

Protein Folding Mechanisms – An Introduction & Novel Suggestion based on Dijkstra’s Algorithm.

D.N.T. Kumar
Independent Consultant - Informatics/Photonics/Nanotechnology
email id : tejdnk@gmail.com

[A] Introduction & Inspiration :

https://rosettacode.org/wiki/Dijkstra%27s_algorithm

http://vixra.org/author/nirmal_tej_kumar

<http://alloy.lcs.mit.edu/alloy/> && <https://aleksandarmilicevic.github.io/hola/>

https://en.wikipedia.org/wiki/Edsger_W._Dijkstra

<https://pdfs.semanticscholar.org/5bd7/450c57dd5278cc5fa380c65890a94d8d298b.pdf>

<https://github.com/wvrossem/Prolog-Dijkstra-Algorithm>

<http://www.jiprolog.com/> && <https://github.com/jiprolog/jiprolog/wiki/Prolog-calls-Java>

<https://www.jikesrvm.org/> && <https://www.cl.cam.ac.uk/teaching/1415/CompGraph/opengl.html>

[B] Informatics Framework & Implementation :

“about alloy*

- retains the syntax of Alloy: predicate logic + relational algebra
- permits higher-order quantification: changes the semantics of Alloy only by expanding the set of specifications that can be analyzed
- implements CEGIS on top of Kodkod: all higher-order quantifiers are analyzable (regardless of their position in the formula)
- applications: synthesis, game theory, minimization/maximization, ...
- papers : [ICSE 2015](#), [FMSD 2017](#) “

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3376395/pdf/nihms359942.pdf>

We have considered the following example to start :

- /*
- * Models how mutexes are grabbed and released by processes, and
- * how Dijkstra's mutex ordering criterion can prevent deadlocks.
- * For a detailed description, see:
- * E. W. Dijkstra, Cooperating sequential processes. Technological
- * University, Eindhoven, The Netherlands, September 1965.
- * Reprinted in Programming Languages, F. Genuys, Ed., Academic
- * Press, New York, 1968, 43-112.
- */

[Source : <https://aleksandarmilicevic.github.io/hola/>]

“Do test new challenging & interesting ideas”.

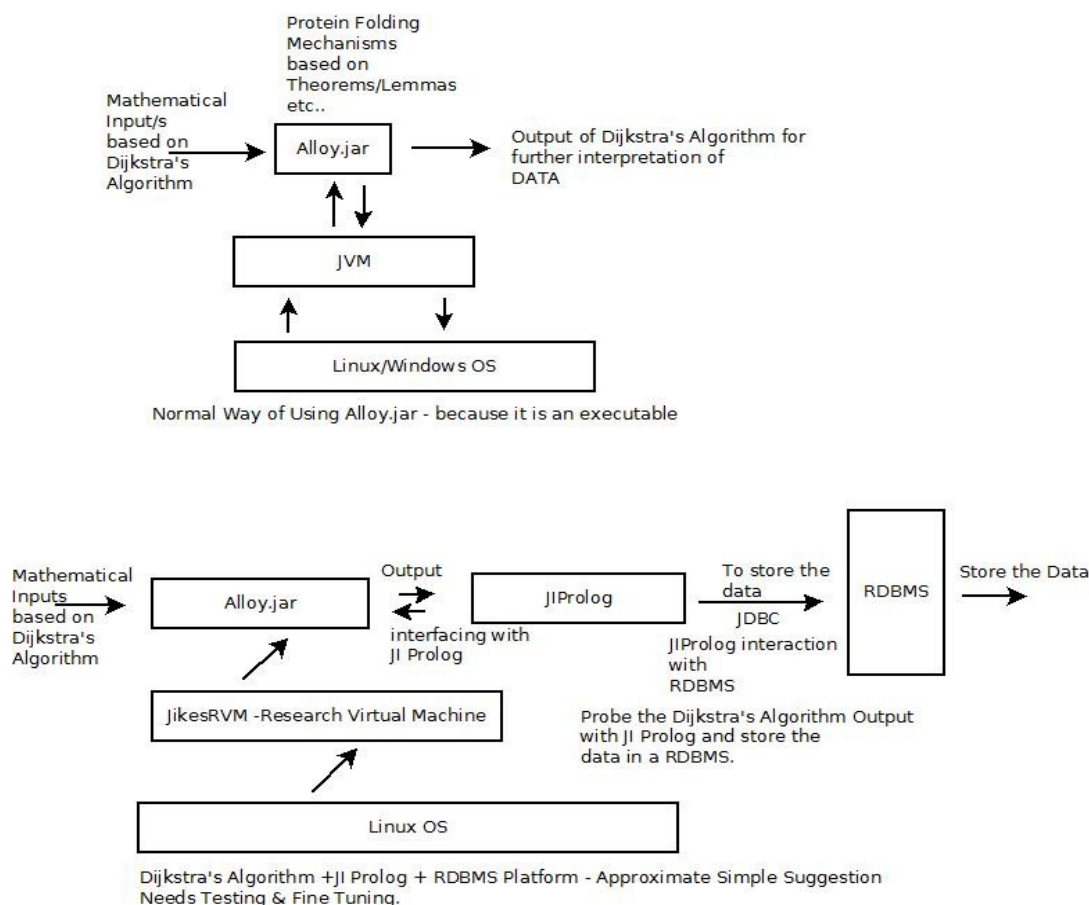


Figure I – Our Simple Suggestion and Informatics Framework to Probe Protein Folding Mechanisms.

*** Readers are requested to satisfy themselves and use the suggestion

This is Short Technical Notes so we are not going into the details of implementation.

Please check before using the proposed informatics frame work. Fine tuning is required.

*Actual implementation will vary. Testing in progress at the time of submission ****

[C] Information on Dijkstra Algorithms Mathematics Software Used :

https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

<http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/GraphAlgor/dijkstraAlgor.htm>

<https://www.cs.auckland.ac.nz/software/AlgAnim/dijkstra.html>

<http://www-math.mit.edu/~rothvoss/18.304.3PM/Presentations/1-Melissa.pdf>

<https://www.baeldung.com/java-dijkstra>

https://www-m9.ma.tum.de/graph-algorithms/spp-dijkstra/index_en.html

“Computation of channels in protein molecules” -

https://is.muni.cz/th/yzbhb/PhD_Thesis_Petr_Medek.pdf

<https://www.gwern.net/docs/biology/1993-fraenkel.pdf>

https://en.wikipedia.org/wiki/Protein_folding

<http://www-inst.eecs.berkeley.edu/~cs61bl/r//cur/graphs/dijkstra-algorithm-runtime.html?topic=lab24.topic&step=4&course=>

[https://en.wikipedia.org/wiki/Alloy_\(specification_language\)](https://en.wikipedia.org/wiki/Alloy_(specification_language))

http://www.doc.ic.ac.uk/project/examples/2007/271j/suprema_on_alloy/Web/

http://www.swi-prolog.org/pldoc/doc/_SWI_/library/jpl.pl

<https://jpl7.org/> - “an API between SWI-Prolog and the Java Virtual Machine - [hosted on GitHub](#)”.

<http://www.declarativa.com/InterProlog/download211.htm>

<https://www.cs.nmsu.edu/ALP/2011/03/concurrent-programming-constructs-and-first-class-logic-engines/>

https://www.ibm.com/support/knowledgecenter/en/SSVSD8_9.0.0/com.ibm.websphere.dtx.funcexp.doc/references/r_funcs_express_jexitexamples.htm

[D] Acknowledgment/s :

Special Thanks to all who made this happen in my LIFE.Non-Profit Academic R&D.

Written in Free Style .

Thank You for reading my communication – Dr.Nirmal.

THE END.