Graph algorithms are in wide use in DoD software applications, including intelligence analysis, autonomous systems, cyber intelligence and security, and logistics optimizations. However, graph algorithms are difficult and costly to implement efficiently on hardware systems. As the size of graphs and the pace at which new hardware is being developed increase, the complexity of developing high performance graph libraries becomes a prohibitive barrier to the work of analyzing the deluge of information.

Currently deep expertise is needed in graph algorithms and hardware tuning to achieve good performance on targeted hardware. It is rare to find this in individuals or even on teams within one organization.

The GraphBLAS Forum – a government, academic and industry consortium – has defined a set of graph primitive objects and operations and is nearing completion of the C Application Programming Interface (API) specification that is able to separate the concerns between:

- the graph expertise needed to develop advanced graph analytics (writing code using the API) and
- the hardware expertise is needed to achieve high levels of performance (implementing efficient versions of the API for specific hardware).

For more information on the GraphBLAS Forum: http://graphblas.org

---

**Operation** | **Description** | **Mathematical Description**
---|---|---
mxm, mxv, vxm | Perform matrix multiplication (e.g., breadth-first traversal, shortest paths) | $C(\sim M) \oplus A^T \odot B^T$
| | | $c_i \oplus = a_{ij} \odot b_{ij}$
eWiseAdd, eWiseMult | Element-wise addition and multiplication of matrices (e.g., graph union, intersection) | $C(\sim M) \oplus A^T \odot B^T$
| | | $c_i \oplus = a_{ij} \odot b_{ij}$
extract | Extract a sub-matrix from a larger matrix (e.g., sub-graph selection) | $C(\sim M) \oplus A(i,j)$
assign | Assign to a sub-matrix of a larger matrix (e.g., sub-graph assignment) | $C(\sim M) (i,j) \oplus = A^T$
apply | Apply unary function to each element of matrix (e.g., edge weight modification) | $C(\sim M) \oplus = f(A^T)$
reduce | Reduce along columns or rows of matrices (vertex degree) | $c(\sim m) \oplus = \bigoplus_i A(\cdot,j)$
transpose | Swaps the rows and columns of a sparse matrix (e.g., reverse directed edges) | $C(\sim M) \oplus = A^T$
buildMatrix | Build an matrix representation from row, column, value tuples | $C(\sim M) \oplus = \bigoplus(i,j,v)A(\sim M)$
extractTuples | Extract the row, column, value tuples from a matrix representation | $(i,j,v) = A(\sim M)$

The table above lists all of the primitive operations supported by the GraphBLAS API along with their mathematical description. These mathematical “requirements” are being captured in a C API Specification as shown for matrix multiplication (mxm) below.

```c
GrB_Info GrB_mxm(GrB_Matrix *C,
    const GrB_Matrix Mask,
    const GrB_BinaryFunction accum,
    const GrB_Semiring op,
    const GrB_Matrix A,
    const GrB_Matrix B,
    [const Descriptor desc]);
```