Type Safe Tensor Combinators

Abstract

We present two higher order combinators for tensor expressions inspired by relational algebra and show how these naturally lead to a nameless representation of expressions. Next we show how to encode affine constraints into these combinators to make our language more expressive. Finally we present a polymorphic type system for shapes of tensors.

Motivation

Most array libraries (e.g. numpy, pandas) and array languages (e.g. APL, Futhark) either lack type safety, higher order operators or support for programming with array shape polymorphism (or a combination of them). We present a language that supports not only all three of these, but also exhibits a nameless representation of terms. We presume this will lead to writing clean, concise and crash free programs.

Syntax

The feature to note in our language is the higher order project operator — which can be used to reshape, resize and permute arrays.

$$
\begin{array}{lll}
L_1\ e ::= a & & \text{(Array identifier)} \\
\quad ::= e\,\overrightarrow{i} & & \text{(Index Reference)} \\
\quad ::= o\ e \ldots e & & \text{(Primitive Operation)} \\
\quad ::= proj_{\overrightarrow{p}}\, e & & \text{(Projection)} \\
\quad ::= \sum_i e & & \text{(Summation)} \\
\quad ::= \sum e & & \text{(Contraction)}
\end{array}
$$

Shape system

The role to note in particular is the PROJ rule, which dictates how the higher order projector works while maintaining safety. MINMAX captures our smallest upper bound and greatest lower bound semantics, which ensures safety is preserved while applying a collective operator.

$$\frac{\psi(a) = \overrightarrow{S}}{\psi \vdash a : \overrightarrow{S}}\ ID$$

$$\frac{\psi \vdash e : \overrightarrow{S}.s'}{\psi \vdash \sum e : \overrightarrow{S}}\ CON$$

$$\frac{\psi \vdash e : \overrightarrow{S} \quad \overrightarrow{R} : [\mathbb{N}] \quad |\overrightarrow{R}| = |\overrightarrow{S}| \quad \forall i \leq |\overrightarrow{R}|, \overrightarrow{R}_i \leq \max(\overrightarrow{S}_i)}{\psi \vdash e\overrightarrow{R} : []}\ REF$$

$$\frac{\psi \vdash e_1 : \overrightarrow{S1} \ldots e_n : \overrightarrow{Sn} \quad \alpha(o) = n \quad \forall i \leq n, |\overrightarrow{Si}| = k}{\psi \vdash o\ e_1 \ldots e_n : [(\max(\overrightarrow{S}i_1.\min), \min(\overrightarrow{S}i_1.\max)), \ldots, (\max(\overrightarrow{S}i_k.\min), \min(\overrightarrow{S}i_k.\max))]}\ MINMAX$$

For example, if $A : [(1,3),(1,5)]$ and $B : [(1,4),(1,2)]$ then $A + B : [(1,3),(1,2)]$

$$\frac{\psi \vdash e : \overrightarrow{S} \quad i \leq |\overrightarrow{S}|}{\psi \vdash \sum_i e : [\overrightarrow{S0} \ldots \overrightarrow{S(i-1)}, \overrightarrow{S(i+1)} \ldots \overrightarrow{S}_{|\overrightarrow{S}|}]}\ SUM$$

$$\frac{\psi \vdash e : \overrightarrow{S} \quad x : S(m,n) \quad G \vdash e : n}{\psi \vdash proj_x e : \overrightarrow{S'}}\ PROJ$$

where for each $1 \leq j \leq |x|$, $\overrightarrow{S'}_{x[j]} = (\overrightarrow{S}_j.\min, \overrightarrow{S}_j.\max)$ and $(-\infty, \infty)$ otherwise.

For example, if $A : [(1,4),(1,5),(1,6)]$ then $proj[4,2,3]A : [(-\infty,\infty),(1,5),(1,6),(1,4),\ldots]$

Authors:Bharathi Ramana Joshi <bharathi.joshi@research.iiit.ac.in>, Venkatesh Choppella <venkatesh.choppella@iiit.ac.in> Research Center : SERC