

Exercise Sheet 3: Compiler Correctness

Due date: May 6th. You can hand in your solutions at the start of the exercise class.

Exercise 1 (AM Semantics)

20%

- (a) [5%] Write a WHILE program c that performs integer division x/y of two positive natural numbers given by variables x, y and stores the result in a variable z .

Solution:

$$c \equiv z := 0; \text{while } (y = x \vee x > y) \text{ do } \{z := z + 1; x := x - y\}$$

- (b) [10%] Translate your program into an AM program using the translation functions $\mathfrak{T}_a$, $\mathfrak{T}_b$ and $\mathfrak{T}_c$ presented in the lecture.

Solution:

$$\begin{aligned} \mathfrak{T}_c[z := 0] &= \mathfrak{T}_a[0]; \text{STO}(z) = \text{PUSH}(0); \text{STO}(z) \\ \mathfrak{T}_c[z := z + 1] &= \mathfrak{T}_a[z + 1]; \text{STO}(z) = \mathfrak{T}_a[z]; \mathfrak{T}_a[1]; \text{ADD}; \text{STO}(z) \\ &= \text{LOAD}(x); \text{PUSH}(1); \text{ADD}; \text{STO}(z) \Rightarrow |\mathfrak{T}_c[z := z + 1]| = 4 \\ \mathfrak{T}_c[x := x - y] &= \mathfrak{T}_a[x - y]; \text{STO}(x) = \mathfrak{T}_a[x]; \mathfrak{T}_a[y]; \text{ADD}; \text{STO}(x) \\ &= \text{LOAD}(x); \text{LOAD}(y); \text{ADD}; \text{STO}(x) \Rightarrow |\mathfrak{T}_c[x := x - y]| = 4 \\ \mathfrak{T}_b[y = x \vee x > y] &= \mathfrak{T}_b[y = x]; \mathfrak{T}_b[x > y]; \text{OR} \\ &= \mathfrak{T}_b[y]; \mathfrak{T}_b[x]; \text{EQ}; \mathfrak{T}_b[x]; \mathfrak{T}_b[y]; \text{GT}; \text{OR} \\ &= \text{LOAD}(y); \text{LOAD}(x); \text{EQ}; \text{LOAD}(x); \text{LOAD}(y); \text{GT}; \text{OR} \\ &\Rightarrow |\mathfrak{T}_b[y = x \vee x > y]| = 7 \\ \mathfrak{T}_c[c] &= \mathfrak{T}_c[z := 0]; \mathfrak{T}_c[\text{while } (y = x \vee x > y) \text{ do } \{z := z + 1; x := x - y\}] \\ &= \mathfrak{T}_c[z := 0]; \mathfrak{T}_b[y = x \vee x > y]; \text{JMPF}(4 + 4 + 2); \\ &\quad \mathfrak{T}_c[z := z + 1; x := x - y]; \text{JMP}(-(7 + 4 + 4 + 1)) \\ &= \text{PUSH}(0); \text{STO}(z); \\ &\quad \text{LOAD}(y); \text{LOAD}(x); \text{EQ}; \text{LOAD}(x); \text{LOAD}(y); \text{GT}; \text{OR} \\ &\quad \text{JMPF}(10); \\ &\quad \text{LOAD}(x); \text{PUSH}(1); \text{ADD}; \text{STO}(z); \\ &\quad \text{LOAD}(x); \text{LOAD}(y); \text{ADD}; \text{STO}(x); \\ &\quad \text{JMP}(-16) \end{aligned}$$

- (c) [5%] Give a run of the AM program, i.e. a sequence of AM configurations, starting in the initial configuration $\mathfrak{T}_c[c] \vdash \langle 0, \varepsilon, \sigma[x \mapsto 5, y \mapsto 3, z \mapsto 0] \rangle$.

Solution: Since $\text{fvc} = \{x, y, z\}$, we denote every state considered in the following run as a triple (x, y, z) to save space. Furthermore, the preceding program P to be executed is

omitted.

$$\begin{aligned}
& \mathfrak{T}_c \llbracket c \rrbracket \vdash \langle 0, \varepsilon, (5, 3, 0) \rangle \triangleright \langle 1, 0, (5, 3, 0) \rangle \triangleright \langle 2, \varepsilon, (5, 3, 0) \rangle \\
& \triangleright \langle 3, 3, (5, 3, 0) \rangle \triangleright \langle 4, 3 : 5, (5, 3, 0) \rangle \triangleright \langle 5, \text{false}, (5, 3, 0) \rangle \\
& \triangleright \langle 6, \text{false} : 5, (5, 3, 0) \rangle \triangleright \langle 7, \text{false} : 5 : 3, (5, 3, 0) \rangle \triangleright \langle 8, \text{false} : \text{true}, (5, 3, 0) \rangle \\
& \triangleright \langle 9, \text{true}, (5, 3, 0) \rangle \triangleright \langle 10, \varepsilon, (5, 3, 0) \rangle \triangleright \langle 11, 0, (5, 3, 0) \rangle \\
& \triangleright \langle 12, 0 : 1, (5, 3, 0) \rangle \triangleright \langle 13, 1, (5, 3, 0) \rangle \triangleright \langle 14, \varepsilon, (5, 3, 1) \rangle \\
& \triangleright \langle 15, 5, (5, 3, 1) \rangle \triangleright \langle 16, 5 : 3, (5, 3, 1) \rangle \triangleright \langle 17, 2, (5, 3, 1) \rangle \\
& \triangleright \langle 18, \varepsilon, (2, 3, 1) \rangle \triangleright \langle 2, \varepsilon, (2, 3, 1) \rangle \triangleright \langle 3, 3, (2, 3, 1) \rangle \\
& \triangleright \langle 4, 3 : 2, (2, 3, 1) \rangle \triangleright \langle 5, \text{false}, (2, 3, 1) \rangle \triangleright \langle 6, \text{false} : 2, (2, 3, 0) \rangle \\
& \triangleright \langle 7, \text{false} : 2 : 3, (2, 3, 1) \rangle \triangleright \langle 8, \text{false} : \text{false}, (2, 3, 1) \rangle \triangleright \langle 9, \text{false}, (2, 3, 1) \rangle \\
& \triangleright \langle 19, \varepsilon, (2, 3, 1) \rangle
\end{aligned}$$

Exercise 2 (Compiler Correctness)

50%

In the lecture, you encountered Theorem 5.9 which intuitively states that every WHILE program is translated by the “compiler” $\mathfrak{T}_c$ into an equivalent AM program, i.e. for every $c \in \text{Cmd}$, $\mathfrak{D} \llbracket c \rrbracket = \mathfrak{M} \llbracket \mathfrak{T}_c \llbracket c \rrbracket \rrbracket$.

- (a) [10%] The following statement is known as the **decomposition lemma** for AM programs. Let $c_1, c_2 \in \text{Cmd}$ and $pc \in \{0, \dots, |\mathfrak{T}_c \llbracket c_1 \rrbracket| - 1\}$. If $\mathfrak{T}_c \llbracket c_1 \rrbracket; \mathfrak{T}_c \llbracket c_2 \rrbracket \vdash \langle pc, e, \sigma \rangle \triangleright^k \langle |\mathfrak{T}_c \llbracket c_1 \rrbracket|; \mathfrak{T}_c \llbracket c_2 \rrbracket, e'', \sigma'' \rangle$, then there exists a configuration $\langle pc', e', \sigma' \rangle$ and natural numbers k_1, k_2 with $k = k_1 + k_2$ such that $\mathfrak{T}_c \llbracket c_1 \rrbracket \vdash \langle pc, e, \sigma \rangle \triangleright^{k_1} \langle |\mathfrak{T}_c \llbracket c_1 \rrbracket|, e', \sigma' \rangle$ and $\mathfrak{T}_c \llbracket c_1 \rrbracket; \mathfrak{T}_c \llbracket c_2 \rrbracket \vdash \langle |\mathfrak{T}_c \llbracket c_1 \rrbracket|, e', \sigma' \rangle \triangleright^{k_2} \langle |\mathfrak{T}_c \llbracket c_1 \rrbracket|; \mathfrak{T}_c \llbracket c_2 \rrbracket, e'', \sigma'' \rangle$. Prove that the decomposition lemma is correct. You may use the following proposition without giving an explicit proof by structural induction:

$$\forall j \in \mathbb{N} : \mathfrak{T}_c \llbracket c \rrbracket \vdash \langle pc, e, \sigma \rangle \triangleright^j \langle pc', e', \sigma' \rangle \text{ implies } pc' \in \{0, 1, \dots, |\mathfrak{T}_c \llbracket c \rrbracket|\}, \quad (1)$$

where $pc \in \{0, \dots, |\mathfrak{T}_c \llbracket c_1 \rrbracket| - 1\}$.

Solution: Assume $\mathfrak{T}_c \llbracket c_1 \rrbracket; \mathfrak{T}_c \llbracket c_2 \rrbracket \vdash \langle pc, e, \sigma \rangle \triangleright^k \langle |\mathfrak{T}_c \llbracket c_1 \rrbracket|; \mathfrak{T}_c \llbracket c_2 \rrbracket, e'', \sigma'' \rangle$.

Since $|\mathfrak{T}_c \llbracket c_1 \rrbracket; \mathfrak{T}_c \llbracket c_2 \rrbracket| = |\mathfrak{T}_c \llbracket c_1 \rrbracket| + |\mathfrak{T}_c \llbracket c_2 \rrbracket| \geq |\mathfrak{T}_c \llbracket c_1 \rrbracket|$, it follows from (1) that there exists $0 \leq k_1 \leq k$ such that $\mathfrak{T}_c \llbracket c_1 \rrbracket \vdash \langle pc, e, \sigma \rangle \triangleright^{k_1} \langle |\mathfrak{T}_c \llbracket c_1 \rrbracket|, e', \sigma' \rangle$. Since the semantics of AM is deterministic, this entails

$$\begin{aligned}
\mathfrak{T}_c \llbracket c_1 \rrbracket; \mathfrak{T}_c \llbracket c_2 \rrbracket \vdash \langle pc, e, \sigma \rangle \triangleright^{k_1} \langle |\mathfrak{T}_c \llbracket c_1 \rrbracket|, e', \sigma' \rangle \\
\triangleright^{k-k_1} \langle |\mathfrak{T}_c \llbracket c_1 \rrbracket|; \mathfrak{T}_c \llbracket c_2 \rrbracket, e'', \sigma'' \rangle.
\end{aligned}$$

Hence, the claim holds for the given k_1 and $k_2 = k - k_1$.

- (b) [35%] Complete the proof of Theorem 5.9 by proving the missing direction, i.e. show that for every $c \in \text{Cmd}$, $\sigma, \sigma' \in \Sigma$ and $e \in \text{Stk}$,

$$\mathfrak{T}_c \llbracket c \rrbracket \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^* \langle |\mathfrak{T}_c \llbracket c \rrbracket|, e, \sigma' \rangle \text{ implies } \langle c, \sigma \rangle \rightarrow \sigma' \text{ and } e = \varepsilon.$$

You may use all theorems and lemmata presented in the lecture so far (except Theorem 5.9 and Lemma 5.11 of course). In particular, note that Lemma 5.7 and Lemma 5.8 already entail the correctness of $\mathfrak{T}_a$ and $\mathfrak{T}_b$ in both directions, because the semantics of **AM** is deterministic and termination is guaranteed. Furthermore, you may assume that the decomposition lemma also works for arithmetic and Boolean expressions.

Solution: Proof by complete induction on the length k of the computation sequence of the abstract machine.

Main Idea: Make a case distinction for each $c \in \text{Cmd}$ and apply the decomposition lemma to separate a single statement from the remaining ones in order to apply the induction hypothesis or the existing lemmata showing correctness for arithmetic and Boolean expressions. For each case, check the corresponding result in operational semantics.

I.B.: The case $k = 0$ can only occur if $c = \text{skip}$ and is straightforward.

I.H.: Assume the claim holds for each computation sequence of length $k \leq k_0$.

I.S.: Consider a computation sequence of length $k_0 + 1$, i.e. assume $\mathfrak{T}_c \llbracket c \rrbracket \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^{k_0+1} \langle |\mathfrak{T}_c \llbracket c \rrbracket|, e, \sigma' \rangle$. We proceed by a case distinction of $\mathfrak{T}_c \llbracket c \rrbracket$ for each $c \in \text{Cmd}$ (except **skip**).

- $\mathfrak{T}_c \llbracket x := a \rrbracket = \mathfrak{T}_a \llbracket a \rrbracket; \text{STO}(x)$. By the decomposition lemma, there exist k_1, k_2 with $k = k_1 + k_2$ such that $\mathfrak{T}_a \llbracket a \rrbracket \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^{k_1} \langle |\mathfrak{T}_a \llbracket a \rrbracket|, e'', \sigma'' \rangle$ and $\mathfrak{T}_a \llbracket a \rrbracket; \text{STO}(x) \vdash \langle |\mathfrak{T}_a \llbracket a \rrbracket|, e'', \sigma'' \rangle \triangleright^{k_2} \langle |\mathfrak{T}_a \llbracket a \rrbracket|, e, \sigma' \rangle$. By Lemma 5.7, we already know that $\mathfrak{T}_a \llbracket a \rrbracket \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^{k_1} \langle |\mathfrak{T}_a \llbracket a \rrbracket|, z, \sigma \rangle$ and $\langle a, \sigma \rangle \rightarrow z$. Hence, in operational semantics, we have $\langle x := a, \sigma \rangle \rightarrow \sigma[x \mapsto z]$. Furthermore, applying the semantics of $\text{STO}(x)$ yields $\mathfrak{T}_a \llbracket a \rrbracket; \text{STO}(x) \vdash \langle |\mathfrak{T}_a \llbracket a \rrbracket|, z, \sigma \rangle \triangleright^{k_2} \langle |\mathfrak{T}_a \llbracket a \rrbracket| + 1, \varepsilon, \sigma[x \mapsto z] \rangle$.
- $\mathfrak{T}_c \llbracket c_1; c_2 \rrbracket = \mathfrak{T}_c \llbracket c_1 \rrbracket; \mathfrak{T}_c \llbracket c_2 \rrbracket$. By the decomposition lemma, there exist k_1, k_2 with $k = k_1 + k_2$ such that $\mathfrak{T}_c \llbracket c_1 \rrbracket \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^{k_1} \langle |\mathfrak{T}_c \llbracket c_1 \rrbracket|, e'', \sigma'' \rangle$ and $\mathfrak{T}_c \llbracket c_1 \rrbracket; \mathfrak{T}_c \llbracket c_2 \rrbracket \vdash \langle |\mathfrak{T}_c \llbracket c_1 \rrbracket|, e'', \sigma'' \rangle \triangleright^{k_2} \langle |\mathfrak{T}_c \llbracket c \rrbracket|, e, \sigma' \rangle$. Now, by I.H., we have $\langle c_1, \sigma \rangle \rightarrow \sigma''$ and $e'' = \varepsilon$. Then, again by I.H., it follows $\langle c_2, \sigma'' \rangle \rightarrow \sigma'$ and $e = \varepsilon$. Applying the sequence rule of operational semantics, we directly obtain $\langle c_1; c_2, \sigma \rangle \rightarrow \sigma'$.
- $\mathfrak{T}_c \llbracket \text{if } (b) \text{ then } \{c_1\} \text{ else } \{c_2\} \rrbracket = \mathfrak{T}_b \llbracket b \rrbracket; \text{JMPF}(|\mathfrak{T}_c \llbracket c_1 \rrbracket| + 2; \mathfrak{T}_c \llbracket c_1 \rrbracket; \text{JMP}(|\mathfrak{T}_c \llbracket c_2 \rrbracket| + 1); \mathfrak{T}_c \llbracket c_2 \rrbracket$. By the decomposition lemma, there exist k_1, k_2 with $k = k_1 + k_2$ such that $\mathfrak{T}_b \llbracket b \rrbracket \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^{k_1} \langle |\mathfrak{T}_b \llbracket b \rrbracket|, e'', \sigma'' \rangle$ and $\mathfrak{T}_c \llbracket \text{if } (b) \text{ then } \{c_1\} \text{ else } \{c_2\} \rrbracket \vdash \langle |\mathfrak{T}_b \llbracket b \rrbracket|, e'', \sigma'' \rangle \triangleright^{k_2} \langle |\mathfrak{T}_c \llbracket \text{if } (b) \text{ then } \{c_1\} \text{ else } \{c_2\} \rrbracket|, e, \sigma' \rangle$. By Lemma 5.8 and determinism of **AM**, we have $\langle b, \sigma \rangle \rightarrow B \in \{\text{true}, \text{false}\}$, $e'' = B$, and $\sigma'' = \sigma$. Two cases arise.
 - $B = \text{true}$. $\mathfrak{T}_c \llbracket \text{if } (b) \text{ then } \{c_1\} \text{ else } \{c_2\} \rrbracket \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^{k_1+1} \langle |\mathfrak{T}_b \llbracket b \rrbracket| + 1, \varepsilon, \sigma \rangle$. Again, by the decomposition lemma, there are k_3, k_4 with $k_3 + k_4 = k_2 - 1$ such that $\mathfrak{T}_c \llbracket c_1 \rrbracket \vdash \langle |\mathfrak{T}_b \llbracket b \rrbracket| + 1, \varepsilon, \sigma \rangle \triangleright^{k_3} \langle |\mathfrak{T}_c \llbracket c_1 \rrbracket|, e''', \sigma''' \rangle$ and $\mathfrak{T}_c \llbracket \text{if } (b) \text{ then } \{c_1\} \text{ else } \{c_2\} \rrbracket \vdash \langle |\mathfrak{T}_b \llbracket b \rrbracket| + 1, \varepsilon, \sigma \rangle \triangleright^{k_4} \langle |\mathfrak{T}_c \llbracket \text{if } (b) \text{ then } \{c_1\} \text{ else } \{c_2\} \rrbracket|, e, \sigma' \rangle$. By I.H., $\langle c_1, \sigma \rangle \rightarrow \sigma'''$ and $e''' = \varepsilon$. Applying the instruction $\text{JMP}(|\mathfrak{T}_c \llbracket c_2 \rrbracket| + 1)$ then yields $\mathfrak{T}_c \llbracket \text{if } (b) \text{ then } \{c_1\} \text{ else } \{c_2\} \rrbracket \vdash$

$$\langle 0, \varepsilon, \sigma \rangle \triangleright^{k_1+1+k_3} \langle |\mathfrak{T}_b[b]| + 1 + |\mathfrak{T}_c[c_1]|, \varepsilon, \sigma''' \rangle \triangleright^1 \langle |\mathfrak{T}_c[\text{if } (b) \text{ then } \{c_1\} \text{ else } \{c_2\}]|, \varepsilon, \sigma''' \rangle.$$

Since **AM** is deterministic, it follows $\sigma''' = \sigma'$. Applying the (if-true) rule of operational semantics yields $\langle \text{if } (b) \text{ then } \{c_1\} \text{ else } \{c_2\}, \sigma \rangle \rightarrow \sigma'$, because $\langle c_1, \sigma \rangle \rightarrow \sigma''$ and $\langle b, \sigma \rangle \rightarrow \text{true}$.

– $B = \text{false}$. Analogous.

- $\mathfrak{T}_c[\text{while } (b) \text{ do } \{c\}] = \mathfrak{T}_b[b]; \text{JMPF}(|\mathfrak{T}_c[c]| + 2); \mathfrak{T}_c[c]; \text{JMP}(-(|\mathfrak{T}_b[b]| + |\mathfrak{T}_c[c]| + 1))$. By the decomposition lemma, there exist k_1, k_2 with $k = k_1 + k_2$ such that $\mathfrak{T}_b[b] \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^{k_1} \langle |\mathfrak{T}_b[b]|, e'', \sigma'' \rangle$ and $\mathfrak{T}_c[\text{while } (b) \text{ do } \{c\}] \vdash \langle |\mathfrak{T}_b[b]|, e'', \sigma'' \rangle \triangleright^{k_2} \langle |\mathfrak{T}_c[\text{while } (b) \text{ do } \{c\}]|, e, \sigma' \rangle$. By Lemma 5.8 and determinism of **AM**, we have $\langle b, \sigma \rangle \rightarrow B \in \{\text{true}, \text{false}\}$, $e'' = B$ and $\sigma'' = \sigma$. Again, two cases arise.

- $B = \text{true}$. Then $\mathfrak{T}_c[\text{while } (b) \text{ do } \{c\}] \langle 0, \varepsilon, \sigma \rangle \triangleright^{k_1+1} \langle |\mathfrak{T}_b[b]| + 1, \varepsilon, \sigma \rangle$. Applying the decomposition lemma again, there are k_3, k_4 with $k_3 + k_4 = k_2 - 1$ such that $\mathfrak{T}_c[c] \vdash \langle |\mathfrak{T}_b[b]| + 1, \varepsilon, \sigma \rangle \triangleright^{k_3} \langle |\mathfrak{T}_b[b]| + 1 + |\mathfrak{T}_c[c]|, e''', \sigma''' \rangle$ and $\mathfrak{T}_c[\text{while } (b) \text{ do } \{c\}] \vdash \langle |\mathfrak{T}_b[b]| + 1 + |\mathfrak{T}_c[c]|, e''', \sigma''' \rangle \triangleright^{k_4} \langle |\mathfrak{T}_c[\text{while } (b) \text{ do } \{c\}]|, e, \sigma' \rangle$. By I.H., we have $e''' = \varepsilon$ and $\langle c, \sigma \rangle \rightarrow \sigma'''$. Now, applying one further computation step yields $\mathfrak{T}_c[\text{while } (b) \text{ do } \{c\}] \vdash \langle |\mathfrak{T}_b[b]| + 1 + |\mathfrak{T}_c[c]|, e''', \sigma''' \rangle \triangleright^1 \langle 0, \varepsilon, \sigma''' \rangle \triangleright^{k_4-1} \langle |\mathfrak{T}_c[\text{while } (b) \text{ do } \{c\}]|, e, \sigma' \rangle$. By I.H., we have $\langle \text{while } (b) \text{ do } \{c\}, \sigma''' \rangle \rightarrow \sigma'$ and $e = \varepsilon$. Applying the (while – true) rule of operational semantics then yields $\langle \text{while } (b) \text{ do } \{c\}, \sigma \rangle \rightarrow \sigma'$.
- $B = \text{false}$. By applying one further step of the semantics of **AM**, we obtain $\mathfrak{T}_c[\text{while } (b) \text{ do } \{c\}] \langle 0, \varepsilon, \sigma \rangle \triangleright^{k_1+1} \langle |\mathfrak{T}_c[\text{while } (b) \text{ do } \{c\}]|, \varepsilon, \sigma \rangle$ and $\langle \text{while } (b) \text{ do } \{c\}, \sigma \rangle \rightarrow \sigma$ by applying the (while – false) rule of operational semantics.

- (c) [5%] Does the decomposition lemma also hold if $\mathfrak{T}_c[c_1]$ and $\mathfrak{T}_c[c_2]$ are replaced by arbitrary **AM** programs P_1 and P_2 ? Explain your answer.

Solution: Let $P_1 := \text{JMP}(1); \text{JMP}(5)$ and $P_2 := \text{PUSH}(5); \text{STO}(x)$. Then, $P_1; P_2 \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^* \langle 5, \varepsilon, \sigma \rangle$, but there exists no corresponding configuration $\langle 3, e, \sigma' \rangle$ as required by the decomposition lemma.

Exercise 3 (Language Extension (again))

30%

- (a) [10%] In the previous exercise sheet, you extended **WHILE** by an additional construct **repeat** $\{c\}$ **until** (b) . Extend the translation function $\mathfrak{T}_c$ such that **repeat** $\{c\}$ **until** (b) statements can be translated directly, i.e. without rewriting them into normal **WHILE** programs first. For reasons of efficiency, we require the statement c to be translated only once.

Solution:

$$\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)] = \mathfrak{T}_c[c]; \mathfrak{T}_b[b]; \text{JMPF}(-(|\mathfrak{T}_c[c]| + |\mathfrak{T}_b[b]|))$$

(b) [20%] Prove that for all $\sigma, \sigma' \in \Sigma$, we have

$$\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)] \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^* \langle |\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)]|, \varepsilon, \sigma' \rangle$$

if and only if $\langle \text{repeat } \{c\} \text{ until } (b), \sigma \rangle \rightarrow \sigma'$.

You may use all theorems and lemmata presented in the lecture so far.

Solution: It suffices to extend the proof of Theorem 5.9 from the lecture by the missing cases for $\text{repeat } \{c\} \text{ until } (b)$.

“if”: Assume $\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)] \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^{k_0+1} \langle |\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)]|, e, \sigma' \rangle$ where k_0 is as in the induction step of exercise 2b. By the decomposition lemma, there exist $k_1, k_2 \in \mathbb{Z}$ with $k_1 + k_2 = k$ such that $\mathfrak{T}_c[c] \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^{k_1} \langle |\mathfrak{T}_c[c]|, e'', \sigma'' \rangle$ and $\mathfrak{T}_c[c]; \mathfrak{T}_b[b]; \text{JMPF}(-(|\mathfrak{T}_c[c]| + |\mathfrak{T}_b[b]| + 1)) \vdash \langle |\mathfrak{T}_c[c]|, e'', \sigma'' \rangle \triangleright^{k_2} \langle |\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)]|, e', \sigma' \rangle$. By I.H., we have $\langle c, \sigma \rangle \rightarrow \sigma''$. Applying the decomposition lemma again, there exist $k_3, k_4 \in \mathbb{Z}$ with $k_3 + k_4 = k_2$ such that $\mathfrak{T}_b[b]; \text{JMPF}(-(|\mathfrak{T}_c[c]| + |\mathfrak{T}_b[b]| + 1)) \vdash \langle |\mathfrak{T}_c[c]|, e'', \sigma'' \rangle \triangleright^{k_3} \langle |\mathfrak{T}_c[c]| + |\mathfrak{T}_b[b]|, e''', \sigma''' \rangle$ and $\mathfrak{T}_b[b]; \text{JMPF}(-(|\mathfrak{T}_c[c]| + |\mathfrak{T}_b[b]| + 1)) \vdash \langle |\mathfrak{T}_c[c]| + |\mathfrak{T}_b[b]|, e''', \sigma''' \rangle \triangleright^{k_4} \langle |\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)]|, e', \sigma' \rangle$. By Lemma 5.8, $\sigma''' = \sigma''$ and $e''' = \text{true}$ iff $\langle b, \sigma'' \rangle \rightarrow \text{true}$ and false otherwise. In the first, case, applying the JMPF(...) instruction yields $\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)] \vdash \langle 0, \varepsilon, \sigma'' \rangle \triangleright^{k_4-1} \langle |\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)]|, e', \sigma' \rangle$. By I.H., $\langle \text{repeat } \{c\} \text{ until } (b), \sigma'' \rangle \rightarrow \sigma'$ and $e' = \varepsilon$. Applying the repeat-until rule of operational semantics yields $\langle \text{repeat } \{c\} \text{ until } (b) \rangle \rightarrow \sigma'$ and $e' = \varepsilon$. The false case is analogous.

“only-if”: By the (repeat-until) rules of operational semantics, $\langle \text{repeat } \{c\} \text{ until } (b), \sigma \rangle \rightarrow \sigma'$ implies $\langle c, \sigma \rangle \rightarrow \sigma''$ and either $\langle b, \sigma'' \rangle \rightarrow \text{false}$ and $\langle \text{repeat } \{c\} \text{ until } (b), \sigma'' \rangle \rightarrow \sigma'$ or $\langle b, \sigma'' \rangle \rightarrow \text{true}$ and $\sigma'' = \sigma'$. In both cases, we can apply the I.H. to obtain $\mathfrak{T}_c[c] \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^* \langle |\mathfrak{T}_c[c]|, \varepsilon, \sigma'' \rangle$. Two cases arise.

1. $\langle b, \sigma'' \rangle \rightarrow \text{true}$. By I.H. $\mathfrak{T}_b[b] \vdash \langle 0, \varepsilon, \sigma'' \rangle \triangleright^* \langle |\mathfrak{T}_b[b]|, \text{true}, \sigma'' \rangle$. Then

$$\begin{aligned} \mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)] &\vdash \langle 0, \varepsilon, \sigma \rangle \\ &\triangleright^* \langle |\mathfrak{T}_c[c]|; \mathfrak{T}_b[b]|, \text{true}, \sigma'' \rangle \\ &\triangleright \langle |\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)]|, \varepsilon, \sigma'' \rangle \\ &\text{and } \sigma'' = \sigma'. \end{aligned}$$

2. $\langle b, \sigma'' \rangle \rightarrow \text{false}$. By I.H. $\mathfrak{T}_b[b] \vdash \langle 0, \varepsilon, \sigma'' \rangle \triangleright^* \langle |\mathfrak{T}_b[b]|, \text{false}, \sigma'' \rangle$ and $\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)] \vdash \langle 0, \varepsilon, \sigma'' \rangle \triangleright^* \langle |\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)]|, \varepsilon, \sigma' \rangle$. Then

$$\begin{aligned} \mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)] &\vdash \langle 0, \varepsilon, \sigma \rangle \\ &\triangleright^* \langle |\mathfrak{T}_c[c]|, \varepsilon, \sigma'' \rangle \\ &\triangleright^* \langle |\mathfrak{T}_c[c]|; \mathfrak{T}_b[b]|, \text{false}, \sigma'' \rangle \\ &\triangleright^* \langle 0, \varepsilon, \sigma'' \rangle \\ &\triangleright^* \langle |\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)]|, \varepsilon, \sigma' \rangle. \end{aligned}$$

In both cases, we have

$$\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)] \vdash \langle 0, \varepsilon, \sigma \rangle \triangleright^* \langle |\mathfrak{T}_c[\text{repeat } \{c\} \text{ until } (b)]|, \varepsilon, \sigma' \rangle.$$