How to give a talk?

Florian Rabe

University Erlangen-Nuremberg

2019

Example: Final Preparation of a Slide

MathML vs. TPTP: Logical Similarities

If we abstract from

- concrete syntax
- intended purpose
- user community
- tool support

the languages are quite similar:

MathML

- MathML objects: constants, variables, application, arbitrary binding
- all operators introduced/specified in content dictionaries

TPTP (since expansion towards higher-order logic)

- TPTP formulas: constants, variables, application, built-in binders $\forall\exists\lambda\Pi\Sigma$
- logic-related operators built-in, specified in various language references no fixed type systems, no fixed calculus
- other operators introduced/specified in TPTP files

MathML vs. TPTP: Logical Similarities

If we abstract from

- concrete syntax
- intended purpose
- user community
- tool support

the languages are quite similar:

MathML

- MathML objects: constants, variables, application, arbitrary binding
- all operators introduced/specified in content dictionaries

TPTP (since expansion towards higher-order logic)

- TPTP formulas: constants, variables, application, built-in binders $\forall\exists\lambda\Pi\Sigma$
- logic-related operators built-in, specified in various language references no fixed type systems, no fixed calculus
- other operators introduced/specified in TPTP files

MathML vs. TPTP: Logical Similarities

If we abstract from

- concrete syntax
- intended purpose
- user community
- tool support

the languages are quite similar:

MathML

- constants, variables, application, arbitrary binding
- all operators introduced/specified in content dictionaries

TPTP (since expansion towards higher-order logic)

- constants, variables, application, built-in binders $\forall\exists\lambda\Pi\Sigma$
- most operators introduced/specified in TPTP files
- built-in logic-related operators, specified in various language references

no fixed type systems, no fixed calculus

MathML vs. TPTP: Logical Similarities

If we abstract from

- concrete syntax
- intended purpose
- user community
- tool support

the languages are quite similar:

MathML

- constants, variables, application, arbitrary binding
- all operators introduced/specified in content dictionaries

TPTP (since expansion towards higher-order logic)

- constant, variables, application, built-in binders $\forall\exists\lambda\Pi\Sigma$
- most operators introduced/specified in TPTP files
- built-in logic-related operators, specified in various language references

no fixed type systems, no fixed calculus

MathML vs. TPTP: Logical Similarities

If we abstract from

- concrete syntax
- intended purpose
- user community
- tool support

the languages are quite similar:

MathML

- constants, variables, application, arbitrary binding
- all operators introduced/specified in content dictionaries

TPTP (since expansion towards higher-order logic)

- constant, variables, application, built-in binders $\forall\exists\lambda\Pi\Sigma$
- most operators introduced/specified in TPTP files
- built-in logic-related operators
  - semantics left open — no fixed type systems, no fixed calculus
  - specified in various language references — fof, tff, thf, thf1, ...

MathML vs. TPTP: Logical Similarities

Languages are quite similar if we abstract from

- concrete syntax
- intended purpose
- user community
- tool support

MathML

- constants, variables, application, arbitrary binding
- all operators introduced/specified in content dictionaries

TPTP (since expansion towards higher-order logic)

- constant, variables, application, built-in binders $\forall\exists\lambda\Pi\Sigma$
- most operators introduced/specified in TPTP files
- built-in logic-related operators
  - semantics left open — no fixed type systems, no fixed calculus
  - various dialects — fof, tff, thf, thf1, ...