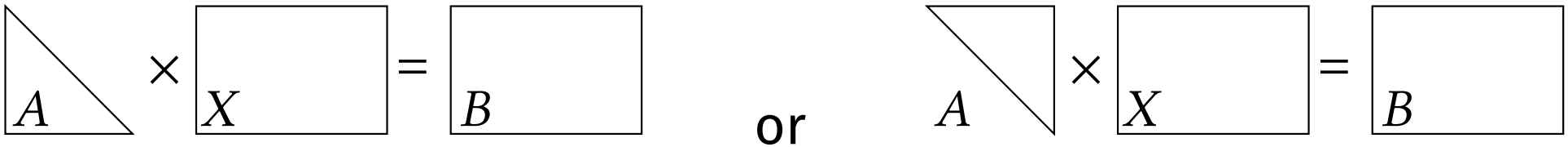
Lecture 4: OpenMP – Homework

- Use OpenMP (not MPI) to implement `xGEMM` in parallel and compare with sequential implementation
  - No need to implement all variants, assume (as an eample) `xGEMM`("No transpose", "No transpose", M, N, K, $\alpha = -1$, A, B, $\beta = 1$, C)
  - You can make the dimensions "convenient"
    - Divisible by number of threads, cores, ...
    - Divisible by 100 or your other favorite integers
  - Make sure though that the matrix dimensions can grow so you can test weak scaling (larger size for more threads).
- **Alternative**: Use OpenMP (not MPI) to implement `xTRSM` in parallel
  - Simplify accordingly in a similar way as you did for `xGEMM`.
- Comment on performance and ease of implementation in comparison to each other and the `pthread` implementation
  - Things to keep in mind is scalability, productivity, potential for concurrency bugs.

Details on TRSM

- TRSM = **TR**iangular **S**olve with **M**atrix
- Reference implementation: http://netlib.org/blas/dtrsm.f
  - If you don't know where to start then you can select one of the loop nests in the reference implementation and focus on parallelizing the selected code.
  - To simplify parallelization you should assume that the if statements are predetermined. That's why I allow you to select particular values for SIDE, UPLO TRANSA, and DIAG input parameters.
- This is a pictorial representation of what the TRSM routine does



- The TRSM routine solves $AX = B$ system where A is either upper or lower triangular. The solution matrix X is returned in the space occupied by B.

Details on the Submission

- Submission file is `first_last_hw2.tar.gz` and its content is:
  - `first_last_hw2/first_last_hw2.pdf`
  - `first_last_hw2/omp_dgemm.c` (`DGEMM` with OpenMP)
  - `first_last_hw2/omp_dtrsm.c` (`DTRSM` with OpenMP)
  - `first_last_hw2/seq_dgemm.c` (sequential `DGEMM`)
  - `first_last_hw2/seq_dtrsm.c` (sequential `DTRSM`)
  - `first_last_hw2/other_file.c` (if needed)
  - `first_last_hw2/Makefile` (if needed)
- If you're using C++ your source files should have extension "`cpp`"; Fortran files: "`f`"
- Do not include code in the PDF file.
- Ideas for writing (you don't need to exceed three pages, a graph per page):
  - What worked? What didn't work?
  - Any difference between `xGEMM` and `xTRSM`? Why or why not?
  - If you had more time, what would you do differently?