Resource Analysis: Problem Set 9

Jan Hoffmann
Carnegie Mellon University

April 19, 2016

Due before 1:30pm on Monday, April 25

9.1 Univariate Polynomial Amortized Analysis (12 Points)

Recall that the type rules of univariate polynomial amortized resource analysis are similar to the rules for the linear type system. The main difference are the rules for pattern matching and constructing lists. All rules can be found in Figure 1.

```
let rec append (l1,l2) =
  match l1 with
  | [] → l2
  | x::xs → x::(append (xs,l2))

let rec partition ((y : int),l) =
  match l with
  | [] → ([],[])
  | x::xs →
     let (cs,bs) = partition (y,xs) in
     if y<x then
       (cs,x::bs)
     else
       (x::cs,bs)

let rec quicksort l =
  match l with
  | [] → []
  | x::xs →
     let (ys,zs) = partition (x,xs) in
     append (quicksort ys, x :: (quicksort zs))
```

Consider the implementation of quick sort that is given above. We are again interested in the number of cons operations that are performed during an evaluation, that is, we use a metric M with $M^{\mathsf{cons}} = 1$ and $M^K = 0$ for all $K \neq \mathsf{cons}$.

a) Provide resource annotated types for the functions *append*, *partition*, and *quicksort*.

b) Give a type derivation for *quicksort*. (You don't have to derive the types for *partition* and *append*.)

9.2 Multivariate Resource Polynomials (18 Points)

Recall that multivariate resource polynomials $R(A)$ for values of type A are non-negative linear combinations of base polynomials $P(A)$ of type A. Base polynomials are defined as follows.

$$\begin{array}{rcl} P(A) & = & \{\lambda a\,.\,1\} \text{ if } A \text{ is an atomic type} \\ P(A_1 * A_2) & = & \{\lambda(a_1,a_2)\,.\,p_1(a_1)\cdot p_2(a_2) \mid p_i \in P(A_i)\} \\ P(L(A)) & = & \left\{\lambda[a_1,\dots,a_n]\,.\, \sum_{1\le j_1<\dots<j_k\le n} \prod_{1\le i\le k} p_i(a_{j_i}) \mid k\in\mathbb{N}, p_i\in P(A)\right\} \end{array}$$

To assign a unique name to each base polynomial we define the *index set* $\mathcal{I}(A)$ to denote resource polynomials for a given data type A. Interestingly, but as I find coincidentally, $\mathcal{I}(A)$ is essentially the meaning of A with every atomic type replaced by *unit*.

$$\begin{array}{rcl} \mathcal{I}(A) & = & \{*\} \text{ if } A\in\{\text{Bool}\}\cup\mathcal{X} \\ \mathcal{I}(A_1 * A_2) & = & \{(i_1,i_2) \mid i_1\in\mathcal{I}(A_1) \text{ and } i_2\in\mathcal{I}(A_2)\} \\ \mathcal{I}(L(B)) & = & \{[i_1,\dots,i_k] \mid k\ge 0, i_j\in\mathcal{I}(B)\} \end{array}$$

For each $i\in\mathcal{I}(A)$, I define a base polynomial $p_i\in P(A)$ as follows: If $A\in\{\mathit{int},\mathit{bool},\mathit{unit}\}$ then

$$p_*(a) = 1\,.$$

If $A=(A_1 * A_2)$ is a pair type and $a=(a_1,a_2)$ then

$$p_{(i_1,i_2)}(a) = p_{i_1}(a_1)\cdot p_{i_2}(a_2)\,.$$

If $A=L(B)$ is a list and $\text{elems}(a)=[a_1,\dots,a_n]$ then

$$p_{[i_1,\dots,i_m]}(a) = \sum_{1\le j_1<\dots<j_m\le n} p_{i_1}(a_{j_1})\cdots p_{i_m}(a_{j_m})\,.$$

A *type annotation* for a data type A is defined to be a family

$$Q_A = (q_i)_{i\in\mathcal{I}(A)} \text{ with } q_i\in\mathbb{Q}_{\ge 0}\,.$$

An *annotated data type* is a pair (A,Q_A) of a data type A and a type annotation Q_A.

Let $a\in[\![A]\!]$ be a value of type A. Then the type annotation Q_A defines the *potential*

$$\Phi(a{:}(A,Q_A)) = \sum_{i\in\mathcal{I}(A)} q_i\cdot p_i(a)\,.$$

Prove the following statements.

a) If $p,p'\in R(A)$ then $p+p'\in R(A)$ and $p\cdot p'\in R(A)$.

b) For every $p\in R(A*A)$ there exists $p'\in R(A)$ with $p'(a)=p(a,a)$ for all $a\in[\![A]\!]$.

c) Let $a\in[\![A]\!]$ and $\ell\in[\![L(A)]\!]$ be a list. Let furthermore $k\ge 0$ and let $i_0,\dots,i_k\in\mathcal{I}(A)$ indexes for type A. Then we have

$$\begin{array}{rcl} p_{[i_0,i_1,\dots,i_k]}([]) & = & 0 \\ p_{[i_0,i_1,\dots,i_k]}(a{::}\ell) & = & p_{i_0}(a)\cdot p_{[i_1,\dots,i_k]}(\ell) + p_{[i_0,i_1,\dots,i_k]}(\ell)\,. \end{array}$$

9.3 Solving Recurrence Relations with RAML

Recall the recurrence relations from Problem 1.3.

a) $T(0) = 10$ and $T(n) = T(n-1) + 12n + 3$ for $n > 0$

b) $T(m,n) = 0$ for $n < 4$and $T(m,n) = T(m,n-4) + 12(m + \binom{m}{2})$ for $n \geq 4$

c) $T_1(0) = T_2(0) = 0$, $T_1(n) \geq T_2(n-1) + 20(n-1)$, and $T_2(n) \geq T_1(n-1) + 8(n-1)$ for $n > 0$

d) $T(0) = T(1) = 0$, $T(2) = 1$, and $T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + n - 1$ for $n > 0$

Use RAML and the built-in tick metric to automatically find solutions these recurrence relations.

Hint: The recurrence $T(0) = 10$ and $T(n) = T(n-1) + 3$ for $n > 0$ can be encoded by the following program.

```
let rec t l =
  match l with
  | [] → Raml.tick(10.0)
  | x::xs → let _ = Raml.tick(3.0) in t xs
```

$\Sigma;\Gamma \vdash^{q}_{q'} e : B$	Given resource metric M, expression e has annotated type A under signature Σ in context Γ.

$$\frac{A \xrightarrow{p/p'} B \in \Sigma(f) \qquad q = p + M^{\mathsf{app}}}{\Sigma; x{:}A \vdash^{q}_{p'} \mathrm{app}(f, x) : B}\ (\text{U:App})$$

$$\frac{\Sigma;\Gamma_1 \vdash^{p}_{p'} e_1 : A \qquad \Sigma;\Gamma_2, x{:}A \vdash^{p'}_{q'} e_2 : B \qquad q = p + M^{\mathsf{let}}}{\Sigma;\Gamma_1,\Gamma_2 \vdash^{q}_{q'} \mathrm{let}(e_1, x.e_2) : B}\ (\text{U:Let})$$

$$\frac{e \in \{\mathrm{true}, \mathrm{false}\} \qquad q = M^{\mathsf{cbool}} + q'}{\Sigma;\cdot \vdash^{q}_{q'} b : \mathrm{Bool}}\ (\text{U:Bconst})$$

$$\frac{\Sigma;\Gamma \vdash^{q_1}_{q'} e_1 : B \qquad \Sigma;\Gamma \vdash^{q_2}_{q'} e_2 : B \qquad q = M_1^{\mathsf{cond}} + q_1 \qquad q = M_2^{\mathsf{cond}} + q_2}{\Sigma;\Gamma, x : \mathrm{Bool} \vdash^{q}_{q'} \mathrm{if}(x, e_1, e_2) : B}\ (\text{U:Cond})$$

$$\frac{q = M^{\mathsf{pair}} + q'}{\Sigma; x_1 : A_1, x_2 : A_2 \vdash^{q}_{q'} \mathrm{pair}(x_1, x_2) : A_1 * A_2}\ (\text{U:Pair})$$

$$\frac{\Sigma;\Gamma, x_1{:}A_1, x_2{:}A_2 \vdash^{p}_{q'} e' : B \qquad q = M^{\mathsf{MatP}} + p}{\Sigma;\Gamma, x : A_1 * A_2 \vdash^{q}_{q'} \mathrm{matP}(e, (x_1, x_2).e') : B}\ (\text{U:MatP}) \qquad \frac{q = M^{\mathsf{nil}} + q'}{\Sigma;\cdot \vdash^{q}_{q'} \mathrm{nil} : L^{\vec{p}}(A)}\ (\text{U:Nil})$$

$$\frac{\vec{p} = (p_1, \ldots, p_k \qquad q = M^{\mathsf{cons}} + p_1 + q'}{\Sigma; x_1 : A, x_2 : L^{\lhd(\vec{p})}(A) \vdash^{q}_{q'} \mathrm{cons}(x_1, x_2) : L^{\vec{p}}(A)}\ (\text{U:Cons})$$

$$\frac{\begin{array}{c}\vec{p} = (p_1, \ldots, p_k) \qquad \Sigma;\Gamma \vdash^{q_1}_{q'} e_1 : B \\ \Sigma;\Gamma, x_1 : A, x_2 : L^{\lhd(\vec{p})}(A) \vdash^{q_2}_{q'} e_2 : B \qquad q = M_1^{\mathsf{matL}} + q_1 \qquad q + p_1 = M_2^{\mathsf{matL}} + q_2\end{array}}{\Sigma;\Gamma, x{:}L^{\vec{p}}(A) \vdash^{q}_{q'} \mathrm{matL}(x, e_1, (x_1, x_2).e_2) : B}\ (\text{U:MatL})$$

$$\frac{\Sigma;\Gamma, x_1 : A_1, x_2 : A_2 \vdash^{q}_{q'} e : B \qquad A \curlyvee (A_1, A_2)}{\Sigma;\Gamma, x : A \vdash^{q}_{q'} \mathrm{share}(x, (x_1, x_2).e) : B}\ (\text{U:Share})$$

$$\frac{\Sigma;\Gamma, x{:}A \vdash^{q}_{q'} e : B \qquad A' <: A}{\Sigma;\Gamma, x{:}A' \vdash^{q}_{q'} e : B}\ (\text{U:Supertype}) \qquad \frac{\Sigma;\Gamma \vdash^{q}_{q'} e : B \qquad B <: B'}{\Sigma;\Gamma \vdash^{q}_{q'} e : B'}\ (\text{U:Subtype})$$

$$\frac{\Sigma;\Gamma \vdash^{p}_{p'} e : B \qquad q \geq p \qquad q - p \geq q' - p'}{\Sigma;\Gamma \vdash^{q}_{q'} e : B}\ (\text{U:Relax}) \qquad \frac{\Sigma;\Gamma \vdash^{q}_{q'} e : B}{\Sigma;\Gamma, x{:}A \vdash^{q}_{q'} e : B}\ (\text{U:Weak})$$

Figure 1: Rules of the univariate polynomial resource-annotated type rules.