Linear Time LCP Computation Algorithm (Kasai et al.)

Robert Bakarić
rbakaric@irb.hr
bakaric@evolbio.mpg.de
02.09.2015
LCPKasai-1.0

Abstract

The longest common prefix (LCP) array is an auxiliary data structure to the suffix array. The array containes lengths of the longest common prefixes (LCPs) between all pairs of consecutive suffixes in a sorted suffix array. The package contains the encapsilated sais-light-2.4.1 library written by Yuta Mori [1] implementing the induced sorting [2] based suffix array construction algorithm and an implementation of Kasai's linear time LCP construction algorithm [3].

Contents

1 Installation

The simplest way to compile this program is to:

1. Unpack the LCPKasai package (`lcpkasai-XXX.tar.gz`):

   ```
   tar -xvzf lcpkasai-XXX.tar.gz
   ```

2. Change the current directory to `lcpkasai-XXX`:

   ```
   cd lcpkasai-XXX/
   ```

3. Configure the program for your system (–bindir is optional):

   ```
   ./configure --bindir=/absolute/directory/path/lcpkasai-xxx/bin
   ```

4. Compile the program:

   ```
   make
   ```

5. Install the program:

   ```
   make install
   ```

Your binaries should be located in your local bin directory if `--bindir` option has been set. Otherwise installation needs to be carried out with root privileges in order to be installed into `/usr/local/bin` directory.

2 Input files

The LCPKasai program takes a simple ASCII txt file and computes an array of indexes corresponding to lexicographically ordered text suffixes from which then longest common prefix array is computed according to the algorithm published by Kasai et al. [3]. An example of the input file can be found in `./lcpkasai-xxx/demo` and it should look like this:

`TestLCP`:

`thisisatesttotestthecorrectnessofLCPKasaialgorithm`

3 Program options

I order to see program options type:

```
./bin/LCPKasai -h
```

Expected output:

```
Usage: ./program [options]

*******************************************************************************
                 LCPKasai - logest common prefix array computation tool
                                        by
                                  Robert Bakaric

CONTACT:
 Code written and maintained by Robert Bakaric,
 email: rbakaric@irb.hr , bakaric@evolbio.mpg.de

ACKNOWLEDGEMENT:
         1. Yuta Mori, 2010. https://sites.google.com/site/yuta256/sais
```

```
        2. Ge Nong, Sen Zhang and Wai Hong Chan, Two Efficient Algorithms for
           Linear Suffix Array Construction, 2008.

        3. Kasai et al. Linear-Time Longest-Common-Prefix Computation in
           Suffix Arrays and Its Applications. 2001.

LICENSE:
 The program is distributed under the GNU General Public License. You should have
 received a copy of the licence together  with this software. If not, see
 http://www.gnu.org/licenses/
*******************************************************************************


Options:
  -h [ --help ]           produce help message
  -v [ --version ]        print version information
  -i [ --input-file ] arg input file
```

4 Functions and classes

LCPArrayKasai class:

`LCPArrayKasai` :	LCPArrayKasai class.
`make` :	Explicit constructor.
`destroy` :	Explicit destructor.
`ComputeLCPArray` :	Function computes the longest common prefix array.
`GetLCPxArray` :	Function returns the longest common prefix array.

5 Example

A minimal example demonstrating the usage of LCPKasai demo program:

```
./bin/LCPKasai -i demo/TestLCP

Text size:51
Suffix Array:
50 34 36 33 35 39 41 37 6 20 25 19 24 28 14 8 32 43 18 1 48 40...
...4 2 46 42 49 27 31 44 21 12 23 45 22 38 5 3 30 29 15 9 13 7...
...17 0 47 26 11 16 10

LCP Array:
0 0 0 0 0 0 1 1 1 0 1 0 2 1 2 4 0 0 0 1 1 0 1 2 1 0 0 0 0 1 2 ...
... 1 0 1 1 0 2 1 1 1 1 3 0 5 1 2 2 1 1 1 2
```

5.1 LCPArrayKasai.hpp

Adding the `LCPArrayKasai.hpp` header file to your program will allow you to include all the functions described in section 4. A minimal example:

```
#include<vector>
#include<string>
#include<SuffixArray.hpp>
#include<LCPArrayKasai.hpp>

string text("This is my text");
```

```
/* Make SuffixArray */

    /* Construction */
    SuffixArray<int|long|unsigned|double> SA(text);
    /*  OR  */
    SuffixArray<int|long|unsigned|double> SA;
    SA.make(text);

/* Get Suffix Array */

   vector<int|long|unsigned|double>SArray = SA.GetSuffixArray();
                                // [15 4 7 10 0 12 1 2 5 8 3 6 14 11 13 9]

/* Compute LCP array */

   LCPArrayKasai<int|long|unsigned|double> LCP(SArray);

/* Get LCP Array */

   vector<int|long|unsigned|double>LCPArray = LCP.GetLCPArray();
                                // [0 0 1 1 0 0 0 0 3 0 0 2 0 1 0 0]
```

6 Acknowledgement

1. Yuta Mori, 2010. https://sites.google.com/site/yuta256/sais
2. Ge Nong, Sen Zhang and Wai Hong Chan, Two Efficient Algorithms for Linear Suffix Array Construction, 2008.
3. Kasai, Toru and Lee, Gunho and Arimura, Hiroki and Arikawa, Setsuo and Park, Kunsoo, Linear-Time Longest-Common-Prefix Computation in Suffix Arrays and Its Applications, Lecture Notes in Computer Science, Combinatorial Pattern Matching, 2001.

7 Future work