Homework Assignment 1

Li Kunle

Exercise 1

1. When deleting the last k elements from the list, we just decrease the length of the list by k (if $k > 0$, or set the length to 0, if $k < 0$). So *length(l)* contributes to time complexity $\theta(1)$ here.

   Besides, according to the lecture, we can shift the c contributions of *delete* to *insert* and *append*, making their contributions to $3c$. And the amortised cost of *deallocation* is 0. Therefore, $delete\_last(l, k)$ has 0 contribution and the amortised complexity is in $\theta(1)$.

Exercise 2

1. If g is associative and e is an identity of g, $src[e, f, g]$ is well-defined.

   Because given a list $l = [l_1, l_2, \dots, l_n]$, we need to map l_i with $f(l_i)$ for every $i \in [1, n]$, and then combine them with $g(f(l_i), f(l_j))$ or $g(f(l_i), e)$. For the first part, since we are not sure how will the list be separate, we need to guarantee that

$$g(g(src[e,f,g](l_1), src[e,f,g](l_2)), src[e,f,g](l_3)) = g(src[e,f,g](l_1), g(src[e,f,g](l_2), src[e,f,g](l_3))) \quad (1)$$

   which means that g must be associative. Otherwise, different separation of lists may yield to different output, in turn, $src[e, f, g]$ will not be well defined.

   For the second part, when g functions on $f(l_1)$ and e, the result must also be in the domain of g for further combination using g. So e must be an identity of g.

2. a. determine the length:

$$src[0,1,+](l) = \begin{cases} 0 & l = [] \\ 1 & l = [x] \\ src[0,1,+](l_1) + src[0,1,+](l_2) & l = l_1 + l_2 \end{cases} \quad (2)$$

   b. apply a function to all elements:

$$src[[],h,+](l) = \begin{cases} [] & l = [] \\ h(x) & l = [x] \\ src[[],h,+](l_1) + src[[],h,+](l_2) & l = l_1 + l_2 \end{cases} \quad (3)$$

   where $h(x)$ is an arbitrary function that maps elements of a set T (the set of list elements) to elements of a set T'.

   c. create a sublist satisfying a condition φ:

$$src[[],k,+](l) = \begin{cases} [] & l = [] \\ k(x) & l = [x] \\ src[[],k,+](l_1) + src[[],k,+](l_2) & l = l_1 + l_2 \end{cases} \quad (4)$$

   where

$$k(x) = \begin{cases} [x] & if\ \varphi \\ [] & else \end{cases} \quad (5)$$

3. Suppose the time complexity of f is F, and the length of the list is n. The structural recursion needs to apply g (n - 1) times, which contributes to time complexity $O(n)$, and f n times, which contributes to time complexity $O(Fn)$. So the total time complexity is in $O(Fn + n) = O(Fn)$.

Exercise 3

1. $pushback(l, x) \Leftrightarrow append(l, x)$

   $pushfront(l, x) \Leftrightarrow insert(l, 1, x)$

   $popback(l) \Leftrightarrow y := get(l, length); delete(l, length)$

   $popfront(l) \Leftrightarrow y := get(l, 1); delete(l, 1)$