Non-linear Associative-Commutative Many-to-One Pattern Matching with Sequence Variables

Manuel Krebber

June 30, 2017

Table of Contents

Term

Function Symbols: f, g

Term

Function Symbols: f, g

Constant Symbols: a, b, c

Term

Function Symbols: f, g

Constant Symbols: a, b, c

Variables: $\boldsymbol{x}, \boldsymbol{y}$

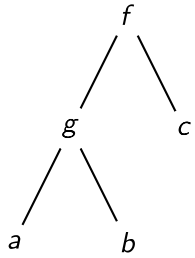
Term

Function Symbols: f, g

Constant Symbols: a, b, c

Variables: $\boldsymbol{x}, \boldsymbol{y}$

Examples: a, $f(\boldsymbol{x}, b)$,
$f(g(a, b), c)$



Pattern Matching

RWTH AACHEN UNIVERSITY

Pattern Matching

Definition: Find substitution $\sigma : \mathcal{X} \nrightarrow \mathcal{T}$ such that
$\hat{\sigma}(\mathit{pattern}) = \mathit{subject}$

HPAC

Pattern Matching

Definition: Find substitution $\sigma : \mathcal{X} \nrightarrow \mathcal{T}$ such that $\hat{\sigma}(\mathit{pattern}) = \mathit{subject}$

Example: Pattern: $f(\boldsymbol{x}, \boldsymbol{y})$

$$\boldsymbol{x} \mapsto a$$
$$\boldsymbol{y} \mapsto g(b)$$

Subject: $f(a, g(b))$

Applications

- Functional programming languages

Applications

- Functional programming languages
- Computer algebra systems (Mathematica)

RWTH AACHEN UNIVERSITY

Applications

- Functional programming languages
- Computer algebra systems (Mathematica)
- Term rewriting systems

RWTH AACHEN UNIVERSITY

Applications

- Functional programming languages
- Computer algebra systems (Mathematica)
- Term rewriting systems
- In this case: Linear Algebra

Table of Contents

Types of Matching

- Syntactic

Types of Matching

- Syntactic
- Linear $(\boldsymbol{x} + \boldsymbol{y})$ vs. non-linear $(\boldsymbol{x} + \boldsymbol{x})$

Types of Matching

- Syntactic
- Linear $(\boldsymbol{x} + \boldsymbol{y})$ vs. non-linear $(\boldsymbol{x} + \boldsymbol{x})$
- Sequence variables
- Associativity
- Commutativity

Types of Matching

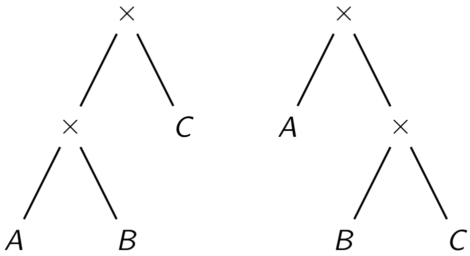
- Syntactic
- Linear $(\boldsymbol{x} + \boldsymbol{y})$ vs. non-linear $(\boldsymbol{x} + \boldsymbol{x})$
- Sequence variables
- Associativity
- Commutativity
- Many-to-one vs. one-to-one

Associativity

$$(A \times B) \times C \quad = \quad A \times (B \times C)$$

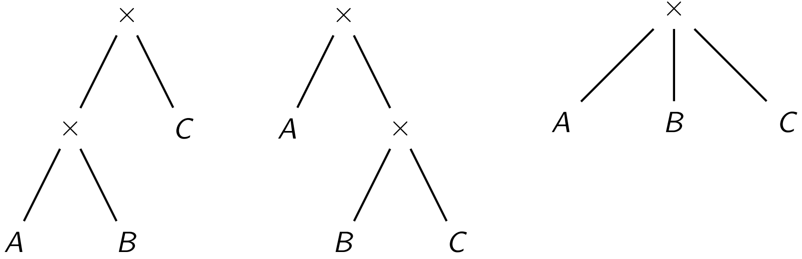
Associativity

$$(A \times B) \times C \quad = \quad A \times (B \times C)$$



Associativity

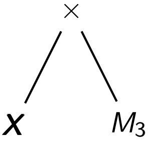
$$(A \times B) \times C \quad = \quad A \times (B \times C) \quad = \quad A \times B \times C$$



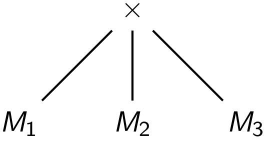
Canonical Variadic Form

Associativity II

$\boldsymbol{X} \times M_3$

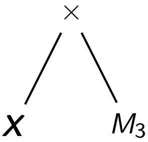


$M_1 \times M_2 \times M_3$

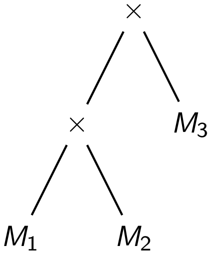


Associativity II

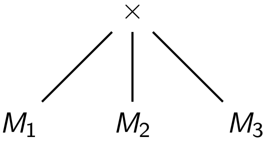
$\boldsymbol{X} \times M_3$



$(M_1 \times M_2) \times M_3$

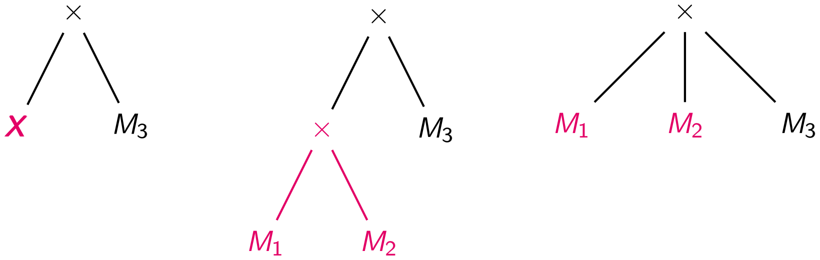


$M_1 \times M_2 \times M_3$



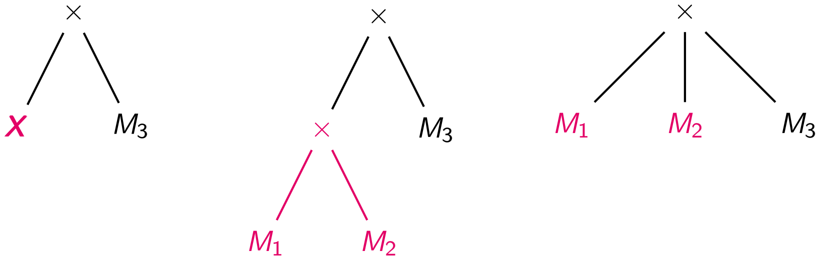
Associativity II

$\boldsymbol{X} \times M_3$ $\quad$ $(M_1 \times M_2) \times M_3$ $\quad$ $M_1 \times M_2 \times M_3$



Associativity II

$\boldsymbol{X} \times M_3$ $\quad$ $(M_1 \times M_2) \times M_3$ $\quad$ $M_1 \times M_2 \times M_3$



$$\sigma = \{\boldsymbol{X} \mapsto (M_1 \times M_2)\}$$

Sequence Variables

Can match a sequence of terms

Sequence Variables

Can match a sequence of terms

Notation: $\boldsymbol{x}^{*}$ star variable, $\boldsymbol{x}^{+}$ plus variable

star variables can match empty sequence

RWTH AACHEN UNIVERSITY

Sequence Variables

Can match a sequence of terms

Notation: $\boldsymbol{x}^*$ star variable, $\boldsymbol{x}^+$ plus variable

star variables can match empty sequence

Example: $\sigma(f(a, \boldsymbol{x}^+, d)) = f(a, b, c, d)$

HPAC

RWTH AACHEN UNIVERSITY

Sequence Variables

Can match a sequence of terms

Notation: $\boldsymbol{x}^*$ star variable, $\boldsymbol{x}^+$ plus variable
star variables can match empty sequence

Example: $\sigma(f(a, \boldsymbol{x}^+, d)) = f(a, b, c, d)$
$\sigma = \{\boldsymbol{x}^+ \mapsto (b, c)\}$

HPAC

RWTH AACHEN UNIVERSITY

Sequence Variables

Can match a sequence of terms

Notation: $\boldsymbol{x}^*$ star variable, $\boldsymbol{x}^+$ plus variable
star variables can match empty sequence

Example: $\sigma(f(a, \boldsymbol{x}^+, d)) = f(a, b, c, d)$
$\sigma = \{\boldsymbol{x}^+ \mapsto (b, c)\}$

Associativity: $\sigma(f_a(a, \boldsymbol{x}, d)) = f_a(a, b, c, d)$

RWTH AACHEN UNIVERSITY

Sequence Variables

Can match a sequence of terms

Notation: $\boldsymbol{x}^*$ star variable, $\boldsymbol{x}^+$ plus variable
star variables can match empty sequence

Example: $\sigma(f(a, \boldsymbol{x}^+, d)) = f(a, b, c, d)$
$\sigma = \{\boldsymbol{x}^+ \mapsto (b, c)\}$

Associativity: $\sigma(f_a(a, \boldsymbol{x}, d)) = f_a(a, b, c, d)$
$\sigma = \{\boldsymbol{x} \mapsto f_a(b, c)\}$

HPAC

Commutativity

- $a + b = b + a$

Commutativity

- $a + b = b + a$
- Sort the arguments: $d + a + c + b \quad \rightarrow \quad a + b + c + d$

RWTH AACHEN UNIVERSITY

Commutativity

- $a + b = b + a$
- Sort the arguments: $d + a + c + b \quad \rightarrow \quad a + b + c + d$
- Every permutation could match:
  $n!$ with $n = |subject|$

HPAC

Complexity

	Synt	Assoc	SeqVar	Comm	All
Max. # matches	1	$\binom{n-1}{m-1}$	$\binom{n+m-1}{m-1}$	$n!$	n^m
NP complete	no	yes	yes	yes	yes

$$n = |subject|, \; m = |pattern|$$

Many-to-one Matching

- Many patterns, one subject

Many-to-one Matching

- Many patterns, one subject
- Speedup by simultaneous matching

RWTH AACHEN UNIVERSITY

Many-to-one Matching

- Many patterns, one subject
- Speedup by simultaneous matching
- Use similarity between patterns

Table of Contents

Table of Contents

RWTH AACHEN UNIVERSITY

Associative and Sequence Variables

Example: Subject $f(a, b)$ and pattern $f(\boldsymbol{x}^*, \boldsymbol{y}^*)$

HPAC

RWTH AACHEN UNIVERSITY

Associative and Sequence Variables

Example: Subject $f(a, b)$ and pattern $f(\boldsymbol{x}^*, \boldsymbol{y}^*)$

- Try every possible distribution

HPAC

RWTH AACHEN UNIVERSITY

Associative and Sequence Variables

Example: Subject $f(a, b)$ and pattern $f(\boldsymbol{x^*}, \boldsymbol{y^*})$

- Try every possible distribution
- Generate (weak) compositions:

$$
\begin{aligned}
0 + 2 &= 2 \qquad && \{\boldsymbol{x^*} \mapsto (), \boldsymbol{y^*} \mapsto (a, b)\} \\
1 + 1 &= 2 \qquad && \{\boldsymbol{x^*} \mapsto (a), \boldsymbol{y^*} \mapsto (b)\} \\
2 + 0 &= 2 \qquad && \{\boldsymbol{x^*} \mapsto (a, b), \boldsymbol{y^*} \mapsto ()\}
\end{aligned}
$$

RWTH AACHEN UNIVERSITY

Associative and Sequence Variables

Example: Subject $f(a, b)$ and pattern $f(\boldsymbol{x^*}, \boldsymbol{y^*})$

- Try every possible distribution
- Generate (weak) compositions:

$$
\begin{aligned}
0 + 2 = 2 \qquad & \{\boldsymbol{x^*} \mapsto (), \boldsymbol{y^*} \mapsto (a, b)\} \\
1 + 1 = 2 \qquad & \{\boldsymbol{x^*} \mapsto (a), \boldsymbol{y^*} \mapsto (b)\} \\
2 + 0 = 2 \qquad & \{\boldsymbol{x^*} \mapsto (a, b), \boldsymbol{y^*} \mapsto ()\}
\end{aligned}
$$

- Backtracking

RWTH AACHEN UNIVERSITY

Steps of Commutative Matching

Pattern: $f_c(a, g(c, d), g(\boldsymbol{x}), \boldsymbol{x}, \boldsymbol{y^*}, \boldsymbol{z})$
Subject: $f_c(a, b, g(b), c, g(c, d), e)$
Match: $\sigma = \{\boldsymbol{z} \mapsto e\}$

Permutations: 6! = 720

Steps of Commutative Matching

Pattern: $f_c(a, g(c, d), g(\boldsymbol{x}), \boldsymbol{x}, \boldsymbol{y^*}, \boldsymbol{z})$
Subject: $f_c(a, b, g(b), c, g(c, d), e)$
Match: $\sigma = \{\boldsymbol{z} \mapsto e\}$

Permutations: 6! = 720

1. Constant terms

RWTH AACHEN UNIVERSITY

Steps of Commutative Matching

Pattern: $f_c(\qquad g(\boldsymbol{x}), \boldsymbol{x}, \boldsymbol{y^*}, \boldsymbol{z})$
Subject: $f_c(\quad b, g(b), c, \qquad e)$
Match: $\sigma = \{\boldsymbol{z} \mapsto e\}$

Permutations: 4! = 24

1. Constant terms
2. Matched variables

HPAC

Steps of Commutative Matching

Pattern: $f_c(\quad g(\boldsymbol{x}), \boldsymbol{x}, \boldsymbol{y^*} \quad)$

Subject: $f_c(\quad b, g(b), c \quad)$

Match: $\sigma = \{\boldsymbol{z} \mapsto e, \boldsymbol{x} \mapsto b\}$

Permutations: 3! = 6

1. Constant terms
2. Matched variables
3. Terms containing variables

RWTH AACHEN UNIVERSITY

Steps of Commutative Matching

Pattern: $f_c(\quad x, y^* \quad)$

Subject: $f_c(\quad b, \quad c \quad)$

Match: $\sigma = \{z \mapsto e, x \mapsto b\}$

Permutations: $2! = 6$

1. Constant terms
2. Matched variables
3. Terms containing variables
4. Repeat step 2

HPAC

RWTH AACHEN UNIVERSITY

Steps of Commutative Matching

Pattern: $f_c(\quad \boldsymbol{y^*} \quad)$
Subject: $f_c(\quad c \quad)$
Match: $\sigma = \{\boldsymbol{z} \mapsto e, \boldsymbol{x} \mapsto b\}$

Permutations: $1! = 1$

1. Constant terms
2. Matched variables
3. Terms containing variables
4. Repeat step 2
5. Regular Variables

HPAC

RWTH AACHEN UNIVERSITY

Steps of Commutative Matching

Pattern: $f_c(\quad \boldsymbol{y^*} \quad)$
Subject: $f_c(\quad c \quad)$
Match: $\sigma = \{\boldsymbol{z} \mapsto e, \boldsymbol{x} \mapsto b, \boldsymbol{y^*} \mapsto \lbag c \rbag\}$

Permutations: $1! = 1$

1. Constant terms
2. Matched variables
3. Terms containing variables
4. Repeat step 2
5. Regular Variables
6. Sequence Variables

HPAC

Commutativity + Sequence Variables

Match $f_c(\boldsymbol{x}^+, \boldsymbol{x}^+, \boldsymbol{y}^*)$ and $f_c(a, a, a, b, b, c)$

Commutativity + Sequence Variables

Match $f_c(\boldsymbol{x}^+, \boldsymbol{x}^+, \boldsymbol{y}^*)$ and $f_c(a, a, a, b, b, c)$

Solve linear diophantine equations (using Extended Euclidean Alg.):

$$
\begin{aligned}
3 &= 2x_a + y_a \\
2 &= 2x_b + y_b \\
1 &= 2x_c + y_c \\
1 &\leq x_a + x_b + x_c
\end{aligned}
$$

Commutativity + Sequence Variables

Match $f_c(\boldsymbol{x}^+, \boldsymbol{x}^+, \boldsymbol{y}^*)$ and $f_c(a, a, a, b, b, c)$

Solve linear diophantine equations (using Extended Euclidean Alg.):

$$
\begin{aligned}
3 &= 2x_a + y_a \\
2 &= 2x_b + y_b \\
1 &= 2x_c + y_c \\
1 &\leq x_a + x_b + x_c
\end{aligned}
$$

Solutions:

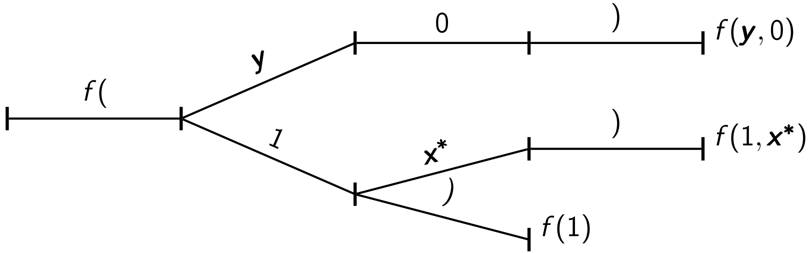
$$
\begin{aligned}
&\{x \mapsto \{a, b\}, && y \mapsto \{a, c\}\} \\
&\{x \mapsto \{a\}, && y \mapsto \{a, b, b, c\}\} \\
&\{x \mapsto \{b\}, && y \mapsto \{a, a, a, c\}\}
\end{aligned}
$$

Table of Contents

RWTH AACHEN UNIVERSITY

Discrimination Net

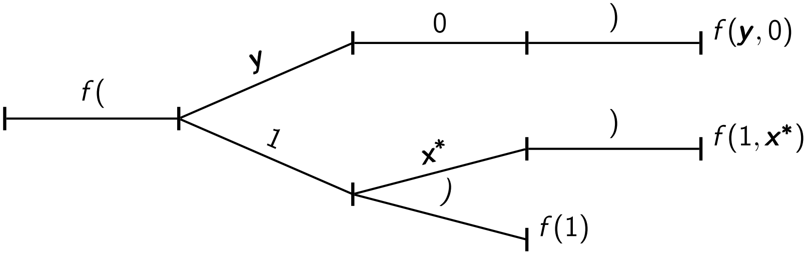
Patterns: $f(1, \boldsymbol{x^*})$, $f(1)$, $f(\boldsymbol{y}, 0)$



HPAC

Discrimination Net

Patterns: $f(1, \boldsymbol{x^*})$, $f(1)$, $f(\boldsymbol{y}, 0)$

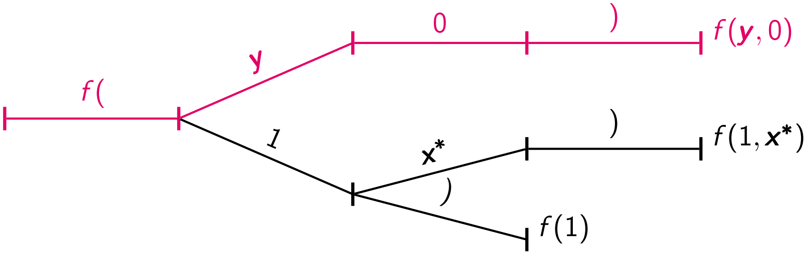


Subject: $f(1, 0)$

RWTH AACHEN UNIVERSITY

Discrimination Net

Patterns: $f(1, \boldsymbol{x^*})$, $f(1)$, $f(\boldsymbol{y}, 0)$

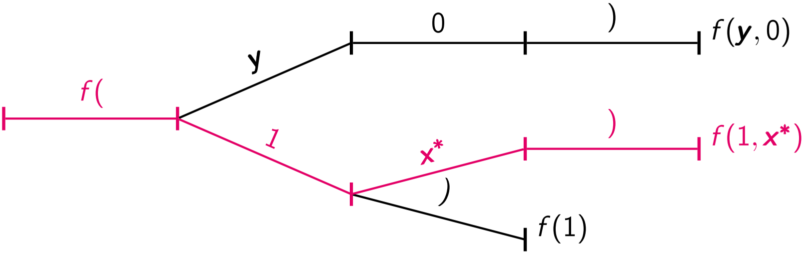


Subject: $f(1, 0)$
Match: $f(\boldsymbol{y}, 0)$ with $\{\boldsymbol{y} \mapsto 1\}$

HPAC

Discrimination Net

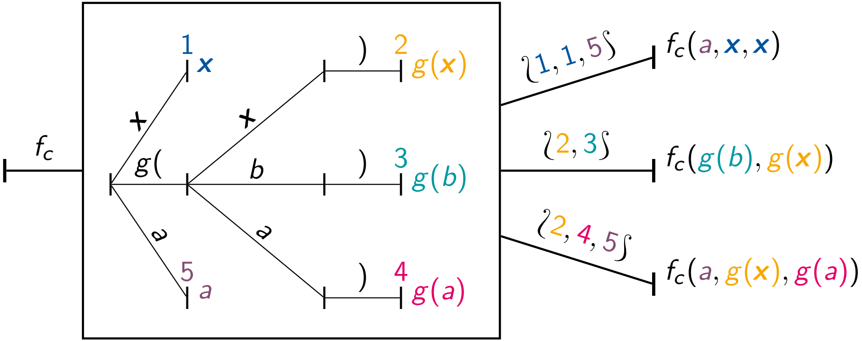
Patterns: $f(1, \boldsymbol{x^*})$, $f(1)$, $f(\boldsymbol{y}, 0)$



Subject: $f(1, 0)$
Match: $f(1, \boldsymbol{x^*})$ with $\{\boldsymbol{x^*} \mapsto (0)\}$

Many-to-One for Commutative

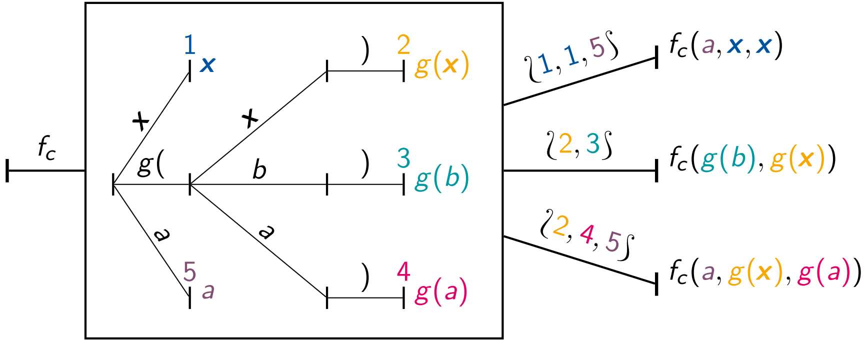
Multi Layer Discrimination Net (MLDN)



RWTH AACHEN UNIVERSITY

Many-to-One for Commutative

Multi Layer Discrimination Net (MLDN)



$$f_c(a, g(a), g(a)) \Rightarrow \{1, 1, 1, 2, 2, 4, 4, 5\}$$

HPAC

Many-to-One for Commutative

1. Match subject arguments with nested DN

Many-to-One for Commutative

1. Match subject arguments with nested DN
2. Build bipartite graph from matches

Many-to-One for Commutative

1. Match subject arguments with nested DN
2. Build bipartite graph from matches
3. Enumerate all maximum matchings (Hopcroft-Karp, Uno)

RWTH AACHEN UNIVERSITY

Many-to-One for Commutative

1. Match subject arguments with nested DN
2. Build bipartite graph from matches
3. Enumerate all maximum matchings (Hopcroft-Karp, Uno)
4. Combine substitutions from matching

HPAC

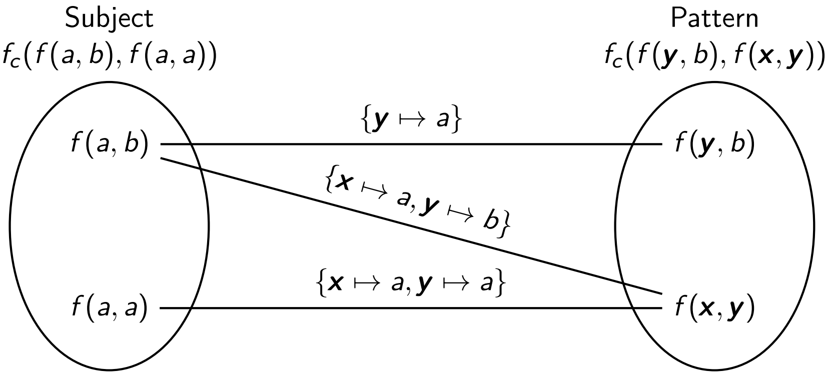
RWTH AACHEN UNIVERSITY

Many-to-One for Commutative

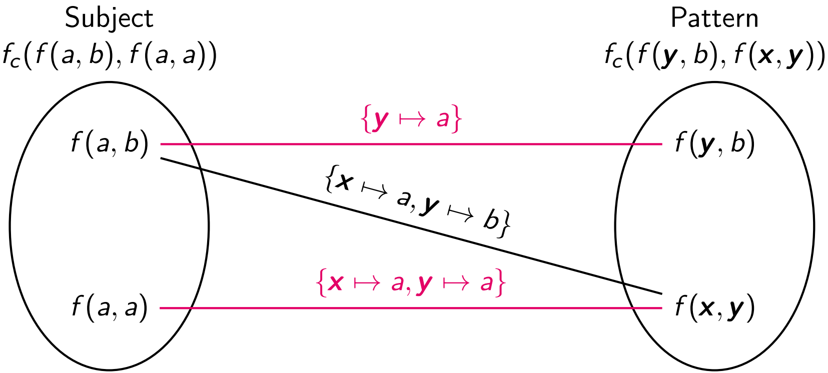
1. Match subject arguments with nested DN
2. Build bipartite graph from matches
3. Enumerate all maximum matchings (Hopcroft-Karp, Uno)
4. Combine substitutions from matching
5. Match sequence variables

HPAC

Bipartite Graph

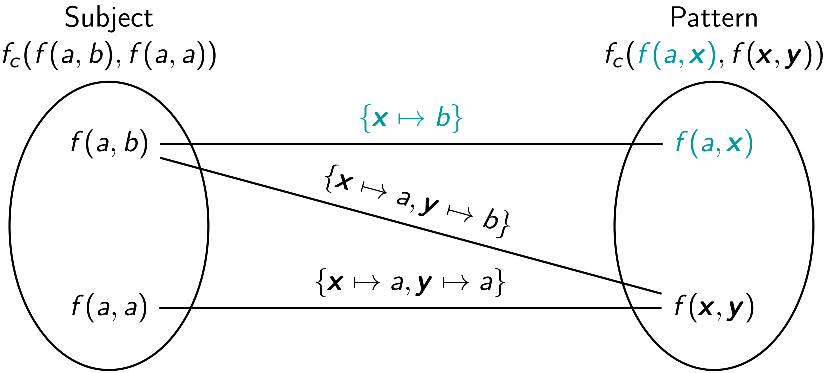


Bipartite Graph



RWTH AACHEN UNIVERSITY

Bipartite Graph



HPAC

Bipartite Graph

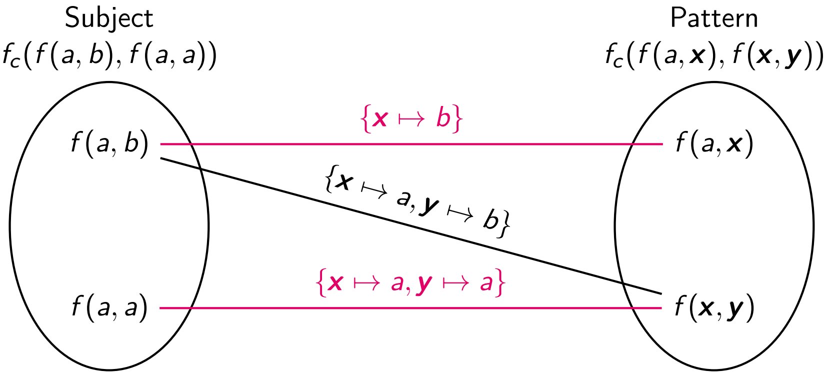


Table of Contents

MatchPy

- Python implementation

MatchPy

- Python implementation
- Open source on Github:
  `https://github.com/hpac/matchpy`

Linear Algebra Example

- $\sim$ 200 patterns for BLAS/LAPACK kernels, e.g. $\boldsymbol{\alpha} \times \boldsymbol{A}^T \times \boldsymbol{B}$

RWTH AACHEN UNIVERSITY

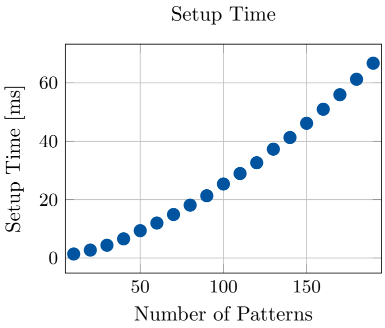
Linear Algebra Example

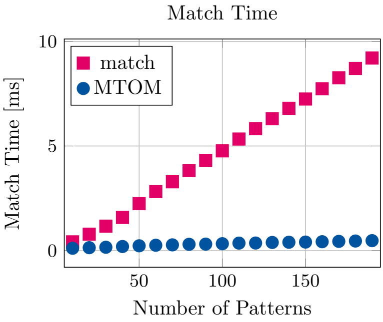
- ~ 200 patterns for BLAS/LAPACK kernels, e.g. $\boldsymbol{\alpha} \times \boldsymbol{A}^T \times \boldsymbol{B}$
- Supported operations:

Operation	Symbol	Arity	Properties
Multiplication	$\times$	variadic	associative
Addition	$+$	variadic	associative, commutative
Transposition	^{T}	unary	
Inversion	$^{-1}$	unary	
Inverse Transposition	$^{-T}$	unary	

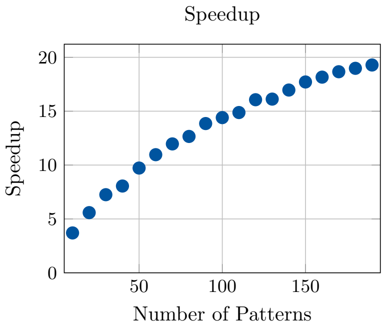
HPAC

Match Times


Setup Time
Setup Time [ms]
0
20
40
60
50
100
150
Number of Patterns


Match Time
Match Time [ms]
0
5
10
match
MTOM
50
100
150
Number of Patterns

Speedup


Speedup
Speedup
0
5
10
15
20
50
100
150
Number of Patterns

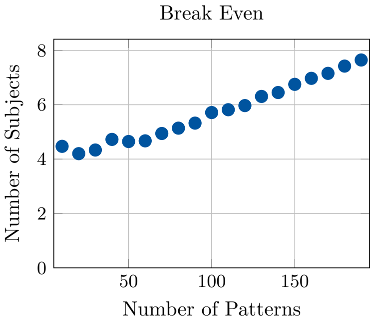

Break Even
Number of Subjects
0
2
4
6
8
50
100
150
Number of Patterns

Table of Contents

RWTH AACHEN UNIVERSITY

Contributions

- Generalized discrimination nets

Contributions

- Generalized discrimination nets
- Many-to-one matching

Contributions

- Generalized discrimination nets
- Many-to-one matching
  - sequence variables

Contributions

- Generalized discrimination nets
- Many-to-one matching
  - sequence variables
  - separate associativity/commutativity

Contributions

- Generalized discrimination nets
- Many-to-one matching
  - sequence variables
  - separate associativity/commutativity
- Open source implementation

RWTH AACHEN UNIVERSITY

Repository

Project is on GitHub:
`https://github.com/hpac/matchpy`

HPAC

Thank you for your attention!