Tensors and n-d Arrays: A Mathematics of Arrays (MoA) and the ψ-calculus

Composition of Tensor and Array Operations

Lenore M. Mullin and James E. Raynolds

Message of This Talk

- An algebra of multi-dimensional arrays (MoA) and an index calculus (the ψ-calculus) allow a series of operations to be composed so as to minimize temporaries.
- An array, and all operations on arrays are defined using shapes, i.e. sizes of each dimension.
- Scalars are 0-dimensional arrays. Their shape is the empty vector.
- Same algebra used to specify the algorithm as well as to map to the architecture.
- Composition of multiple Kronecker Products will be discussed.

Historical Background

- ***Universal Algebra - Joseph Sylvester, late 19th Century***
- ***Matrix Mechanics - Werner Heisenberg, 1925***
  - ***Basis of Dirac's bra-ket notation***
- ***Algebra of Arrays - APL – Ken Iverson, 1957***
  - *Languages: Interpreters & Compilers*
  - *Phil Abrams: An APL Machine(1972) with Harold Stone*
    - *Indexing operations on shapes, open questions, not algebraic closed system. Furthered by Hassett and Lyon, Guibas and Wyatt. Used in Fortran.*
  - *Alan Perlis: Explored Abram's optimizations in compilers for APL. Furthered by Miller, Minter, Budd.*
  - *Susan Gerhart: Anomalies in APL algebra, can not verify correctness.*
  - *Alan Perlis with Tu: Array Calculator and Lambda Calculus 1986*

Historical Background

- ***MoA and Psi Calculus*: *Mullin (1988)***
  - o ***Full closure* on Algebra of Arrays and Index Calculus based on shapes.**
  - o *Klaus Berkling: Augmented Lambda Calculus with MoA*
    - – *Werner Kluge and Sven-Bodo Scholz: SAC*
  - o *Built prototype compilers: output C, F77, F90, HPF*
  - o *Modified Portland Groups HPF, HPF Research Partner*
  - o *Introduced Theory to Functional language Community*
    - – *Bird-Meertens, SAC, …*
    - – *Applied to Hardware Design and Verification*
      - • *Pottinger (ASICS), IBM (Patent, Sparse Arrays), Savaria (Hierarchical Bus Parallel Machines)*
  - o *Introduce Theory to OO Community (Sabbatical MIT Lincoln Laboratory).*
    - – *Expression Templates and C++ Optimizations for Scientific Libraries*

Tensors and Language Issues

- *Minimizing materialization* of array valued temporaries…

$$D = ((A + B) \otimes C)^T$$

  where A, B, C, and D are huge 3-d (or higher dimensional) arrays
- How to generally map to processor/memory hierarchies
- Verification of both semantics and operation
- Equivalence of programs
- No language today has an array algebra and index calculus without problems with boundary conditions.

Notation

- **Dirac:**
  - Inner product: $\langle a \mid b \rangle$
  - Outer product: $|a\rangle\langle b|$
  - Tensor-vector multiply: $\left(|a\rangle\langle b|\right)|c\rangle = |a\rangle\left(\langle b|c\rangle\right)$
- Dirac notation unified Linear Algebra and Hilbert Space: ***Tensors are ubiquitous across many disciplines: Quantum and Classical***

Tensors and Multi-Particle States

"The Hilbert space describing the n-particle system is that spanned by all n-th rank tensors of the form:

$$|\psi> = |\psi_1> |\psi_2> \ldots |\psi_n>$$

The zero-particle states (i.e. n = 0) are tensors of rank zero, that is, scalars (complex numbers)."

Quote: Richard Feynman, "Statistical Mechanics" pp. 168-170

Manipulation of an Array

- Given a 3 by 5 by 4 array:

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 \end{bmatrix}, \begin{bmatrix} 20 & 21 & 22 & 23 \\ 24 & 25 & 26 & 27 \\ 28 & 29 & 30 & 31 \\ 32 & 33 & 34 & 35 \\ 36 & 37 & 38 & 39 \end{bmatrix}, \begin{bmatrix} 40 & 41 & 42 & 43 \\ 44 & 45 & 46 & 47 \\ 48 & 49 & 50 & 51 \\ 52 & 53 & 54 & 55 \\ 56 & 57 & 58 & 59 \end{bmatrix}$$

- Shape vector: $\rho A = < 3\ 5\ 4 >$
- Index vector: $\vec{i} = < 2\ 1\ 3 >$
- Used to select: $\vec{i}\ \psi A = < 2\ 1\ 3 > \psi A = 47$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } B = \begin{bmatrix} 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

The Kronecker Product: $A \bigotimes B$ Is defined by:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \bigotimes \begin{bmatrix} 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} =$$

$$\begin{bmatrix} 1\times 5 & 1\times 6 & 1\times 7 & 1\times 8 & 2\times 5 & 2\times 6 & 2\times 7 & 2\times 8 \\ 1\times 9 & 1\times 10 & 1\times 11 & 1\times 12 & 2\times 9 & 2\times 10 & 2\times 11 & 2\times 12 \\ 1\times 13 & 1\times 14 & 1\times 15 & 1\times 16 & 2\times 13 & 2\times 14 & 2\times 15 & 2\times 16 \\ 3\times 5 & 3\times 6 & 3\times 7 & 3\times 8 & 4\times 5 & 4\times 6 & 4\times 7 & 4\times 8 \\ 3\times 9 & 3\times 10 & 3\times 11 & 3\times 12 & 4\times 9 & 4\times 10 & 4\times 11 & 4\times 12 \\ 3\times 13 & 3\times 14 & 3\times 15 & 3\times 16 & 4\times 13 & 4\times 14 & 4\times 15 & 4\times 16 \end{bmatrix}$$

Kronecker Product of A and B

Kronecker Product of A and B

$$(A \otimes B)_{I,J} = A_{i,j} B_{l,m}$$

where

$$I \equiv < i \ l > \qquad J \equiv < j \ m >$$

MoA would write:

$$< i \ j \ l \ m > \psi \, (A \ op_x \ B) = (< i \ j > \psi A) \times (< l \ m > \psi B)$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} op_{\times} \begin{bmatrix} 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} =$$

$$\left[\left[\begin{bmatrix} 1\times 5 & 1\times 6 & 1\times 7 & 1\times 8 \\ 1\times 9 & 1\times 10 & 1\times 11 & 1\times 12 \\ 1\times 13 & 1\times 14 & 1\times 15 & 1\times 16 \\ 3\times 5 & 3\times 6 & 3\times 7 & 3\times 8 \\ 3\times 9 & 3\times 10 & 3\times 11 & 3\times 12 \\ 3\times 13 & 3\times 14 & 3\times 15 & 3\times 16 \end{bmatrix} \begin{bmatrix} 2\times 5 & 2\times 6 & 2\times 7 & 2\times 8 \\ 2\times 9 & 2\times 10 & 2\times 11 & 2\times 12 \\ 2\times 13 & 2\times 14 & 2\times 15 & 2\times 16 \\ 4\times 5 & 4\times 6 & 4\times 7 & 4\times 8 \\ 4\times 9 & 4\times 10 & 4\times 11 & 4\times 12 \\ 4\times 13 & 4\times 14 & 4\times 15 & 4\times 16 \end{bmatrix} \right] \right]$$

MoA Outer Product of A and B

$$\langle\, 1\times 5 \quad 1\times 6 \quad 1\times 7 \quad 1\times 8 \quad 2\times 5 \quad 2\times 6 \quad 2\times 7 \quad 2\times 8 \;\ldots \rangle$$

Kronecker Product flattened using row-major layout

$$\langle\, 1\times 5 \quad 1\times 6 \quad 1\times 7 \quad 1\times 8 \quad 1\times 9 \quad 1\times 10 \quad 1\times 11 \quad 1\times 12 \;\ldots \rangle$$

MoA Outer Product flattened using row-major layout

Multiple Kronecker Products

$$A \otimes (B \otimes C)$$

Standard approach suffers from:

- Large temporaries: $TEMP = B \otimes C$; $A \otimes TEMP$
- Complicated index calculations
- Processor/memory optimizations difficult

Shapes and the Outer Product

Definition 1 *Given A, B, C and D are n by n arrays. That is, their shape, i.e.*

$$\rho A = \rho B = \rho C = \rho D = < \; n \, n \; > .$$

Assume the existence of the ψ operator and that it is well defined for n-dimensional arrays. The ψ operator takes as left argument an index vector and an array as the right argument and returns the corresponding component of the array. For

Note: The general definition takes an array of indicies as its left argument.

$$D = A\ op_{\times}(B\ op_{\times}C)$$

is defined when the shape of D *is equal to the shape of* $A\ op_{\times}(B\ op_{\times}C)$. *And the shape of* $A\ op_{\times}(B\ op_{\times}C)$ *is equal to the shape of* A *concatenated to the shape of* $(B\ op_{\times}C)$ *which is equivalent to the shape of* A *concatenated to the shape of* B *concatenated to the shape of* C. *i.e.*

$$\rho D = \rho(A\ op_{\times}(B\ op_{\times}C)) = \rho A +\!\!\!+\, \rho(B\ op_{\times}C) = \rho A +\!\!\!+\, \rho B +\!\!\!+\, \rho C = < n\ n\ n\ n\ n\ n >$$

Then, $\forall\ i_0, j_0, k_0, l_0, m_0.n_0$ *s.t.*
$0 \leq i_0 < n;\ 0 \leq j_0 < n;\ 0 \leq k_0 < n;\ 0 \leq l_0 < n;\ 0 \leq m_0 < n;\ 0 \leq n_0 < n$

$$\begin{aligned} < i_0\ j_0\ k_0\ l_0\ m_0\ n_0 > \psi\ D &= (< i_0\ j_0 > \psi\ A) \times (< k_0\ l_0\ m_0\ n_0 > \psi\ (B\ op_{\times}C)) \\ &= (< i_0\ j_0 > \psi\ A) \times (< k_0\ l_0 > \psi\ B) \times (< m_0\ n_0 > \psi\ C) \end{aligned}$$

Multiple Kronecker Products

- We want: $$E = (A \otimes B) \otimes A$$

with: $$C = A \otimes B$$

The input arrays A and B are

$$A = \begin{matrix} 0 & 1 \\ 2 & 3 \end{matrix}$$

and

$$B = \begin{matrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{matrix}$$

Multiple Kronecker Products

- We want: $E = (A \otimes B) \otimes A$

with: $C = A \otimes B$

Shape of C is < 6 6 > because we are combining a < 2 2 > array with a < 3 3 >

Typo: +ed instead of Xed

$$C = \begin{matrix} 0 & 1 & 2 & 1 & 2 & 3 \\ 3 & 4 & 5 & 4 & 5 & 6 \\ 6 & 7 & 8 & 7 & 8 & 9 \\ 2 & 3 & 4 & 3 & 4 & 5 \\ 5 & 6 & 7 & 6 & 7 & 8 \\ 8 & 9 & 10 & 9 & 10 & 11 \end{matrix}$$

Note the use of the generalized binary operation + rather than * (times) in this “product”

$$E = C \otimes A$$

$$E = \begin{matrix} 0 & 1 & 2 & 1 & 2 & 3 \\ 3 & 4 & 5 & 4 & 5 & 6 \\ 6 & 7 & 8 & 7 & 8 & 9 \\ 2 & 3 & 4 & 3 & 4 & 5 \\ 5 & 6 & 7 & 6 & 7 & 8 \\ 8 & 9 & 10 & 9 & 10 & 11 \end{matrix} \quad \otimes A$$

$$E = \begin{matrix} 0 & 1 & 1 & 2 & 2 & 3 & 1 & 2 & 2 & 3 & 3 & 4 \\ 2 & 3 & 3 & 4 & 4 & 5 & 3 & 4 & 4 & 5 & 5 & 6 \\ 3 & 4 & 4 & 5 & 5 & 6 & 4 & 5 & 5 & 6 & 6 & 7 \\ 5 & 6 & 6 & 7 & 7 & 8 & 6 & 7 & 7 & 8 & 8 & 9 \\ 6 & 7 & 7 & 8 & 8 & 9 & 7 & 8 & 8 & 9 & 9 & 10 \\ 8 & 9 & 9 & 10 & 10 & 11 & 9 & 10 & 10 & 11 & 11 & 12 \\ 2 & 3 & 3 & 4 & 4 & 5 & 3 & 4 & 4 & 5 & 5 & 6 \\ 4 & 5 & 5 & 6 & 6 & 7 & 5 & 6 & 6 & 7 & 7 & 8 \\ 5 & 6 & 6 & 7 & 7 & 8 & 6 & 7 & 7 & 8 & 8 & 9 \\ 7 & 8 & 8 & 9 & 9 & 10 & 8 & 9 & 9 & 10 & 10 & 11 \\ 8 & 9 & 9 & 10 & 10 & 11 & 9 & 10 & 10 & 11 & 11 & 12 \\ 10 & 11 & 11 & 12 & 12 & 13 & 11 & 12 & 12 & 13 & 13 & 14 \end{matrix}$$

Multiple Outer Products

$$C = A \, op_x \; B$$

$$C = \begin{array}{ccc} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \\ \\ 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ \\ 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \\ \\ 3 & 4 & 5 \\ 6 & 7 & 8 \\ 9 & 10 & 11 \end{array}$$

Typo: +ed instead of Xed

Multiple Outer Products

Notice the shape. The shape is <2 2 3 3>.
From this shape we can easily index each 3 by 3.
<0 0> ψ gets the first one.
Notice how easily we can use these indicies
to map to processors. Now perform the outer product of
C with A. Now the shape is <2 2 3 3 2 2>.

With the shape we can perform index compositions.

The indices are composed as follows: Given $0 \leq^* \; i, j <^* 2, 2$; $0 \leq^* k, l <^* 3, 3$; and $0 \leq^* m, n <^* 2, 2$ and for all $0 \leq^* i, j, k, l, m, n <^* 2, 2, 3, 3, 2, 2$;

$$
\begin{aligned}
< i\ j\ k\ l\ m\ n > \psi\ ((A\ op_\times\ B)\ op_\times\ A) &= (< i\ j\ k\ l > \psi\ (A\ op_\times\ B)) \times (< m\ n > \psi A) \\
&= (< i\ j > \psi A)\ \times (< k\ l > \psi B)\ \times (< m\ n > \psi A)
\end{aligned}
$$

- This is the *Denotational Normal Form* (DNF) expressed in terms of Cartesian coordinates.

- Convert to the Operational Normal Form (ONF) expressed in terms of *start, stop and stride*, *the ideal machine abstraction*

- Break up over 4 processors…need to restructure the array shape from $\langle 2\ 2\ 3\ 3\ 2\ 2 \rangle$ to $\langle 4\ 3\ 3\ 2\ 2 \rangle$

- Thus for $0 \leq p < 4$

$$\begin{aligned} <i\ j\ k\ l\ m\ n>\psi\ ((A\ op_{\times}\ B)\ op_{\times}\ A) &= \ <p\ k\ l\ m\ n>\psi\ (A\ op_{\times}\ B)\ op_{\times}\ A \\ &= ((<p>\psi\vec{a})\ \times (<k\ l>\psi B))\ \times (<m\ n>\psi A) \end{aligned}$$

$$\forall\ p, q, r\ \ s.t.\ 0 \leq p < 4\ ;\ 0 \leq r < 9;\ \ 0 \leq s < 4$$

```
( avec[p] x bvec[q ]) x avec[r]
```

Conclusions

- We’ve discussed how an algebra can describe an algorithm, its decomposition, and its mapping to processors.
- We’ve discussed how to compose array operations
- We’ve built prototype compilers and hardware
- What is next? How can the applications drive the research?
- How can the research drive the funding?
- How do we continue to have fun either way?