Options for a consistent GraphBLAS notation

Tim Davis, Gabor Szarnyas

September 4, 2020

1 Creating matrices

in the C API: `GrB_Matrix_new (&A, m, n, GrB_INT64)`

in ASCII: `A = int16(m,n)` (I'm open to suggestions for this)

in LaTeX: let: $\mathbf{A} \in \mathbb{Z}_{16}^{m \times n}$

2 Mask computation

In the following table the "mask" column is the value of the mask after applying the value/structural option and optionally complementing the mask. If no mask is present (and not complemented) then mask is 1. If no mask is present (yet complemented) then the mask is always 0; this has no effect unless the Replace option is also used.

The repl column in the table is "yes" if Replace is enabled, "no" otherwise. The accum column in the table is "yes" if the accum is present, "no" otherwise. The **C** column is listed as c_{ij} if that entry is present, and a dash otherwise, likewise for the **A** column.

repl	accum	$\mathbf{C}$	$\mathbf{A}$	mask	action taken by $\mathbf{C}\langle\mathbf{M}\rangle = \mathbf{C} \odot \mathbf{A}$
					`C<M>=A` and `C<!M>=A`
-	-	c_{ij}	a_{ij}	1	$c_{ij} = a_{ij}$, update
-	-	-	a_{ij}	1	$c_{ij} = a_{ij}$, insert
-	-	c_{ij}	-	1	delete c_{ij} because a_{ij} not present
-	-	-	-	1	
-	-	c_{ij}	a_{ij}	0	
-	-	-	a_{ij}	0	
-	-	c_{ij}	-	0	
-	-	-	-	0	
					`C<M,repl>=A` and `C<!M,repl>=A`
yes	-	c_{ij}	a_{ij}	1	$c_{ij} = a_{ij}$, update
yes	-	-	a_{ij}	1	$c_{ij} = a_{ij}$, insert
yes	-	c_{ij}	-	1	delete c_{ij} because a_{ij} not present
yes	-	-	-	1	
yes	-	c_{ij}	a_{ij}	0	delete c_{ij} (because of `GrB_REPLACE`)
yes	-	-	a_{ij}	0	
yes	-	c_{ij}	-	0	delete c_{ij} (because of `GrB_REPLACE`)
yes	-	-	-	0	
					`C<M>+=A` and `C<!M>+=A`
-	yes	c_{ij}	a_{ij}	1	$c_{ij} = c_{ij} \odot a_{ij}$, apply accumulator
-	yes	-	a_{ij}	1	$c_{ij} = a_{ij}$, insert
-	yes	c_{ij}	-	1	
-	yes	-	-	1	
-	yes	c_{ij}	a_{ij}	0	
-	yes	-	a_{ij}	0	
-	yes	c_{ij}	-	0	
-	yes	-	-	0	
					`C<M,repl>+=A` and `C<!M,repl>+=A`
yes	yes	c_{ij}	a_{ij}	1	$c_{ij} = c_{ij} \odot a_{ij}$, apply accumulator
yes	yes	-	a_{ij}	1	$c_{ij} = a_{ij}$, insert
yes	yes	c_{ij}	-	1	
yes	yes	-	-	1	
yes	yes	c_{ij}	a_{ij}	0	delete c_{ij} (because of `GrB_REPLACE`)
yes	yes	-	a_{ij}	0	
yes	yes	c_{ij}	-	0	delete c_{ij} (because of `GrB_REPLACE`)
yes	yes	-	-	0	

Table 1: Results of the mask/accumulator phase

3 For Mask/Accum/Replace: there are 32 variants

All these options can affect each other so there are 32 variants total, except that when no mask is present, the value/structural option makes no difference.

- the mask **M** can be present, or not present.
- the mask can be complemented, or not (even when no mask is present)
- the mask can be valued or structural
- the replace option can be enabled, or not.
- the accumulator can be present, or not.

Here are some options:

- plain mask, with accum:

  in ASCII: `C<M>+=A`
  in LaTex: $\mathbf{C}\langle\mathbf{M}\rangle\odot= \mathbf{A}$

- negated mask, with accum. Several options for LaTeX. I like the $\overline{\mathbf{M}}$ since it is more compact, but it has problems when **M** is also structural (see below).

  in ASCII: `C<!M>+=A`
  in LaTex: $\mathbf{C}\langle\neg\mathbf{M}\rangle\odot= \mathbf{A}$
  in LaTex: $\mathbf{C}\langle\overline{\mathbf{M}}\rangle\odot= \mathbf{A}$

- structural mask, not complemented: using $s(\mathbf{M})$ where s stands for structural. This feels wordy to me. Not elegant. The letter s could be confused for a vector or scalar.

  in ASCII: `C<s(M)>+=A`
  in LaTex: $\mathbf{C}\langle s(\mathbf{M})\rangle\odot= \mathbf{A}$

- structural mask alternatives. The letter s could be confused for a scalar or vector, so a Greek letter could be used instead, such as sigma (σ). But this is also wordy.

in ASCII: `C<s(M)>+=A`

in LaTex: $\mathbf{C}\langle\sigma(\mathbf{M})\rangle\odot= \mathbf{A}$

- structural mask alternative. The notation $s(\mathbf{M})$ is a little wordy. The Mask cannot be transposed in a single call to `GrB_anything` so a superscript would work. But a subscript might be better, because someone might want to indicate a transposed mask (which would have to be done as a separate call to `GrB_transpose`). This doesn't look good in ASCII however:

  in ASCII: `C<M_s>+=A`

  in LaTex: $\mathbf{C}\langle\mathbf{M}_s\rangle\odot= \mathbf{A}$

- structural mask: why not use set notation? The structure of $\mathbf{M}$ is a set of the entries of $\mathbf{M}$. In a sense, selecting the structural option in the descriptor converts $\mathbf{M}$ into a set. So how about set notation, with curly brackets? We don't use curly brackets anywhere else in GraphBLAS, so this is unambigous. I like this the best.

  in ASCII: `C<{M}>+=A`

  in LaTex: $\mathbf{C}\langle\{\mathbf{M}\}\rangle\odot= \mathbf{A}$

- complemented structural mask, for $s(\mathbf{M})$:

  in ASCII: `C<!s(M)>+=A`

  in LaTex: $\mathbf{C}\langle\neg s(\mathbf{M})\rangle\odot= \mathbf{A}$

- complemented structural mask, for the curly bracket option: It is important to note that the structure of $\mathbf{M}$ is taken first, and then complemented. Not the other way around. I like this option the best (with $\neg$).

  ASCII: `C<!{M}>+=A`

  LaTeX: $\mathbf{C}\langle\neg\{\mathbf{M}\}\rangle\odot= \mathbf{A}$

  LaTeX: $\mathbf{C}\langle\overline{\{\mathbf{M}\}}\rangle\odot= \mathbf{A}$ ouch. That is a long overline, and could be confused with $\mathbf{C}\langle\{\overline{\mathbf{M}}\}\rangle\odot= \mathbf{A}$

(Gabor) I like the set notation for the "structure" option and the $\neg$ symbol for negation.

4 Replace option

The examples consider all options except for Replace. Here are some ideas; I am not really happy with any of them.

- simple case: a mask with replace

  in ASCII: `C<M,replace>+=A`

  in LaTex: $\mathbf{C}\langle\mathbf{M}, \text{replace}\rangle\odot= \mathbf{A}$

- complemented structural mask with replace

  ASCII: `C<!{M},replace>+=A`

  LaTeX: $\mathbf{C}\langle\neg\{\mathbf{M}\}, \text{replace}\rangle\odot= \mathbf{A}$

- no mask, yet complemented, with replace. This does actual work that does not depend on the input matrix **A** or input scalar.

  ASCII: `C<!,replace>=A`

  LaTeX: $\mathbf{C}\langle\neg, \text{replace}\rangle\odot= \mathbf{A}$

4.1 (Gabor) Alternative

How about double angle brackets for "replace"? This could symbolize that the computation only keeps that part by the mask and discards the rest.

- simple case: a mask with replace

  in ASCII: `C<<M>>+=A`

  in LaTex: $\mathbf{C}\langle\!\langle\mathbf{M}\rangle\!\rangle\odot= \mathbf{A}$

- complemented structural mask with replace

  ASCII: `C<<!{M}>>+=A`

  LaTeX: $\mathbf{C}\langle\!\langle\neg\{\mathbf{M}\}\rangle\!\rangle\odot= \mathbf{A}$

- no mask, yet complemented, with replace

  ASCII: `C<<!>>=A`

  LaTeX: $\mathbf{C}\langle\!\langle\neg\rangle\!\rangle\odot= \mathbf{A}$

5 Different accum operators

The accum operator can be any function, such as PLUS, TIMES, FIRST, MIN, MAX, ANY, logical OR, logical AND etc. I propose using using symbols from C/C++ if available, or spelling out the function otherwise. Assuming no mask/replace/complement, in ASCII:

`C +=A` plus

`C *= A` times

`C &&= A` logical AND

`C &= A` bitwise AND

`C ||= A` logical OR

`C |= A` bitwise OR

`C ^= A` logical XOR, but see the XOR in LaTeX below

`C /= A` divide

`C max= A` maximum

`C any= A` , this is Tim's `GxB_ANY_*` operator.

`C first= A`

and so on

In LaTeX, it is very similar. Note the spacing:

$\mathbf{C} \mathrel{+}= \mathbf{A}$, plus

Latex source of the above: `$\bf C \,+\!\!= A$`

$\mathbf{C} \mathrel{*}= \mathbf{A}$, times

$\mathbf{C} \mathrel{\wedge}= \mathbf{A}$, logical AND

$\mathbf{C} \mathrel{\&}= \mathbf{A}$, bitwise AND

$\mathbf{C} \mathrel{\vee}= \mathbf{A}$, logical OR

$\mathbf{C} \mathrel{|}= \mathbf{A}$, bitwise OR

$\mathbf{C} \oplus= \mathbf{A}$, logical XOR? but this conflicts with the use of $\oplus$ to denote a generic additive operator. So perhaps logical XOR should be written $\mathbf{C}\,\mathrm{xor}= \mathbf{A}$.

$\mathbf{C} /= \mathbf{A}$, divide, or perhaps $\mathbf{C} \div= \mathbf{A}$

$\mathbf{C}\,\mathrm{max}= \mathbf{A}$, maximum

$\mathbf{C}\,\mathrm{any}= \mathbf{A}$, ANY

$\mathbf{C}\,\mathrm{first}= \mathbf{A}$, FIRST

and so on

(Gabor) I propose to use `\mathbin{}` for spacing around the operator, e.g., $\mathbf{C} \mathbin{+}= \mathbf{A}$.