RNA Stem Loop Visualizer

Pieter Lauwers – Jakob Struye – Ruben Van Assche – Stijn Wouters

University of Antwerp

Belgium

pieter.lauwers@student.uantwerpen.be – jakob.struye@student.uantwerpen.be
ruben.vanassche@student.uantwerpen.be – stijn.wouters2@student.uantwerpen.be

For phase I, we've decided to implement the following algorithms:

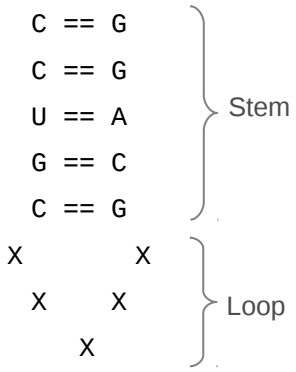
- Converting PDA to CFG and vice versa (Ruben)
- Converting CFG to CNF and the CYK algorithm (Stijn)
- Simulating Turing Machines (Jakob)
- Either an LL or an LR parser (Pieter)

For phase II, we will design a “RNA Stem Loop Visualizer”. This is a software that recognizes RNA Stem Loops and visualizes them. This is useful in fields as cellular biology, where it can be used as an aid to analyze and understand the RNA structure of genes and viruses.

For now, it's sufficient to know that RNA is actually a chain of nucleotides, abbreviated as G, A, U and C. Also keep in mind that G is a complement of C and U of A. RNA will form a stem loop if there are two parts in a RNA sequence consisting of complements of each other (the stem) while there are other non-complementing nucleotides in between (the loop).

The software in the basic version works roughly as follows:

1) The input is a RNA sequence, something like: `CCUGCXXXXXXXGCAGG` (X is any of C, U, A, G).
2) The software will test whether this is a stem loop (cfr. palindromes, but with a matching complement).
    1. If so, visualize the stem loop:



    2. If not, reject.

Note that the loop starts where X does **not** match with a complementary X anymore (e.g. G can not match with an A, so then the loop starts from here).

We'll use a PDA for parsing the sequence and maybe we'll extended it to using a LL or LR parser or TM instead.

In case of sufficient time, we'll allow the following extensions:

- Input RNA sequence where the initial nucleotide is not necessarily the beginning of the stem loop.
- Support sequences with multiple stem loops.
- Input which elements should be in the loop, so you can track down specific stem loops.

- Use something fancier than ASCII to visualize the stem loop.
- …

The software will be implemented using C++ programming language and will be Open Sourced soon after the release on the final presentation at the Conference Call for Projects under GPL v3 License.

We've selected this project because **it is actually something useful** for (bio)scientists.

There is no consensus yet how we will divide the work for phase II because it depends on how much time phase I will take for each of us, but below we've made some estimate:

- 20h phase I
- 20h phase II
- 4h workshop
- 6h conference
- 8h testing
- 12h bugfixing

We've also in-calculated a buffer of 6 hours, just in case when something took us longer than expected or for extending the functionality of phase II further.

[1] Durbin, R., Eddy, S., Krogh, A., Mitchison, G. *Biological Sequence Analysis. Probabilistic models of proteins and nucleic acids.* Cambridge, Cambridge University Press, 1998, 350 p.
[2] https://en.wikipedia.org/wiki/Stem-loop (2013-12-28)

Thanks to the following scientists who encouraged us and gave background information for this Project Proposal.

[3] EL AJJOURI, Y., (2 Ba Bio Ir. – University of Antwerp)
[4] DE PAUW, I., (2 Ma Molecular and Cel Biology – University of Antwerp)
[5] WOUTERS, I., (2 Ma Bio-Medical Science – University of Antwerp)
[6] LIEKENS, A. (Bio-Computer-Scientist – Educated in University of Antwerp)