

# Complex invariants in structured documents

Dario Teixeira

nleyten.com

April 2013

# Talk overview

- Leveraging type system to encode document invariants:
  - The nested link problem.
  - The phantom-type approach.
  - The problem with phantom-types.
  - Generalized Algebraic Data Types (GADTs) to the rescue.
  - GADTs as a limited form of dependent typing.
  - Type-system trickery in practice.

# Talk overview

- Context:
  - The [Nleyten](#) project.
  - The [Lambddoc](#) library.
- Leveraging type system to encode document invariants:
  - The nested link problem.
  - The phantom-type approach.
  - The problem with phantom-types.
  - Generalized Algebraic Data Types (GADTs) to the rescue.
  - GADTs as a limited form of dependent typing.
  - Type-system trickery in practice.

# Talk overview

- Context:
  - The [Nleyten](#) project.
  - The [Lambdoc](#) library.
- OCaml for Haskellers:
  - A brief overview.
  - Polymorphic variants (a.k.a. *open unions*).
  - Phantom-types in OCaml.
- Leveraging type system to encode document invariants:
  - The nested link problem.
  - The phantom-type approach.
  - The problem with phantom-types.
  - Generalized Algebraic Data Types (GADTs) to the rescue.
  - GADTs as a limited form of dependent typing.
  - Type-system trickery in practice.

# Context: the Nleyten project

- Take web communities with user-contributed content to the next level.
- Make web communities resilient and scalable.
- Build on a strong foundation: OCaml and Ocsigen.
- First community:  $\lambda^3$ .

# Lambdoc: overview and features

- Open-source library offering support for semantically rich documents in web applications.
- Built in [OCaml](#), with [Ocsigen/Eliom](#) integration in mind.
- Includes parsers for three distinct markups: Lambtex, Lambxml, and Lambwiki. Support for Markdown also planned (pending progress on Markdown standardisation!)
- Supports runtime customisation of available features (e.g. based on user permissions).
- Offers control over macro/inline/block depth.
- Detailed error messages.
- Ships with CLI utility lambcmd:

```
lambcmd -f lambtex -t html -i source.tex -o index.html
```

# Lambtex

- Syntax inspired by L<sup>A</sup>T<sub>E</sub>X (but without the hairy stuff).
- Designed with brevity and consistency in mind.
- L<sup>A</sup>T<sub>E</sub>X users should grok it immediately.
- Newbies will find it much simpler than L<sup>A</sup>T<sub>E</sub>X.
- Supports all the features of the Lambdoc library.
- Complete manual available!

```
\section[sec:intro]{Introduction}
```

This is a paragraph with some `\emph{emphasised}` text and a link to the `\link{http://www.wikipedia.org/}{Wikipedia}`.

This is another paragraph. Note that paragraphs are separated by a blank line. Below is an unordered list:

```
\begin{itemize}
\item Alpha
\item Beta
\item Gamma
\end{itemize}
```

# Lambwiki

- Light-weight syntax in the guise of Markdown or Wiki markups.
- Largely compatible with [WikiCreole](#).
- Does not cover full-range of Lambdoc features.
- Well-suited for writing comments.
- [Complete manual available!](#)

## = Introduction

This is a paragraph with some //emphasised// text and \*\*bold\*\* text and a link to [[<http://www.wikipedia.org/>|Wikipedia]].

> This is a quotation.

- Alpha
- Beta
- Gamma

# Lambxml

- XML-based markup, largely compatible with XHTML.
- Not particularly suited for manual input, but easily exported by XML-based tools.
- Element names identical to Lambtex.
- Supports all the features of the Lambdoc library.

```
<h1>Introduction</h1>

<p>This is a paragraph with some <em>emphasised</em> text and a link
to the <a href="http://www.wikipedia.org/">Wikipedia</a>.</p>

<p>This is another paragraph. Below is an unordered list:</p>

<ul>
<li>Alpha</li>
<li>Beta</li>
<li>Gamma</li>
</ul>
```

# Lambdoc elements (I)

## A COMPLETE SAMPLE OF LAMBTEX

DARIO TEIXEIRA

MARCH 2013

### Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

### Table of Contents

- I [Inline features](#)
- II [Block features](#)
  - 1 [Text paragraphs](#)
  - 2 [Unordered lists](#)
  - 3 [Ordered lists](#)
  - 4 [Description lists](#)
  - 5 [Q&A environments](#)
  - 6 [Verse environments](#)

# Lambdoc elements (II)

## I Inline Features

**L**orem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

Here we have examples of text in **bold** and *emphasised* and **monospaced** and in SMALL-CAPS and with superscripts and subscripts. You may also correct ~~perfectbroken~~ text. Also, *all of this emphasised text should appear on the same line*.

It is possible to enter HTML entities either by name (`&euro;` produces '€'), by decimal code point (`&#8364;` produces '€'), or by hexadecimal code point (`&#x20ac;` produces '€'). You can also enter en-dashes ( -- produces '-'), em-dashes ( --- produces '--'), and proper double quotes (ex: "hello"). The input charset is UTF-8, so you can — for example — enter Japanese characters directly: おなかがいっぱいですからにもたべたくないです。

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. It is possible to embed images like  into an inline context. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa.

Here is a link to part number I, or smartly, [Part I](#). Similarly, here is a link to section number 1, or smartly, [Section 1](#). And here is a link to [that same section](#). You can also link to Tip #1, or smartly [Tip #1](#), or a link to [that same tip](#). The bibliography is at the end [\[1\]\[2\]\[3\]](#). You can also place multiple citations in the same square brackets [\[1,2,3\]](#). There is also [\(1\)](#) a list [\(2\)](#) of notes [\(3\)](#), which if you prefer you can reference them with a single command [\(1,2,3\)](#).

# Lambdoc elements (III)

## 2 Unordered lists

  Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

## Lambdoc elements (IV)

### 3 Ordered lists

Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdierit. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.
  2. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdierit. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.
  3. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor.

## Lambdoc elements (V)

## 4 Description lists

alpha

beta

# Lambdoc elements (VI)

## 5 Q&A environments

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

**QUESTION:** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa.

**“ ANSWER:** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

**Q:** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa.

**“ A:** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus

# Lambdoc elements (VII)

## 7 Quote environments

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

## Lambdoc elements (VIII)

## 8 MathTeX blocks

$$x = \frac{a^2 + \sqrt{a^2 + b^2}}{1 + \sqrt{b^2}}$$

## 9 Presentation MathML blocks

## Lambdoc elements (IX)

## 11 Source environment: default with syntax highlighting

```
type 'a tree =
| Leaf
| Node of 'a * 'a tree * 'a tree

let rec count = function
| Leaf           -> 0
| Node (node, left, right) -> 1 + count left + count right
```

# Lambdoc elements (X)

## 18 Source environment: zebra with syntax highlighting and line numbers

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

```

1  type 'a tree =
2    | Leaf
3    | Node of 'a * 'a tree * 'a tree
4
5  let rec count = function
6    | Leaf                  -> 0
7    | Node {node, left, right} -> 1 + count left + count right

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

# Lambdoc elements (XI)

## 21 Tabular environments

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

Scientific name		
Common name	Genus	Species
Wolf	<i>Canis</i>	<i>lupus</i>
Cat	<i>Felis</i>	<i>catus</i>
Chicken	<i>Gallus</i>	<i>gallus</i>
Lion	<i>Panthera</i>	<i>leo</i>
Bonobo	<i>Pan</i>	<i>paniscus</i>

Common name	Genus	Species
Scientific name		

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec

# Lambdoc elements (XII)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

“ The quick brown fox jumps over the lazy dog.

— John McAuthor

# Lambdoc elements (XIII)

## 27.11 Floating right

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

Lorem ipsum dolor sit amet, consectetur

 <b>WARNING</b>
<p>         Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.       </p>

# Lambdoc elements (XIV)

## 28 Custom theorem environments

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

**Theorem 1.** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

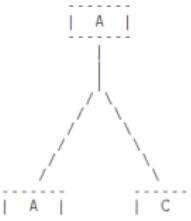
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

## Lambdoc elements (XV)

### 34 Figure environments around verbatim blocks

### 34.1 Centered

Lore ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.



**Fig. 1:** The quick brown fox jumps over the lazy dog.

## Lambdoc elements (XVI)

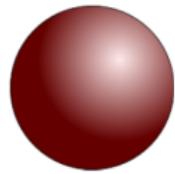
### 35.3 Floating right

*Lore ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.*



**Fig. 6:** The quick brown fox jumps over the lazy dog.

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec iudeo iusto fringilla vel aliquet nec vulnoulate eget arcu. In enim justo, rhoncus ut imperdiet, a*



**Fig. 6:** The quick brown fox jumps over the lazy dog.

# Lambdoc elements (XVII)

## III Sectioning

### 38 Introduction

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

#### 38.1 Sub-introduction

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

##### 38.1.1 Sub-sub-introduction

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec

# Lambdoc elements (XVIII)

---

## IV Backmatter

---

### Bibliography

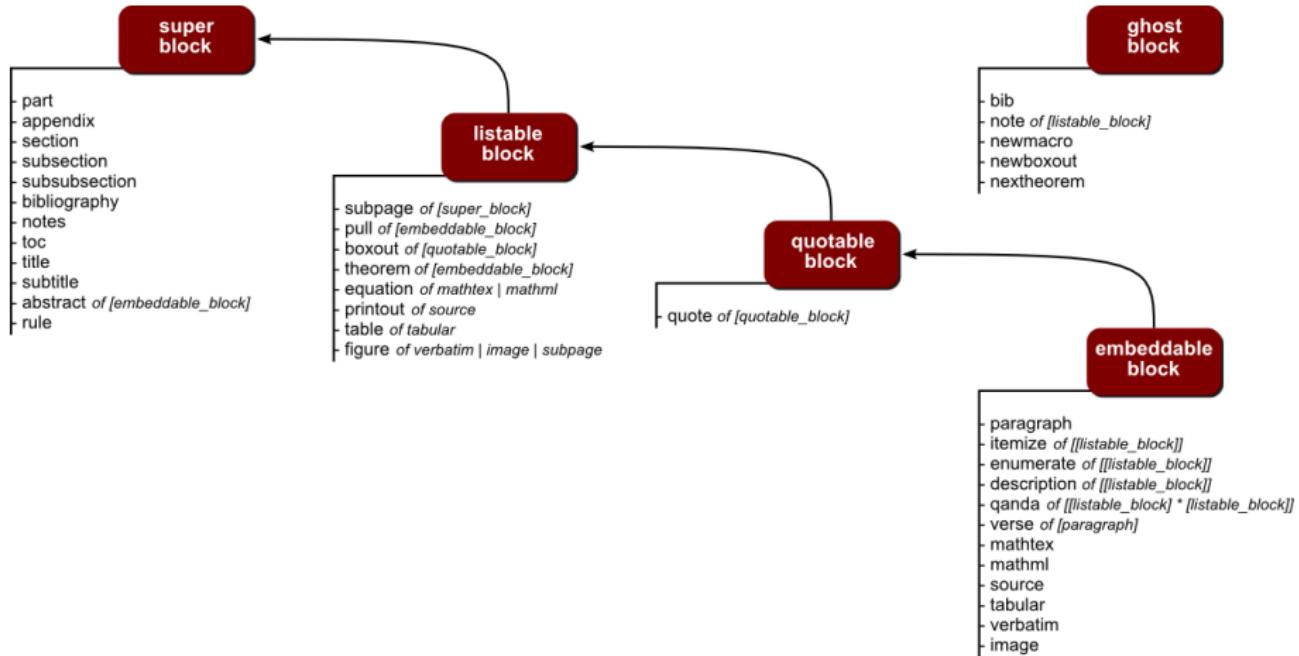
- [1] John McAuthor: *Title of the article*  
<http://www.wikipedia.org/>
- [2] John McAuthor: *Title of the article*  
<http://www.wikipedia.org/>
- [3] John McAuthor: *Title of the article*  
<http://www.wikipedia.org/>
- [4] John McAuthor: *Title of the article*  
<http://www.wikipedia.org/>
- [5] John McAuthor: *Title of the article*  
<http://www.wikipedia.org/>
- [6] John McAuthor: *Title of the article*  
<http://www.wikipedia.org/>
- [7] John McAuthor: *Title of the article*  
<http://www.wikipedia.org/>
- [8] John McAuthor: *Title of the article*  
<http://www.wikipedia.org/>
- [9] John McAuthor: *Title of the article*  
<http://www.wikipedia.org/>
- [10] John McAuthor: *Title of the article*

# Lambdoc elements (XIX)

## Notes

- (1) Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.
- (2) Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.
- (3) Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet.

# Block hierarchy



# OCaml for Haskellers: overview

- OCaml is a member of ML family, developed at INRIA.
- Multi-paradigm: functional, imperative, object-oriented (functional is the idiomatic approach; language is impure).
- Statically-typed (with type inference) and featuring a rich type system: (generalised) algebraic data types, parametric polymorphism, structural subtyping, rich module system (with functors and first-class modules).
- Eager by default, lazy on demand (contrast with Haskell).
- Language design and evolution consciously aimed at *the ML sweet-spot*.
- Many compilation targets: bytecode, native, Javascript, JVM.
- Native code compiler generates high (and predictable!) performance.
- Meta-programming: base syntax of the language can be extended with Camlp4.
- Fairly sizeable ecosystem: plenty of high-quality tools and libraries.

# OCaml for Haskellers: a small sample

```
type 'a tree = Leaf | Node of 'a * 'a tree * 'a tree

(* val count: 'a tree -> int *)
let rec count tree = match tree with
  | Leaf -> 0
  | Node (x, left, right) -> 1 + count left + count right

(* val incr: int tree -> int tree *)
let rec incr = function
  | Leaf -> Leaf
  | Node (x, left, right) -> Node (x+1, incr left, incr right)
```

# OCaml for Haskellers: polymorphic variants (a.k.a. open variants, a.k.a. open unions)

```
type foo = [ 'A | 'B ]           type foo = [ 'A | 'B ]
type bar = [ 'A | 'B | 'C ]       type bar = [ 'A | 'B | 'C ]
type zoo = [ foo | 'C | 'D ]      type zoo = [ 'A | 'B | 'C | 'D ]

let sprint = function             val sprint : [< 'A | 'B | 'C ] -> string
| 'A -> "alpha"                  | 'A -> "alpha"
| 'B -> "bravo"                 | 'B -> "bravo"
| 'C -> "charlie"                | 'C -> "charlie"

let foo_of_string = function       val foo_of_string : string -> [> 'A | 'B ]
| "alpha"    -> 'A
| "bravo"    -> 'B
| x          -> invalid_arg x
```

# How to prevent writes on read-only connections?

```
module type CONN =
sig
  type t

  val open_ro: string -> t
  val open_rw: string -> t
  val close: t -> unit
  val read: t -> string
  val write: t -> string -> unit
end
```

# Introduction to phantom types (I)

- Phantom-types are a well-known trick that leverages abstraction + parametric polymorphism to enforce invariants.
- The phantom type has no realisation on the implementation side: hence the *phantom* moniker.
- Discarded after type-checking: zero runtime cost!
- Two common methods of implementing phantom-types in OCaml:
  - Uninhabited types.
  - Polymorphic variants (a.k.a. *open unions*).

# Introduction to phantom types (II)

## (Uninhabited type version)

```
module type CONN2 =
sig
  type ro
  type rw
  type 'a t

  val open_ro: string -> ro t
  val open_rw: string -> rw t
  val close: 'a t -> unit
  val read: 'a t -> string
  val write: rw t -> string -> unit
end
```

# Introduction to phantom types (II)

(Uninhabited type version)

```
module type CONN2 =
sig
  type ro
  type rw
  type 'a t

  val open_ro: string -> ro t
  val open_rw: string -> rw t
  val close: 'a t -> unit
  val read: 'a t -> string
  val write: rw t -> string -> unit
end
```

```
module Conn2: CONN2 =
struct
  type ro
  type rw
  type 'a t = int

  let open_ro loc = ...
  let open_rw = open_ro
  let close hnd = ...
  let read hnd = ...
  let write hnd str = ...
end
```

# Introduction to phantom types (III)

## (Polymorphic variant version)

```
module type CONN3 =
sig
  type +'a t
  val open_ro: string -> [> 'RO ] t
  val open_rw: string -> [> 'RW ] t
  val close: 'a t -> unit
  val read: 'a t -> string
  val write: [< 'RW ] t -> string -> unit
end
```

# Introduction to phantom types (III)

(Polymorphic variant version)

```
module type CONN3 =
sig
  type +'a t

  val open_ro: string -> [> 'RO ] t
  val open_rw: string -> [> 'RW ] t
  val close: 'a t -> unit
  val read: 'a t -> string
  val write: [< 'RW ] t -> string -> unit
end
```

```
module Conn3: CONN3 =
struct
  type 'a t = int

  let open_ro loc = ...
  let open_rw = open_ro
  let close hnd = ...
  let read hnd = ...
  let write hnd str = ...
end
```

# The nested link problem

```
type elem =
  | Text of string
  | Bold of elem list
  | Link of string
  | Mref of string * elem list
```

# The nested link problem

```
type elem =
  | Text of string
  | Bold of elem list
  | Link of string
  | Mref of string * elem list

let seq1 = [Text "Welcome"; Bold [Text "user"]]

let seq2 = [Text "Here's an external link:"; Link "http://example.com/"]

let seq3 = [Mref ("label1", [Text "See this section"])]]

let seq4 = [Mref ("label2", [Link "http://example.com/"])]

let seq5 = [Mref ("label2", [Bold [Link "http://example.com/"]])]
```

# Solving the nested link problem with phantom-types

(Polymorphic variant version)

```
module type ELEM =
sig
  type +'a t

  val text: string -> [> 'Nonlink] t
  val bold: 'a t list -> 'a t
  val link: string -> [> 'Link] t
  val mref:
    string ->
    [< 'Nonlink] t list ->
    [> 'Link] t
end
```

# Solving the nested link problem with phantom-types

(Polymorphic variant version)

```
module type ELEM =
sig
  type +'a t
  val text: string -> [> 'Nonlink] t
  val bold: 'a t list -> 'a t
  val link: string -> [> 'Link] t
  val mref:
    string ->
    [< 'Nonlink] t list ->
    [> 'Link] t
end
```

```
module Elem: ELEM =
struct
  type elem =
    | Text of string
    | Bold of elem list
    | Link of string
    | Mref of string * elem list
  type 'a t = elem
  let text txt = Text txt
  let bold xs = Bold xs
  let link url = Link url
  let mref url xs = Mref (url, xs)
end
```

# The problem with phantom-types

- Loss of pattern-matching (though possible to recover with some hairy trickery with polymorphic variants).
- Hard to manipulate outside defining module.
- All problems boil down to phantom-types requiring constructors outside the core data structure.
- We need to make phantom-types first-class citizens.

# Generalized Algebraic Data Types (GADTs)

- First-class phantom types.
- Generalization of parameterised algebraic data types where the parameter of return type may differ for each constructor.
- Significantly extend the range of “correct” programmes we were previously unable to type-check (in OCaml, a fair amount of `Obj.magic` usage was due to lack of GADTs).
- Allow for limited forms of dependent typing without having to rely on clever encodings or switching to a full-blown dependently typed language.
- Available as a language extension in GHC since version 6.4 (March 2005).
- Available in OCaml since version 4.00 (July 2012).

# Solving the nested link problem with GADTs

```
module Elem:  
sig  
  type 'a t =  
    | Text: string -> [> 'Nonlink ] t  
    | Bold: 'a t list -> 'a t  
    | Link: string -> [> 'Link ] t  
    | Mref: string * [< 'Nonlink ] t list -> [> 'Link ] t  
end
```

# Solving the nested link problem with GADTs

```
module Elem:  
sig  
  type 'a t =  
    | Text: string -> [> 'Nonlink ] t  
    | Bold: 'a t list -> 'a t  
    | Link: string -> [> 'Link ] t  
    | Mref: string * [< 'Nonlink ] t list -> [> 'Link ] t  
end  
  
=  
  
struct  
  type 'a t =  
    | Text: string -> [> 'Nonlink ] t  
    | Bold: 'a t list -> 'a t  
    | Link: string -> [> 'Link ] t  
    | Mref: string * [< 'Nonlink ] t list -> [> 'Link ] t  
end
```

# Does this really work?

```
type 'a t =
| Text: string -> [> 'Nonlink ] t
| Bold: 'a t list -> 'a t
| Link: string -> [> 'Link ] t
| Mref: string * [< 'Nonlink ] t list -> [> 'Link ] t

let seq1 = [Text "welcome"; Bold [Text "user"]]
```

# Does this really work?

```
type 'a t =
| Text: string -> [> 'Nonlink ] t
| Bold: 'a t list -> 'a t
| Link: string -> [> 'Link ] t
| Mref: string * [< 'Nonlink ] t list -> [> 'Link ] t

let seq1 = [Text "welcome"; Bold [Text "user"]]

val seq1 : [> 'Nonlink ] Elem.t list
```

# Does this really work?

```
type 'a t =
| Text: string -> [> 'Nonlink ] t
| Bold: 'a t list -> 'a t
| Link: string -> [> 'Link ] t
| Mref: string * [< 'Nonlink ] t list -> [> 'Link ] t

let seq2 = [Link "http://example.com/"]
```

# Does this really work?

```
type 'a t =
| Text: string -> [> 'Nonlink ] t
| Bold: 'a t list -> 'a t
| Link: string -> [> 'Link ] t
| Mref: string * [< 'Nonlink ] t list -> [> 'Link ] t

let seq2 = [Link "http://example.com/"]

val seq2 : [> 'Link ] Elemt.list
```

# Does this really work?

```
type 'a t =
| Text: string -> [> 'Nonlink ] t
| Bold: 'a t list -> 'a t
| Link: string -> [> 'Link ] t
| Mref: string * [< 'Nonlink ] t list -> [> 'Link ] t

let seq3 = [Text "Here's an external link:"; Link "http://example.com/"]
```

# Does this really work?

```
type 'a t =
| Text: string -> [> 'Nonlink ] t
| Bold: 'a t list -> 'a t
| Link: string -> [> 'Link ] t
| Mref: string * [< 'Nonlink ] t list -> [> 'Link ] t

let seq3 = [Text "Here's an external link:"; Link "http://example.com/"]

val seq3 : [> 'Link | 'Nonlink ] Elemt.list
```

# Does this really work?

```
type 'a t =
| Text: string -> [> 'Nonlink ] t
| Bold: 'a t list -> 'a t
| Link: string -> [> 'Link ] t
| Mref: string * [< 'Nonlink ] t list -> [> 'Link ] t

let seq4 = [Mref ("label", [Text "see this section"])]
```

# Does this really work?

```
type 'a t =
| Text: string -> [> 'Nonlink ] t
| Bold: 'a t list -> 'a t
| Link: string -> [> 'Link ] t
| Mref: string * [< 'Nonlink ] t list -> [> 'Link ] t

let seq4 = [Mref ("label", [Text "see this section"])]
val seq4 : [> 'Link ] Elemt.list =
```

# Does this really work?

```
type 'a t =
| Text: string -> [> 'Nonlink ] t
| Bold: 'a t list -> 'a t
| Link: string -> [> 'Link ] t
| Mref: string * [< 'Nonlink ] t list -> [> 'Link ] t

let seq5 = [Mref ("label", [Link "http://example.com/"])]
```

# Does this really work?

```
type 'a t =
| Text: string -> [> 'Nonlink] t
| Bold: 'a t list -> 'a t
| Link: string -> [> 'Link] t
| Mref: string * [< 'Nonlink] t list -> [> 'Link] t
```

```
let seq5 = [Mref ("label", [Link "http://example.com/"])]
```

```
Error: This expression has type [> 'Link] Elemt
      but an expression was expected of type [< 'Nonlink] Elemt
      The second variant type does not allow tag(s) 'Link
```

# Does this really work?

```
type 'a t =
| Text: string -> [> 'Nonlink] t
| Bold: 'a t list -> 'a t
| Link: string -> [> 'Link] t
| Mref: string * [< 'Nonlink] t list -> [> 'Link] t
```

```
let seq6 = [Mref ("label", [Bold [Link "http://example.com/"]])]
```

# Does this really work?

```
type 'a t =
| Text: string -> [> 'Nonlink] t
| Bold: 'a t list -> 'a t
| Link: string -> [> 'Link] t
| Mref: string * [< 'Nonlink] t list -> [> 'Link] t
```

```
let seq6 = [Mref ("label", [Bold [Link "http://example.com/"]])]
```

```
Error: This expression has type [> 'Link] Elemt
      but an expression was expected of type [< 'Nonlink] Elemt
      The second variant type does not allow tag(s) 'Link
```

# GADTs pose no problem for pattern-matching

```
module Ele =  
  struct  
    type 'a t =  
      | Text: string -> [> 'Nonlink] t  
      | Bold: 'a t list -> 'a t  
      | Link: string -> [> 'Link] t  
      | Mref: string * [< 'Nonlink] t list -> [> 'Link] t  
  end  
  
module M =  
  struct  
    open Ele  
  
    let rec to_upper: type a. a Ele.t -> a Ele.t = function  
      | Text txt          -> Text (String.uppercase txt)  
      | Bold seq          -> Bold (List.map to_upper seq)  
      | Link url          -> Link url  
      | Mref (url, seq)   -> Mref (url, List.map to_upper seq)  
  end
```

# GADTs: a poor man's substitute for dependent types (I)

```
type literal =
  | ILit of int
  | FLit of float
  | SLit of string

let get_int = function
  | ILit i -> i
  | _ -> assert false

let get_float = function
  | FLit x -> x
  | _ -> assert false

let get_string = function
  | SLit s -> s
  | _ -> assert false

let () =
  let v1 = ILit 1 in
  let v2 = FLit 2.0 in
  Printf.printf "Int=%d, Float=%f\n" (get_int v1) (get_float v2)
```

# GADTs: a poor man's substitute for dependent types (II)

```

type literal =
  | ILit of int
  | FLit of float
  | SLit of string

type 'a getter =
  | IGet: int getter
  | FGet: float getter
  | SGet: string getter

let get: type a. literal -> a getter -> a = fun lit sing ->
  match (lit, sing) with
  | (ILit i, IGet) -> i
  | (FLit x, FGet) -> x
  | (SLit s, SGet) -> s
  | _                      -> assert false

let () =
  let v1 = ILit 1 in
  let v2 = FLit 2.0 in
  Printf.printf "Int=%d, Float=%f\n" (get v1 IGet) (get v2 FGet)

```

# GADTs: a poor man's substitute for dependent types (III)

```
type 'a expr =
| ILit : int -> int expr
| BLit : bool -> bool expr
| Add : int expr * int expr -> int expr
| Eq : int expr * int expr -> bool expr
| Not : bool expr -> bool expr

let rec eval: type a. a expr -> a = function
| ILit n          -> n
| BLit b          -> b
| Add (e1, e2)    -> eval e1 + eval e2
| Eq (e1, e2)     -> eval e1 = eval e2
| Not e           -> not (eval e)
```

# Type system trickery in practice

- Leveraging type system for enforcing complex invariants can be useful in many domains.
- Beware that extra complexity implies code that is harder to understand and harder to refactor.
- Beware of diminishing returns: there's a reason why the ML type system is referred to as a *sweet spot*!
- Always keep an eye on the ultimate goals: reliability and expressiveness.
- Finding proper trade-off is a craft; like all crafts, it is informed by experience.

## For more information

- Nleyten blog:  
<http://nleyten.com/>
- Home of the various open-source components:  
<http://forge.ocamlcore.org/users/dario/>
- OCaml home page:  
<http://ocaml.org/>
- Ocsigen project home page:  
<http://ocsigen.org/>

# Questions?