Automatic Generation of PMEs for Matrix Operations

Diego Fabregat-Traver and Paolo Bientinesi

AICES, RWTH Aachen
fabregat@aices.rwth-aachen.de

Automated Computing - ICNAAM 2010
23rd September
Rhodos, Greece

Objective: generation of algorithms

$LU = A \quad \longrightarrow$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$
where A_{TL} is 0×0

While $n(A_{TL}) < n(A)$ **do**

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

$$\begin{aligned} A_{01} &= L_{00}^{-1} A_{01} \\ A_{10} &= A_{10} U_{01}^{-1} \\ A_{11} &= LU(A_{11} - A_{10} A_{01}) \end{aligned}$$

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

endwhile

Objective: generation of algorithms

RWTH AACHEN UNIVERSITY

$$LU = A \quad \longrightarrow$$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$
where A_{TL} is 0×0

While $n(A_{TL}) < n(A)$ **do**

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

$$\begin{aligned} A_{01} &= L_{00}^{-1} A_{01} \\ A_{10} &= A_{10} U_{01}^{-1} \\ A_{11} &= LU(A_{11} - A_{10} A_{01}) \end{aligned}$$

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

endwhile

- We aim at loop-based algorithms

AI ces

$$LU = A \quad \longrightarrow$$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$
where A_{TL} is 0×0

While $n(A_{TL}) < n(A)$ **do**

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

$$\begin{aligned} A_{01} &= L_{00}^{-1} A_{01} \\ A_{10} &= A_{10} U_{01}^{-1} \\ A_{11} &= LU(A_{11} - A_{10} A_{01}) \end{aligned}$$

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

endwhile

- We aim at loop-based algorithms
- Correct by construction

Objective: generation of algorithms

$$LU = A \quad \longrightarrow$$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$

where A_{TL} is 0×0

While $n(A_{TL}) < n(A)$ **do**

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

$$\begin{aligned} A_{01} &= L_{00}^{-1} A_{01} \\ A_{10} &= A_{10} U_{01}^{-1} \\ A_{11} &= LU(A_{11} - A_{10} A_{01}) \end{aligned}$$

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

endwhile

- We aim at loop-based algorithms
- Correct by construction
- Loop invariants needed beforehand

Objective: generation of algorithms

$$LU = A \quad \longrightarrow$$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$

where A_{TL} is 0×0

While $n(A_{TL}) < n(A)$ **do**

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

$$\begin{aligned} A_{01} &= L_{00}^{-1} A_{01} \\ A_{10} &= A_{10} U_{01}^{-1} \\ A_{11} &= LU(A_{11} - A_{10} A_{01}) \end{aligned}$$

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

endwhile

- We aim at loop-based algorithms
- Correct by construction
- Loop invariants needed beforehand
- Loop invariants come from the PME

Cholesky Factorization

$$LL^T = A$$

Cholesky Factorization

$$LL^T = A$$

or in explicit form:

$$L := \Gamma(A)$$

Cholesky Factorization

$$LL^T = A$$

or in explicit form:

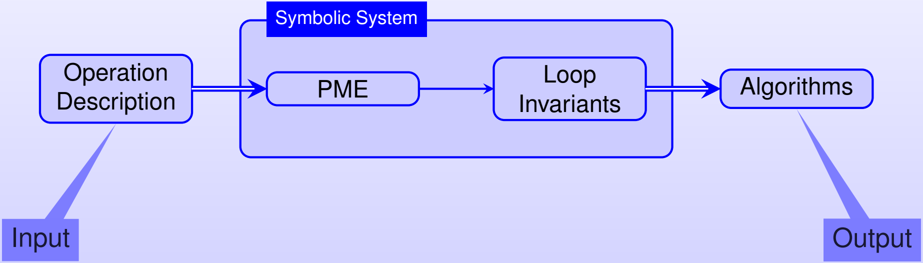
$$L := \Gamma(A)$$

Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \star \\ \hline L_{BL} = A_{BL} L_{TL}^{-T} & L_{BR} = \Gamma(A_{BR} - L_{BL} L_{BL}^T) \end{array}\right)$$

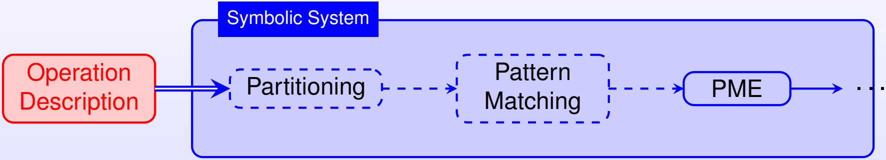
Automation

Mechanical process: can be automated



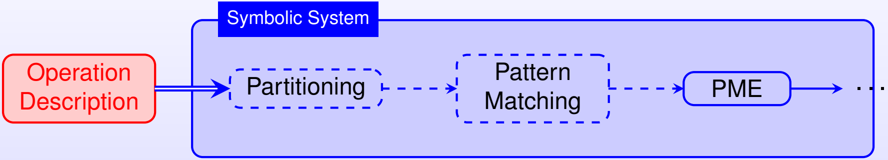
Outline

An example: LU Factorization

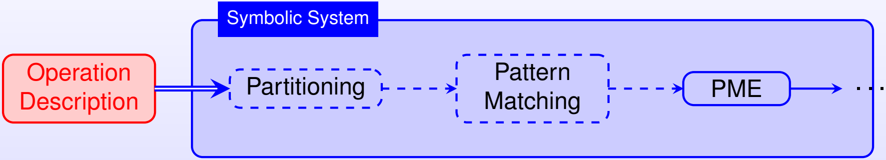


$$LU = A \text{ ?}$$

An example: LU Factorization



$$XY = Z \text{ ?}$$

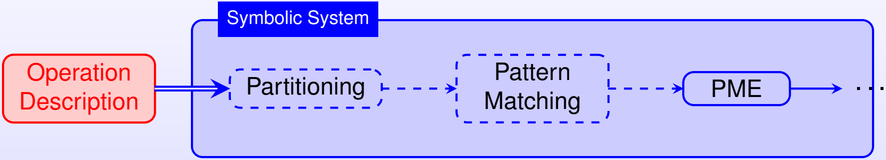


$XY = Z$?

$Z \leftarrow XY$

Operation Description

An example: LU Factorization

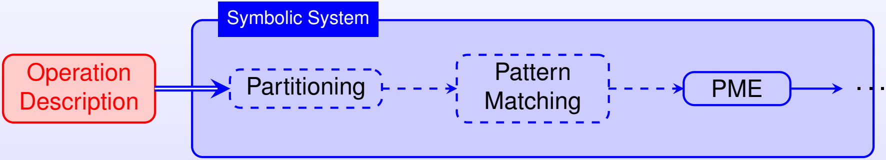


$$XY = Z\ ?$$

$$Z \leftarrow XY \qquad X \leftarrow ZY^{-1} \qquad Y \leftarrow X^{-1}Z$$

Operation Description

An example: LU Factorization

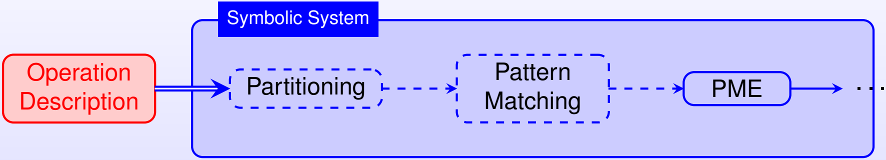


$$XY = Z \text{ ?}$$

$$Z \leftarrow XY \qquad X \leftarrow ZY^{-1} \qquad Y \leftarrow X^{-1}Z \qquad XY \leftarrow Z$$

Operation Description

An example: LU Factorization



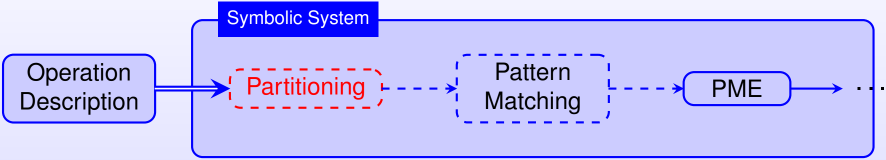
$$LU = A$$

$$\Downarrow$$

$$f: \quad \{L, U\} := LU(A) \quad \equiv \left\{ \begin{array}{ll} f_{\text{Pre}}: & \{Input(A) \ \wedge ExistsLU(A) \wedge \\ & Output(L) \ \wedge LowTriUnit(L) \wedge \\ & Output(U) \ \wedge UppTri(U)\} \\ f_{\text{Post}}: & \{LU = A\} \end{array} \right.$$

RWTH AACHEN UNIVERSITY

An example: LU Factorization



$$L \rightarrow (L)$$

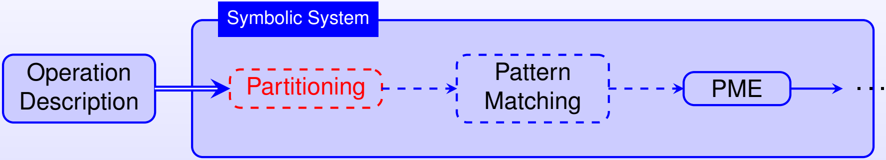
$$L \rightarrow \left(\begin{array}{c|c} L_L & L_R \end{array} \right)$$

$$L \rightarrow \left(\begin{array}{c} L_T \\ \hline L_B \end{array} \right)$$

$$L \rightarrow \left(\begin{array}{c|c} L_{TL} & L_{TR} \\ \hline L_{BL} & L_{BR} \end{array} \right)$$

Partitioning

An example: LU Factorization

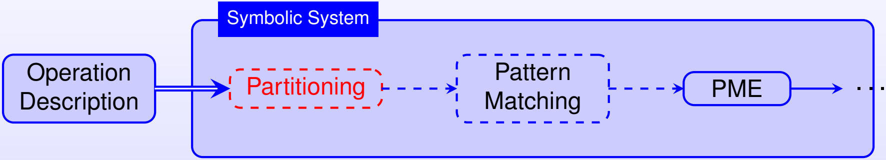


$$L \rightarrow (L) \quad \text{where } L \text{ is lower triangular}$$

$$L \rightarrow \left(L_L \,\middle|\, L_R \right)$$

$$L \rightarrow \left(\frac{L_T}{L_B} \right)$$

$$L \rightarrow \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array} \right) \quad \text{where } L_{TL} \text{ \& } L_{BR} \text{ are lower triangular}$$



$$L \rightarrow (L)$$ where L is lower triangular

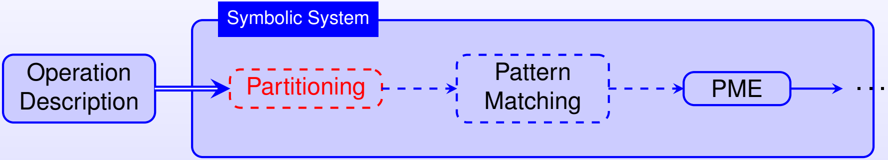
$$L \rightarrow \xcancel{\left(L_L \,\middle|\, L_R \right)}$$

$$L \rightarrow \xcancel{\left(\begin{array}{c} L_T \\ \hline L_B \end{array} \right)}$$

$$L \rightarrow \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array} \right)$$ where L_{TL} & L_{BR} are lower triangular

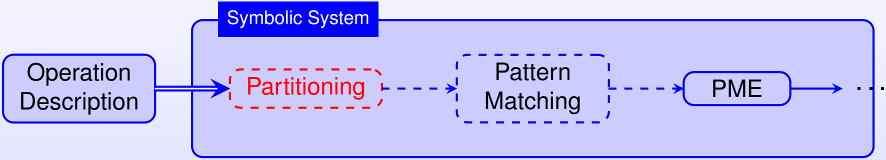
Our system keeps track of the properties

An example: LU Factorization



Partitioned postcondition:

$$LU = A$$



Partitioned postcondition:

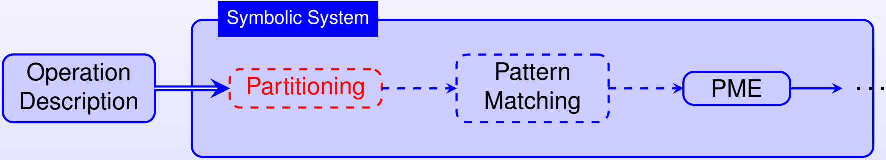
$$LU = A$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right)\left(U\right) = \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

Partitioning

An example: LU Factorization



Partitioned postcondition:

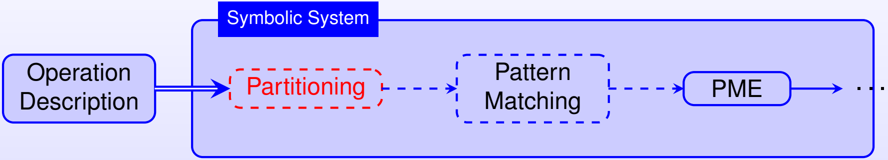
$$LU = A$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \left(U \right) = \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

Non-Conformal Partitioning

An example: LU Factorization

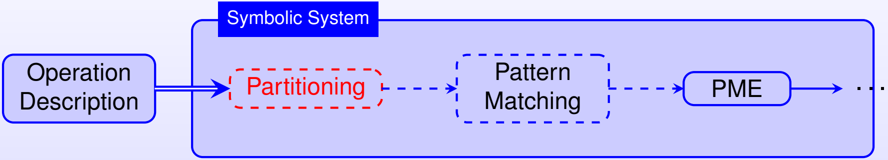


Partitioned postcondition:

$$LU = A$$

$$\Downarrow$$

$$(L)\,(U) = (A)$$



Partitioned postcondition:

$$LU = A$$

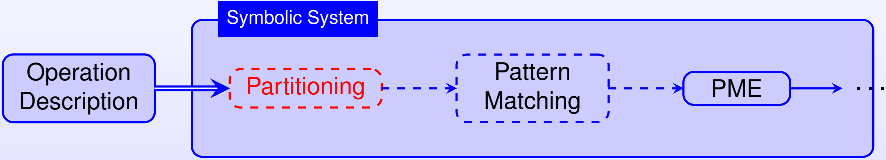
$$\Downarrow$$

$$\left(L\right)\left(U\right) = \left(A\right)$$

Does not decompose the operation

Partitioning

An example: LU Factorization

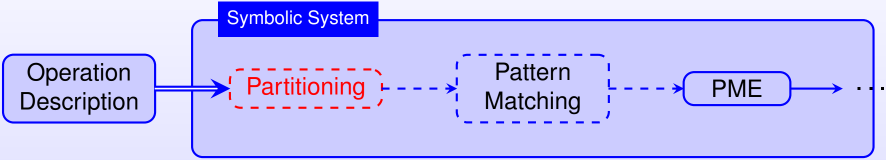


Partitioned postcondition:

$$LU = A$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right) = \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$$



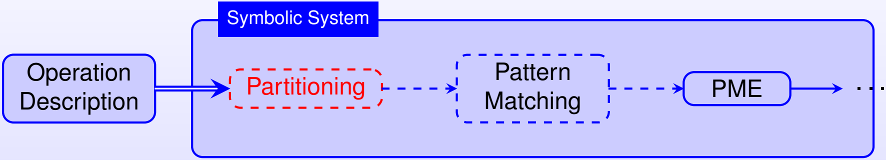
Partitioned postcondition:

$$LU = A$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right)\left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right) = \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

Only the feasible ones are generated!



Symbolic computation:

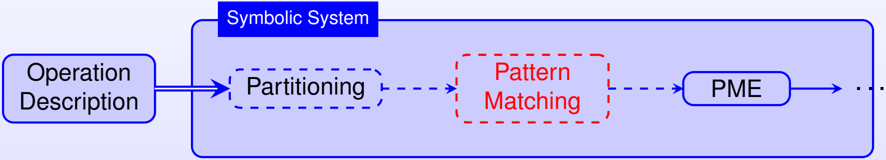
$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right)\left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right) = \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} L_{TL}U_{TL} = A_{TL} & L_{TL}U_{TR} = A_{TR} \\ \hline L_{BL}U_{TL} = A_{BL} & L_{BL}U_{TR} + L_{BR}U_{BR} = A_{BR} \end{array}\right)$$

Pattern Matching

An example: LU Factorization



Input / Output:

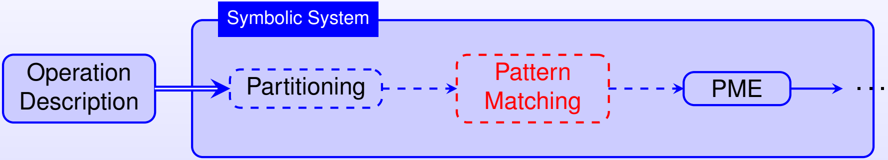
$$\left(\begin{array}{c|c} L_{TL}U_{TL} = A_{TL} & L_{TL}U_{TR} = A_{TR} \\ \hline L_{BL}U_{TL} = A_{BL} & L_{BL}U_{TR} + L_{BR}U_{BR} = A_{BR} \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} L_{TL}U_{TL} = A_{TL} & L_{TL}U_{TR} = A_{TR} \\ \hline L_{BL}U_{TL} = A_{BL} & L_{BL}U_{TR} + L_{BR}U_{BR} = A_{BR} \end{array}\right)$$

Pattern Matching

An example: LU Factorization

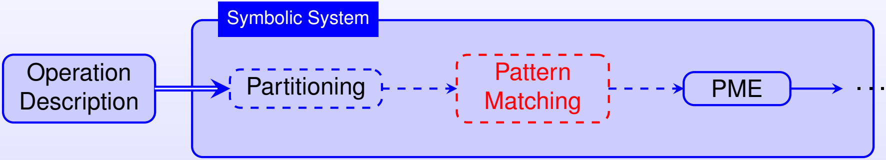


Pattern matching and algebraic manipulation:

$$\left(\begin{array}{c|c} L_{TL}U_{TL} = A_{TL} & L_{TL}U_{TR} = A_{TR} \\ \hline L_{BL}U_{TL} = A_{BL} & L_{BL}U_{TR} + L_{BR}U_{BR} = A_{BR} \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & L_{TL}U_{TR} = A_{TR} \\ \hline L_{BL}U_{TL} = A_{BL} & L_{BL}U_{TR} + L_{BR}U_{BR} = A_{BR} \end{array}\right)$$



Pattern matching and algebraic manipulation:

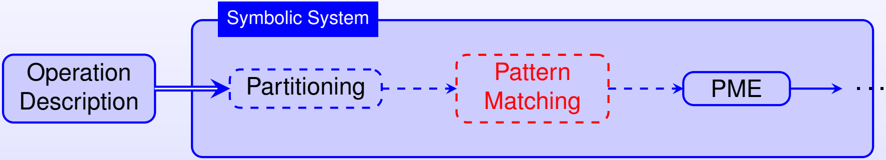
$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & L_{TL}U_{TR} = A_{TR} \\ \hline L_{BL}U_{TL} = A_{BL} & L_{BL}U_{TR} + L_{BR}U_{BR} = A_{BR} \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & L_{TL}U_{TR} = A_{TR} \\ \hline L_{BL}U_{TL} = A_{BL} & L_{BL}U_{TR} + L_{BR}U_{BR} = A_{BR} \end{array}\right)$$

Pattern Matching

An example: LU Factorization



Pattern matching and algebraic manipulation:

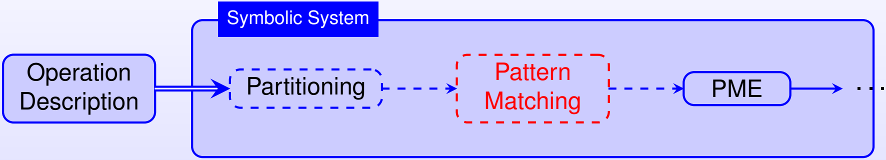
$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & L_{TL}U_{TR} = A_{TR} \\ \hline L_{BL}U_{TL} = A_{BL} & L_{BL}U_{TR} + L_{BR}U_{BR} = A_{BR} \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1}A_{TR} \\ \hline L_{BL} = A_{BL}U_{TL}^{-1} & L_{BL}U_{TR} + L_{BR}U_{BR} = A_{BR} \end{array}\right)$$

Pattern Matching

An example: LU Factorization

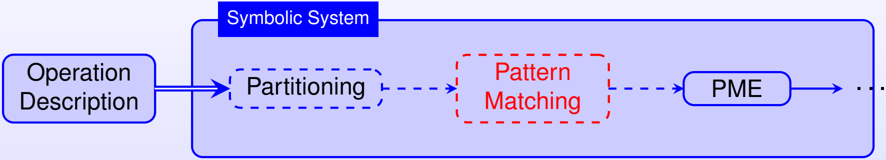


Pattern matching and algebraic manipulation:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & L_{BL} U_{TR} + L_{BR} U_{BR} = A_{BR} \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = TRSM(L_{TL}^{-1}, A_{TR}) \\ \hline L_{BL} = TRSM(A_{BL}, U_{TL}^{-1}) & L_{BL} U_{TR} + L_{BR} U_{BR} = A_{BR} \end{array}\right)$$



Pattern matching and algebraic manipulation:

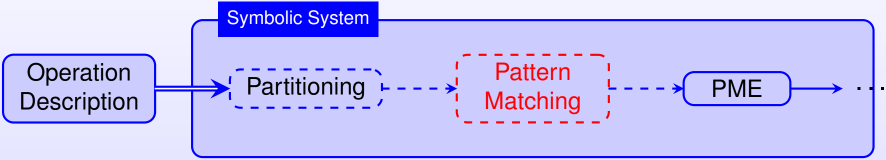
$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = TRSM(L_{TL}^{-1}, A_{TR}) \\ \hline L_{BL} = TRSM(A_{BL}, U_{TL}^{-1}) & L_{BL}U_{TR} + L_{BR}U_{BR} = A_{BR} \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = TRSM(L_{TL}^{-1}, A_{TR}) \\ \hline L_{BL} = TRSM(A_{BL}, U_{TL}^{-1}) & L_{BL}U_{TR} + L_{BR}U_{BR} = A_{BR} \end{array}\right)$$

Pattern Matching

An example: LU Factorization



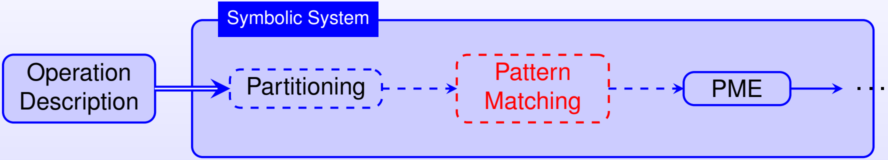
Pattern matching and algebraic manipulation:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = TRSM(L_{TL}^{-1}, A_{TR}) \\ \hline L_{BL} = TRSM(A_{BL}, U_{TL}^{-1}) & L_{BL}U_{TR} + L_{BR}U_{BR} = A_{BR} \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = TRSM(L_{TL}^{-1}, A_{TR}) \\ \hline L_{BL} = TRSM(A_{BL}, U_{TL}^{-1}) & L_{BR}U_{BR} = A_{BR} - L_{BL}U_{TR} \end{array}\right)$$

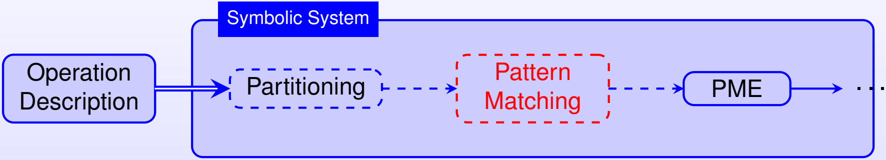
An example: LU Factorization



Pattern matching and algebraic manipulation:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = TRSM(L_{TL}^{-1}, A_{TR}) \\ \hline L_{BL} = TRSM(A_{BL}, U_{TL}^{-1}) & L_{BR}U_{BR} = A_{BR} - L_{BL}U_{TR} \end{array}\right)$$

An example: LU Factorization



Pattern matching and algebraic manipulation:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = TRSM(L_{TL}^{-1}, A_{TR}) \\ \hline L_{BL} = TRSM(A_{BL}, U_{TL}^{-1}) & L_{BR}U_{BR} = A_{BR} - L_{BL}U_{TR} \end{array}\right)$$

$$\boxed{ExistsLU(A_{BR} - L_{BL}U_{TR})\ ?}$$

RWTH AACHEN UNIVERSITY

$ExistsLU(A_{BR} - L_{BL}U_{TR})$?

Theorem

$$A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

$$\underset{\text{full rank(A)}}{ExistsLU(A)} \Longrightarrow \begin{cases} ExistsLU(A_{TL}) \wedge \\ ExistsLU(A_{BR} - A_{BL}A_{TL}^{-1}A_{TR}) \end{cases}$$

AICES

Existence of LU

$$ExistsLU(A_{BR} - L_{BL}U_{TR})\ ?$$

Theorem

$$A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

$$\begin{array}{c} ExistsLU(A) \\ \text{full rank(A)} \end{array} \Longrightarrow \begin{cases} ExistsLU(A_{TL}) \wedge \\ ExistsLU(A_{BR} - A_{BL}A_{TL}^{-1}A_{TR}) \end{cases}$$

$$ExistsLU(A_{BR} - L_{BL}U_{TR}) \equiv ExistsLU(A_{BR} - A_{BL}A_{TL}^{-1}A_{TR})\ ?$$

Existence of LU

$$ExistsLU(A_{BR} - L_{BL}U_{TR})\ ?$$

Theorem

$$A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

$$\begin{array}{c} ExistsLU(A) \\ \text{full rank(A)} \end{array} \Longrightarrow \begin{cases} ExistsLU(A_{TL}) \wedge \\ ExistsLU(A_{BR} - A_{BL}A_{TL}^{-1}A_{TR}) \end{cases}$$

$$ExistsLU(A_{BR} - L_{BL}U_{TR}) \equiv ExistsLU(A_{BR} - A_{BL}A_{TL}^{-1}A_{TR})\ ?$$

$$L_{BL} \rightarrow A_{BL}U_{TL}^{-1} \ ; \ U_{TR} \rightarrow L_{TL}^{-1}A_{TR} \ ; \ L_{TL} * U_{TL} \rightarrow A_{TL}$$

Existence of LU

$$ExistsLU(A_{BR} - L_{BL}U_{TR})\ ?$$

Theorem

$$A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

$ExistsLU(A)$
full rank(A)
$$\Longrightarrow \begin{cases} ExistsLU(A_{TL}) \wedge \\ ExistsLU(A_{BR} - A_{BL}A_{TL}^{-1}A_{TR}) \end{cases}$$

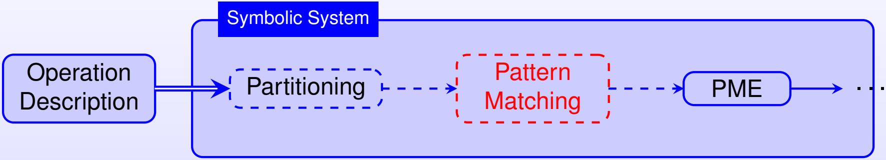
$$ExistsLU(A_{BR} - L_{BL}U_{TR}) \equiv ExistsLU(A_{BR} - A_{BL}A_{TL}^{-1}A_{TR})\ ?$$

$$L_{BL} \rightarrow A_{BL}U_{TL}^{-1} \ ; \ U_{TR} \rightarrow L_{TL}^{-1}A_{TR} \ ; \ L_{TL} * U_{TL} \rightarrow A_{TL}$$

$$L_{BL}U_{TR} \equiv A_{BL}U_{TL}^{-1}L_{TL}^{-1}A_{TR} \equiv A_{BL}(L_{TL}U_{TL})^{-1}A_{TR} \equiv A_{BL}A_{TL}^{-1}A_{TR}$$

Pattern Matching

An example: LU Factorization



Pattern matching and algebraic manipulation:

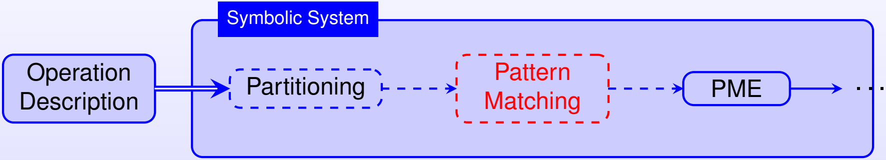
$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = TRSM(L_{TL}^{-1}, A_{TR}) \\ \hline L_{BL} = TRSM(A_{BL}, U_{TL}^{-1}) & L_{BR}U_{BR} = A_{BR} - L_{BL}U_{TR} \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = TRSM(L_{TL}^{-1}, A_{TR}) \\ \hline L_{BL} = TRSM(A_{BL}, U_{TL}^{-1}) & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL}U_{TR}) \end{array}\right)$$

Pattern Matching

An example: LU Factorization



Pattern matching and algebraic manipulation:

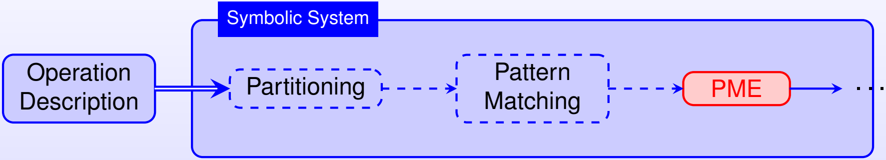
$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = TRSM(L_{TL}^{-1}, A_{TR}) \\ \hline L_{BL} = TRSM(A_{BL}, U_{TL}^{-1}) & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL}U_{TR}) \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = TRSM(L_{TL}^{-1}, A_{TR}) \\ \hline L_{BL} = TRSM(A_{BL}, U_{TL}^{-1}) & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL}U_{TR}) \end{array}\right)$$

Partitioned Matrix Expression

An example: LU Factorization



PME:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = TRSM(L_{TL}^{-1}, A_{TR}) \\ \hline L_{BL} = TRSM(A_{BL}, U_{TL}^{-1}) & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL}U_{TR}) \end{array}\right)$$

Outline

Example 1 - Sylvester

Sylvester Equation:
$LX + XU = C$

$$f: \quad X := \Omega(L,U,C) \quad \equiv \begin{cases} f_{\text{Pre}}: & \{Input(L,U,C) \wedge Output(X) \wedge \\ & \quad LowTri(L) \wedge UppTri(U)\} \\ f_{\text{Post}}: & \{LX + XU = C\} \end{cases}$$

Sylvester Equation:
$LX + XU = C$

$$f: \quad X := \Omega(L, U, C) \;\equiv \begin{cases} f_{\text{Pre}}: & \{Input(L,U,C) \,\wedge\, Output(X) \,\wedge \\ & \;LowTri(L) \,\wedge\, UppTri(U)\} \\ f_{\text{Post}}: & \{LX + XU = C\} \end{cases}$$

$$\Downarrow$$

$$\left(\frac{C_T}{C_B}\right) = \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right)\left(\frac{X_T}{X_B}\right) + \left(\frac{X_T}{X_B}\right) U$$

RWTH AACHEN UNIVERSITY

Sylvester Equation:
$LX + XU = C$

$$f: \quad X := \Omega(L, U, C) \quad \equiv \begin{cases} f_{\text{Pre}}: & \{Input(L, U, C) \ \wedge \ Output(X) \ \wedge \\ & \ LowTri(L) \ \wedge \ UppTri(U)\} \\ f_{\text{Post}}: & \{LX + XU = C\} \end{cases}$$

$$\Downarrow$$

$$\left(\frac{C_T}{C_B}\right) = \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \left(\frac{X_T}{X_B}\right) + \left(\frac{X_T}{X_B}\right) U$$

$$\Downarrow$$

$$\left(\frac{X_T = \Omega(L_{TL}, U, C_T)}{X_B = \Omega(L_{BR}, U, C_B - L_{BL} X_T)}\right)$$

AICES

Sylvester Equation:
$$LX + XU = C$$

$$f: \quad X := \Omega(L, U, C) \quad \equiv \begin{cases} f_{\mathrm{Pre}}: & \{Input(L,U,C) \wedge Output(X) \wedge \\ & \quad LowTri(L) \wedge UppTri(U)\} \\ f_{\mathrm{Post}}: & \{LX + XU = C\} \end{cases}$$

$$\Downarrow$$

$$\left(C_L \,\middle|\, C_R \right) = L \left(X_L \,\middle|\, X_R \right) + \left(X_L \,\middle|\, X_R \right) \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array} \right)$$

Sylvester Equation:
$LX + XU = C$

$$f: \quad X := \Omega(L,U,C) \;\equiv \begin{cases} f_{\mathrm{Pre}}: & \{Input(L,U,C) \wedge Output(X) \wedge \\ & \quad LowTri(L) \wedge UppTri(U)\} \\ f_{\mathrm{Post}}: & \{LX + XU = C\} \end{cases}$$

$$\Downarrow$$

$$\left(\; C_L \;\middle|\; C_R \; \right) = L \left(\; X_L \;\middle|\; X_R \; \right) + \left(\; X_L \;\middle|\; X_R \; \right) \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array} \right)$$

$$\Downarrow$$

$$\left(\; X_L = \Omega(L, U_{TL}, C_L) \;\middle|\; X_R = \Omega(L, U_{BR}, C_R - X_L U_{TR}) \; \right)$$

Sylvester Equation:
$LX + XU = C$

$$f: \quad X := \Omega(L,U,C) \quad \equiv \begin{cases} f_{\text{Pre}}: & \{Input(L,U,C) \ \wedge \ Output(X) \ \wedge \\ & \ LowTri(L) \ \wedge \ UppTri(U)\} \\ f_{\text{Post}}: & \{LX + XU = C\} \end{cases}$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} C_{TL} & C_{TR} \\ \hline C_{BL} & C_{BR} \end{array}\right) = \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \left(\begin{array}{c|c} X_{TL} & X_{TR} \\ \hline X_{BL} & X_{BR} \end{array}\right) + \left(\begin{array}{c|c} X_{TL} & X_{TR} \\ \hline X_{BL} & X_{BR} \end{array}\right) \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right)$$

Sylvester Equation:
$LX + XU = C$

$$f: \quad X := \Omega(L, U, C) \quad \equiv \begin{cases} f_{\text{Pre}}: & \{Input(L,U,C) \wedge Output(X) \wedge \\ & \; LowTri(L) \wedge UppTri(U)\} \\ f_{\text{Post}}: & \{LX + XU = C\} \end{cases}$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} C_{TL} & C_{TR} \\ \hline C_{BL} & C_{BR} \end{array}\right) = \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \left(\begin{array}{c|c} X_{TL} & X_{TR} \\ \hline X_{BL} & X_{BR} \end{array}\right) + \left(\begin{array}{c|c} X_{TL} & X_{TR} \\ \hline X_{BL} & X_{BR} \end{array}\right) \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} X_{TL} = \Omega(L_{TL}, U_{TL}, C_{TL}) & X_{TR} = \Omega(L_{TL}, U_{BR}, C_{TR} - X_{TL}U_{TR}) \\ \hline X_{BL} = \Omega(L_{BR}, U_{TL}, C_{BL} - L_{BL}X_{TL}) & X_{BR} = \Omega(L_{BR}, U_{BR}, C_{BR} - L_{BL}X_{TR} - X_{BL}U_{TR}) \end{array}\right)$$

Example 2 - Reduction

Reduction to Standard Form:
$LBL^T = A$

$$f: \quad B := \Phi(A, L) \;\equiv \begin{cases} f_{\text{Pre}}: & \{Input(A, L) \;\wedge\; Output(B) \;\wedge \\ & \;LowTri(L) \;\wedge\; Symmetric(A, B)\} \\ f_{\text{Post}}: & \{LBL^T = A\} \end{cases}$$

Reduction to Standard Form:
$LBL^T = A$

$$f: \quad B := \Phi(A, L) \quad \equiv \begin{cases} f_{\text{Pre}}: & \{Input(A, L) \wedge Output(B) \wedge \\ & \;LowTri(L) \wedge Symmetric(A, B)\} \\ f_{\text{Post}}: & \{LBL^T = A\} \end{cases}$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \left(\begin{array}{c|c} B_{TL} & B_{BL}^T \\ \hline B_{BL} & B_{BR} \end{array}\right) \left(\begin{array}{c|c} L_{TL}^T & L_{BL}^T \\ \hline 0 & L_{BR}^T \end{array}\right) = \left(\begin{array}{c|c} A_{TL} & A_{BL}^T \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

Reduction to Standard Form:
$LBL^T = A$

$$f: \quad B := \Phi(A, L) \quad \equiv \begin{cases} f_{\text{Pre}}: & \{Input(A, L) \ \wedge \ Output(B) \ \wedge \\ & \ LowTri(L) \ \wedge \ Symmetric(A, B)\} \\ f_{\text{Post}}: & \{LBL^T = A\} \end{cases}$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \left(\begin{array}{c|c} B_{TL} & B_{BL}^T \\ \hline B_{BL} & B_{BR} \end{array}\right) \left(\begin{array}{c|c} L_{TL}^T & L_{BL}^T \\ \hline 0 & L_{BR}^T \end{array}\right) = \left(\begin{array}{c|c} A_{TL} & A_{BL}^T \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{c|c} B_{TL} = \Phi(A_{TL}, L_{TL}) & \star \\ \hline B_{BL} = L_{BR}^{-1}(A_{BL} L_{TL}^{-1} - L_{BL} B_{TL}) & \begin{array}{c} B_{BR} = \Phi(A_{BR} - L_{BL} B_{TL} L_{BL}^T - L_{BR} B_{BL} L_{BL}^T \\ - L_{BL} B_{BL}^T L_{BR}^T, \quad L_{BR}) \end{array} \end{array}\right)$$

Outline

Conclusions

- Minimum amount of knowledge:
  - Input and output operands
  - Size of the operands
  - Structure of the operands: triangularity, symmetry, ...

Conclusions

- Minimum amount of knowledge:
  - Input and output operands
  - Size of the operands
  - Structure of the operands: triangularity, symmetry, . . .
- Knowledge implemented IN the system:
  - Basic matrix algebra
  - Pattern matching
  - Properties inheritance
  - Theorems

Conclusions

- Minimum amount of knowledge:
  - Input and output operands
  - Size of the operands
  - Structure of the operands: triangularity, symmetry, ...
- Knowledge implemented IN the system:
  - Basic matrix algebra
  - Pattern matching
  - Properties inheritance
  - Theorems
- One step closer to the full automatic generation of algorithms.

Thanks to:

- Dr. Edoardo Di Napoli
- Matthias Petschow
- Roman Iakymchuk
- FLAME Group in UT, Austin (Texas)

Funding from DFG is gratefully acknowledged

Deutsche
Forschungsgemeinschaft
DFG