



MathWebSearch: Low-Latency Unification-based Search



Michael Kohlhase [<m.kohlhase@jacobs-university.de>](mailto:m.kohlhase@jacobs-university.de)
Corneliu Prodescu [<c.prodescu@jacobs-university.de>](mailto:c.prodescu@jacobs-university.de)

MathWebSearch is a content-based search engine that focuses on fast query answering for interactive applications. It is currently restricted to exact formula search via unification queries, i.e. no similarity search and no full-text search.

Formula Search with **Named Wildcards** on ZBMath

Zentralblatt MATH

Query: $\int_a^b |f(x)g(x)| dx \leq r$

Found 4 results

1. <http://opal.eecs.jacobs-university.de/zb-sandbox/26/70/1704151/1704151.xhtml#S0.Ex1.m1.1>
Opial inequalities for fractional derivatives.

$$\int_0^a |f(x)f'(x)| dx \leq \frac{a}{4} \int_0^a |f'(x)|^2 dx$$

language: EN class: 26A33 26D10 26D15 keywords: fractional derivative; Opial inequality doctype: serial article published: 2001

a → 0
b → a
f → f
g → f^{normal}
r → $\frac{a}{4} \times \int_0^a |f^{\text{normal}}(x)|^2 dx$

2. <http://opal.eecs.jacobs-university.de/zb-sandbox/6/118/5707565/5707565.xhtml#S0.Ex1.m1.1>
Caputo fractional multivariate Opial type inequalities on spherical shells.

Unification Queries: Applicable Theorem Search

Approximate $\int_{\mathbb{R}^2} |\sin(t) \cos(t)| dt$ from above? \rightsquigarrow Ask MATHWEBSEARCH!
It finds Hölder's inequality with universal variables in the index

$$\int_D |f(x)g(x)| dx \leq (\int_D |f(x)|^p dx)^{\frac{1}{p}} (\int_D |g(x)|^q dx)^{\frac{1}{q}}$$

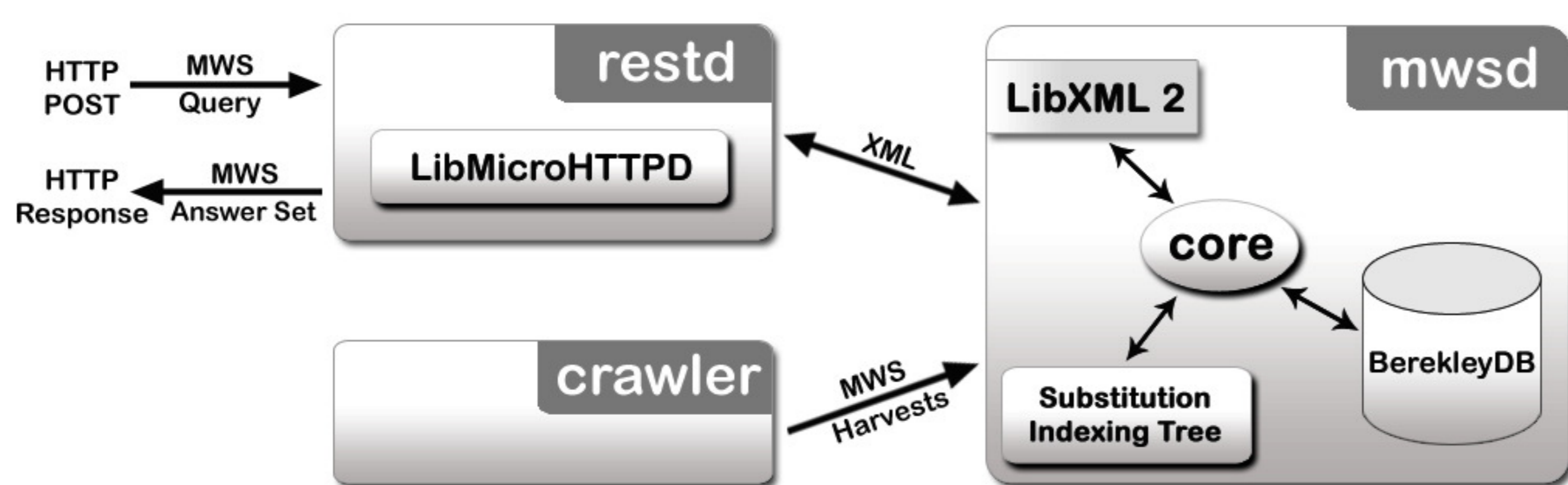
with substitution $x \mapsto t, f \mapsto \sin, g \mapsto \cos, D \mapsto \mathbb{R}^2 \rightsquigarrow$ Solution:

$$\int_{\mathbb{R}^2} |\sin(t) \cos(t)| dt \leq (\int_{\mathbb{R}^2} |\sin(t)|^p dt)^{\frac{1}{p}} (\int_{\mathbb{R}^2} |\cos(t)|^q dt)^{\frac{1}{q}}$$

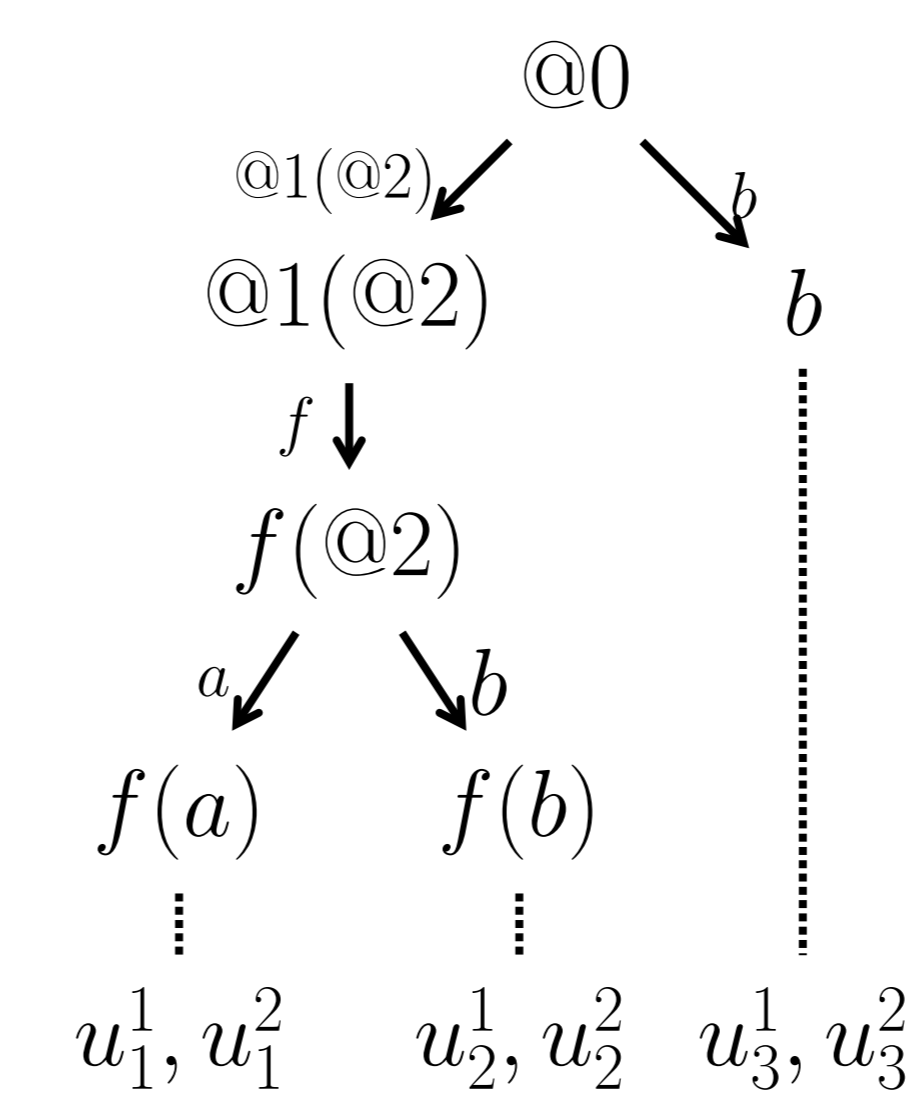
Variant query $\int_{\mathbb{R}^2} |\sin(t) \cos(2t)| dt$ will not find Hölder's inequality since that would introduce inconsistent substitutions $x \mapsto t$ and $x \mapsto 2t$.

The MathWebSearch backend is realized as a RESTful web service that keeps a formula index in memory and hit URIs in database. MathWebSearch front-ends post MathML queries via HTTP and receive XML results.

System Architecture: Web Service with multiple Front-Ends



Substitution Tree Indexing



- Represent Mathematical Formulae in Content MathML extended with query variables
- Insert them into an in-memory "index": a formula structure tree that shares common substructures
- unification by "dropping queries through tree"
- leaves correspond to unifiable formulae
- leaves are mapped to result occurrence URIs u_i^j (in database)

Results Evaluation: NTCIR dataset (100,000 XHTML+MathML documents: 63GBs, 297 MFormulae)
 \rightsquigarrow 10GBs RAM + 43 GBs URIs (on disk) \rightsquigarrow query answer times 3 – 70ms (avg = 11ms)

MATHWEBSEARCH aims at high-quality hits only, reported 434 hits:

1	2	3	4	5	6	7	8	9	10	11	12	13*	14	15	16	17*	18	19	20*	21	22
0	9	21	4	49	0	51	100+	100+	0	0	0	0	0	0	0	100+	0	0	0	0	0

- lots of hits (100+): general queries with multiple query variables
- few hits: specific queries with precise expressions.
- no hits: errors* in query or missing exact matches

Current Work: Full-Text Search, Ranking, Extensions, Embedding

Full Text Search (TeMaSearch)

Idea: Formulae as words in Lucene, MATHWEBSEARCH for Query Expansion

- Replace text formulae by their index id \rightsquigarrow index in Lucene
- Unify query formulae via MATHWEBSEARCH \rightsquigarrow replace by index ids in query
- Augment Lucene for math results presentation (as above)

Searching other Corpora

e.g. Spreadsheet formulae \rightsquigarrow just encode in content MathML!

Searching the Mathematical Knowledge Space

FlatSearch DEMO

Variables that start with "*" are converted to MWS query variables, the rest are literal.

Query: $(X + Y) + Z == ?q$

Result: <http://latin.omdoc.org/math?IntArith?c/assoc>

assoc: $(X + Y) + Z == X + (Y + Z)$

Justification:
Induced statement found in <http://latin.omdoc.org/math?IntArith>
IntArith is a AbelianGroup if we interpret over view c
AbelianGroup contains the statement assoc

assoc: $(X + Y) + Z == X + (Y + Z)$