8.1 (20 Points) Resource-Polymorphic Recursion

In this problem we are interested in the number of cons operations. We use a metric $M$ with $M_{\text{cons}} = 1$ and $M_K = 0$ for all $K \neq \text{cons}$.

Consider the following OCaml functions.

```ocaml
let rec rev_append (l1,l2) =  
  match l1 with  
  | [] -> l2  
  | x::xs -> rev_append (xs, x::l2)

let rec skip l =  
  match l with  
  | [] -> []  
  | x1::xs ->  
    match xs with  
    | [] -> []  
    | x2::xs -> x2::skip(xs)

let rec f1 l =  
  match l with  
  | [] -> []  
  | x::xs ->  
    let l' = skip l in  
    rev_append (f1 l', l')
```

a) Give linear resource-annotated types for the functions `rev_append`, `skip`, and `f1`.

b) Provide annotated type derivations for the functions `skip` and `f1` that justify your types. Transform the functions to share-let-normal form for the type derivation.

c) Argue informally why the type inference algorithm that we discussed in class cannot derive a resource-annotated type for `f1`.

d) Now give a resource-annotated type derivation for `f2`, which is defined below. Why can the type inference algorithm derive this bound?
let rec f2 l = 
  match l with 
  | [] → [] 
  | x::xs → 
      let l' = skip l in 
      rev_append (l', f2 l')
e) Finally, consider again the original program in which we replace skip with the following implementation.

let rec skip l = 
  match l with 
  | [] → [] 
  | x1::xs → 
      match xs with 
      | [] → [x1] 
      | x2::xs → x2::skip(xs)

Explain informally why your type derivation from (b) does not work for the new variant of skip. Can you informally derive a bound for f1 with the new variant of skip?

8.2 (16 Points) Non-Negative Polynomials

The potential functions of univariate polynomial amortized resource analysis are a generalization of non-negative linear combinations of binomial coefficients (binomials)

\[ \mathcal{B} = \left\{ \lambda n. \sum_{i=0,...,k} q_i \binom{n}{i} \mid k \in \mathbb{N}, q_i \in \mathbb{Q}_{\geq 0} \right\} \]

Recall that \( \binom{n}{k} = 1 \) for every \( n \in \mathbb{N} \).

For a function \( f : \mathbb{N} \rightarrow \mathbb{Q}_{\geq 0} \), the discrete derivative \( \Delta f : \mathbb{N} \rightarrow \mathbb{Q}_{\geq 0} \) is defined by

\[ (\Delta f)(n) = f(n+1) - f(n). \]

As usual, we define \( \Delta^0 f = f \) and \( \Delta^k f = \Delta(\Delta^{k-1} f) \) if \( k > 0 \).

The set of polynomials is defined as

\[ \mathcal{P} = \left\{ \lambda n. \sum_{i=0,...,k} q_i n^i \mid k \in \mathbb{N}, q_i \in \mathbb{Q} \right\} \]

We call a function \( f : \mathbb{N} \rightarrow \mathbb{Q} \) hereditary non-negative if \( \Delta^i f \geq 0 \) for all \( i \in \mathbb{N} \).

a) Prove that if \( f \in \mathcal{B} \) and \( k \in \mathbb{N} \) then \( \Delta^k f \in \mathcal{B} \). Note that it follows that \( \mathcal{B} \) is a set of hereditary non-negative polynomials.

b) Show that \( \mathcal{B} \) is the largest set of hereditary non-negative polynomials: \( \mathcal{P} \cap \{ f \mid \forall i \Delta^i f \geq 0 \} = \mathcal{B} \).

Hint: If \( p : \mathbb{N} \rightarrow \mathbb{Q} \) is a hereditary non-negative polynomial then \( p(n) = \sum_i q_i \binom{n}{i} \) where \( q_i = (\Delta^i p)(0) \).

c) Let \( \mathcal{C} \) be a set of non-negative polynomials that is closed under discrete differentiation, that is, \( p \in \mathcal{C} \implies \Delta p \in \mathcal{C} \). Show that \( \mathcal{C} \subseteq \mathcal{B} \).
8.3  (12 Points) Resource Aware ML

Resource Aware ML (RAML) is an implementation of multivariate polynomial amortized resource analysis for OCaml. A web interface for RAML is available at

http://raml.co

1. Use the web interface of RAML to derive evaluation-step bounds on the functions defined in Problem 8.1 and report the derived bounds.

2. Use the template in the file search.raml and the functional queue from Problem 2.3 to implement a breadth-first search and a depth-first search in RAML. Derive and report evaluation-step bounds using the web interface.
Given resource metric $M$, expression $e$ has annotated type $A$ under signature $\Sigma$ in context $\Gamma$.

$$\Sigma; \Gamma \vdash e : B \quad \text{Given resource metric } M, \text{expression } e \text{ has annotated type } A \text{ under signature } \Sigma \text{ in context } \Gamma.$$

$$\frac{q = q' + M^{\text{var}}}{\Sigma; x : B \vdash e : B} \quad (\text{L:VAR}) \quad \frac{A \cdot \epsilon^M_B \cdot B \in \Sigma(f)}{\Sigma; x : A \vdash e : B} \quad (\text{L:APP})$$

$$\frac{\Sigma; \Gamma_1 \vdash e_1 : A \quad \Sigma; \Gamma_2 \vdash e_2 : B}{\Sigma; \Gamma_1, \Gamma_2 \vdash \text{let}(e_1, e_2) : B} \quad (\text{L:LET}) \quad \frac{q = p + M^{\text{let}}}{\Sigma; x : B \vdash e : B} \quad (\text{L:LET})$$

$$\frac{e \in \{\text{true, false}\}}{\Sigma; \vdash q' : B \text{Bool}} \quad (\text{L:CONST}) \quad \frac{\Sigma; \vdash \text{pair}(x_1, x_2) : A_1 \ast A_2}{\Sigma; \vdash q' : B \text{Pair}} \quad (\text{L:PAIR})$$

$$\frac{\Sigma; \vdash \text{matP}(e, (x_1, x_2), e') : B}{\Sigma; \vdash q' : B \text{Pair}} \quad (\text{L:MATP}) \quad \frac{\Sigma; \vdash \text{matL}(x, e_1, (x_1, x_2), e_2) : B}{\Sigma; \vdash q' : B \text{Pair}} \quad (\text{L:MATL})$$

$$\frac{\Sigma; \vdash \text{cons}(x_1, x_2) : L^P(A)}{\Sigma; \vdash q' : L^P(A)} \quad (\text{L:CONS}) \quad \frac{\Sigma; \vdash \text{cons}(x_1, x_2) : L^P(A)}{\Sigma; \vdash q' : L^P(A)} \quad (\text{L:CONS})$$

$$\frac{\Sigma; \vdash \text{share}(x_1, (x_1, x_2), e) : B}{\Sigma; \vdash q' : B \text{Pair}} \quad (\text{L:SHARE}) \quad \frac{\Sigma; \vdash A' <: A}{\Sigma; \vdash q' : A \text{Sup}} \quad (\text{L:SUPERTY}) \quad \frac{\Sigma; \vdash e : B \quad B <: B'}{\Sigma; \vdash q' : B' \text{Sub}} \quad (\text{L:SUBTY})$$

$$\frac{q = p + M^{\text{let}}}{\Sigma; \vdash q' : B} \quad (\text{L:LET}) \quad \frac{\Sigma; \vdash e : B}{\Sigma; \vdash q' : B} \quad (\text{L:LET})$$

$$\Sigma; \vdash e : B \quad q \geq p \quad q - p \geq p' \quad (\text{L:RELAX}) \quad \frac{\Sigma; \vdash e : B}{\Sigma; \vdash q' : B} \quad (\text{L:WEAK})$$

Figure 1: Linear resource-annotated type rules.