Matita 0.99.1

Andrea Asperti
(Wilmer Ricciotti, Claudio Sacerdoti Coen)

Department of Computer Science, University of Bologna
Mura Anteo Zamboni 7, 40127, Bologna, ITALY
asperti@cs.unibo.it

Foundation of Mathematics for Computer-Aided Formalization
Padova, 9-11 January 2013
Matita (pencil) is an implementation of the Calculus of Inductive Constructions alternative to Coq.

Distinctive features (already in version 0.5.2 [CADE ’11])

- light
- completely functional
- native open terms [Matita team ’09 (a)]\(^1\)
- strong disambiguation facilities [Sacerdoti Coen, Zacchiroli ’04]
- small step execution of structured tactics (tinycals) [Sacerdoti Coen, Tassi, Zacchiroli ’06]
- good documentation of system’s internals

\(^1\)Matita Team: Asperti, Ricciotti, Sacerdoti Coen, Tassi
Training and sperimentation

A good environment for

- learning the practice of formal development and the internals of interactive provers.
- experimenting innovative ideas
A Mature System

Some Matita developments:

- **Number theory**: Properties of Möbius $\mu$, Euler $\varphi$ and Chebyshev $\Theta$ functions; Bertrand’s postulate [Asperti, Ricciotti ’12 (b)]

- **Constructive analysis**: Lebesgue’s dominated convergence theorem [Sacerdoti Coen, Tassi ’08]

- **Formal topology**: elements of pointless topology [Sacerdoti Coen, Tassi ’011]

- **Programming languages metatheory**: solution to the POPLmark challenge [Matita team ’11]

- **Compilers verification**: EU Project CerCo (Certified Complexity) for the verification of a formally certified complexity preserving compiler for the C programming language [CerCo].

- **Formal Complexity**: Formalization of aspects of Complexity Theory (reverse computational Complexity) [Asperti, Ricciotti ’12].
Main novelties in version 0.99.1

Huge refactoring and simplification effort

- bidirectional type inference [Matita team ’12]
- enhanced mechanism of unifications hints [Sacerdoti Coen, Tassi ’011]
- a new type for tactics [Matita team ’09 (b)]
- new “compact” syntax (partially inspired by SSReflect)
## New Syntax

<table>
<thead>
<tr>
<th>tactic</th>
<th>old syntax</th>
<th>new syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>introduction</td>
<td>intro aaa</td>
<td>#aaa</td>
</tr>
<tr>
<td>application</td>
<td>apply aaa</td>
<td>@aaa</td>
</tr>
<tr>
<td>rewriting</td>
<td>rewrite &gt; aaa</td>
<td>&gt;aaa</td>
</tr>
<tr>
<td>constructor</td>
<td>constructor n</td>
<td>%n</td>
</tr>
<tr>
<td>automation</td>
<td>auto depth=n</td>
<td>/n/</td>
</tr>
<tr>
<td>proof leaves</td>
<td>reflexivity/assumption</td>
<td>//</td>
</tr>
<tr>
<td>anonymous elim.</td>
<td>intro H; elim H</td>
<td>*</td>
</tr>
</tbody>
</table>

* behaviour:

\[
\Delta \vdash (\exists x : A.B) \rightarrow C \quad \Rightarrow \quad \Delta \vdash \forall x : A.B \rightarrow C
\]

\[
\Delta \vdash A \land B \rightarrow C \quad \Rightarrow \quad \Delta \vdash A \rightarrow B \rightarrow C
\]
An example (old syntax)

```latex
\textbf{theorem} \texttt{le}\_\texttt{exp}: \forall n, m, p: \texttt{nat}. \ O < p \rightarrow n \leq m \rightarrow p^n \leq p^m.
\texttt{apply} \ \texttt{nat}\_\texttt{elim2}
\texttt{[intros.}
\texttt{\qquad \texttt{apply} \ \texttt{lt}\_\texttt{O}\_\texttt{exp}. \texttt{assumption}}
\texttt{\qquad | \texttt{intros.}}
\texttt{\qquad \texttt{apply} \ \texttt{False}\_\texttt{ind}.}
\texttt{\qquad \texttt{apply} \ ( \texttt{le}\_\texttt{to}\_\texttt{not}\_\texttt{lt} \ ? \ ? \ ? \ H1).}
\texttt{\qquad \texttt{apply} \ \texttt{le}\_\texttt{O}\_\texttt{n}}
\texttt{\qquad | \texttt{intros.}}
\texttt{\qquad \texttt{simplify.}}
\texttt{\qquad \texttt{apply} \ \texttt{le}\_\texttt{times}}
\texttt{\qquad \qquad \texttt{[apply} \ \texttt{le}\_\texttt{n}}
\texttt{\qquad \qquad \qquad \texttt{| apply} \ H[\texttt{assumption}|\texttt{apply} \ \texttt{le}\_\texttt{S}\_\texttt{S}\_\texttt{to}\_\texttt{le}. \texttt{assumption}]}
\texttt{\qquad \qquad \]}\texttt{]}
\texttt{\]}\texttt{qed.}
```
An example (new syntax)

\textbf{theorem} le\_exp: \( \forall n,m,p: \text{nat}. \ O < p \rightarrow n \leq m \rightarrow p^n \leq p^m. \)

@nat\_elim2 \#n \#m

[\#ltm \#len @lt\_O\_exp //
|\#_ \#len @False\_ind /2/
|\#Hind \#p \#posp \#lenm normalize @le\_times // @Hind /2/
]

\textbf{qed}.

\textit{compact and elegant}
## Size comparison

<table>
<thead>
<tr>
<th>file</th>
<th>Matita 0.5.2</th>
<th>Matita 0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>logarithms</td>
<td>413 (20)</td>
<td>223 (21)</td>
</tr>
<tr>
<td>square root</td>
<td>217 (13)</td>
<td>221 (19)</td>
</tr>
<tr>
<td>binomial coeff.</td>
<td>259 (9)</td>
<td>192 (12)</td>
</tr>
<tr>
<td>order of primes</td>
<td>656 (33)</td>
<td>411 (37)</td>
</tr>
<tr>
<td>big operators</td>
<td>978 (30)</td>
<td>425 (27)</td>
</tr>
<tr>
<td>sigma and pi</td>
<td>526 (26)</td>
<td>188 (9)</td>
</tr>
<tr>
<td>factorial</td>
<td>325 (14)</td>
<td>145 (12)</td>
</tr>
<tr>
<td>chebyshev’s theta</td>
<td>486 (13)</td>
<td>213 (13)</td>
</tr>
<tr>
<td>chebishev’s psi</td>
<td>294 (11)</td>
<td>143 (13)</td>
</tr>
<tr>
<td>factorization</td>
<td>927 (25)</td>
<td>629 (32)</td>
</tr>
<tr>
<td>psi bounds</td>
<td>1123 (37)</td>
<td>507 (30)</td>
</tr>
<tr>
<td>bertrand (up)</td>
<td>683 (18)</td>
<td>446 (27)</td>
</tr>
<tr>
<td>bertrand (down)</td>
<td>526 (22)</td>
<td>240 (19)</td>
</tr>
<tr>
<td><strong>total</strong></td>
<td><strong>7413 (271)</strong></td>
<td><strong>3983 (271)</strong></td>
</tr>
</tbody>
</table>

Matita 0.5.2  27 lines per theorem  
Matita 0.99.1  15 lines per theorem
## Tactic invocations

<table>
<thead>
<tr>
<th>Tactic</th>
<th>Matita 0.5.2</th>
<th>Matita 0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>tactic name no.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>application</td>
<td></td>
<td></td>
</tr>
<tr>
<td>assumption</td>
<td>apply 2203</td>
<td>@ 1792</td>
</tr>
<tr>
<td>rewriting</td>
<td>rewrite 1110</td>
<td>&lt; / &gt; 984</td>
</tr>
<tr>
<td>assumption</td>
<td>rewrite 2203</td>
<td></td>
</tr>
<tr>
<td>reflexivity</td>
<td>rewrite 244</td>
<td></td>
</tr>
<tr>
<td>simplification</td>
<td>simplify 255</td>
<td>normalize 122</td>
</tr>
<tr>
<td>introduction</td>
<td>intro/intros 435</td>
<td># 1904</td>
</tr>
<tr>
<td>elimination</td>
<td>cases 306</td>
<td>cases 190</td>
</tr>
<tr>
<td></td>
<td>elim 131</td>
<td>elim 92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* 62</td>
</tr>
<tr>
<td>cut</td>
<td>cut 89</td>
<td>cut 148</td>
</tr>
<tr>
<td>automation</td>
<td>auto 10</td>
<td>// 943</td>
</tr>
</tbody>
</table>
Pushing automation

Arithmetics with (2) and without automation (1) [Asperti, Sacerdoti Coen ’09]
Philosophy

Representing a proof as a set of (possibly annotated, possibly structured) names (the \textit{relevant} facts used in the proof).

Portable!?!?

Exploit the interpretative capabilities of systems.
To be tested for scalability (it certainly works for small proofs).

Looking for partners to test the idea.
Representing a proof as a set of (possibly annotated, possibly structured) names (the *relevant* facts used in the proof).

**Portable!?!**

Exploit the interpretative capabilities of systems.
To be tested for scalability (it certainly works for small proofs).

Looking for partners to tests the idea.


