Resource Analysis: Lecture 2

Farzaneh Derakhshan, Tao Gu, Aditya Oak

21 June 2019

1 Resource Monoid

We define the resource monoid for the rule described in the previous lecture.

$$(q, q').(p, p') = \begin{cases} (q + p - q', p'), & \cdots \quad q' \leq p \\ (q, p' + q' - p), & \cdots \quad q' \geq p \end{cases}$$

2 Language Constructs

2.1 Type System

We consider following type system to bound the resources.

$$\begin{array}{rcll} \tau & ::= & \mathsf{unit} & \\ & | & \tau_1 \to \tau_2 & \\ & | & L(\tau) & \cdots \text{(List)} \end{array}$$

2.2 Expressions in the language :

$$\begin{array}{rcll} e & ::= & \textbf{nil}\{\tau\} & [\,] \\ & | & \textbf{cons}(e_1; e_2) & e_1 :: e_2 \\ & | & \textbf{matL}(e; e_1; x_1, x_2.e_2) & \text{match } e \text{ with } [] \leftrightarrow e_1, x_1 :: x_2 \to e_2 \\ & | & \textbf{tick}\{q\}(e) & \text{tick } q \text{ in } e \cdots (q \in \mathbb{Q}) \\ & | & \textbf{fix}\{\tau\}(x.e) & \text{fix } x : \tau \text{ as } e \end{array}$$

User inputs ticks expression to denote the number of resources consumed.

2.3 Values

$$\frac{}{\textbf{nil val}}$$

$$\frac{e_1 \textbf{ val} \qquad e_2 \textbf{ val}}{\textbf{cons}(e_1; e_2) \textbf{ val}}$$

2.4 Cost Dynamics

$$\langle e, q \rangle \;\mapsto\; \langle e', q' \rangle \qquad \cdots q, q' \geq 0$$

Reads as "with q available resources, e evaluates to e' and after evaluation, q' resources are available".

2.5 Dynamic Rules

1. $\dfrac{q - p \geq 0}{\langle \textbf{tick}\{p\}(e), q \rangle \mapsto \langle e, q - p \rangle}$

2. $\dfrac{\langle e, q \rangle \mapsto \langle e', q' \rangle}{\langle \textbf{matL}(e; e_1; x_1, x_2.e_2), q \rangle \mapsto \langle \textbf{matL}(e'; e_1; x_1, x_2.e_2), q' \rangle}$

3. $\dfrac{}{\langle \textbf{matL}(\textbf{nil}; e_1; x_1, x_2.e_2), q \rangle \mapsto \langle e_1, q \rangle}$

4. $\dfrac{e'_1 \ \textbf{val} \qquad e'_2 \ \textbf{val}}{\langle \textbf{matL}(\textbf{cons}(e'_1, e'_2); e_1; x_1 \cdot x_2 \cdot, e_2), q \rangle \mapsto \langle [e'_1/x_1, e'_2/x_2]e_2, q \rangle}$

2.6 Static Rules

1. $\dfrac{\Gamma, x\colon \tau \vdash e\colon \tau}{\Gamma \vdash \textbf{fix}\{\tau\}(x.e)\colon \tau}$

2. $\dfrac{\Gamma \vdash e\colon \tau}{\Gamma\colon \textbf{tick}\{q\}(e)\colon \tau}$

2.7 Observations

For the above calculus we can prove the following type safety theorems:

Proposition 1 (Preservation)**.** *If $e\colon \tau$ and $\langle e, q \rangle \mapsto \langle e', q' \rangle$, then $e'\colon \tau$.*

Proposition 2 (Progress)**.** *Suppose $\langle \textbf{tick}\{1\}(\langle\rangle), 0 \not\mapsto e \rangle$. If $e\colon \tau$, then there either e* **val***, or there exists q such that $\langle e, q \rangle \mapsto \langle e', q' \rangle$ for some e', q'.*

Consider the following example:

Example 3. Define *id* as

$$id = fix\ id : L(\text{unit}) \to L(\text{unit})\ as\ \lambda(x : L(\text{unit}))\ match\ x\ with\ [] \mapsto []\ y :: ys \mapsto\ tick\ 2\ in\ y :: id(ys)$$

and let v_n =<>:: $\cdots$ ::<>:: [], where <> is concatanted to itself n times. Then the question is how many resources q do we need to get $\langle id, \langle v_n \rangle, q \rangle \mapsto \langle v_n, 0 \rangle$. With some calculations, we can see that for $q = 2n$:

$$\langle id, \langle v_n \rangle, 2n \rangle \mapsto \langle v_n, 0 \rangle$$

Our next goal is to do analysis in the above example with a type system. The idea is to have types that carry potentials that has to be used to pay for ticks. We introduce type-based amortized resource analysis in the next section to acheive this goal.

3 Type-based Amortized Resource Analysis

3.1 Type System

We define types τ and context Γ as follows:

$$
\begin{array}{llll}
A, B & ::= & pot(\tau, q) & \langle \tau, q \rangle \\
\\
\tau & ::= & arr(A, B) & \tau \xrightarrow{q,q'} \tau' \quad \textit{where } A = \langle \tau, q \rangle, B = \langle \tau', q' \rangle \\
 & \mid & L(A) & L^q(\tau) \qquad\qquad \textit{where } A = \langle \tau, q \rangle \\
 & \mid & \mathsf{unit} & 1 \\
\\
\Gamma & ::= & \cdot \mid x : \tau &
\end{array}
$$

As illustrated in the following example, we have a set of annotations for each function in this system.

Example 4. In the expression

$$\lambda(x : L(\mathsf{unit}))id(id(x)) : L^4(\mathsf{unit}) \xrightarrow{0/0} L^0(\mathsf{unit}),$$

function id has two different types: $id : L^2(\mathsf{unit}) \xrightarrow{0/0} L^0(\mathsf{unit})$ and $id : L^4(\mathsf{unit}) \xrightarrow{0/0} L^2(\mathsf{unit})$.

Definition 5 (Potential)**.** For arbitrary expression $v : \tau$, its *potential* $\Phi(v : \tau)$ is defined as follows:

- $\Phi(v : \mathsf{unit}) = 0$
- $\Phi(\mathbf{lam}\{\tau\}(x : e) : A \to B) = 0$
- $\Phi(\mathbf{cons}(v_1, v_2) : L(A)) = \Phi(v_1 : A) \to \Phi(v_2 : L(A))$.
- $\Phi(v : \langle \tau, q \rangle) = q + \Phi(v, \tau)$

Example 6. The following hold according to the definition above:

- $\Phi(<>: \langle \mathsf{unit}, 10 \rangle) = 10$
- $\Phi(<>::<>:: [] : L^5(\mathsf{unit})) = 10$
- $\Phi(a_1 :: \cdots :: a_n :: [] : L^1(\tau)) = q.n + \Sigma_{1 \le i \le n} \Phi(a_i : \tau)$

3.2 Type system

The type judgements are of the form $\Gamma \vdash^{q}_{q'} e : \tau$ meaning that under context Γ with potential q, expression e has annotated type τ and potential q'. The rules for judgements are as follows:

- $$\frac{}{x : \tau \vdash_0^0 x : \tau}$$

  Note that in this rule the only thing allowed in the context is variable x, i.e. we need to consume all other things in the context. Also, context and expression potentials are restricted to 0, instead of an arbitrary q.

- $$\frac{\Gamma_1 \vdash_r^q e_1 : \tau \xrightarrow{p/q'} \tau' \quad \Gamma_2 \vdash_p^r e_2 : \tau}{\Gamma_1 \Gamma_2 \vdash_{q'}^q e_1(e_2) : \tau'}$$

  The potential in Γ_1 is q and $e_1 : \tau \xrightarrow{p/q'}$ is a function with potential r. This potential can be used in Γ_2 to evaluate e_2. The potential we have around after evaluating $e_2 : \tau$ is p and we can plug it into the function body e_1 with encoded potential p/q'. Then we are left with potential q' and $e_1(e_2) : \tau'$.

- $$\frac{\Gamma, x : \tau \vdash_{p'}^p e : \tau' \quad |\Gamma| = \Gamma}{\Gamma \vdash_0^0 \mathbf{lam}\{\tau\}(x \cdot e) : \tau \to \tau'}$$

  We write $|\tau|$ for τ in which all annotations q are replaced by 0. And $|\Gamma|$ is defined point-wise on the types. We will see by an example in the next lecture that the extra assumption $|\Gamma| = \Gamma$ assures possibility of using a function more than once.

- $$\frac{\Gamma, x : \tau \vdash_0^0 e : \tau \quad |\tau| = \tau \quad |\Gamma| = \Gamma}{\Gamma \vdash_0^0 \mathbf{fix}\{\tau\}(x \cdot e) : \tau}$$