Compiler Construction

Mini Project

A BibTeX2HTML

The aim of this mini project is to implement a suite of small command-line utilities for processing bibliographic databases in BibTeX-format and producing logical documents in HTML-format.

BibTeX is a tool for generating bibliographies and including them in LaTeX-documents. A general description is given at `http://en.wikipedia.org/wiki/BibTeX`. When producing a bibliography, BibTeX reads bibliographic data from a database written in a small domain-specific language. Here is an example of such a database:

```
@book{pierce02types,
  author    = "Pierce, Benjamin C.",
  title     = "Types and Programming Languages",
  publisher = "The MIT Press",
  address   = "Cambridge, Massachusetts",
  year      = 2002}

@inproceedings{loeh03dependency,
  author    = "L{\"o}h, Andres and Clarke, Dave and Jeuring,
               Johan",
  title     = "Dependency-style {G}eneric {H}askell",
  editor    = "Runciman, Colin and Shivers, Olin",
  booktitle = "Proceedings of the Eighth ACM SIGPLAN
               International Conference on Functional
               Programming, ICFP 2003, Uppsala, Sweden,
               August 25--29, 2003",
  pages     = "141--152",
  publisher = "ACM Press",
  year      = 2003}
```

In general, a BibTeX-database contains of zero or more *entries*. Each entry consists of three parts: a *type specifier* (marked by an `@`-sign), a *key*, and the actual *data* for the entry. The example database above holds two such entries: the first has `book` as its type specifier, the second `inproceedings`, while the keys read `pierce02types` and `loeh03dependency`. The data part of an entry amounts to a comma-separated list of fields, each consisting of a field name and a value. Every entry type comes with a set of required and optional fields. More detailed descriptions of the BibTeX-format and which fields are required and optional to which entry type can be found on the web.

The overall objective of the set of tools to be implemented is to enable a user to produce an HTML-rendering of a BibTeX-database. For instance,

for the database above, we want to obtain an HTML-document similar to the following:

```
<html>
  <head><title>Bibliography</title></head>
  <body>
    <a href="loeh03dependency">[LCJ03]</a> |
    <a href="pierce02types">[P02]</a>
    <hr>
    <table border="0">
      <tr valign="top">
        <td><a name="loeh03dependency">[LCJ03]</a></td>
        <td>
          Andres L&ouml;h, Dave Clarke, and Johan
          Jeuring. Dependency-style Generic Haskell. In:
          Colin Runciman and Olin Shivers, editors,
          <em>Proceedings of the Eighth ACM SIGPLAN
          International Conference on Functional
          Programming, ICFP 2003, Uppsala, Sweden,
          August 25&ndash;29, 2003</em>, pages
          141&ndash;152. ACM Press, 2003.
        </td>
      </tr>
      <tr valign="top">
        <td><a name="pierce02types">[P02]</a></td>
        <td>
          Benjamin C. Pierce. <em>Types and Programming
          Languages</em>. The MIT Press, Cambridge,
          Massachusetts, 2002.
        </td>
      </tr>
    </table>
  </body>
</html>
```

which, when rendered in a browser will look like

[LCJ03] | [P02]

[LCJ03] Andres Löh, Dave Clarke, and Johan Jeuring. Dependency-style Generic Haskell. In: Colin Runciman and Olin Shivers, editors, *Proceedings of the Eighth ACM SIGPLAN International Conference on Functional Programming, ICFP 2003, Uppsala, Sweden, August 25–29, 2003*, pages 141–152. ACM Press, 2003.

[P02] Benjamin C. Pierce. *Types and Programming Languages*. The MIT Press, Cambridge, Massachusetts, 2002.

Architecture

The implementation should comprise (at least) three main components:

1. A program `parse-bib` that consumes and parses a bibliographic database in BibTeX-format and produces an ATerm that describes the structure of the database.

2. A program `bib2html` that

   (a) consumes an ATerm as produced by `parse-bib`;

   (b) checks, for each entry in the bibliographic database, whether all required fields associated with the entry's type are present, issueing helpful error messages in case this check fails; and

   (c) produces an ATerm describing the HTML-rendering of the database in which the entries of the database are sorted first by author and then by year and title.

3. A program `pp-html` that consumes an ATerm for an HTML-document as produced by `bib2html` and that produces a pretty printing of the document.

A typical use of the resulting pipeline from the command line is

```
cat biblio.bib | parse-bib | bib2html | pp-html > biblio.html
```

Details

Note that while the bare essentials of the BibTeX-format are straightforward, a full and faithful implementation of the format will need to deal with many subtleties, such as possible variations in bracketing (parentheses instead of curly braces, nested curly braces instead of quotation marks); optionally leaving out quotation marks in numeric fields, formatting rules for last names, given names, etc. in the `author` and `editor` fields; accents in fields (`\"o`, `\'e`, etc); cross

references; comments; `@string`-declarations; `@preamble`-declarations. You will be probably not be able to implement all these features, but you are expected to support at least some of them, so that a reasonable subset of the BibTeX-format can be handled.

Your implementation should issue warning messages if, in the data part of an entry, fields are used that are neither required or optional for the type of the entry. When generating HTML, these fields are to be ignored.

For guidance on how to format your HTML-renderings (which fields to put in `<em>`-tags, which words to convert to lower case, etc.) you could inspect the LaTeX-output generated by the actual BibTeX program.

For the More Ambitious

You may extend the toolset with additional programs that support rendering bibliographic databases in other formats, such as plain text and TWiki formatting commands.

Submitting

The source code of your implementation should be handed in according to the submission instructions on the website of this course.

Include in your submission a number of example databases.