FuL: A Logic processor to aid design and validate virological experiments

Alejandro Kondrasky[1] - Daniel Gutson[1] - Carlos Areces[1,2]

[1]FuDePAN: Fundación para el Desarrollo de la Programación en Ácidos Nucleicos, X5002AOO, Córdoba, Argentina.
[2]FaMAF, Universidad Nacional de Córdoba, Ciudad Universitaria, X5000HUA, Córdoba, Argentina.

Introduction

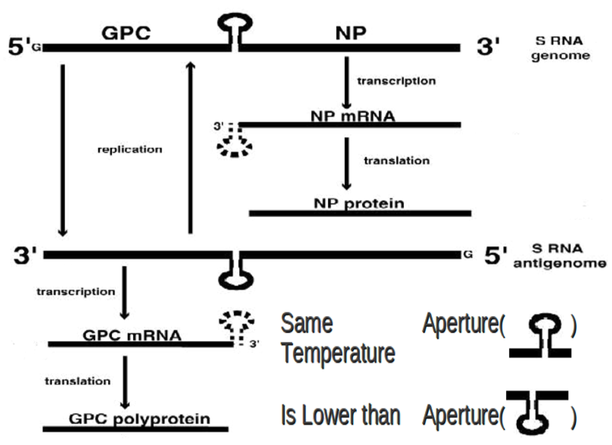
The body of knowledge in biology, particularly in virology and immunology, is increasing in volume and complexity. This is why it would be useful to have these knowledge represented in a formal language inside a knowledge base. Subsequently, different methodologies for analysis and manipulation could be developed, allowing validity checks to be performed on conclusions obtained in experiments.

Objectives

FuDePAN's Logic processor (FuL) is being developed to organize, interpret, verify and explore knowledge in molecular biology, applied to virology and immunology in particular. This will help find inconsistencies and automatically derive new information.
Its main function will be the verification of conclusions obtained by results from experiments using queries, as well as assisting in the design of new experiments.

Case Study: Junín Virus

Our initial test case will be the following conclusion obtained from experiments done by FuDePAN, which will be the validation of the conclusions obtained in the Junín experiment about the temperature-change effects over the virus secondary structure[3]:

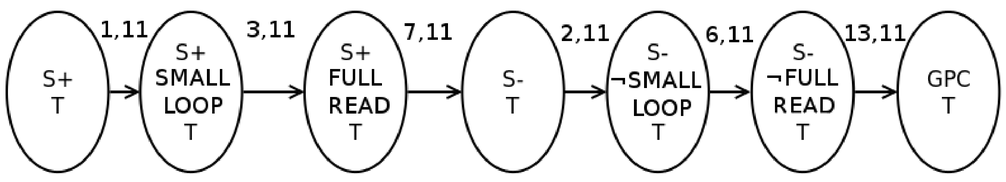


Corroborate that the line of thought that includes the predictions of the effects of febrile state over the Junín RNA secondary structure, in which it is hypothesized that the temperature increment reduces the production of nucleoproteins because the hairpin loop in the intergenic region presents dissimilar characteristics when it is compared on the two ambisense genome strings when the temperature is increased.

By modelling it a set composed with definitions and another set with rules was obtained:

Definitions	Rules
$S^+ \equiv$ Genomic RNA produced	1 $S^+ \wedge T \implies x(S^+ \wedge small\_loop)$
$S^- \equiv$ Antigenomic RNA produced	2 $S^- \wedge T \implies x(S^- \wedge \neg small\_loop)$
$NP \equiv$ Nucleoprotein produced	3 $S^+ \wedge small\_loop \implies x(S^+ \wedge fully\_readable)$
$GPC \equiv$ Glycoprotein produced	4 $S^+ \wedge \neg small\_loop \implies x(S^+ \wedge \neg fully\_readable)$
$small\_loop \equiv$ Flat intergenic zone	5 $S^- \wedge small\_loop \implies x(S^- \wedge fully\_readable)$
$\neg small\_loop \equiv$ Normal intergenic zone	6 $S^- \wedge \neg small\_loop \implies x(S^- \wedge \neg fully\_readable)$
$fully\_readable \equiv$ RNA fully readable without cuts	7 $S^+ \wedge fully\_readable \implies x(S^- \wedge \neg S^+)$
$\neg fully\_readable \equiv$ RNA not fully readable without cuts	8 $S^+ \wedge \neg fully\_readable \implies x(NP \wedge \neg S^+)$
$T \equiv$ Temperature increase	9 $S^- \wedge fully\_readable \implies x(S^+ \wedge \neg S^-)$
$\neg T \equiv$ Temperature decrease	10 $S^- \wedge \neg fully\_readable \implies x(GPC \wedge \neg S^-)$
	11 $T \implies x(T)$
	12 $S^+ \wedge \neg T \implies x(NP \wedge \neg S^+)$
	13 $S^- \wedge \neg T \implies x(GPC \wedge \neg S^-)$

Combining this knowledge with the one in a selected knowledge base (KB), the application should be able to validate of the conclusions obtained in the Junín experiment[3], using a planner and a semantic reasoner that will process each state of the plan along with the KB and return a new state that will extend it.



It will be repeated until a conclusion is reached or a set of questions is obtained for which their answers must be included in the KB in order to advance in the deduction process for answer the original query.

Materials and Methods

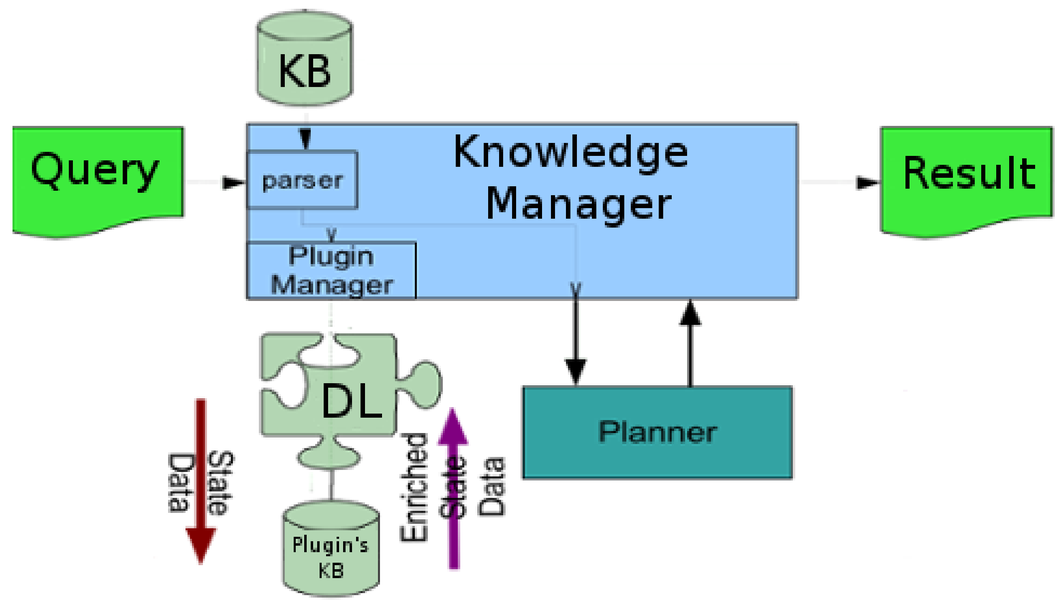
- C++: Programming language `http://www.open-std.org/jtc1/sc22/wg21/`
- FF: Fast Forward Planner `http://arxiv.org/abs/1106.0675`
- DL: Description Logics `http://dl.kr.org/`

Repository

The repository with the development documentation and soon source code is available at `code.google.com/p/ful/`

Early Design

FuL has a plug-in architecture, simplifying the inclusion of new kinds of reasoning services. An API will be provided, which defines the way in which knowledge flows between the plug-ins and FuL's core reasoning engine. An SDK composed of libraries and tools required for building plug-ins will also be made available.



The kernel of the tool will be composed of a planner that can handle PDDL (Planning Domain Definition Language) input, and a KM (knowledge manager) that will be the interface between the plug-ins registered in that session and the planner. During each planner state the deriving process of the next state is as follows :

1. The KM take the actual state from the planner and sends it to each registered plug-in.
2. Each plug-in try to enrich that state by derivation of new knowledge by consulting the shared KB and their own, the plug-in can return a enriched state.
3. The knowledge manager generate the derived state by combining the enriched states received from the plug-ins. If there incongruities between the states received from the plug-ins the KM will report this to the user, otherwise the KM will return the derived state.

FuL will include a semantic reasoner for Description Logics (DL) as one of the plug-ins and we will also provide a knowledge representation language for the virology domain based on DL. This language will allow the development of an ontology of virology knowledge that will be available for querying during a FuL session.

Features

Via a XML file provided by the user, FuL will register the plug-ins that will be utilized during that session and configure different session parameters. Also a initial KB file will be used during the session. FuL will process the KB file with the plug-ins register in the XML file. Afterwards a prompt-like CLI will be gived to the user that allow to access the following functionalities:

- **Extend the actual KB by join it with a Δ KB :** It will check for incongruities and saturate new KB with atomic concepts derived from actual KB $\cup \Delta$KB.
- **Queries :** The answer of this queries will be consistent with the actual KB and register plug-ins. If FuL can't arrive to a conclusion it will return a set of questions that are unanswerable by FuL, to which answers can be added by extending the actual KB in order to return a define answer.

References

[1] Franz Baader, Deborah L. McGuinness, Daniele Nardi, Peter F. Patel-Schneider. The Description Logic Handbook: Theory, implementation, and applications.
[2] The Seventh International Planning Competition Description of Participant Planners of the Deterministic Track. In *2011*
[3] Daniel Gutson, Agustín March, Maximiliano Combina, Daniel Rabinovich. Prediction of consequences of the febrile status on the RNA secondary structure of the Junín Virus In *2006* `www.fudepan.org.ar/node/71`