`mathanno.sty`: Annotation Macros for Math Expressions

Deyan Ginev
Jacobs University, Bremen
`http://kwarc.info/dginev`

March 29, 2012

Abstract

This package provides macros for annotating LaTeX-authored mathematical expressions, with a focus on structural and syntactic properties.

Contents

1 Introduction

EdN:1

[1]

2 User Interface

EdN:2

[2]

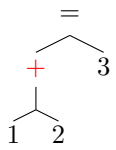
```
\documentclass{article}

An annotation for $1+2=3$:
@=(@+(1,2)) ???
==================
First try: $1+2\is{infix,relation,expression}{=}3$
```

Figure 1: Example of a Basic Annotation

Linear: $1 + 2{=}3$

Tree:



3 Exhaustive Feature List

4 Implementation

We proceed to doing the actual work on the LaTeX side of affairs.

To start things off, we provide Tikz-based tree building macros.

```
⟨*package⟩
\usepackage{color}
\usepackage{qtree}

% PDF annotation internals:
\def\tooltiptarget{\phantom{\rule{1mm}{1mm}}}
\newbox\tempboxa
\setbox\tempboxa=\hbox{}
\immediate\pdfxform\tempboxa
\edef\emptyicon{\the\pdflastxform}

```

[1] EdNote: we need this for the arXiv case study.

[2] EdNote: talk about keywords, trees, tikz

```
% Magic for ,
\begingroup
\lccode`\~=`\,%
\lowercase{\endgroup%
 \def~{\string\r}%
}%

\def\activatecomma{\begingroup\catcode`,=\active}
\def\deactivatecomma{\endgroup}
\def\MathTree{\Tree}

% Annotation macro:
\def\is{\activatecomma\annoi}
\newcommand\annoi[2]{%
\pdfstartlink user{%
  /Subtype /Text
  /Contents  (#1)
  /AP <<
    /N \emptyicon\space 0 R
  >>
}%
{\color{red}#2}%
\pdfendlink%
\deactivatecomma
}

\newcommand{\labelentry}{.}
\newcounter{entryi}
%%% ENTRIES for expression case study:
\newenvironment{nextentries}[2]%
{\begin{table}[hp]\def\capentries{#1}\def\labelentries{#2}%
 \begin{tabular}{|llll|}\hline  & Expression & Denotation & Annotation \\}%
{\hline\end{tabular}\caption{\capentries}\label{\labelentries}\end{table}}%

\newenvironment{entries}[2]%
{\setcounter{entryi}{1}\begin{nextentries}{#1}{#2}}
{\end{nextentries}}

\newcommand\entry[4]{\hline\\[-4mm] {\theentryi\labelentry}\stepcounter{entryi} & #1 & #2 & {\s
⟨/package⟩
```