\begin{array}{lcll} z[x \mapsto e_1][y \mapsto e_2] & = & e_1[y \mapsto e_2] & (\textit{defn. substitution}) \\ & = & e_1 & (\textit{lemma from question, } y \notin FV(e_1)) \\ & = & z[x \mapsto e_1] & (\textit{defn. substitution}) \\ & = & z[y \mapsto e_2][x \mapsto e_1] & (\textit{defn. substitution}) \end{array}$$

$(z = y)$**:** By symmetric argument to the previous case.

$(z \neq x \wedge z \neq y)$**:** Then $z[x \mapsto e_1][y \mapsto e_2] = z = z[y \mapsto e_2][x \mapsto e_1]$, as required.

Case $e \equiv (e_3 \ \mathit{op} \ e_4)$:

To Show: $P(e_3 \ \mathit{op} \ e_4)$, i.e., that:
$(e_3 \ \mathit{op} \ e_4)[x \mapsto e_1][y \mapsto e_2] = (e_3 \ \mathit{op} \ e_4)[y \mapsto e_2][x \mapsto e_1]$
IH: $P(e_3), P(e_4)$, i.e., that: $e_3[x \mapsto e_1][y \mapsto e_2] = e_3[y \mapsto e_2][x \mapsto e_1]$, and $e_4[x \mapsto e_1][y \mapsto e_2] = e_4[y \mapsto e_2][x \mapsto e_1]$. We can show the required result as follows:

$$
\begin{array}{lll}
 & (e_3 \ \mathit{op} \ e_4)[x \mapsto e_1][y \mapsto e_2] & \\
= & (e_3[x \mapsto e_1] \ \mathit{op} \ e_4[x \mapsto e_1])[y \mapsto e_2] & (\textit{defn. substitution}) \\
= & (e_3[x \mapsto e_1][y \mapsto e_2] \ \mathit{op} \ e_4[x \mapsto e_1][y \mapsto e_2]) & (\textit{defn. substitution}) \\
= & (e_3[y \mapsto e_2][x \mapsto e_1] \ \mathit{op} \ e_4[y \mapsto e_2][x \mapsto e_1]) & (\textit{IH, twice}) \\
= & (e_3[y \mapsto e_2] \ \mathit{op} \ e_4[y \mapsto e_2])[x \mapsto e_1] & (\textit{defn. substitution}) \\
= & (e_3 \ \mathit{op} \ e_4)[x \mapsto e_1][y \mapsto e_2] & (\textit{defn. substitution})
\end{array}
$$

(iii) Let x, y, e_1, e_2, σ be arbitrary. We assume $x \neq y$ and $y \notin FV(e_1)$ and $x \notin FV(e_2)$ and seek to prove that $\forall e.\ \mathcal{A}[\![e[x \mapsto e_1][y \mapsto e_2]]\!]\sigma = \mathcal{A}[\![e[y \mapsto e_2][x \mapsto e_1]]\!]\sigma$. Let e also be arbitrary. We can then prove $\mathcal{A}[\![e[x \mapsto e_1][y \mapsto e_2]]\!]\sigma = \mathcal{A}[\![e[y \mapsto e_2][x \mapsto e_1]]\!]\sigma$ by applying the substitution lemma for arithmetic expressions (mentioned in Assignment 2), the result proved in Assignment 3, and the result proved in Assignment 1(ii). The argument goes as follows:

$$
\begin{array}{lll}
 & \mathcal{A}[\![e[x \mapsto e_1][y \mapsto e_2]]\!]\sigma & \\
= & \mathcal{A}[\![e[x \mapsto e_1]]\!]\sigma[y \mapsto \mathcal{A}[\![e_2]\!]\sigma] & (\textit{substitution lemma}) \\
= & \mathcal{A}[\![e]\!](\sigma[y \mapsto \mathcal{A}[\![e_2]\!]\sigma])[x \mapsto \mathcal{A}[\![e_1]\!]\sigma[y \mapsto \mathcal{A}[\![e_2]\!]\sigma]] & (\textit{substitution lemma}) \\
= & \mathcal{A}[\![e]\!](\sigma[y \mapsto \mathcal{A}[\![e_2]\!]\sigma])[x \mapsto \mathcal{A}[\![e_1]\!]\sigma] & (\textit{Ass. 3}, y \notin FV(e_1)) \\
= & \mathcal{A}[\![e]\!](\sigma[x \mapsto \mathcal{A}[\![e_1]\!]\sigma])[y \mapsto \mathcal{A}[\![e_2]\!]\sigma] & (\textit{Ass. 1(ii)}, x \neq y) \\
= & \mathcal{A}[\![e]\!](\sigma[x \mapsto \mathcal{A}[\![e_1]\!]\sigma])[y \mapsto \mathcal{A}[\![e_2]\!]\sigma[x \mapsto \mathcal{A}[\![e_1]\!]\sigma]] & (\textit{Ass. 3}, x \notin FV(e_2)) \\
= & \mathcal{A}[\![e[y \mapsto e_2]]\!]\sigma[x \mapsto \mathcal{A}[\![e_1]\!]\sigma] & (\textit{substitution lemma}) \\
= & \mathcal{A}[\![e[y \mapsto e_2][x \mapsto e_1]]\!]\sigma & (\textit{substitution lemma})
\end{array}
$$