Formats for Topics and Result Submissions for the MathIR Task at NTCIR-12

Michael Kohlhase
Jacobs University
http://kwarc.info/kohlhase

January 12, 2016

Abstract

This document presents the formats of the challenge queries and results for the MathIR Task at NTCIR-12. The document will be specified in further and clarified in a discussion process with the NTCIR12-MathIR participants.

Contents

1 Introduction

NTCIR is an evaluation project and conference aimed at advancing the research of Information Access technologies such as Information Retrieval, Text Summarization, Information Extraction, and Question Answering [NTC]. NTCIR-12 features the MathIR (Math Information Retrieval; MIR) task as a successor to the very successful Math pilot task at NTCIR-10; and the subsequent Math-2 task at NTCIR-11 see [Aiz+14; AKO13] for an overview.

This document presents the format of topics (challenge queries) and result submissions for the MathIR Task at NTCIR-12 [NTM].

Participants receive the NTCIR-Math dataset which contains two sub-corpora:

1. The **wikipedia** subcorpus of over 300,000 articles. 1/10th (roughly 30,000 articles) contain explicit `<math>` tags for formulae, while the remaining 9/10ths do not. Over 560,000 formulae are annotated with `<math>` tags in the corpus. These articles were extracted from a snapshot of English Wikipedia (see CorpusOverview.md file in the corpus for additional details).
2. the **arXiv** subcorpus of 100 000 HTML5 full texts of articles from the Cornell e-print arXiv [ArX].

The first contains topic-centered articles about well-known mathematics, addressed at a literate, but non-specialist audience. The second is a collection of research articles aimed at mathematics researchers.

The arXiv subcorpus was created by transforming the LaTeX from `http://arxiv.org` with the LaTeXML system [LTX]. The wikipedia subcorpus was converted from an English Wikipedia snapshot in mediawiki format, with formulae in `<math>` tags and (some) mediawiki math templates converted to MathML using LaTeXML (see the CorpusOverview.md file for additional details).

Formulae are marked up in three forms *i*) content MathML *ii*) presentation MathML, and *iii*) LaTeX source; see 2 for details. Figure 1 shows simplified version of the XML representation of the simple arithmetic expression $\left(\frac{p-2}{p-1}\right)^{p-1}$. Section 2 discusses the details of formula markup.

The NTCIR-12 datasets contain the original documents in HTML5 and XHTML5 (HTML5 serialized as XML) form as well as the NTCIR-12 **retrieval units**: XHTML5 files that contain the smallest sectioning units (usually logical paragraphs; ca. 80/paper in the arXiv subcorpus).

The MathIR Task at NTCIR-12 is a full-text information retrieval task. Participating IR systems obtain a list of queries consisting of words and formulae (possibly) with wildcards (query variables and similarity regions) and return for every query an ordered list retrieval units that are claimed to match the query, plus possible supporting evidence (e.g. the identifiers of formulae and the substitution for query variables). Results will be evaluated using a standard IR evaluation measures, Precision@k and Mean Average Precision (MAP).

2 Formula Markup

The MathML **m:math** element indicates a formula, the **m:semantics** element (see [Aus+10, Chapter 5] for a discussion) combines the three representations of the formula:

i) The first child contains the **content MathML**, which marks up the *content structure* of the formula in an "operator tree" using **m:apply** (function/relation application), **m:ci** (identifiers), **m:cn** (numbers), as well as the specific operators **m:exp**, **m:divide**, and **m:minus**.

For a full list of structural elements and operators see [Aus+10, Chapter 4].

ii) The **m:annotation−xml** element contains the **presentation MathML**, which marks up the *visual appearance* in a "layout tree", using **m:mrow** (horizontal rows) **m:msup** (upper index) **m:mfrac** (fractions), **m:mo** (mathematical operators), **m:mn** (numbers), and **m:mi** (identifiers). For a full list of layout primitives see [Aus+10, Chapter 3].

iii) The **m:annotation** element contains LaTeX source of the formula – this is the source format the author provided the formula in.

```
<m:math id="fid1">
  <m:semantics>
    <m:apply>
      <m:exp/>
      <m:apply>
        <m:divide/>
        <m:apply><m:minus/><m:ci>p</m:ci><m:cn>2</m:cn></m:apply>
        <m:apply><m:minus/><m:ci>p</m:ci><m:cn>1</m:cn></m:apply>
      </m:apply>
      <m:apply><m:minus/><m:ci>p</m:ci><m:cn>1</m:cn></m:apply>
    </m:apply>
    <m:annotation−xml id="fid2" encoding="MathML−Presentation">
      <m:msup>
        <m:mrow>
          <m:mo fence="true">(</m:mo>
          <m:mfrac>
            <m:mrow><m:mi>p</m:mi><m:mo>−</m:mo><m:mn>2</m:mn></m:mrow>
            <m:mrow><m:mi>p</m:mi><m:mo>−</m:mo><m:mn>1</m:mn></m:mrow>
          </m:mfrac>
          <m:mo fence="true">)</m:mo>
        </m:mrow>
        <m:mrow><m:mi>p</m:mi><m:mo>−</m:mo><m:mn>1</m:mn></m:mrow>
      </m:msup>
    </m:annotation−xml>
    <m:annotation encoding="application/x−tex">
      \left(\frac{p−2}{p−1}\right)^{p−1}
    </m:annotation>
  </m:semantics>
</m:math>
```

Figure 1: XML encoding for MathML Formulae

Note that the two MathML representations are generated from the LaTeX source by a heuristic process (concretely the LaTeXML program [LTX]). As LaTeX is a presentation-oriented format, the generated presentation MathML usually leads to a visual appearance (e.g. when viewed in Firefox) that is quite close to the one generated by LaTeX itself, but the "semantic features" of presentation MathML (grouping of subexpressions via **m:mrow**, invisible operators, etc. are artefacts of the "semantic parsing" in LaTeXML). The heuristic nature of the transformation has a much more profound effect in the content MathML which is a semantic format. Mathematical notation is ambiguous: An expression like x^2 could mean "x squared" or "the second element in the sequence $x^1, x^2, x^3, \ldots$. Similarly, an expression $f(x+3)$ could

be "the application of the function f to the sum of x and 3" or "f times the sum of x and 3". In all such cases, LaTeXML makes a heuristic choice in generating the content MathML, sometimes resorting to underspecified content MathML of the form

```
<m:apply>
  <m:csymbol cd="ambiguous">superscript</m:csymbol>
  <m:ci>x</m:ci>
  <m:cn>2</m:cn>
</m:apply>
```

Note finally, that the example in Figure 1 is a didactically motivated example. The actual formulae in the NTCIR dataset contain more metadata (source references), identifiers, and cross-references.[1] EdN:1

3 Query Language and Topic Format for the NTCIR-12 MathIR Evaluation Task

NTCIR Topics are specifications of an information need of an individual[2] A MathIR topic consists of EdN:2

- a *machine-understandable query* – given to the task participants as a basis for generating results and
- further *specification of the information need* that enables human judges to evaluate the submitted results. This information is reserved to the evaluators, i.e. it is not given to the participants.

Note that the query language employed need not be able to fully express the information need. In fact the semantics of the query language does not even need to be fully specified, as the evaluators judge the relevance of the results with respect to the full information need. This flexibility allows us to gain insights on the suitability and interpretation for query languages for a special domain like mathematics.

As the NTCIR-12 MathIR Task query language, we use a mixture of key phrases and formula queries. The interpretation of the key phrases and their combination with formula queries is left unspecified (see above). Formula queries may contain

1. named **query variables** that act as wildcards (see Section 3.1)
2. named **similarity regions** induce relaxed matching (see Section 3.2)

Query variables and similarity regions were included in the MathIR task even though they are non-standard for information retrieval because they were determined to be an important feature for adequately expressing information needs by the previous evaluation tasks and other user studies: The judge's feedback and participants discussion at the MIR Happening at CICM 2013 [MIRH] positively confirmed the necessity of formula queries with wildcards for expressing information needs; The "abysmal" performance of "bag-of-symbol approaches" in NTCIR-10/11, e.g. [LRG13], negatively confirms this assumption.

Before we present the NTCIR-12 MathIR topic format in Section 3.3, we present the mathematical intuition behind the two query primitives.

[1] EdNote: MK@MK: maybe put an example into the appendix and explain things in more detail there?
[2] EdNote: MK@MK: continue: classify the search task here; what is the search scenario?

3.1 Query Variables

We will use mathematical notation with MIR annotations to convey the concepts and leave the actual XML-based query format to the Section 3.3. Consider for instance:

$$\frac{f(v+d)-f(v)}{d} \tag{1}$$

(1) is a typical example for a formula query with query variables f, v, and i[1]. We will write query a query variable with name "x" as a red identifier x in our examples. Query variables stand for arbitrary subexpressions in formulae, thus the query (1) matches the equation

$$g'(cx) = \lim_{h\to 0} \frac{g(cx+h)-g(cx)}{h} \tag{2}$$

as we can substitute g for f, cx for v, and h for i in (1) to obtain the subformula $\frac{g(cx+h)-g(cx)}{h}$ at position π[2] of (2). In particular, the subformula occurrence π and the substitution $f \mapsto g, v \mapsto cx, i \mapsto h$ fully describe the match – we call them a **justification**; it should be provided as part of the result.

As we have seen above, query variables allow to express complex information needs by abstracting over subformulae. Note that query variables with the same name must be instantiated by the same subformula, in our example above this would have precluded unwanted numerators like $g(cx+h)-h(cx)$ or $g(cx+h)-g(dx)$.

Also note that the name of a query variable does not carry any query information, it is simply an identifier for the query variable that allows to specify substitutions for justifications.

Finally note furthermore that depending on whether we express the query in content or presentation MathML we may obtain different results: presentation MathML distinguishes the variants $\frac{n}{d}$, $n:d$ and n/d of a fraction, while content MathML only sees them as applications of the division function to n and d.

3.2 Similarity Regions (Experimental)

In the query (1), matching was assumed to be exact, which is not always what we want. Using similarity region, we can adapt this behavior to similarity search. For instance, we can use the query of the form

$$\frac{\boxed{a}\,\underline{g(cx+h)-g(cx)}}{h} \tag{3}$$

where $\boxed{a}\,\underline{A}$ denotes a similarity region, with **name** "a" and **body** A. Depending on the notion of similarity intended – which we do not specify – query (3) might match (4) – if addition is similar to subtraction, or (5) – assuming c is somehow similar to d.

$$\frac{g(cx+h)+g(cx)}{h} \tag{4}$$

[1] Actually in the LaTeX sense, ? is an active character that expands to an internal macro \qvar@construct that marks up a query variable. In fact in the form is still visible in the LaTeX facet of the query (see ⟨LaTeX⟩ in Figure 2)

[2] We assume some way of identifying subformulae by position here. As formulae are represented in MathML, this can be done by XPath or subformula ids.

$$\frac{g(cx+h)-g(dx)}{h} \tag{5}$$

The justification of a similarity region is also a substitution – here $a \mapsto g(cx+h)+g(cx)$ for (4) or $a \mapsto g(cx+h)-g(dx)$ for (5) which allow to compare the region and the subformula of the hit.

Of course, query variables and similarity regions can be combined, e.g. in

$$\frac{\boxed{a}\ f(v+d)+f(v)}{d} \tag{6}$$

which matches the equation in (2). Here a joint justification for would be the subformula $\mathcal{S}_1$ together with the substitution $f \mapsto g, v \mapsto cx, i \mapsto h, a \mapsto g(cx+h)-g(cx)$: after applying this substitution to $\mathcal{Q}_3$, we end up with $\frac{g(cx+h)+g(cx)}{h}$, which is similar to $\mathcal{S}_1$. Again, the name of a similarity region does not carry any query information.

Of course, sometimes we want to insist that certain parts of a similarity region should match exactly, e.g. the subtraction operator in (3) to exclude a hit like (4). Then we can specify **regions of exact matching** inside a similarity region e.g.

$$\frac{\boxed{a}\ g(cx+h)-g(cx)}{h} \tag{7}$$

In (7) we show the exact matching region as a white hole in the similarity region. Note that this allows to arbitrarily nest similarity and exact regions. Note furthermore that exact regions do not need a separate justification, and thus do not have a name of their own.

3.3 Topic Format

The general form of queries is shown in Figure 2 – see appendices A.1 and A.2 for RelaxNG schemata; and appendices A.3 and A.4 for examples. The top-level element is the topics element that specifies the three XML namespaces involved and their prefixes: the default namespace (no prefix) for NTCIR MathIR topics, and the MathML namespace with prefix **m:** and the MathIR namespace with prefix mws:. The children of this element are topic elements.

Each MIR topic has an identifier ⟪qid⟫ of the form NTCIR12−⟪part⟫−⟪num⟫ in the num element. The query itself is given in the query element and consists of a list of words and formula schemata given the keyword and formula elements – they can appear in any order; in particular their order may matter for the search.

The contents of the keyword element is a UniCode string in UTF-8 encoding which is interpreted as a single search keyword. The keyword and formula elements carry an id element with an identifier that can be used for identification in the results (see Section 4).

As above the content of the formula element is a MathML formula query in parallel markup (see section 5.4 in [Aus+10]); ⟪CMML⟫ in content MathML, ⟪PMML⟫ in presentation MathML, and ⟪LaTeX⟫ as the LaTeX source. A MathML **formula query** is a MathML expression that additionally allows mws:qvar elements for query variables, mws:simto, and mws:exact elements for similarity/exact regions. The first two carry a name attribute that for the name used in the justifications. The mws:qvar elements are empty and the latter two contain a MathML expression. For the query $\mathcal{Q}_3$ above would have the form given in Figure 3.

In ⟪LaTeX⟫, we simply use ? for query variables and the \simto and \exact macros for similarity/exact regions. With this, ⟪LaTeX⟫becomes **frac**{\simto{a}{?f(?v+?d)\exact{+}?f(?v)}}{?d}

```
<?xml version="1.0" encoding="utf−8"?>
<topics xmlns="http://ntcir−math.nii.ac.jp/"
        xmlns:m="http://www.w3.org/1998/Math/MathML"
        xmlns:mws="http://search.mathweb.org/ns">
  <topic>
    <num>⟪qid⟫</num>
    <query>
      <keyword id="⟪id⟫">⟪keyword 1⟫</keyword>
      <formula id="⟪id⟫">
        <m:math>
          <m:semantics>
            ⟪PMML⟫
            <m:annotation−xml encoding="MathML−Presentation">
              ⟪CMML⟫
            </m:annotation−xml>
            <m:annotation encoding="application/x−tex">⟪LaTeX⟫</m:annotation>
          </m:semantics>
        </m:math>
      </formula>
      ...
      <formula id="⟪id⟫">...</formula>
      ...
      <keyword id="⟪id⟫">⟪keyword n⟫</keyword>
    </query>
  </topic>
  <topic>...</topic>
</topics>
```

Figure 2: Machine-Readable form of Queries

We discussed above that the conversion to MathML is heuristic and not always optimal. But if we use the same conversion in the dataset and queries, errors often cancel out in practice. In any case, the **<m:annotation>** child of any MathML expression in formula queries and the data set has the original LaTeX formula, so alternative conversions can be employed.

4 Reporting Results

Participants report the results of up to four "runs" of their search engine with the given queries (see Section 3) over the dataset supplied. A "run" is your system outcome for a given approach. In general, you can imagine a run as the outcome of a specific variant of your system testing a specific hypothesis. Each run contains exactly 1000 search results. Note that even if the search method only finds fewer results, then the 1000 have to be filled up – e.g. by random selection.

See Section 5 for details on the submission. Results can be reported in two forms:

⟪CMML⟫

```
<m:mfrac>
  <mws:simto name="a">
    <m:mrow>
      <mws:qvar name="f"/>
      <m:mrow>
        <m:mo fence="true">(</m:mo>
        <mws:qvar name="v"/>
        <m:mo>+</m:mo>
        <mws:qvar name="d"/>
        <m:mo fence="true">)</m:mo>
      </m:mrow>
      <mws:exact>
       <m:mo>+</m:mo>
      </mws:exact>
      <m:mrow>
      <mws:qvar name="f"/>
        <m:mo fence="true">(</m:mo>
        <mws:qvar name="v"/>
        <m:mo fence="true">)</m:mo>
      </m:mrow>
    </m:mrow>
  </mws:simto>
  <mws:qvar name="d"/>
</m:mfrac>
```

⟪PMML⟫

```
<m:apply>
  <m:divide/>
  <mws:simto name="a">
    <m:apply>
      <mws:exact>
        <m:plus/>
     </mws:exact>
      <m:apply>
        <mws:qvar name="f"/>
        <m:apply>
          <m:plus/>
          <mws:qvar name="v"/>
          <mws:qvar name="d"/>
        </m:apply>
      </m:apply>
      <m:apply>
        <mws:qvar name="f"/>
        <mws:qvar name="v"/>
      </m:apply>
    </m:apply>
  </mws:simto>
  <mws:qvar name="d"/>
</m:apply>
```

Figure 3: MathML representations for the Query $\frac{\boxed{a}}{d}$ in Figure 2

4.1 XML Reports

Results should be reported in an XML file structured as in Figure 4 (see Appendix A.5 for a RelaxNG schema). Use naming convention above, with ⟪ext⟫=xml.

For each run there is a run element whose runtag attribute specifies the run identifier of the form ⟪group-id⟫_⟪run-id⟫, where ⟪run-id⟫ can be chosen freely. Runs can be manual or automatic, correspondingly the value of ⟪type⟫ is one of manual or automatic. For each query there is a result element, whose for attribute identifies the query. The total run time in milliseconds used to answer the query – including query processing and result generation – is specified in the runtime attribute; its value should be an integer. This attribute is mandatory for automatic runs.

The result element contains a list of hit elements that identify a file from the NTCIR-12 dataset purported to match that query via the its name ⟪filename⟫ in the xref attribute. ⟪rank⟫ and ⟪score⟫ are as above and underlie the same constraints.

1. ⟪filename⟫ the full path of the file that purportedly matches that query in the NTCIR12-MathIR dataset, e.g. `xhtml/1/hep-th0007174/hep-th0007174_1_1.xhtml`.
2. ⟪rank⟫ gives the rank of the "hit" in the answers to the particular query. ⟪rank⟫ should start from 1 and should be always incremental (by one, i.e. the rank numbers just represent the number of lines; no documents have the same rank).

```
<?xml version="1.0" encoding="utf-8"?>
<results xmlns="http://ntcir-math.nii.ac.jp/">
  <run runtag="⟪runtag⟫" runtime="⟪ms⟫" run_type="⟪type⟫">
    <result for="⟪qid⟫" runtime="⟪ms⟫">
      <hit id="⟪id⟫" xref="⟪filename⟫" score="⟪score⟫" rank="⟪rank⟫">
      ⟪see Figure 5⟫
      </hit>
      <hit>...</hit>
    </result>
    ...
    <result>...</result>
  </run>
  ...
  <run>...</run>
</results>
```

Figure 4: Returning Results in XML

3. ⟪score⟫ specifies how "happy" your system is with the result. Make sure that ⟪score⟫ and ⟪rank⟫ are consistent; for example, $\langle\!\langle \text{rank} \rangle\!\rangle_i < \langle\!\langle \text{rank} \rangle\!\rangle_j$ iff $\langle\!\langle \text{score} \rangle\!\rangle_i > \langle\!\langle \text{score} \rangle\!\rangle_j$.

```
<hit id="⟪id⟫" xref="⟪filename⟫"
     score="⟪score⟫" rank="⟪rank⟫">
  <formula id="⟪id⟫" for="⟪idref⟫"
           xref="⟪fref⟫" score="⟪score⟫">
    <qvar for="⟪name⟫" xref="⟪sub-fref⟫"/>
    ...
    <qvar .../>
  </formula>
  <keyword id="⟪id⟫" for="⟪idref⟫"
           xref="⟪kref⟫" score="⟪score⟫"/>
  ...
  <formula ...>...</formula>
</hit>
```

e problem of estimating the Lebesgue c
proven to be finite for all $k \geq 1$ and $1 \leq$
ber estimates for the numbers $c_{k,m}$ for v
rem 1. The estimate $\mu_k \leq c_{k,k} \leq A\,\mu_k$ hol

$$\mu_k = \frac{(k+1)(k+2)}{2} \int_{-1}^{1} |I_k(t)|\,dt,$$

nial, resp. a Jacobi polynomial. Theoren
extensive use of the theory of classical

Figure 5: Justifying and Highlighting Hits

A hit should be further specified by formula and keyword elements in the hit element. Their for attributes identify the respective keyword or formula schema from the query by referencing its id attribute and specify the exact occurrences in the file via their xref attributes. The value of the xref attribute is a URI reference of the form ⟪filename⟫#⟪id⟫, where ⟪filename⟫ is the file in the dataset with the hit and ⟪idref⟫ the value id attribute of the keyword or formula in the source. Individual keyword/formula occurrences can be given a ⟪score⟫ as well in the optional score attribute.

formula elements can be (optionally[3]) further justified by giving a substitution for the query variables: For each wildcard (query variable mws:qvar whose name attribute has value ⟪name⟫) in the query, the subformula it matches can be given in a qvar element. Its for attribute specifies

[3]Note that the justifications will not be judged per se, but will make life of the evaluators easier and allow them to understand why the hit is justified.

the ⟪name⟫ and its xref attribute points to the id of the subformula.

4.2 Reporting Results in CSV Format

Teams who have problems generating valid XML described above, can report results as lists of tuples (one line per hit, columns separated by blanks) in a text file (naming conventions as above, but with ⟪ext⟫=csv).

⟪qid⟫ 1 ⟪filename⟫ ⟪rank⟫ ⟪score⟫ ⟪runtag⟫ ⟪ms⟫ ⟪type⟫ ⟪subst⟫

where

1. The first column contains the query identifier ⟪qid⟫
2. The second one is fixed to 1.
3. ⟪filename⟫, ⟪rank⟫, and ⟪score⟫ are as above
4. ⟪runtag⟫/⟪type⟫ identify the "system run" (see above) and ⟪ms⟫ its runtime.
5. Finally, ⟪subst⟫ gives a substitution, i.e. a list of scored assignments from keyword/formula identifiers to references

   [⟪name 1⟫:=⟪idref 1⟫!⟪score 1⟫,...,⟪name n⟫:=⟪idref n⟫!⟪score n⟫]

   Here ⟪name i⟫ is the name of a keyword or formula, ⟪idref i⟫ a reference to the identifier in the source, and (optionally; leave out the ! if no score is given) ⟪score⟫ the local score.

   Note that the representation of the substitution may not contain blanks to ease parsing.

The MathIR task organizers will provide a converter to XML format described in Section 4.1, so that the RelaxNG schema can be used for validation.

5 Submission

Submission of NTCIR-21 MathIR results is personalized; each participant can only see their own submission. An account and password will be provided to the teams by the NTCIR-12 MathIR organizers.

Participants can submit the results of four runs in XML format (preferred see 4.1) or CSV format (see 4.2) to `https://ntcir-math.nii.ac.jp/upload`; follow the instructions there. Files should be named as ⟪group-id⟫.⟪ext⟫ where ⟪group-id⟫ is the group identifier you have chosen upon NTCIR-12 registration.

The current individual file size limit is 128M. Should that prove insufficient, please let the organizers know by e-mail to `ntcadm-math@nii.ac.jp`.

6 Conclusion

We have specified the query and result reporting formats for the NTCIR-12 MathIR task. This specification is a best-effort description of the intended format, but may be incomplete or contain errors. In such cases, please contact the organizers at `ntcadm-math@nii.ac.jp`.

References

[Aiz+14] Akiko Aizawa et al. "NTCIR-11 Math-2 Task Overview". In: *NTCIR Workshop 11 Meeting*. Ed. by Noriko Kando, Hideo Joho, and Kazuaki Kishida. Tokyo, Japan: NII, Tokyo, 2014, pp. 88–98. URL: http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/OVERVIEW/01-NTCIR11-OV-MATH-AizawaA.pdf.

[AKO13] Akiko Aizawa, Michael Kohlhase, and Iadh Ounis. "NTCIR-10 Math Pilot Task Overview". In: *NTCIR Workshop 10 Meeting*. Ed. by Noriko Kando and Kazuaki Kishida. Tokyo, Japan: NII, Tokyo, 2013, pp. 1–8. URL: http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings10/pdf/NTCIR/OVERVIEW/01-NTCIR10-OV-MATH-AizawaA.pdf.

[ArX] *arxiv.org e-Print archive*. URL: http://www.arxiv.org (visited on 06/12/2012).

[Aus+10] Ron Ausbrooks et al. *Mathematical Markup Language (MathML) Version 3.0*. W3C Recommendation. World Wide Web Consortium (W3C), 2010. URL: http://www.w3.org/TR/MathML3.

[Jing] *Jing — Relax NG Validator in Java*. URL: http://www.thaiopensource.com/relaxng/jing.html (visited on 01/15/2015).

[KK13] Noriko Kando and Kazuaki Kishida, eds. *NTCIR Workshop 10 Meeting*. Tokyo, Japan: NII, Tokyo, 2013.

[LRG13] Ray R. Larson, Chloe J. Reynolds, and Fredric C. Gey. "The Abject Failure of Keyword IR for Mathematics Search: Berkeley at NTCIR-10 Math". In: *NTCIR Workshop 10 Meeting*. Ed. by Noriko Kando and Kazuaki Kishida. Tokyo, Japan: NII, Tokyo, 2013, pp. 662–666. URL: http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings10/pdf/NTCIR/MATH/02-NTCIR10-MATH-LarsonRR.pdf.

[LTX] Bruce Miller. *LaTeXML: A LaTeX to XML Converter*. URL: http://dlmf.nist.gov/LaTeXML/ (visited on 03/12/2013).

[MIRH] *Math IR Happening at MIR 2012*. URL: http://cicm2012.cicm-conference.org/cicm.php?event=mir&menu=happening (visited on 03/18/2012).

[NTC] *Workshop Aims | NTCIR-11 | NTCIR*. URL: http://research.nii.ac.jp/ntcir/ntcir-11/aims.html (visited on 05/14/2014).

[NTM] *NTCIR-12 MathIR*. URL: http://ntcir-math.nii.ac.jp/ (visited on 12/18/2015).

[Vei] Daniel Veillard. *The XML C parser and toolkit of Gnome; libxml*. URL: http://xmlsoft.org (visited on 07/11/2011).

[Vli03] Eric van der Vlist. *Relax NG*. O'Reilly, 2003.

A Appendix

We provide the RelaxNG schemata and examples for the XML query and result formats for convenience, they can be found `https://kwarc.info/kohlhase/event/NTCIR12/topics/lib`.

RelaxNG schemata in XML (RNG) can be used to validate XML files with standard tools like Jing [Jing] or `xmllint` [Vei]. Schemata in compact form (RNC; as given below) can be used as human-oriented documentation if read like extended BNF grammars

- element defines XML elements, and their content model, e.g. element foo {bar$*$} defines an element foo that can have an arbitrary number of bar children
- attribute defines XML attributes with their content model, e.g. attribute foo {text} defines a text attribute foo.
- text allows arbitrary sequences of UniCode characters.
- = is used for creating abbreviations/macros, e.g. bar = element bar {foo$*$} allows to use the macro name bar, instead of the grammar specification element bar {foo$*$}.
- ?, +, and $*$ allow iteration: zero or one, one or more, and zero or more.
- ,, |, and & used for sequence, disjunction, and interleaving. a,b specifies a followed by b, a|b allows (one of) a or b, for interleaving consider a&b$*$, which allows any combination of one a with arbitrarily many b.
- start specifies the start symbol of the grammar, essentially giving the rule for the top element of the XML expressions accepted by this grammar.

A full description of RelaxNG can be found in [Vli03]. We use some of the modularity features in our schemata, in particular

- grammar encapsulates partial grammars, where parent allows to use macros from outside the encapsulated grammar
- include includes an external schema with adaptions: include ⟪file⟫ {⟪rules⟫| includes the grammar in ⟪file⟫ literally, but adds ⟪rules⟫ to it – possibly overwriting the rules in ⟪file⟫.

A.1 RelaxNG Schema for NTCIR Queries (Participants)

We first have a schema for the participants, see Appendix A.3 for an example that conforms to this schema.

```
# A RelaxNG for NTCIR−12 Math Task Topics (common part)
# (c) 2014 Michael Kohlhase, released under the GNU Public License (GPL)

namespace mws = "http://search.mathweb.org/ns"
namespace m = "http://www.w3.org/1998/Math/MathML"
namespace xsd = "http://www.w3.org/2001/XMLSchema−datatypes"
default namespace = "http://ntcir−math.nii.ac.jp/"

start = element topics {topic∗}

id.att = attribute id {xsd:ID}?
formula = element formula {id.att, math}
keyword = element keyword {id.att, inline.model}

name.att = attribute name {text}
qvar = element mws:qvar {name.att}

pmath = grammar {include "../../lib/mathml3/mathml3.rc" {start=PresentationExpression}
                                 PresentationExpression |= parent qvar | parent p−simto | parent p−exact
                                 ContExp |= parent qvar | parent c−simto | parent c−exact}
p−simto = element mws:simto {name.att,pmath}
p−exact = element mws:exact {pmath}

cmath = grammar {include "../../lib/mathml3/mathml3.rc" {start=ContExp}
                                 PresentationExpression |= parent qvar | parent p−simto | parent p−exact
                                 ContExp |= parent qvar | parent c−simto | parent c−exact}
c−simto = element mws:simto {name.att,cmath}
c−exact = element mws:exact {cmath}

math = grammar {include "../../lib/mathml3/mathml3.rc"
                                 PresentationExpression |= parent qvar | parent p−simto | parent p−exact
                                 ContExp |= parent qvar | parent c−simto | parent c−exact}

inline.class = math
inline.model = (text | inline.class)∗

query = element query {formula∗ & keyword∗}
num = element num {text}

topic.model = num & query
topic = element topic {topic.model}
```

A.2 RelaxNG Schema for NTCIR Queries (Judges)

And then we extend it for the judges with the private parts. See Appendix A.4 for an example that conforms to this schema.

```
# A RelaxNG for NTCIR−12 Math Task Topics
# (c) 2014 Michael Kohlhase, released under the GNU Public License (GPL)

namespace mws = "http://search.mathweb.org/ns"
namespace m = "http://www.w3.org/1998/Math/MathML"
namespace xsd = "http://www.w3.org/2001/XMLSchema−datatypes"
default namespace = "http://ntcir−math.nii.ac.jp/"

include "NTCIR12−topic−participants.rnc"

private = element private {examplehit* & contributor* & title? & note* & relevance? & reference? }
examplehit = element examplehit {attribute href {xsd:anyURI}}
contributor = element contributor {inline.model}
title = element title {inline.model}
relevance = element relevance {inline.model}
reference = element reference {xsd:anyURI}
note = element note {inline.model}

topic.model &= private
```

A.3 An Example Query (Participants)

We first have a an example of what the participants see in a query

```
<?xml version="1.0" encoding="UTF-8"?>
<topics xmlns="http://ntcir-math.nii.ac.jp/" xmlns:m="http://www.w3.org/1998/Math/MathML">
  <topic>
    <num>NTCIR12-MathIR-1</num>
    <query>
      <formula id="f.0">
        <m:math>
          <m:semantics xml:id="m1.1a">
            <m:apply xml:id="m1.1.4" xref="m1.1.4.pmml">
              <m:plus xml:id="m1.1.2" xref="m1.1.2.pmml"/>
              <m:ci xml:id="m1.1.1" xref="m1.1.1.pmml">a</m:ci>
              <mws:qvar xmlns:mws="http://search.mathweb.org/ns" name="b"/>
            </m:apply>
            <m:annotation-xml encoding="MathML-Presentation" xml:id="m1.1b">
              <m:mrow xml:id="m1.1.4.pmml" xref="m1.1.4">
                <m:mi xml:id="m1.1.1.pmml" xref="m1.1.1">a</m:mi>
                <m:mo xml:id="m1.1.2.pmml" xref="m1.1.2">+</m:mo>
                <mws:qvar xmlns:mws="http://search.mathweb.org/ns" name="b"/>
              </m:mrow>
            </m:annotation-xml>
            <m:annotation encoding="application/x-tex" xml:id="m1.1c">a+\qvar@construct{b}</m:annotation>
          </m:semantics>
        </m:math>
      </formula>
      <keyword id="w.0">Jack</keyword>
      <keyword id="w.1">Ripper</keyword>
    </query>

  </topic>
</topics>
```

A.4 An Example Query (Judges)

and then one of what the judges see. The private element with a title, a description of the information need (in the relevance element), and metadata about the contributor (this plays no role in the evaluation) and an example hit in the data set to show that the query is solvable.

```
<?xml version="1.0" encoding="UTF-8"?>
<topics xmlns="http://ntcir-math.nii.ac.jp/" xmlns:m="http://www.w3.org/1998/Math/MathML">
  <topic>
    <num>NTCIR12-MathIR-1</num>
    <query>
      <formula id="f.0">
        <m:math>
          <m:semantics xml:id="m1.1a">
            <m:apply xml:id="m1.1.4" xref="m1.1.4.pmml">
              <m:plus xml:id="m1.1.2" xref="m1.1.2.pmml"/>
              <m:ci xml:id="m1.1.1" xref="m1.1.1.pmml">a</m:ci>
              <mws:qvar xmlns:mws="http://search.mathweb.org/ns" name="b"/>
            </m:apply>
            <m:annotation-xml encoding="MathML-Presentation" xml:id="m1.1b">
              <m:mrow xml:id="m1.1.4.pmml" xref="m1.1.4">
                <m:mi xml:id="m1.1.1.pmml" xref="m1.1.1">a</m:mi>
                <m:mo xml:id="m1.1.2.pmml" xref="m1.1.2">+</m:mo>
                <mws:qvar xmlns:mws="http://search.mathweb.org/ns" name="b"/>
              </m:mrow>
            </m:annotation-xml>
            <m:annotation encoding="application/x-tex" xml:id="m1.1c">a+\qvar@construct{b}</m:annotation>
          </m:semantics>
        </m:math>
      </formula>
      <keyword id="w.0">Jack</keyword>
      <keyword id="w.1">Ripper</keyword>
    </query>
    <private>
      <title>Jack the Ripper adds up</title>
      <relevance>
The hits should give an answer to the question whether there is any connection
between Jack the Ripper and sums that start with a variable <m:math><m:semantics xml:id="m2.1a"><m:ci xml:id="m2.1.1" xref="
</relevance>
      <examplehit href="http://example.org/files/4711/0815.xhtml"/>
      <contributor>Michael Kohlhase</contributor>
    </private>
  </topic>
</topics>
```

A.5 RelaxNG Schema for NTCIR Results

```
# A RelaxNG for NTCIR-12 Math Task Results
# (c) 2014 Michael Kohlhase, released under the GNU Public License (GPL)

namespace xsd = "http://www.w3.org/2001/XMLSchema-datatypes"
default namespace = "http://ntcir-math.nii.ac.jp/"

start = element results {run+}

id.att = attribute id {xsd:ID}
runtag.att = attribute runtag {text}
for.att = attribute for {text}
xref.att = attribute xref {xsd:anyURI}
score.att = attribute score {xsd:decimal}?
runtime.att = attribute runtime {xsd:nonNegativeInteger}
manualrun.att = attribute run_type {"manual"}
autorun.att = attribute run_type {"automatic"}
run.atts = (autorun.att & runtime.att) | (manualrun.att & runtime.att?)
rank.att = attribute rank {xsd:positiveInteger}

run = element run {runtag.att & run.atts & result+}
result = element result {id.att & for.att & runtime.att & hit+}
hit = element hit {id.att & score.att & xref.att & rank.att & formula* & keyword*}
qvar = element qvar {for.att & xref.att}
kwform.att = id.att & score.att & xref.att & for.att
formula = element formula { kwform.att & qvar*}
keyword = element keyword {kwform.att}
```

A.6 An Example of Justified Results

```
<?xml version="1.0" encoding="utf-8"?>
<results xmlns="http://ntcir-math.nii.ac.jp/">
  <run runtag="ex1" runtime="34567" run_type="automatic">
    <result id="ex1.1" for="NTCIR12-Math-3" runtime="245">
      <hit id="foo" xref="filename" score=".7777" rank="1">
        <formula id="foof" for="idref1" xref="#fref" score=".555">
          <qvar for="name" xref="#fref.0.0.7.3"/>
        </formula>
        <keyword id="fook" for="idref2" xref="#kref" score="1.0"/>
      </hit>
    </result>
  </run>
</results>
```