1 Control Flow Analysis

Consider the following lambda calculus program.

\[(\lambda f. (f\,76) + (f\,77))\,(\lambda a. a)\]

(a) Add labels to the program. That is, make it an expression in the labeled lambda calculus of Lecture 24, where every label is unique.

Answer:

\[(((\lambda f. ((f^1\,76^2)^3 + (f^4\,77^5)^6)^7)^8\,(\lambda a. a)^9)^{10})^{11}\]

To be clear about what exactly the subexpressions are, here is the result of \text{exprof}(e, i), where \(e\) is the labeled lambda calculus program.

\[
\begin{align*}
\text{exprof}(e, 1) &= f^1 \\
\text{exprof}(e, 2) &= 76^2 \\
\text{exprof}(e, 3) &= (f^1\,76^2)^3 \\
\text{exprof}(e, 4) &= f^4 \\
\text{exprof}(e, 5) &= 77^5 \\
\text{exprof}(e, 6) &= (f^4\,77^5)^6 \\
\text{exprof}(e, 7) &= ((f^1\,76^2)^3 + (f^4\,77^5)^6)^7 \\
\text{exprof}(e, 8) &= (\lambda f. ((f^1\,76^2)^3 + (f^4\,77^5)^6)^7)^8 \\
\text{exprof}(e, 9) &= a^9 \\
\text{exprof}(e, 10) &= (\lambda a. a^9)^{10} \\
\text{exprof}(e, 11) &= ((\lambda f. ((f^1\,76^2)^3 + (f^4\,77^5)^6)^7)^8\,(\lambda a. a^9)^{10})^{11}
\end{align*}
\]

(b) Let \(e\) be your labeled lambda calculus program. Write out \(C[e]_e\), i.e., the set of constraints for the program \(e\). (Hint, you should have 20 constraints in total. In particular, for each of the 3 applications, you should have 4 constraints, 2 for each of the lambda terms in the program.)

Answer: The constraints are as follows. Make sure you understand where each constraint comes from. We
have generated the constraints first for labeled expression 1, then for labeled expression 2, and so on.

\[ C[e] = \{ r(f) \subseteq C(1), \]
\[ 2 \in C(2), \]
\[ 8 \in C(1) \Rightarrow C(2) \subseteq r(f), \]
\[ 10 \in C(1) \Rightarrow C(2) \subseteq r(a), \]
\[ 8 \in C(1) \Rightarrow C(7) \subseteq C(3), \]
\[ 10 \in C(1) \Rightarrow C(9) \subseteq C(3), \]
\[ r(f) \subseteq C(4), \]
\[ 5 \in C(5), \]
\[ 8 \in C(4) \Rightarrow C(5) \subseteq r(f), \]
\[ 10 \in C(4) \Rightarrow C(5) \subseteq r(a), \]
\[ 8 \in C(4) \Rightarrow C(7) \subseteq C(6), \]
\[ 10 \in C(4) \Rightarrow C(9) \subseteq C(6), \]
\[ 7 \in C(7), \]
\[ 8 \in C(8), \]
\[ r(a) \subseteq C(9), \]
\[ 10 \in C(10), \]
\[ 8 \in C(8) \Rightarrow C(10) \subseteq r(f), \]
\[ 10 \in C(8) \Rightarrow C(10) \subseteq r(a), \]
\[ 8 \in C(8) \Rightarrow C(7) \subseteq C(11), \]
\[ 10 \in C(8) \Rightarrow C(9) \subseteq C(11) \}

(c) Find \( C^* \) and \( r^* \), the smallest functions that satisfy the constraints you generated in the question above.

**Answer:** We show the result of the iterative algorithm described in the lecture notes. Note that iteration 7 is the same as iteration 6, so that is a fixed point.

<table>
<thead>
<tr>
<th></th>
<th>( C(1) )</th>
<th>( C(2) )</th>
<th>( C(3) )</th>
<th>( C(4) )</th>
<th>( C(5) )</th>
<th>( C(6) )</th>
<th>( C(7) )</th>
<th>( C(8) )</th>
<th>( C(9) )</th>
<th>( C(10) )</th>
<th>( C(11) )</th>
<th>( r(f) )</th>
<th>( r(a) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>2.5</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>2.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>2</td>
<td>2.5</td>
<td>10</td>
<td>5</td>
<td>2.5</td>
<td>7</td>
<td>2.5</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>2.5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>2</td>
<td>2.5</td>
<td>10</td>
<td>5</td>
<td>2.5</td>
<td>7</td>
<td>2.5</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>2.5</td>
<td></td>
</tr>
</tbody>
</table>

(d) Check that your functions \( C^* \) and \( r^* \) make sense. That is, if an expression labeled \( l \) can evaluate to an expression labeled \( l' \), do you have \( l' \in C^*(l) \)?

**Answer:** Yes, it does make sense! Consider \( r^*(f) \), i.e., what values variable \( f \) can be bound to. \( r^*(f) = \{10\} \), meaning that the only value that \( f \) can be bound to is the expression labeled 10: \((\lambda a. a^9)^{10}\). That is indeed the
Consider also \( r^*(a) = \{2, 5\} \). The variable \( a \) can be bound to expressions \( 76^2 \) and \( 77^5 \), which is indeed the case. Check some of the other results, and make sure that they correctly reflect the program execution.

(e) Consider adding the expression \((\text{let } x = e_1 \text{ in } e_2)^l\) to the language. Define \( C[(\text{let } x = e_1 \text{ in } e_2)^l]_e \). Try rewriting the program above using one or more let expressions, and make sure that the constraints you generate for the modified program produce the same solution \( C^* \) and \( r^* \).

**Answer:**

\[
C[(\text{let } x = e_1^i \text{ in } e_2^j)^l]_e = C[e_1^i]_e \cup C[e_2^j]_e \cup \{C(l_1) \subseteq r(x), C(l_2) \subseteq C(l)\}
\]

We leave the rewriting of the program and the generation of constraints as an exercise.