PROLOG

16 mars 2022

SOMMAIRE

PROLOG n'est pas qu'un langage universitaire, mais a servi et sert encore dans des applications réelles, où il joue un rôle clef.

WORK IN PROGRESS

Personnes ayant participé à cette rédaction

- Laurent Gouzènes
- Laurent Cervoni
- Via Eric de la Clergerie
- Jean Rohmer

1. Quelques éléments préliminaires sur Prolog aujourdhui

Le classement Tiobe des langages cite Prolog en 21 place.
et avec quelques questions complémentaires

- Is Prolog professionally useful?
- prolog nécessite un état d'esprit particulier et de l'expérience (beaucoup ?)
- logic programming career ???
- Prolog is a crazy language : Unless you use it constantly, you end up forgetting it
- limitations of prolog (on wikipedia)
- comment documenter des programmes prolog ?

Notes

Comparison of Prolog implementations

1.1. Some history

The history of prolog and derivates can be understood at different scales, and according to different views (theoretic, implementation, system features, syntex, etc, ...)
More details can be read

- here.
- Colmerauer's blog

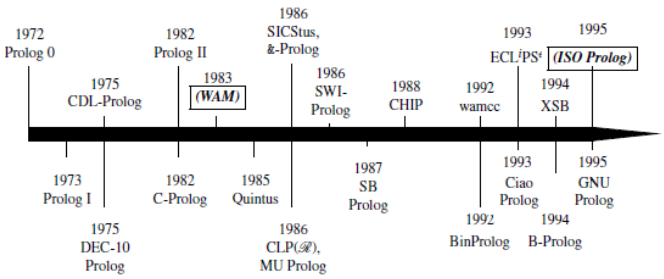
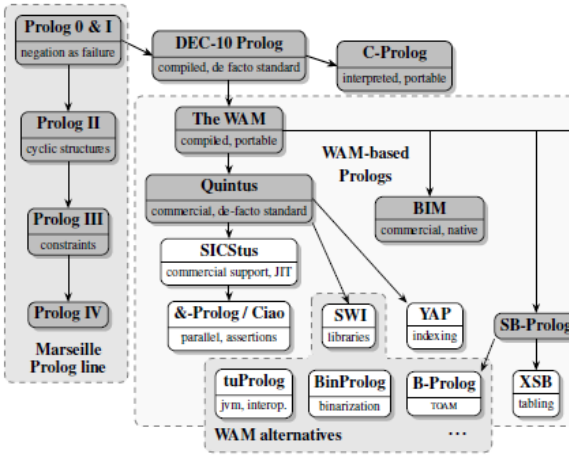
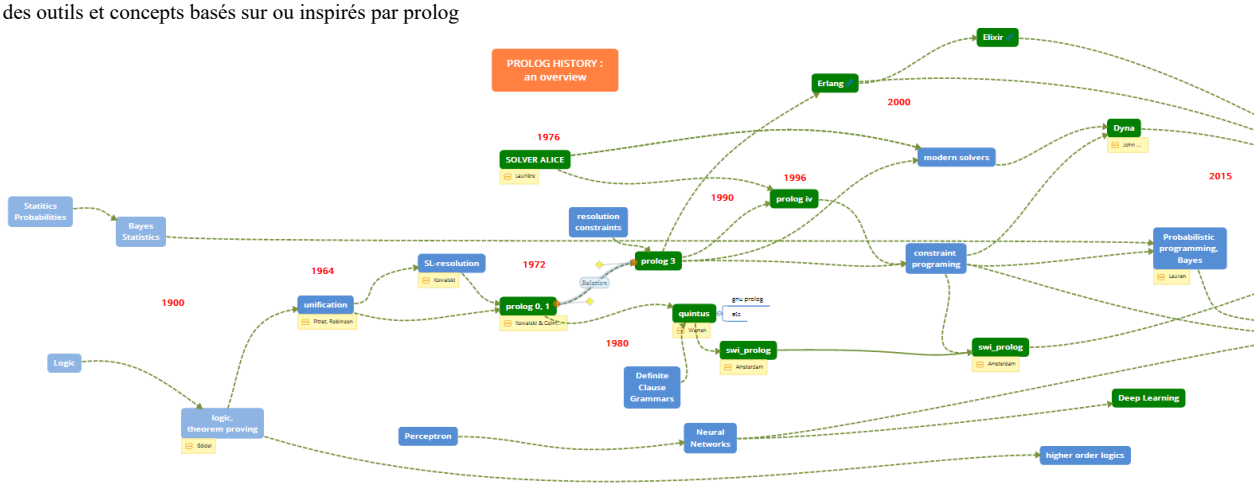


Figure 1. Approximate timeline of some early Prolog systems (up to the ISO Standard).



A global view of the surrounding fields is depicted by the following image : Cartographie et typologie

des outils et concepts basés sur ou inspirés par prolog



2. LES APPLICATIONS

Ce paragraphe ne recense que des applications qui ont réellement tourné dans des contextes opérationnels.

2.1. Difficultés du recensement

Le recensement présente plusieurs difficultés

- peu/pas de publicité des solutions mises en oeuvre.
- solutions remplacées après quelques années par des outils plus efficaces dans d'autres langages.
- ne prend pas en compte les applications "jouet" de laboratoires.

Les applications prolog souffrent de :

- outils d'interface limités
  - entrées sorties
  - graphique
  - modules de calcul auxiliaire échanges de données efficaces .. ou complexes à mettre en oeuvre.
- programmeurs maîtrisant
  - l'outil
  - capacité d'abstraction nécessaire

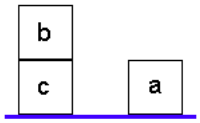
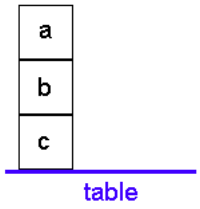
Some requests

- applications in prolog
  - https://www.metalevel.at/prolog/web
  - applications of prolog quite complete set of algorithms and basic examples
  - XPROLOG : prolog on android
  - https://ieeexplore.ieee.org/document/5387386
  - web development
    - http://pbrown.me/blog/swi_spa/
  - https://avaxgfx.com/ebooks/275088-programmer-passport-prolog.html
  - formal european policies in prolog (project Light)
  - Formal Models and Techniques for Analyzing Security Protocol

2.2. Les programmes jouets

On désigne par "programme jouet" un programme destiné à démontrer un principe ou un outil, mais qui n'a pas été jusqu'au bout d'une implémentation en vraie grandeur ou opérationnelle dans le monde réel.
Il y a beaucoup de programmes jouets intéressants pour des raisons de pédagogie :

APPLICATION	Objectif	notes	techniques utilisées
le monde des cubes	Des cubes empilables sont sur une table. Il s'agit • de trouver une séquence qui permet d'une configuration à une autre • de guider un "robot manipulateur" pour déplacer les cubes Exemple	peu de contraintes (ex : cubes vs Hanoï) ou place limitée sur la table se cantone souvent à des cas triviaux (3 cubes, pas 1000 cubes) les programmes présentés ne passent pas à l'échelle.	résolution de problèmes traitement du langage naturel graphique

	 		formalisme pas toujours évident si beaucoup plus de cubes documentation pauvre possible d'avoir plus d'objets (boules, pyramides, etc), et couleurs ?	

On peut même constater que les programmes "jouets" n'ont pas permis la constitution de librairies ré-utilisables

2.3. Exemples applications du passé

Le tableau suivant présente un recensement de quelques applications:

Thème	Année	Contexte	Note	Résultats
traitement de textes	1995	traitement des doublons et fautes d'orthographe dans les adresses postales d'une base de données. plusieurs dizaines de millions d'adresses.	Solutions avec programme impératif classique(/java)	temps de traitement = ...
conception de pièces en matériaux composites	1988	Besoin d'un outil de CAO pour produire des plans de fabrication de pièces en matériaux composites d'épaisseur variable (et donc le nomre de couches et leur orientations). problème combinatoire	??	Arrangement des couches produit en quelques secondes. Production des plans identiques avec plans d'atelier du processus de fabrication fabrication de centaines de pièces pour l'A320.
conception de molécules	1988-1992	recherche de structures ayant les bonnes propriétés de spectre et d'organisation moléculaire. problème combinatoire	solution réécrite complètement en C, puis en python et C	identification de centaines de molécules
windows NT	1994	Along with its C++ code and SProlog interpreter, the NCPA.CPL file has a 700-line SProlog program embedded into it as a textual Windows "resource." cf here • The SProlog algorithm has several features, but primarily it exploits Prolog's inherent backtracking mechanism by exhaustively checking each extant component with every other to determine compatibility. A set of potential "bindings" (one-directional links) is asserted into the Prolog database in the first pass. Then the other		not in use anymore https:\\www.drdobbs.com\\cpp\\extending-c-with-prolog\\184409294 code available here prolog en C https:\\www.drdobbs.com\\cpp\\extending-c-with-prolog\\184409294#00a1_003a

		constraints are checked, and any associations which would violate negative constraints are retracted. Finally, the remaining database information is used to construct NT namespace device names for each component's "bindings." • This network configuration methodology was designed to facilitate the installation of component ensembles which could not be foreseen in 1991, when the work began. Since components declare their classes and constraints themselves, the Prolog interpreter can be counted upon to perform correctly without requiring updates to the NT binaries themselves. • This program is the actual network-configuration algorithm; a C++ class "wrapper" encapsulates the SProlog engine and exposes C++ member functions for facilities such as consulting and querying. Network configuration is performed by consulting the "rules" (the resource-based algorithm), consulting the "facts" (the Registry-derived declarative information), and performing a single query which runs the configure everything predicate. Then the C++ code			

		performs many smaller queries to enumerate the results of the main query from the SProlog database. This information is rearranged and written back into each network component's Registry area. Then, when the network is started, this updated information is used by each software process to determine what other software modules are to be dynamically connected to it.		

2.4. Exemples actuels

Thème	Année	Contexte	Note	Résultats
compréhension du langage naturel	2020 ->	analyse de pages html et suivi des news des agences	cf causality link	analyse quotidienne de milions de news au dessus d'une couche de TLN en python. après spacy.
production				
Marque commerciale		Watson / IBM • surtout Java et C • des gros morceaux en prolog Articles : ◦ https://www.cs.nmsu.edu/ALP/2011/03/natural-language-processing-with-prolog-in-the-ibm-watson-system/		
Logical English		traduction d'un mini langage de règles en prolog (Kowalski) usages • représentation standardisée de règles de comportement et calcul the most popular approach to building Computational Law systems today is based on Computational Logic", (Genesereth), Genesereth M (2015) Computational law: the cop in the backseat. In third annual futurelaw conference		
programmation low code / GeneXUs		GeneXus is a Low Code, cross-platform, knowledge representation-based development tool,[1][2] mainly oriented towards enterprise-class applications for web applications, smart devices, and the Microsoft Windows platform. GeneXus site		
Knowledge Graphs (TerminusDB		TerminusDB is an open source knowledge graph and document store. It is used to build versioned data products. It is a native revision control database that is architecturally similar to Git. It is listed on DB-Engines. cf WOQL as a web object query language		

3. ATTENDUS ET REALITES DE PROLOG

3.1. Les prolog modernes

Il y a de nombreuses variantes modernes de prolog

- caractéristiques communes
- différences

Les avantages des prolog modernes :

- rapidité d'exécution : les machines de 2002 sont 1 millird de fois plus puissantes que les machines de 1985.
- solvers autres que unification
  - avec contraintes
    - avec intervalles
    - sur ensembles

3.2. Critères d'évaluation

critère	Attendu	Remarque	Réel
lisibilité	formalisme compact pour problèmes complexes	Sous-sujet 3	programmes souvent illisibles pour celui qui ne les a pas écrit. programmes mal documentés Liens entre formalisme et réalité peu clairs.
simplicité			
efficacité			
dialogue avec utilisateur			
généralité d'usage	la logique devrait résoudre tous les problèmes	existence de prologs flous ? lenteur ? modèles à base de NN plus efficaces	besoin de flou besoin d'évaluer des solutions avec une/des fonction/S d'évaluation calcul fonctionnel ou impératif nécessaire
langage naturel	production de textes pour explications	modèles avec des RN plus souples ?	toujours un sujet plutôt absent, peut-être par manque de systèmes capables de raisonner de façon non triviale.
apprentissage	en opposition à la programmation de règles		problème bien compris dans les cas simples, mais pas dans les cas complexes. impact sur le coût du calcul souvent non mesurée.
intégration avec des processus algorithmiques.			

NOTES

- dès qu'une application est bien comprise et maîtrisée en prolog, elle est réécrite pour plus d'éfficacité.
- difficulté de maîtriser des centaines de règles
- le noyau prolog, qui est concentré sur l'unification, ne fournit pas d'outils d'interface homme-machine confortables. Les programmeurs se tournent vers d'autres outils.

3.3. Nouveaux axes

3.3.1. stats + neuronal

Via Eric de la Clergerie

Nom	Outil	Lien	Info	notes
Pedro Domingos	Alchemy	https:\\alchemy.cs.washington.edu\\	software package providing a series of algorithms for statistical relational learning and probabilistic logic inference, based on the Markov logic representation. Alchemy allows you to easily develop a wide range of AI applications, including: • Collective classification	

			• Link prediction • Entity resolution • Social network modeling • Information extraction	
Jason Eisner	Dyna	https:\\dyna.readthedocs.io\\en\\latest\\	Dyna is an new declarative programming language developed at Johns Hopkins University.	logic + solver
ProbLog2 was developed in the DTAI group of KULeuven,	Problog	https:\\dtai.cs.kuleuven.be\\problog\\	Probabilistic logic programs are logic programs in which some of the facts are annotated with probabilities. ProbLog is a tool that allows you to intuitively build programs that do not only encode complex interactions between a large sets of heterogenous components but also the inherent uncertainties that are present in real-life situations.	tutorial
ProbLog2 was developed in the DTAI group of KULeuven,	DeepProbLog	https:\\github.com\\ML-KULeuven\\deepproblog	**DeepProbLog is an extension of ProbLog** that integrates Probabilistic Logic Programming with deep learning by introducing the neural predicate. The neural predicate represents probabilistic facts whose probabilites are parameterized by neural networks. For more information, consult the papers listed below. • exemples • tutorial 42 mn	en python besoin de • ProbLog • PySDD • PyTorch • TorchVision • PySwip

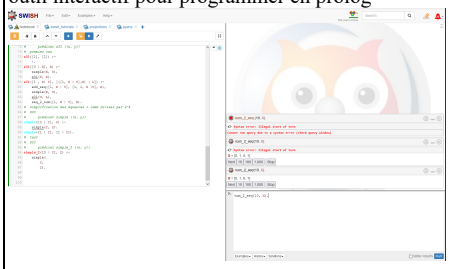
3.3.2. usages au sein de frameworks

Exemple	notes
Jupyter	au point ou pas ?

4. LIBRAIRIES EXISTANTES

4.1. Usages

prolog system	description	web server	graphics	doc generator	sql interface	web client server	constraints
swi_prolog		yes	XPCE	yes	??		

			library only linux ? rustique				
tau prolog	An open source Prolog interpreter in JavaScript	by definition				manipulation du DOM	
visual prolog							
Prolog IV							
SWISH	outil interactif pour programmer en prolog 						
other ?			GUI				

USAGES	note
TTS Text to speech	swi prolog pack_install(cowsay)

Open Source projects used by SWI-Prolog

- CMake Configuration and build management
- dtoa.c (Float <-> text conversion by David M. Gay).
- GMP (Unbounded and rational arithmetic)
- GNU (Linux, development tools, libraries, etc.)
- LibYAML (A C library for parsing and emitting YAML)
- MacPorts (Porting environment for MacOS)
- MurmurHash (string hashing)
- Nullsoft (Windows installer)
- NetBSD (crypt implementation for Windows)
- OpenSSL (secure sockets)
- PCRE (Perl Compatible Regular Expressions)
- SHA routines by Brian Gladman
- Snowball NLP stemmer library
- UnixODBC (Database connectivity on Unix)
- utf8proc (unicode normalization)
- X.org (Graphics on Unix)
- Zlib (compression)
- Qt GUI console

Il y a un paradoxe : alors que Prolog est censé être excellent en génération de texte, il n'y a pas de module Text-To-Speech dans prolog
==> Il faut utiliser des outils externes

mais il yen a un en python ! ou librairies C, etc

4.2. implémentation

prolog system	company	price	platforms	other
swi_prolog	university	free	Unix, Mac, Windows	interface -> C/C++, python, java, JPL, C# Edinburgh Prolog standard
tau prolog	university/GIT	free	on line tool	
visual prolog	PDC	free professional edition - 100$		
Prolog IV	university	free		
SWISH				
other ?				

4.3. Voir aussi

5. Recommandations pour un futur de prolog

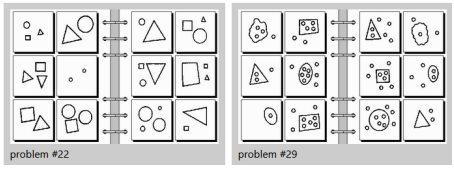
1. développer un git spécialisé prolog
    - librairies de résolutions de problèmes
    - pousser les jouets à l'extrème. Par exemple :
        - monde des cubes = 1000 cubes, pas juste 3 cubes
        - dialogue en langage naturel avec centaines de phrases et mots pas juste 20 mots de vocabulaire

    - librairies d'interfaces
2. publiciser les solutions développées.
    - chaque conférence prolog doit avoir des papiers invités sur les applications
3. install an interactive Prolog insinde the Jupyter framewordk
    - it seems that existing solutions are not fully ok .... but you can find more answers here
    - http://alexriina.com/2019/02/11/ipython-prolog/ works ?

6. Bibliographie

6.1. via Laurent Cervoni

- sémantique pour le ML https:\\arxiv.org\\pdf\\2202.09868v1.pdf 2022
- Bâtment https:\\doi.org\\10.1016\\j.autcon.2021.103756 accessible si paywall ou ici logic representation for BIM building offsite construction.pdf
- (psy et démonstrateur de théories)
    - https://www.sciencedirect.com/science/article/pii/S0732118X20302130 accessible si paywall ou ici using logic programming for theory representation and scientific inference (2021).pdf
    - https://jeanchristopherohner.github.io/theory-toolbox-2 https:\\jeanchristopherohner.github.io\\theory-toolbox-2\\#writtenDoc cf discussion sur "Advantages of Logic Programming"
- explicabilité en IA Prolog-based agnostic explanation module for structured pattern classification (2021)
- (modélisation empreinte carbone) Green Application Placement in the Cloud-IoT Continuum(2021)
- Résolution des puzzles de Bongard (1967) en Prolog et en DeepLearning :
    - Using Program Synthesis and Inductive Logic Programming to solve Bongard Problems - 2021



    - https://k10v.github.io/2018/02/25/Solving-Bongard-problems-with-deep-learning/ http://www.foundalis.com/res/diss_research.html
        - http://www.foundalis.com/soc/why_no_more_Bongard.html
        - Bongard Problems - Critique of RF4 : The RF4 approach to input representation, and why it should be avoided in cognitive science.

6.2. Doc générale

▶ Détails

6.3. Notes pour le challenge

- Connaissances de base quelques sites intéressants
    - méta site avec plein de liens
    - sur github
        - https://github.com/klaussinani/awesome-prolog#math

Faire une hiérarchie de problèmes du plus simple au plus complexe

- Prolog de base
    - pour apprendre
        - revers d'une liste
        - compter en prolog cf Jean Rohmer
    - simples
        - tours de hanoi
        - jeu de permutations
        - dessiner des fractales en prolog
        - mini eliza
    - moyens
        - cubes avec <= 5 cubes
        - fonction dérivation
        - eliza mieux
    - difficiles
        - cubes avec <= 50 cubes
            - manipulation interactive
        - résolution de problèmes
    - utilisation des librairies
        - graphiques
            - animation des tours de Hanoi
            - animation des cubes
        - mini serveur web
    - très difficiles
        - graphique plus sophistiqué
        - cubes avec > 100 cubes variantes
            - espace contraint
            - sur un plan
            - optimisation des trajets
            - etc
        - fonction intégration
        - agent conversationnel pour aider à écrire un programme
- prolog avec solver et autres variantes ???

7. APPENDIX

main source

7.1. Prolog implementations

Name	C Interface[3]	Code Profiler	Compiled Code	Debugger	Interactive Interpreter	Java Interface[3]	Licence	Native Graphics	Native OS Control	OS
AllegroProlog	Yes, via Lisp	Yes, via Lisp	Yes	Yes	Yes	Yes, via Lisp	Commercial, limited free edition available		Yes, via Lisp	Unix, Windows, Mac OS X
BProlog	Yes	Yes	Yes	Yes	Yes	Yes	Free for non-commercial uses		Yes	Unix, Windows, Mac OS X
Ciao	Yes	Yes	Yes	Yes	Yes	Yes	GPL, LGPL		Yes	Unix, Windows, Mac OS X
DOS-PROLOG			Yes	Yes			Shareware	Yes	Yes	MS-DOS
ECLiPSe	Yes	Yes	Yes	Yes	Yes	Yes	MPL		Yes	Linux, Windows, Solaris, macOS
GNU Prolog	Yes		Yes	Yes	Yes		GPL, LGPL		Yes	Unix, Windows, Mac OS X
JIProlog	Yes via Java			Yes	Yes	Yes	Shareware/Commercial and AGPL	Yes	Yes via Java	JVM, Android
JLog			Yes		Yes	Yes	GPL	Yes		JVM
JScriptLog					Yes		GPL			Web Browser
jTrolog				Yes	Yes	Yes	LGPL			JVM
LPA-PROLOG	Yes	Yes	Yes	Yes	Yes	Yes	Commercial	Yes	Yes	Windows
Open Prolog				Yes			Freeware			Mac OS
Poplog Prolog	Yes		Yes	Yes	Yes		Free Open Source	Only through POP-11, on Linux	Yes	Linux (32- and 64-bit), Unix, Windows
SICStus Prolog	Yes	Yes	Yes	Yes	Yes	Yes	Commercial	Yes	Yes	Unix, Linux, Windows, macOS
Strawberry Prolog			Yes	Yes			Freeware, Commercial	Yes		Windows, Unix
SWI-Prolog	Yes	Yes	Yes	Yes	Yes	Yes	BSD License	Yes	Yes	Unix, Linux, Windows, macOS
tuProlog	Yes			Yes	Yes	Yes	LGPL	Yes		JVM, Android
Visual Prolog	Yes	Yes	Yes	Yes			Freeware, Commercial	Yes	Yes	Windows
XSB Prolog	Yes	Yes	Yes	Yes	Yes	Yes	LGPL		Yes	Linux, Windows, Solaris, macOS
YAP-Prolog	Yes		Yes	Yes	Yes	Yes	GPL or Artistic (user choice)		Yes	Linux, Windows, Solaris, Mac OS X, HP-UX

7.2. Operating system and Web-related features

Name	Conditional compilation	HTML Parser	HTTP client	HTTP server	Multi-threading	RDF Triple store	Sockets	Tabling
BProlog								Yes
Ciao	Yes	Yes	Yes	Yes	Yes		Yes	Yes
ECLiPSe	Yes	Yes	Yes		Yes		Yes	
GNU Prolog							Yes	
LPA-Prolog		Yes	Yes	Yes			Yes	
SICStus Prolog	Yes				Yes		Yes	
SWI-Prolog	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Visual Prolog	Yes	Yes	Yes	Yes	Yes		Yes	
XSB	Yes		Yes		Yes		Yes	Yes
YAP-Prolog	Yes				Yes		Yes	Yes

7.3. Optimizations

Name	Choice Point Elimination	Environment Trimming	Just-in-Time Indexing	Tail-Call Optimization
Ciao	Yes	Yes	?	Yes
ECLiPSe	Yes	Yes	multi-argument (compile time)	Yes
GNU Prolog	Yes	Yes	?	Yes
SICStus Prolog	Yes	Yes		Yes
SWI-Prolog	Yes	Yes	Yes	Yes
Visual Prolog	Yes (compile time)	N/A	N/A (compile time)	Yes (compile time)
XSB	Yes	Yes	?	Yes
YAP-Prolog	Yes	Yes	Yes	Yes

7.4. Static analysis

Name	Call-pattern checker	Determinacy checker	Type checker
Ciao	Yes	Yes	Yes
GNU Prolog			
SICStus Prolog		Yes	
SWI-Prolog		Yes	
Visual Prolog	Yes	Yes	Yes
XSB			
YAP-Prolog			

SOMMAIRE