Fifty Years of Prolog and Beyond

Robert Kowalski
RuleMl Webinar 26 Oct 2022

https://prologyear.logicprogramming.org/PrologYear.html

Disclaimer:

This presentation is a personal perspective

2022: The Year of Prolog
Celebrating the 50th anniversary of Prolog

2022: The Year of Prolog

Organized by The Association for Logic Programming and The Prolog Heritage Association

In the summer of 1972, Alain Colmerauer and his team in Marseille developed and implemented the first version of the logic programming language Prolog. Together with both earlier and later collaborations with Robert Kowalski and his colleagues in Edinburgh, this work laid the practical and theoretical foundations for the Prolog and logic programming of today. Prolog and its related technologies soon became key tools of symbolic programming and Artificial Intelligence.

The birth of Prolog

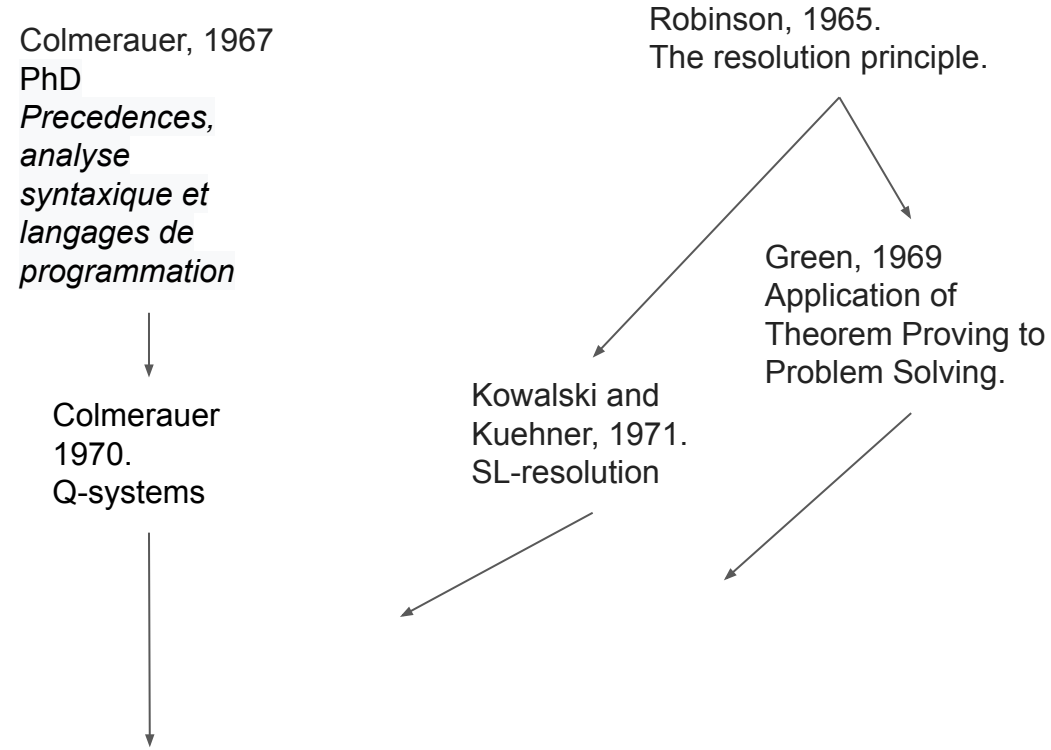
Alain Colmerauer and Philippe Roussel

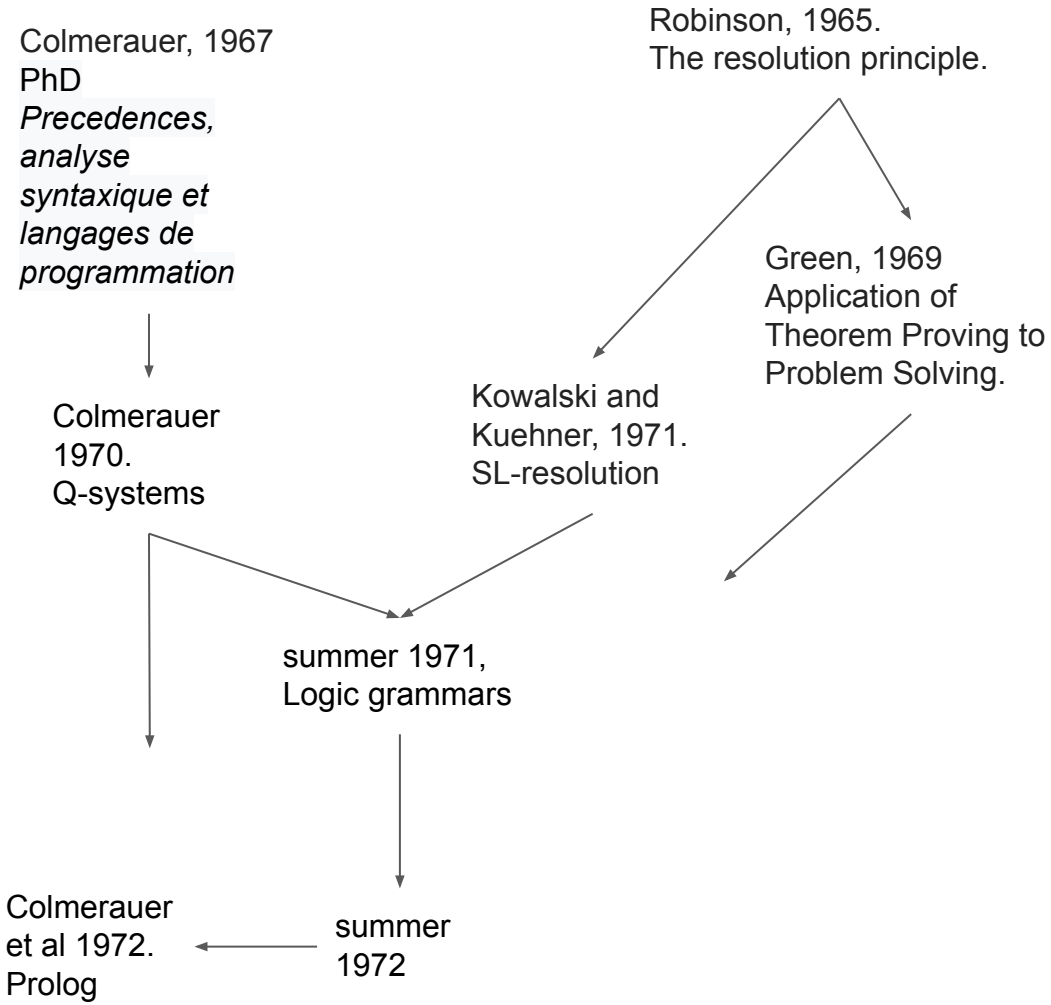
November 1992

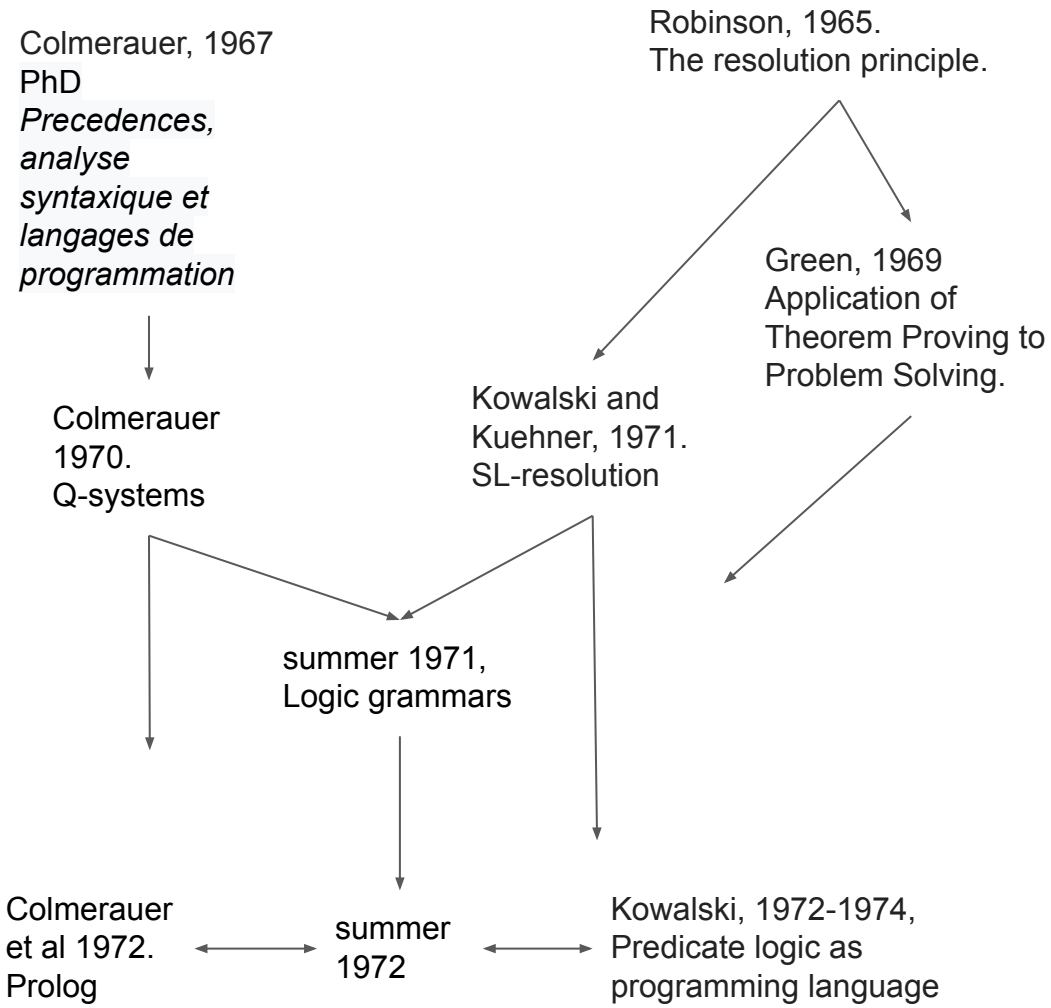
Abstract

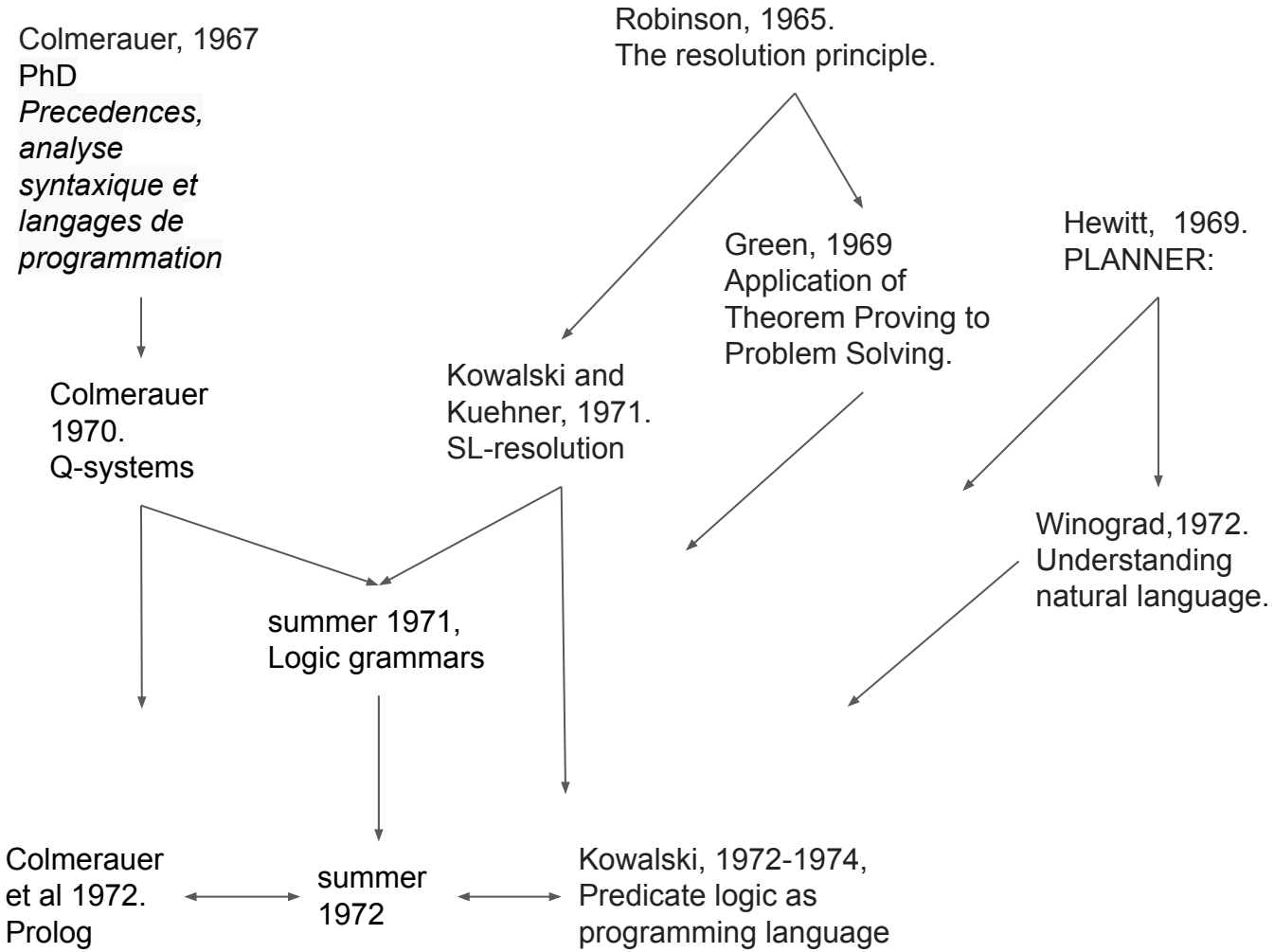
The programming language, Prolog, was born of a project aimed not at producing a programming language but at processing natural languages; in this case, French. The project gave rise to a preliminary version of Prolog at the end of 1971 and a more definitive version at the end of 1972. This article gives the history of this project and describes in detail the preliminary and then the final versions of Prolog. The authors also felt it appropriate to describe the Q-systems since it was a language which played a prominent part in Prolog's genesis.

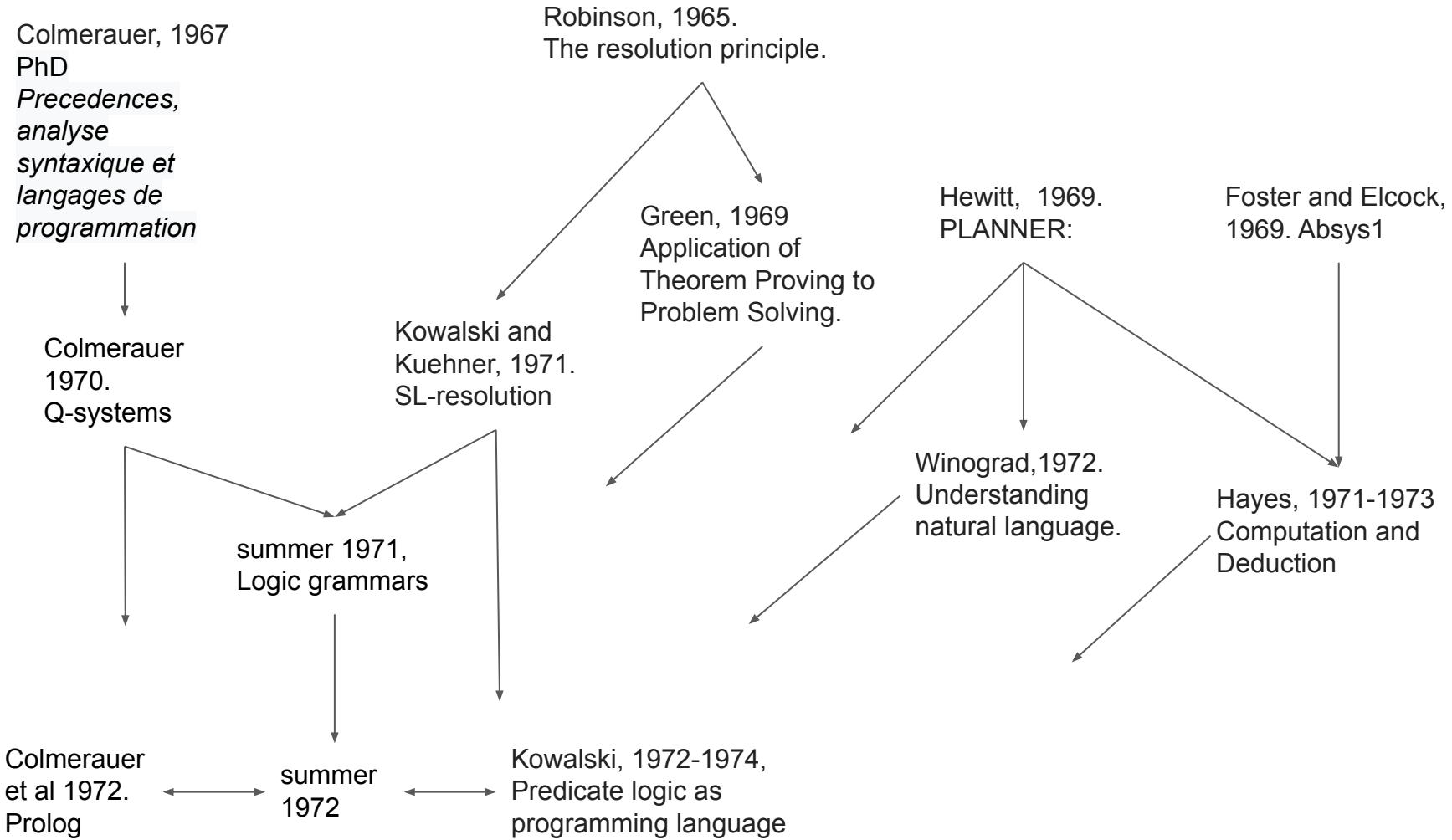
Contents


Colmerauer, 1967
PhD
Precedences, analyse syntaxique et langages de programmation
Colmerauer 1970.
Q-systems
Colmerauer et al 1972.
Prolog
Robinson, 1965.
The resolution principle.
Kowalski and Kuehner, 1971.
SL-resolution
Green, 1969
Application of Theorem Proving to Problem Solving.


Colmerauer, 1967
PhD
Precedences, analyse syntaxique et langages de programmation
Robinson, 1965.
The resolution principle.
Green, 1969
Application of Theorem Proving to Problem Solving.
Colmerauer 1970.
Q-systems
Kowalski and Kuehner, 1971.
SL-resolution
summer 1971,
Logic grammars
summer 1972
Colmerauer et al 1972.
Prolog


Colmerauer, 1967
PhD
Precedences, analyse syntaxique et langages de programmation
Robinson, 1965.
The resolution principle.
Green, 1969
Application of Theorem Proving to Problem Solving.
Colmerauer 1970.
Q-systems
Kowalski and Kuehner, 1971.
SL-resolution
summer 1971,
Logic grammars
Colmerauer et al 1972.
Prolog
summer 1972
Kowalski, 1972-1974,
Predicate logic as programming language


Colmerauer, 1967
PhD
Precedences, analyse syntaxique et langages de programmation
Robinson, 1965.
The resolution principle.
Green, 1969
Application of Theorem Proving to Problem Solving.
Hewitt, 1969.
PLANNER:
Colmerauer 1970.
Q-systems
Kowalski and Kuehner, 1971.
SL-resolution
Winograd,1972.
Understanding natural language.
summer 1971,
Logic grammars
Colmerauer et al 1972.
Prolog
summer 1972
Kowalski, 1972-1974,
Predicate logic as programming language


Colmerauer, 1967
PhD
Precedences, analyse syntaxique et langages de programmation
Robinson, 1965.
The resolution principle.
Green, 1969
Application of Theorem Proving to Problem Solving.
Hewitt, 1969.
PLANNER:
Foster and Elcock, 1969. Absys1
Colmerauer 1970.
Q-systems
Kowalski and Kuehner, 1971.
SL-resolution
Winograd,1972.
Understanding natural language.
Hayes, 1971-1973
Computation and Deduction
summer 1971,
Logic grammars
Colmerauer et al 1972.
Prolog
summer 1972
Kowalski, 1972-1974,
Predicate logic as programming language

Groupe de recherche en

Intelligence Artificielle

U.E.R. de Luminy

Université d'Aix-Marseille

Rapport de recherche
sur le contrat
CRI n° 72-18 de
février 72 à juin 73

UN SYSTEME DE COMMUNICATION HOMME-MACHINE EN FRANCAIS

A. COLMERAUER

H. KANOUI

P. ROUSSEL

R. PASERO

Texte soumis

TOUT PSYCHIATRE EST UNE PERSONNE.
CHAQUE PERSONNE QU'IL ANALYSE, EST MALADE.
*JACQUES EST UN PSYCHIATRE A *MARSEILLE.
EST-CE QUE *JACQUES EST UNE PERSONNE?
OU EST *JACQUES?
EST-CE QUE *JACQUES EST MALADE?

Réponse

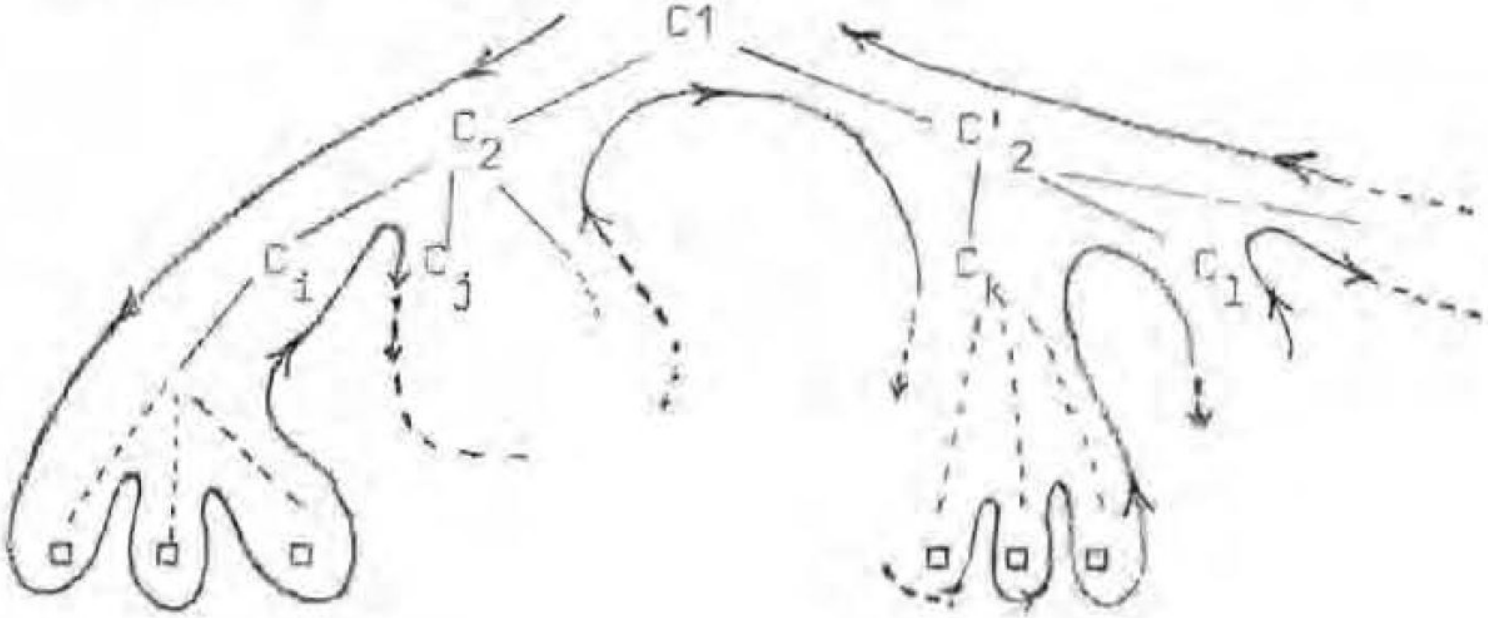
OUI

A MARSEILLE

JE NE SAIS PAS

```
+Frere(*y,*z)-Pere(*x,*y)-Pere(*x,*z).

+Pere(Paul,Pierre)

+Mere(Marie,Jacques)

+Mari(Paul,Marie)

+Pere(*x,*y)-Mari(*x,*z)-Mere(*z,*y)


** CONCATENATION DE LISTES ..

+CONC(*X.NIL,*X) ..
+CONC((*X.*Y).*Z,*X.*U) -CONC(*Y.*Z,*U) ..
+CONC(NIL.*X,*U) -CONC(*X,*U) ..
```

On peut représenter les descendants d'une clause C_1 de <données> sous forme d'un arbre:



What is Prolog?

Procedure = Horn clause + Top-down reasoning (SL-resolution)

(Algorithm = Logic + Control)

So:

- Computational procedures can be given a logical form.
- Horn clause reasoning can be performed as efficiently as computation.

What is Prolog?

Prolog is not only a programming language,
but also a computer language for

- databases
- modelling (and specification)
- AI Knowledge Representation and Problem Solving

And vice-versa.

What is the relationship with Datalog and ASP?

- The same basic syntax.
- The same basic semantics.
- But no use of the procedural interpretation.

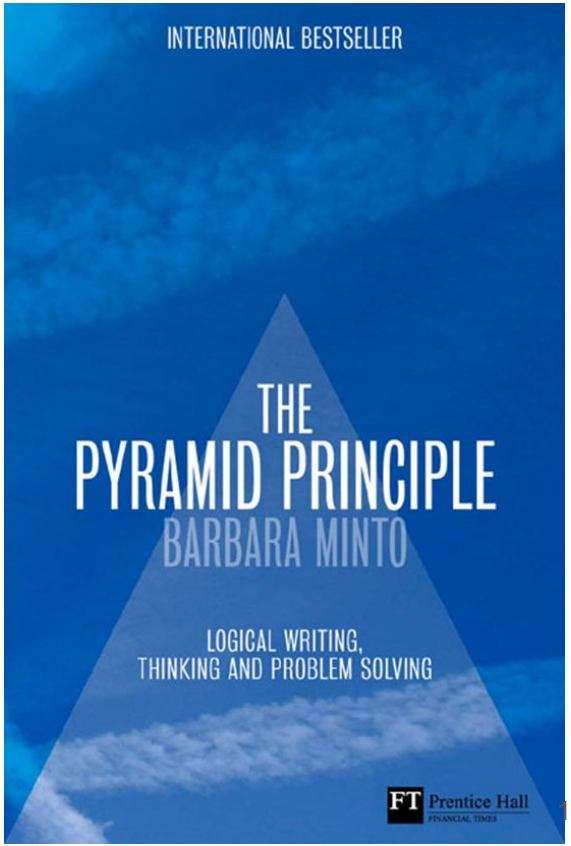
So:

- No concern with choosing an efficient representation.

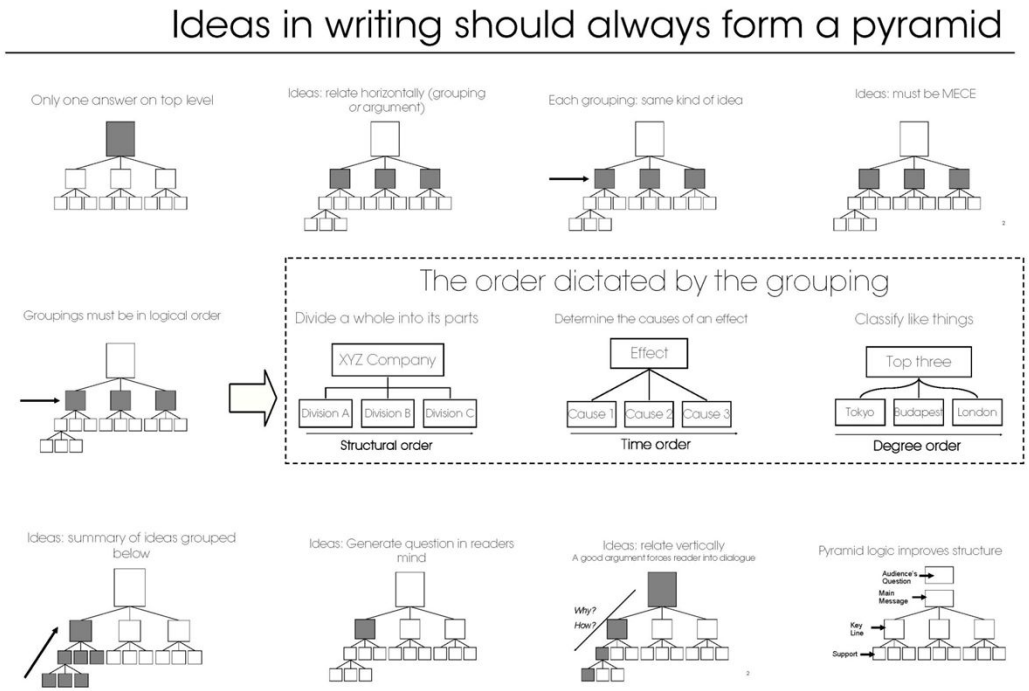
- Limited ability to reason with infinite structures, e.g. append.

What is Prolog and Why is it important?

Pure Prolog is the only computing paradigm
based on a logical view of
human thinking.



The Pyramid Principle



JAZZCODE™
jazzcode.org

Computing with logical relations is much more natural (and more efficient) than computing with conventional procedures

One rule in Prolog:

```
grandparent(Elder, Younger) :- Parent(Elder, Person), Parent(Person, Younger).
```

Many procedures in conventional programming languages:

to test the grandparent relationship between two persons, given as input.

to find grandparents as output, given grandchildren as input.

to find grandchildren as output, given grandparents as input.

etc.

What are some of the weakness of ISO Prolog and how can they be remedied?

- Positional notation for arguments of predicates
  e.g. what does parent(X, Y) :- mother(X, Y) mean?

  solution1: named arguments
  e.g. parent(older -> X younger -> Y) :- mother(older -> X younger -> Y).

  solution 2: infix predicates
  e.g X is_parent_of Y :- X is_mother_of Y.

  solution 3: controlled natural language
  e.g. a person X is a parent of a person Y if X is the mother of Y.

What are some of the weakness of ISO Prolog and how can they be remedied?

- Relational notation for functions
  e.g. bigger(X, Y) :- size(X, U), size(Y,V), U>V.

  solution1: nested notation:
  bigger(X, Y) :- size(X) > size(Y).

  solution2: controlled natural language
  a thing X is bigger than a thing Y if the size of X is greater than the size of Y.

What are some of the weakness of ISO Prolog and how can they be remedied?

Nested notation can be generalised from functions to relations:

Example: grandparent(X, Y) :- parent(X, Z), parent(Z, Y).

Rewrite: a person X is a grandparent of a person Y
if X is a parent of a parent of Y.

What are some of the weakness of ISO Prolog and how can they be remedied?

- Assert and retract (and input-output) do not have a logical interpretation.

  solution0: Embrace full Prolog with its impure features.

  solution1: Keep Prolog pure and add extra arguments to predicates.

  solution2: Combine pure Prolog with an existing imperative programming language, e.g Python.

  solution3: Extend pure Prolog to include mutable state with a purely logical semantics, e.g.

  Transaction Logic
  Epilog with active rules
  LPS with reactive rules
  DALI with reactive rules
  CHR with propagation rules

  solution4: Embed pure Prolog into an agent architecture, where Prolog is the agent's language of thought.

  solution5: Combine Deliberation RuleML with Reaction RuleML?

What are some of the weakness of ISO Prolog and how can they be remedied?

- Horn clauses and first-order logic alone are not adequate for reasoning/computing with exceptions.

  solution0: Negation as failure (weak negation)

  solution1: Combine NAF with explicit (strong, but not classical) negation

The Prolog Day Symposium

November 10, 2022 ⊢ Save the date and Register Here!

This Prolog-Day Symposium will present the highlights of the Year of Prolog, celebrating the **50th anniversary of the birth of Prolog**. The Symposium will include the award of the inaugural edition of the **ALP Alain Colmerauer Prolog Heritage Prize** (in short: the Alain Colmerauer Prize) for *recent practical accomplishments that highlight the benefits of Prolog-inspired computing for the future*, and presents the Prolog education initiative for introducing young people to Prolog. The Symposium will visit the past, present, and future of Prolog and explore its advantages as a language for explainable AI and for computing more generally. The day includes four sessions; the last session is devoted to the Alain Colmerauer prize. The other three sessions consist of presentations and round tables.

The Prolog Day Symposium

November 10, 2022 ⊢ Save the date and Register Here!

- 09h15 – 10h45: Session 1 **What is Prolog and why is it important?**

  This session, chaired by Robert Kowalski will include a talk by Manuel Hermenegildo and a round table whose members will include Stefania Constantini, Randy Goebel, Gopal Gupta, Michael Genesereth, Manuel Hermenegildo, Jean Rohmer, and David S. Warren.

- *10h45 – 11h15: Break*
- 11h15 – 12h45: Session 2 **Prolog Thinking and Education**

  This session, chaired by Veronica Dahl will include a talk by Veronica Dahl and a round table whose members will include Julien Brasseur, Laura Cecchi, José Francisco Morales, Asya Stoyanova-Doycheva, and Robert Kowalski.

- *12h45 – 14h15: Lunch*

elcome and ringmaster of the afternoon: Celestin Sedogbo

- 14h15 – 14h45: **Alain Colmerauer and the Prolog Adventure (film)**
- 14h45 – 15h00: On stage: **Several generations of Prolog Heroes**
- 15h00 – 16h00: Session 3: **Prolog-powered applications**

  This session, chaired by Laurent Gouzènes and Annie Liu, will include a talk by Laurent Gouzènes and a round table chaired by David S. Warren and whose members will include Mats Carlsson, Laurent Cervoni, John Gallagher, Laurent Gouzènes, Benjamin Grosof, and Jan Wielemaker.

- Session 4: **Alain Colmerauer Prize** Chair: Francesca Rossi
- 16h00 – 16h40: *Short-listed candidate presentations*
  - *ProB: Harnessing the Power of Prolog to Bring Formal Models and Mathematics to Life.* Michael Leuschel.
  - *Logic Model Processing.* Pierre Dissaux.
- *16h40 – 17h00: Break*
- 17h00 – 18h00: *Short-listed candidate presentations*
  - *Symbium: The Computational Law Company.* Michael Genesereth, Abhijeet Mohapatra, and Leila Banijamali.
  - *PROLEG: Practical Legal Reasoning System.* Ken Satoh.
  - *Pacioli: a PROLOG system for financial report validation*. Miguel Calejo and and Charles Hoffman
- 18h00 – 18h15: **Announcement of the Prize winner and Prize ceremony**
- 18h15 – 18:20: **Launching of the "Future of Prolog Initiative"** (Robert Kowalski)
- 18h20 – 18h30: **Closing Ceremony**

- 09h00 – 09h15: **Opening Addresses**
- 09h15 – 10h45: Session 1 **What is Prolog and why is it important?**

  This session, chaired by Robert Kowalski will include a talk by Manuel Hermenegildo and a round table whose members will include Stefania Constantini, Randy Goebel, Gopal Gupta, Michael Genesereth, Manuel Hermenegildo, Jean Rohmer, and David S. Warren.

- *10h45 – 11h15: Break*
- 11h15 – 12h45: Session 2 **Prolog Thinking and Education**

  This session, chaired by Veronica Dahl will include a talk by Veronica Dahl and a round table whose members will include Julien Brasseur, Laura Cecchi, José Francisco Morales, Asya Stoyanova-Doycheva, and Robert Kowalski.

- *12h45 – 14h15: Lunch*

Welcome and ringmaster of the afternoon: Celestin Sedogbo

- 14h15 – 14h45: **Alain Colmerauer and the Prolog Adventure (film)**
- 14h45 – 15h00: On stage: **Several generations of Prolog Heroes**

- 15h00 – 16h00: Session 3: **Prolog-powered applications**

  This session, chaired by Laurent Gouzènes and Annie Liu, will include a talk by Laurent Gouzènes and a round table chaired by David S. Warren and whose members will include Mats Carlsson, Laurent Cervoni, John Gallagher, Laurent Gouzènes, Benjamin Grosof, and Jan Wielemaker.

- Session 4: **Alain Colmerauer Prize** Chair: Francesca Rossi
- 16h00 – 16h40: *Short-listed candidate presentations*
  - *ProB: Harnessing the Power of Prolog to Bring Formal Models and Mathematics to Life.* Michael Leuschel.
  - *Logic Model Processing.* Pierre Dissaux.
- *16h40 – 17h00: Break*
- 17h00 – 18h00: *Short-listed candidate presentations*
  - *Symbium: The Computational Law Company.* Michael Genesereth, Abhijeet Mohapatra, and Leila Banijamali.
  - *PROLEG: Practical Legal Reasoning System.* Ken Satoh.
  - *Pacioli: a PROLOG system for financial report validation*. Miguel Calejo and and Charles Hoffman
- 18h00 – 18h15: **Announcement of the Prize winner and Prize ceremony**
- 18h15 – 18:20: **Launching of the "Future of Prolog Initiative"** (Robert Kowalski)
- 18h20 – 18h30: **Closing Ceremony**
- *18h30 – 20h00: Cocktail*

ICLP at Imperial College 9-15 July 2023

- End of Fifty-Year celebrations
- Launch of the State of the Art and Directions for the Future Prolog Book
- Continuation of the ALP Alain Colmerauer Prize for
  recent practical accomplishments that highlight
  the benefits of Prolog-inspired computing for the future,