Escola Politécnica da USP - Depto. de Enga. Mecatrônica

PMR-3510 Inteligência Artificial

Aula 2 - Resolução de problemas em IA

Prof. José Reinaldo Silva

reinaldo@usp.br

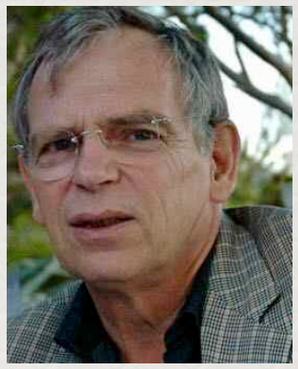
Aula passada: uma introdução sobre o surgimento da IA

O tema do processamento computacional de processos racionais continuou na pauta de vários pesquisadores e engenheiros até que em 1956, John McCarthy propôs um evento com os principais interessados no Dartmouth College, financiado pelo Rockefeller Foundation. Organizadores foram, além de McCarthy, Marvin Minsky (Harvard), Nathaniel Rochester (IBM) e Claude Shannon (Bell Labs). O termo "Inteligência Artificial" usado pela primeira vez com o significado atual neste evento.

... uma linguagem para "resolver problemas"!

Sem dúvida esta linguagem tem que ser capaz de expressar "conhecimento", isto é, ações que não são puramente comandos (procedurais) de forma sucinta e fácil de interpretar. A base matemática pra isso é a Lógica, e, um sub-conjunto da lógica de primeira ordem pode ser transformada em linguagem interpretada para servir a este propósito.

A linguagem Prolog (de Programação Lógica) foi criada para servir ao propósito de expressar de forma declarativa (e não procedural) ações cognitivas, dentro de um sub-conjunto da lógica de primeira ordem. Foi criada por Alain Colmerauer e Robert Kowalski e lançada em 1972.



Alain Colmerauer (1941-2017)

Robert Kowalski (1945-2017)

```
mother_child(trude, sally).

father_child(tom, sally).
father_child(tom, erica).
father_child(mike, tom).

sibling(X, Y)      :- parent_child(Z,
X), parent_child(Z, Y).

parent_child(X, Y) :- father_child(X,
Y).
parent_child(X, Y) :- mother_child(X,
Y).
```

SWI Prolog

Robust, mature, free. **Prolog for the real world.**

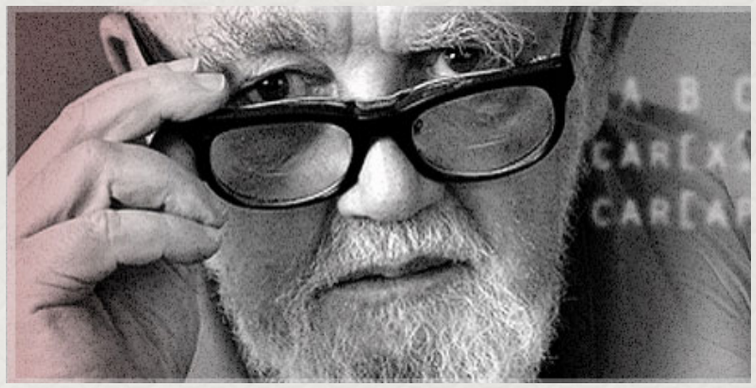
HOME | DOWNLOAD | DOCUMENTATION | TUTORIALS | COMMUNITY | USERS | WIKI

SWI-Prolog offers a comprehensive free Prolog environment. Since its start in 1987, SWI-Prolog development has been driven by the needs of real world applications. SWI-Prolog is widely used in research and education as well as commercial applications. Join over a million users who have downloaded SWI-Prolog. more ...

Download SWI-Prolog Get Started Try SWI-Prolog online

Programação nos anos 50

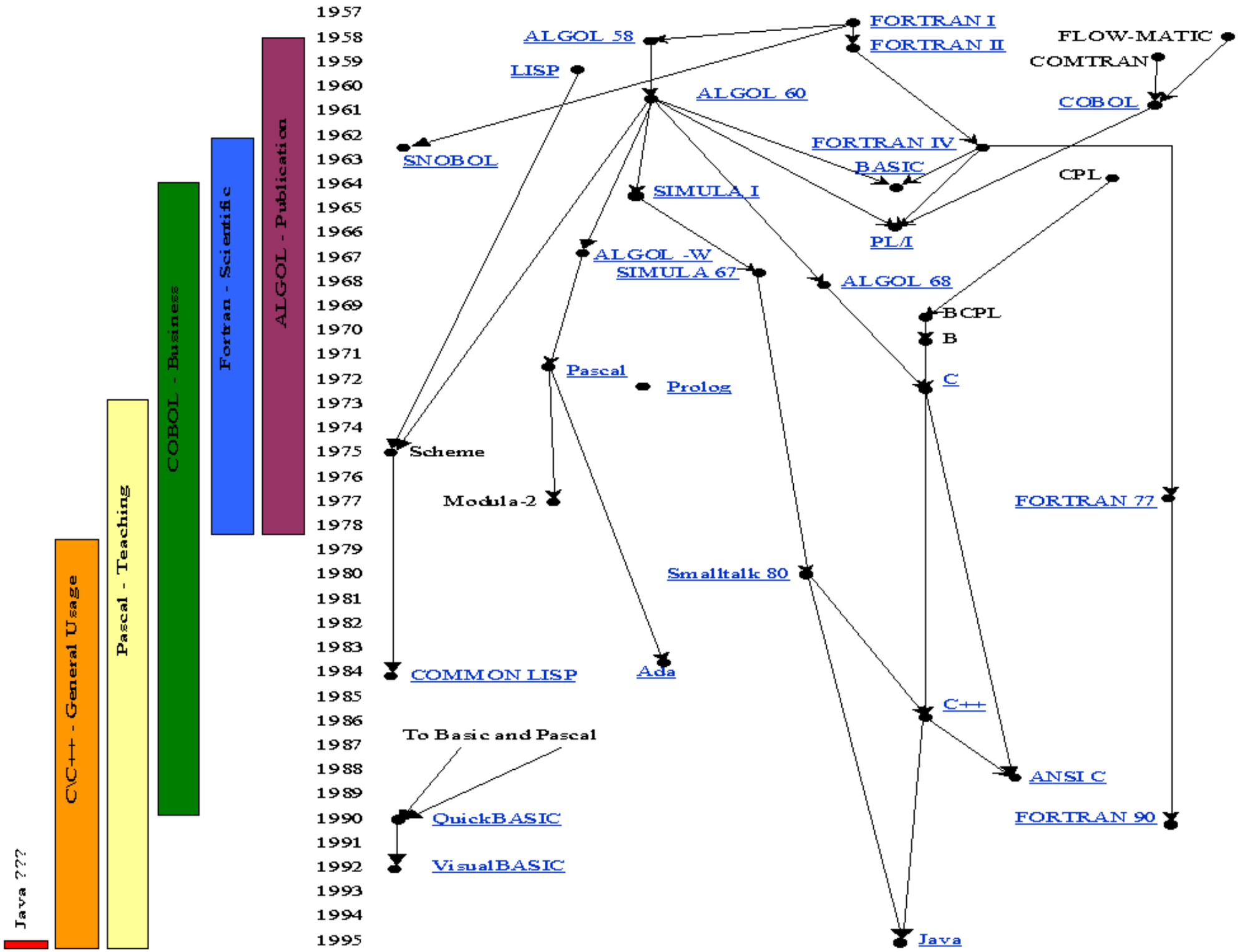
As linguagens de programação mais difundidas eram o FORTRAN e o COBOL, e existia o LISP, criado por John McCarthy em 1958



John McCarthy

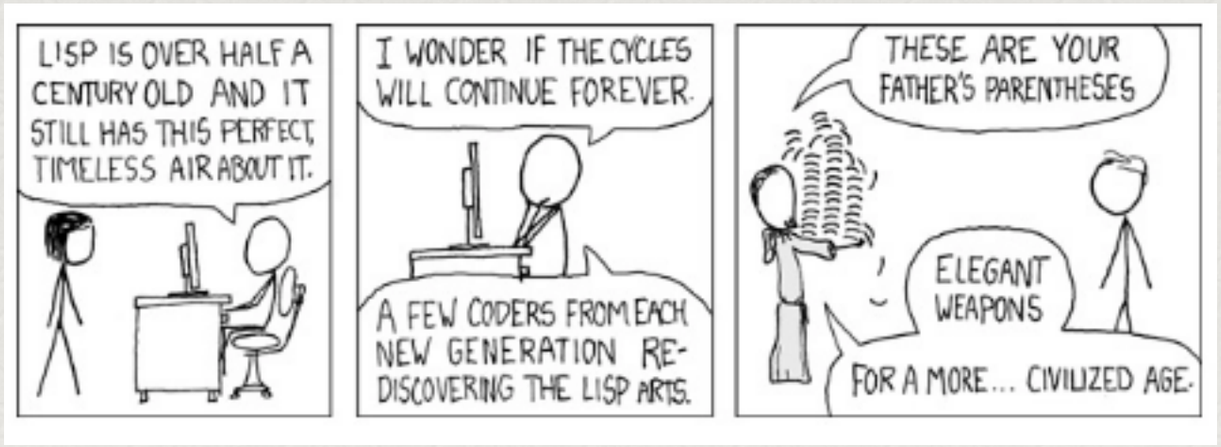
McCarthy e seu grupo passaram os dois anos - desde o verão de 1956 até o verão de 1958 - desenvolvendo uma nova linguagem de programação e desafiando os que foram ao evento de 1956 a fazerem "programas inteligentes". Herbert Simon and Allan Newell foram os primeiros a aceitar o desafio e apresentaram uma nova versão do Logist Theorist, já mostrado no evento do Dartmouth College.

History of Programming Languages



A linguagem LISP

O LISP foi a primeira linguagem usada para IA. Foi lançada em 1958 pelo próprio John McCarthy. É usada até hoje e apareceram sistemas mais modernos como o Allegro Common Lisp (v. 10.1) e o CLOS (Common Lisp Object System).



A linguagem Prolog

Alain Colmerauer
Aix-Marseille

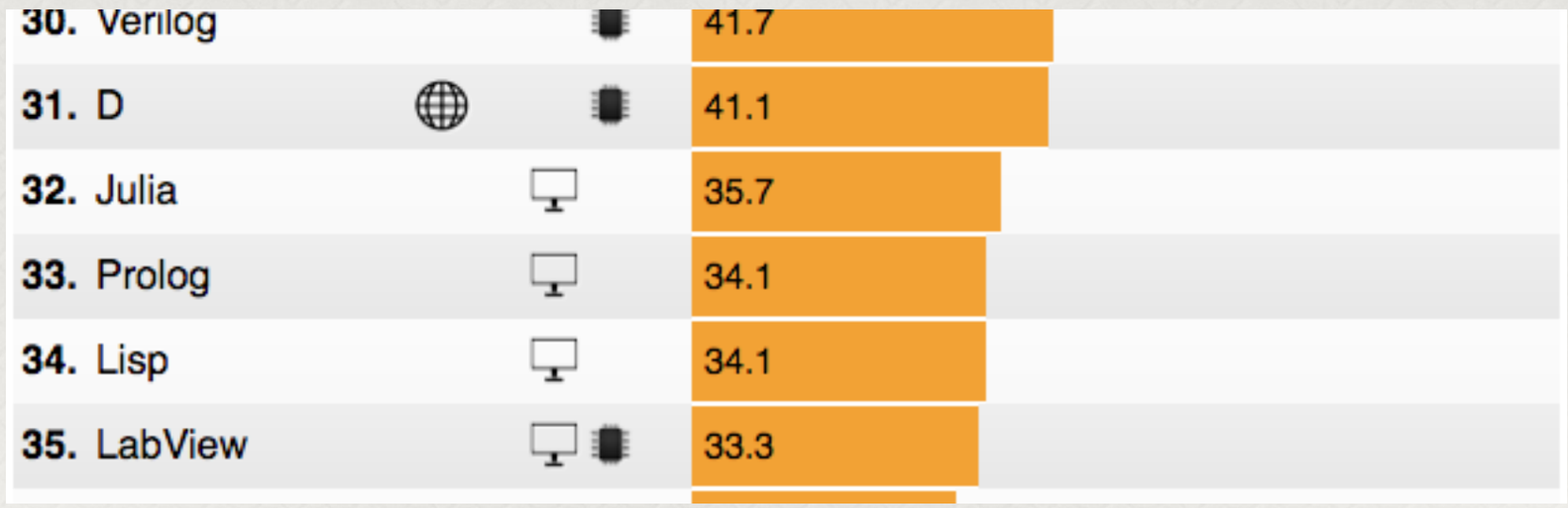
Robert Kowalski
Univ. of Edinburgh

O Prolog foi criado à lançada no inicio dos anos 70, à partir do trabalho de Alain Colmerauer em cooperação com Robert Kowalski. Embora seja uma linguagem que traduz fielmente os objetivos da IA clássica (deep knowledge) o Prolog tem sido usado na prática mas não ocupa hoje os primeiros postos entre as linguagens mais difundidas e usadas no mercado. O termo significa "Programação Lógica", e tem como objetivo facilitar a representação do conhecimento, desde o formato mais simples e especialmente a programação de processos de inferência.

https://www.metalevel.at/prolog/videos/predicate_logic

O uso comercial da linguagem Prolog

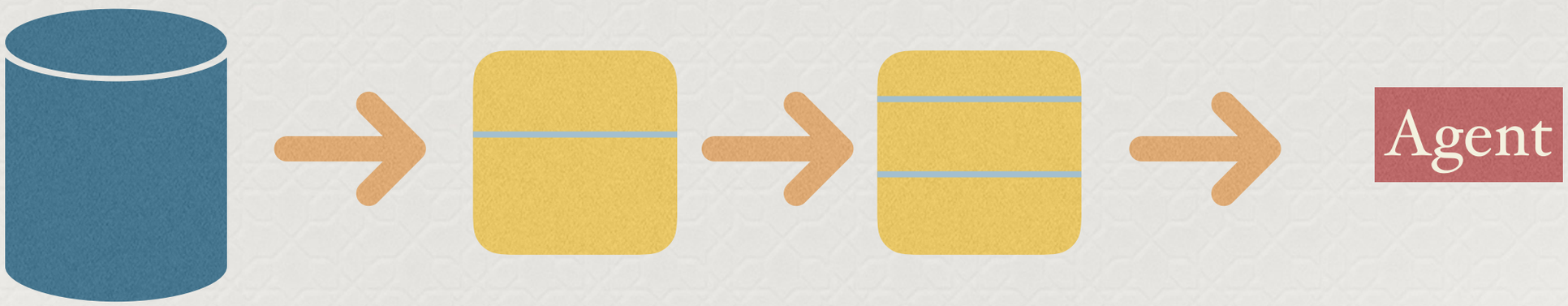
Existem vários desenvolvedores de ambientes de programação Prolog no mercado, como Quintus Prolog, SICStus Prolog, Visual Prolog, etc. Segundo o site da SICStus cerca de 30% dos processos de emissão de passagem aérea são feitos hoje em Prolog (ver site da disciplina).



Introduction

From Data to Intelligent Agents

Portanto temos como objetivo representar conhecimento e processos racionais de forma computável. Assim, temos como ponto de partida a forma mais simples de conhecimento que são os dados, especialmente se tidos como confiáveis e verdadeiros. Mesmo a programação convencional tem dedicado atenção na representação de dados, tendo como tendência a unificação entre a parte estática e a parte dinâmica associada a estes dados.



```
sibling(X, Y)      :- parent_child(Z, X), parent_child(Z, Y).
```

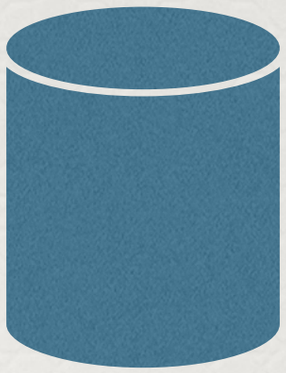
Regras

Relações entre
fatos, Dados e variáveis

pai(maria, X)

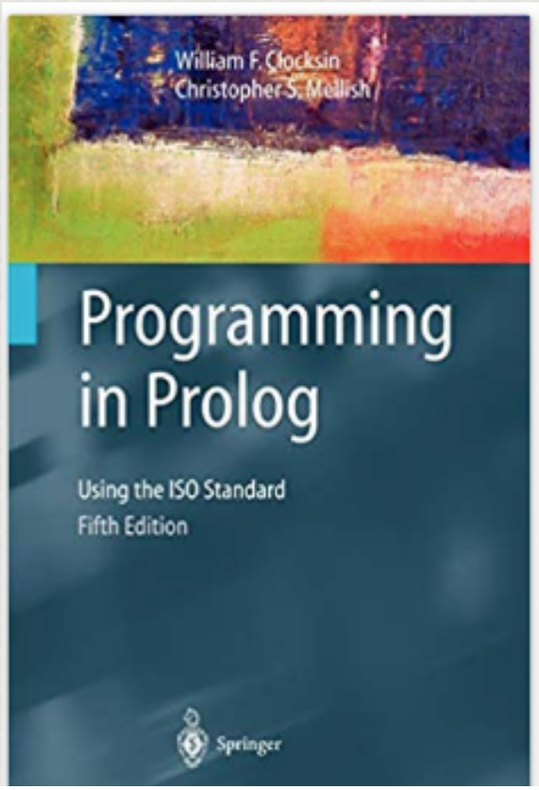
Relações entre
Dados

pai(maria, jose)



homem(jose)

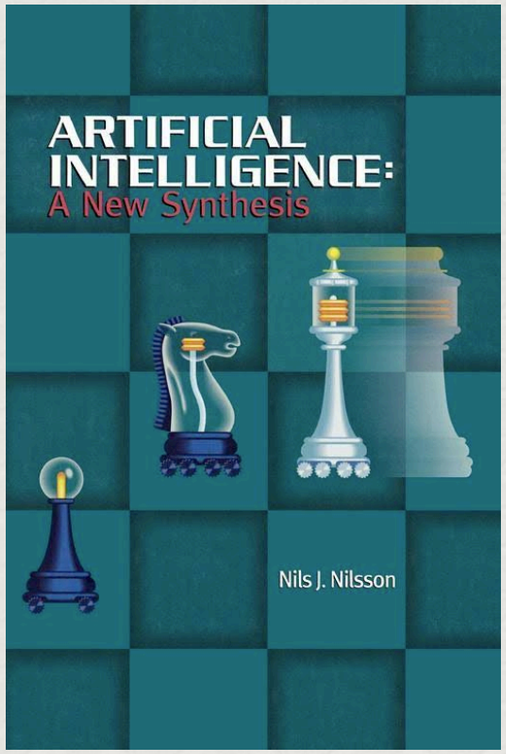
Dados
(fatos)

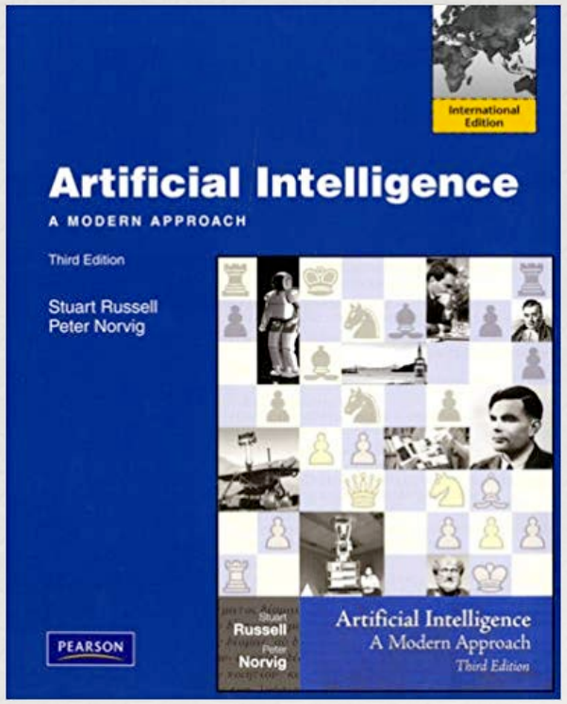


Programming in PROLOG: Using the ISO Standard

25 jul 2003, 5th. Edition

William F. Clocksin and Christopher S. Hellish





Objetos e relacionamentos em Prolog

Cloksin, W. S., Mellish, C. S.; Programming in Prolog, Springer, 2003.

A forma mais simples de se referir a um objeto é pela sua classificação genérica ou classe. Assim, podemos falar sobre um "livro" - sem especificar qual livro, sobre que assunto, de qual autor, etc. Nesse caso não precisamos de uma estrutura e o "nome" ou classificação é o suficiente. Entretanto, esta não é uma informação relevante (ser um "livro") exceto se relacionamos objetos. Por exemplo podemos dizer que "o libro está sobre a mesa", acrescenta algo mais substancial. Também podemos dizer que "o livro pertence ao José" (objeto aqui pode também ser uma pessoa específica, e aí não se trata mais de uma classe e sim de um objeto específico).

Fatos

```
sobre(livro, mesa)
pertence(livro, José)
```

Objetos e relacionamentos em Prolog

Cloksin, W. S., Mellish, C. S.; Programming in Prolog, Springer, 2003.

Portanto a primeira forma de representação de "conhecimento" é através de fatos e admitimos que os fatos inseridos em um programa lógico são verdadeiros. A omissão de um fato significa portanto a sua negação. Podemos ter uma atribuição direta a um objeto sem relacioná-lo com outro objeto: por exemplo, podemos dizer que "Maria é corintiana", o que simplesmente atribui uma propriedade ao objeto "Maria".

obs: Prolog não é uma linguagem orientada a objetos e o termo aqui tem somente parte do significado hoje dado a "objetos".

Fatos

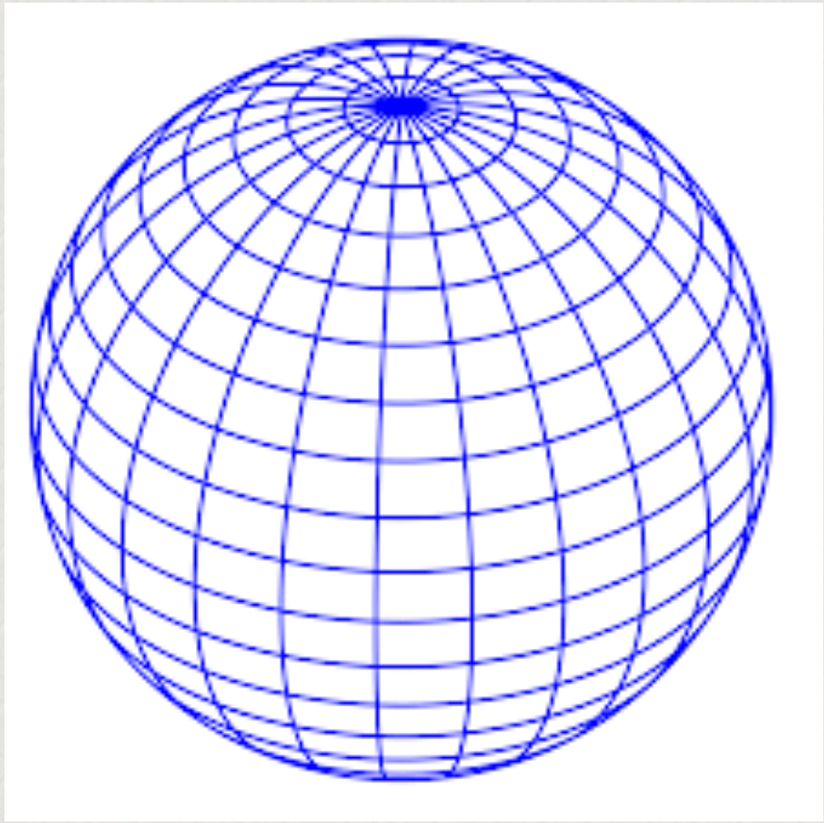
```
sobre(livro, mesa)
pertence(livro, José)

corintiana(Maria)
```

A verdade na teoria pragmatista

A primeira teoria pragmatista foi publicada por Charles Sanders Peirce no artigo "How to Make Our Ideas Clear" ("Como tornar claras nossas idéias"), no número de janeiro de 1878, da revista Popular Science Monthly. Peirce desenvolveu a máxima pragmatista: "Para averiguar o significado de um conceito intelectual, é preciso considerar que suas conseqüências práticas podem ser inferidas como resultantes, necessariamente, da verdade desse conceito. A soma dessas conseqüências constituiria o significado do conceito."

Em outras palavras, postula que, para ter significado, um conceito ou idéia deve apresentar, em primeiro lugar, um correlato prático suscetível de comprovação experimental; segundo, que suas "conseqüências" se diferenciem claramente das de outro conceito. Dessa forma, a verdade de um conceito seria seu processo de verificação.

A hipótese do mundo fechado



...o que não é conhecido não é verdade.

Objetos e relacionamentos em Prolog

Cloksin, W. S., Mellish, C. S.; Programming in Prolog, Springer, 2003.

Podemos agora escrever nossa primeiras linhas de um "programa inteligente" usando como tema a relação entre elementos de uma familia hipotética.

Base de conhecimento

```
mae(maria, paulo)
mae(maria, carla)
mae(susana, jose)
mae(vania, mara)
mae(carla, antonio)

pai(flavio, jose)
pai(flavio, beatriz)

corintiana(maria)
```

Objetos e relacionamentos em Prolog

Cloksin, W. S., Mellish, C. S.; Programming in Prolog, Springer, 2003.

Podemos elaborar queries usando variáveis e estas agora vão em letra maiúscula. Por exemplo, podemos perguntar à nossa base de conhecimento quem é a mãe de jose. A forma de fazer isso é colocar no local reservado às perguntas (o prompt para isso é ?-)

mae(X, jose).

Base de conhecimento

```
mae(maria, paulo)
mae(maria, carla)
mae(susana, jose)
mae(vania, mara)
mae(carla, antonio)

pai(flavio, jose)
pai(flavio, beatriz)

corintiana(maria)
```

www.swi-prolog.org

SWI Prolog

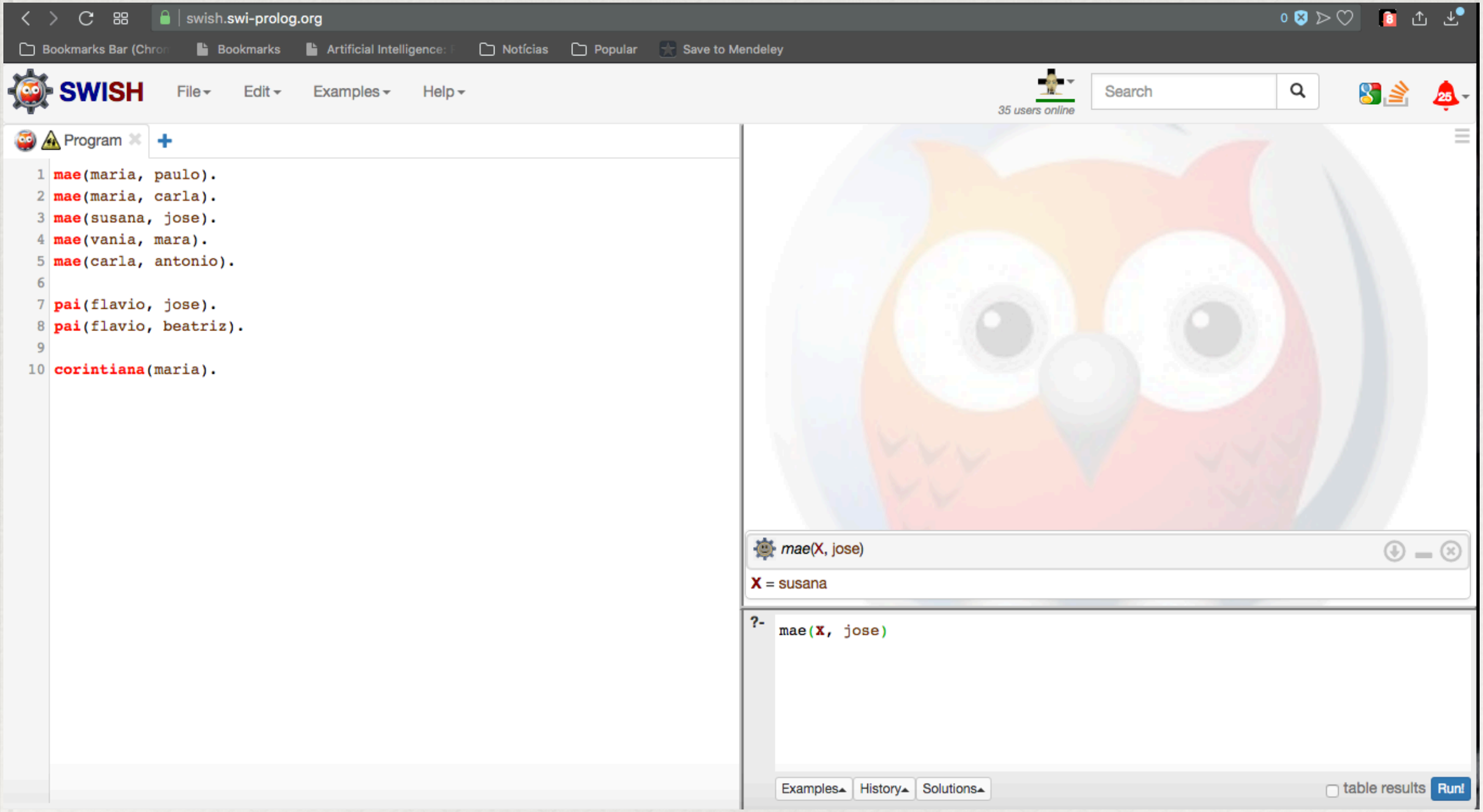
Robust, mature, free. **Prolog for the real world.**

HOME | DOWNLOAD | DOCUMENTATION | TUTORIALS | COMMUNITY | USERS | WIKI

SWI-Prolog offers a comprehensive free Prolog environment. Since its start in 1987, SWI-Prolog development has been driven by the needs of real world applications. SWI-Prolog is widely used in research and education as well as commercial applications. Join over a million users who have downloaded SWI-Prolog. more ...

Download SWI-Prolog Get Started Try SWI-Prolog online

swish.swi-prolog.org



Objetos e relacionamentos em Prolog

Cloksin, W. S., Mellish, C. S.; Programming in Prolog, Springer, 2003.

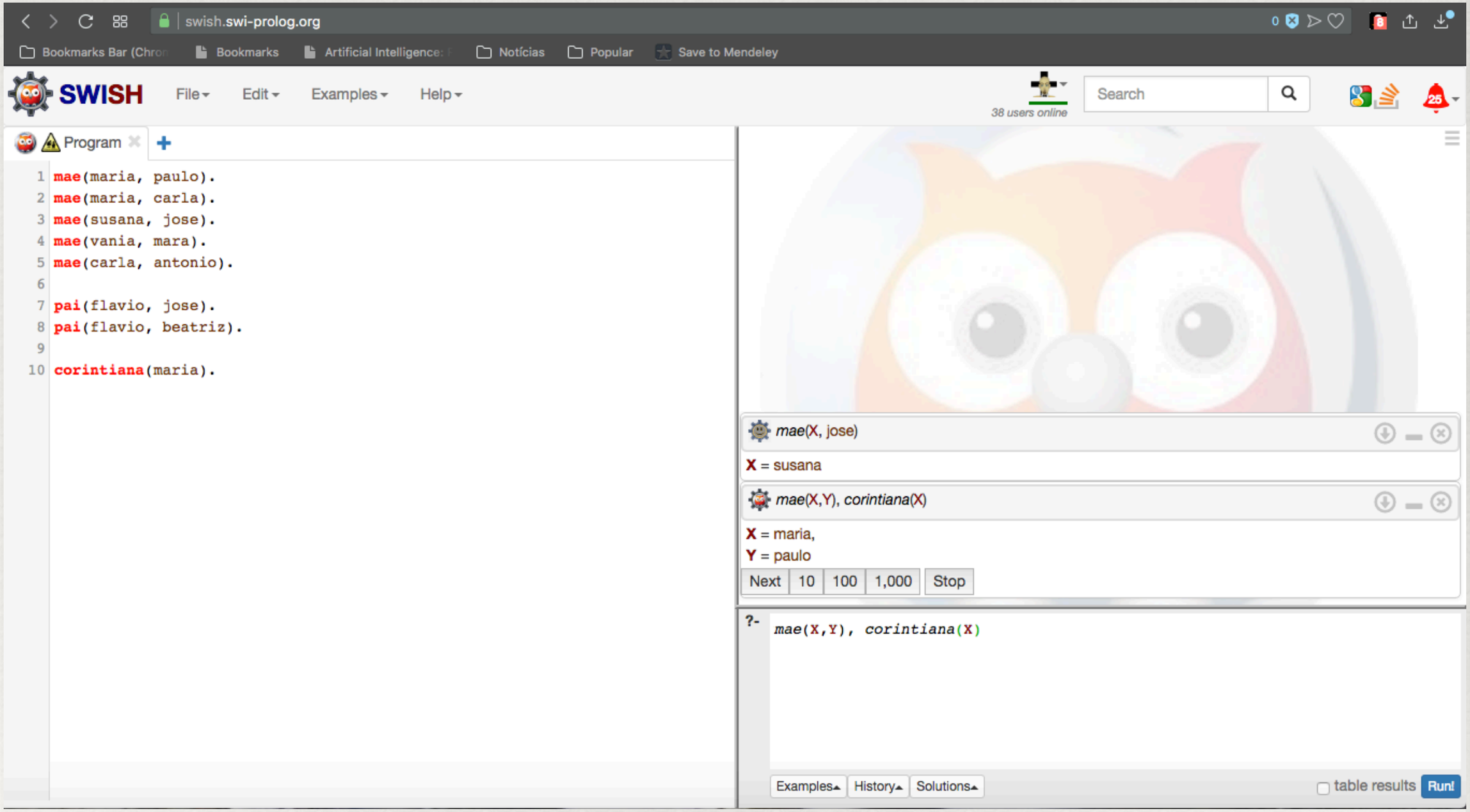
Podemos elaborar queries ainda mais sofisticados perguntando se alguém na nossa base de conhecimento tem uma mãe que seja corintiana. A pergunta seria agora
mae(X, Y), corintiana(X) - note que a virgula aqui funciona como o AND.

Base de conhecimento

```
mae(maria, paulo)
mae(maria, carla)
mae(susana, jose)
mae(vania, mara)
mae(carla, antonio)

pai(flavio, jose)
pai(flavio, beatriz)
```

```
corintiana(maria)
```

swish.swi-prolog.org



```
sibling(X, Y)      :- parent_child(Z, X), parent_child(Z, Y).
```

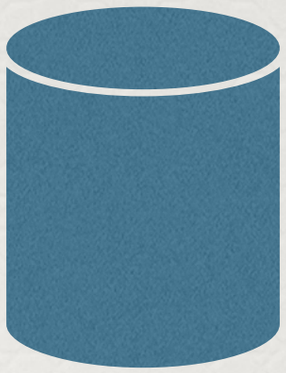
Regras

Relações entre
fatos, Dados e variáveis

pai(maria, X)

Relações entre
Dados

pai(maria, jose)



homem(jose)

Dados
(fatos)

Regras

Sofisticando a base de conhecimento

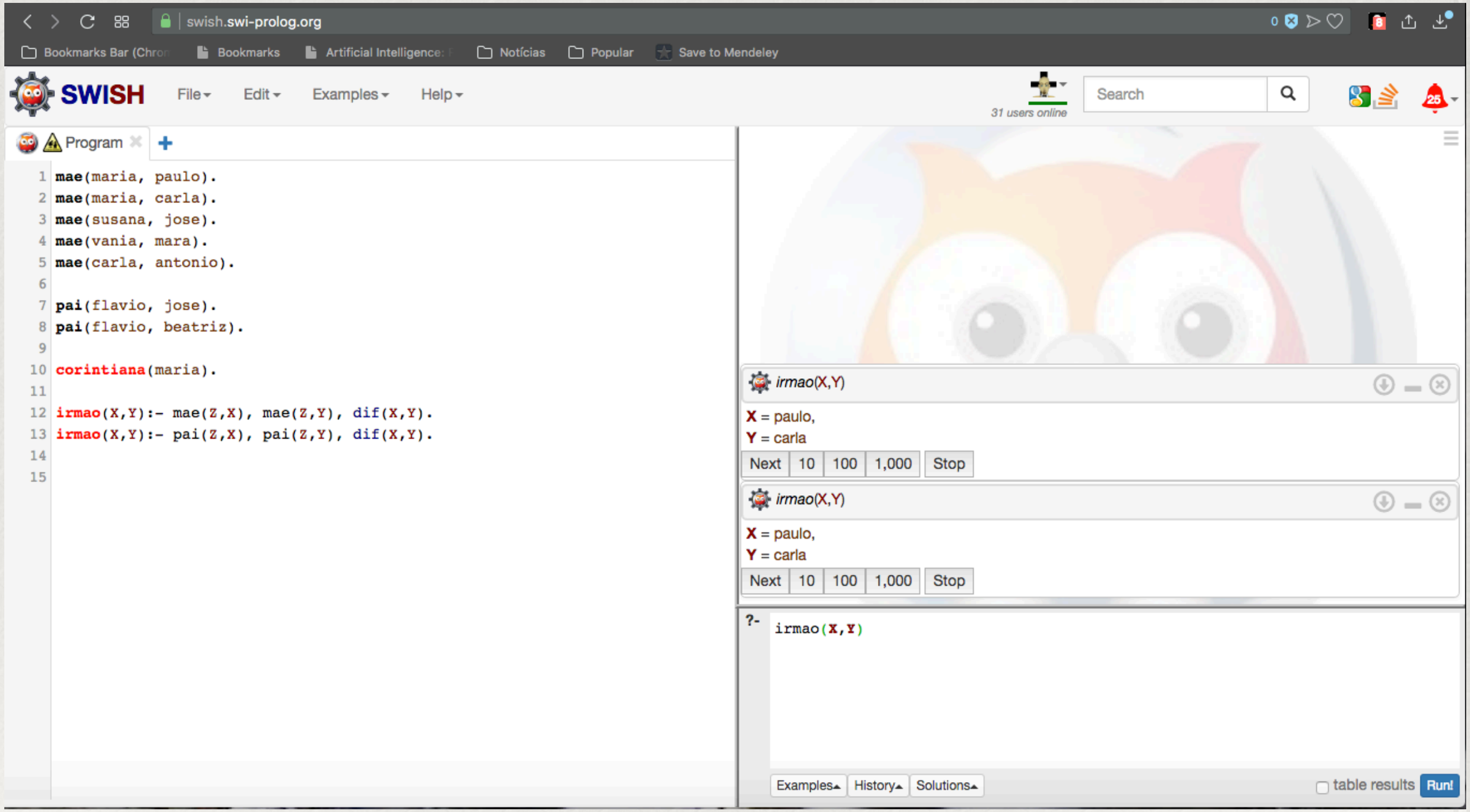
Vamos dar um passo adiante e pensar em relacionar fatos, ou, em outras palavras, ou criar novos fatos (deduzir) à partir de fatos já existentes e de regras de relacionamento. Por exemplo, na nossa base familiar vamos definir o que é o irmão ou irmã. Teoricamente é aquele ou aquela que tem um dos pais em comum, para se bem abrangente. Na nossa base vamos inserir o "conceito" de irmão. E vamos precisar de uma relação pre-definida do Prolog.

Base de conhecimento

```
mae(maria, paulo).
mae(maria, carla).
mae(susana, jose).
mae(vania, mara).
mae(carla, antonio).

pai(flavio, jose).
pai(flavio, beatriz).
```

```
corintiana(maria)

irmao(X, Y) :- mae(Z, X), mae(Z, Y), dif(X, Y)

irmao(X, Y) :- pai(Z, X), pai(Z, Y), dif(X, Y)
```

swish.swi-prolog.org



Regras

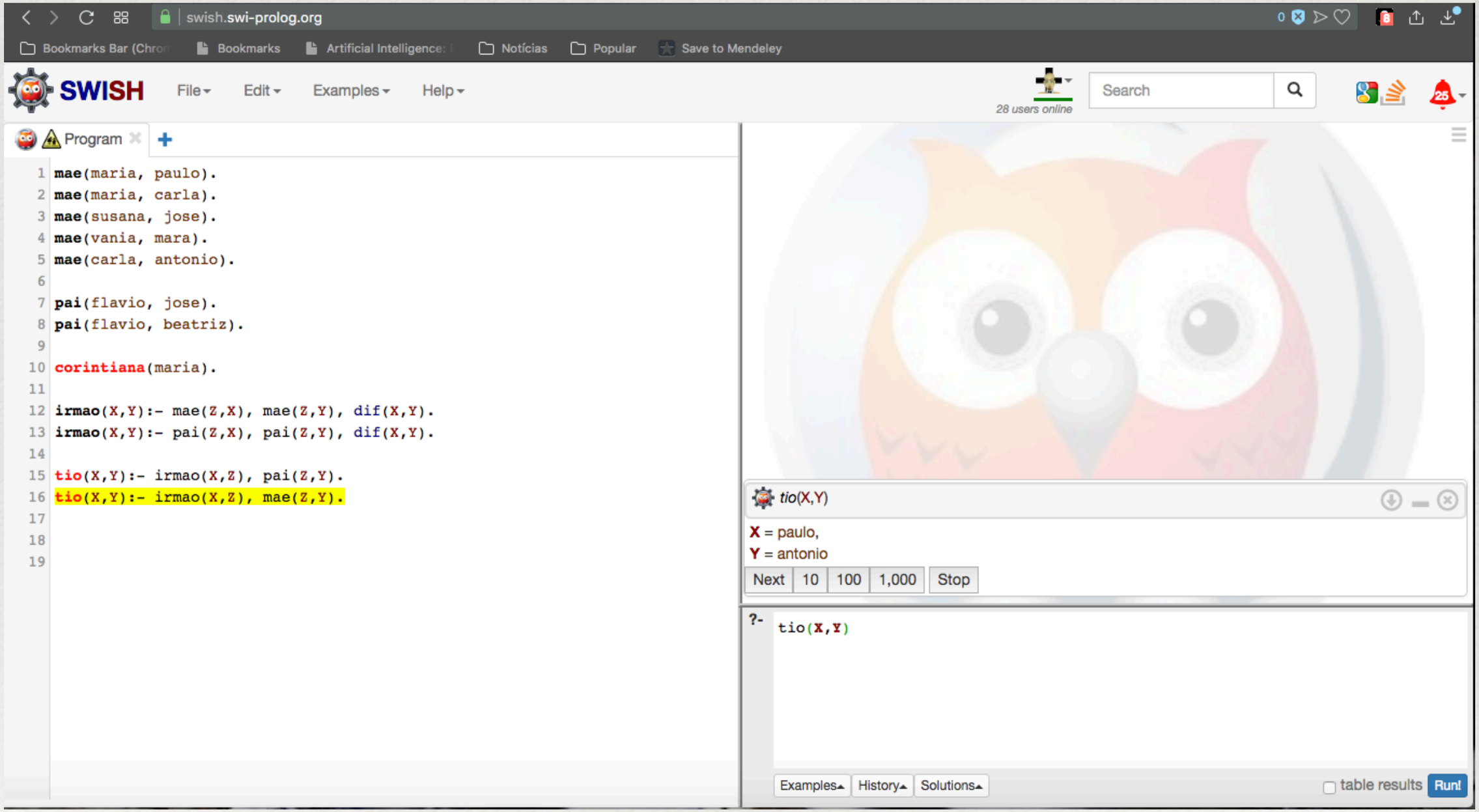
Sofisticando a base de conhecimento

Mais um passo: vamos agora definir o "conceito" de tio (mais uma relação familiar). O tio (tia) é de alguém que é irmão (irmã) do pai ou da mãe deste indivíduo (objeto). Como poderemos representar isso em regras de produção? Veja abaixo.

Base de conhecimento

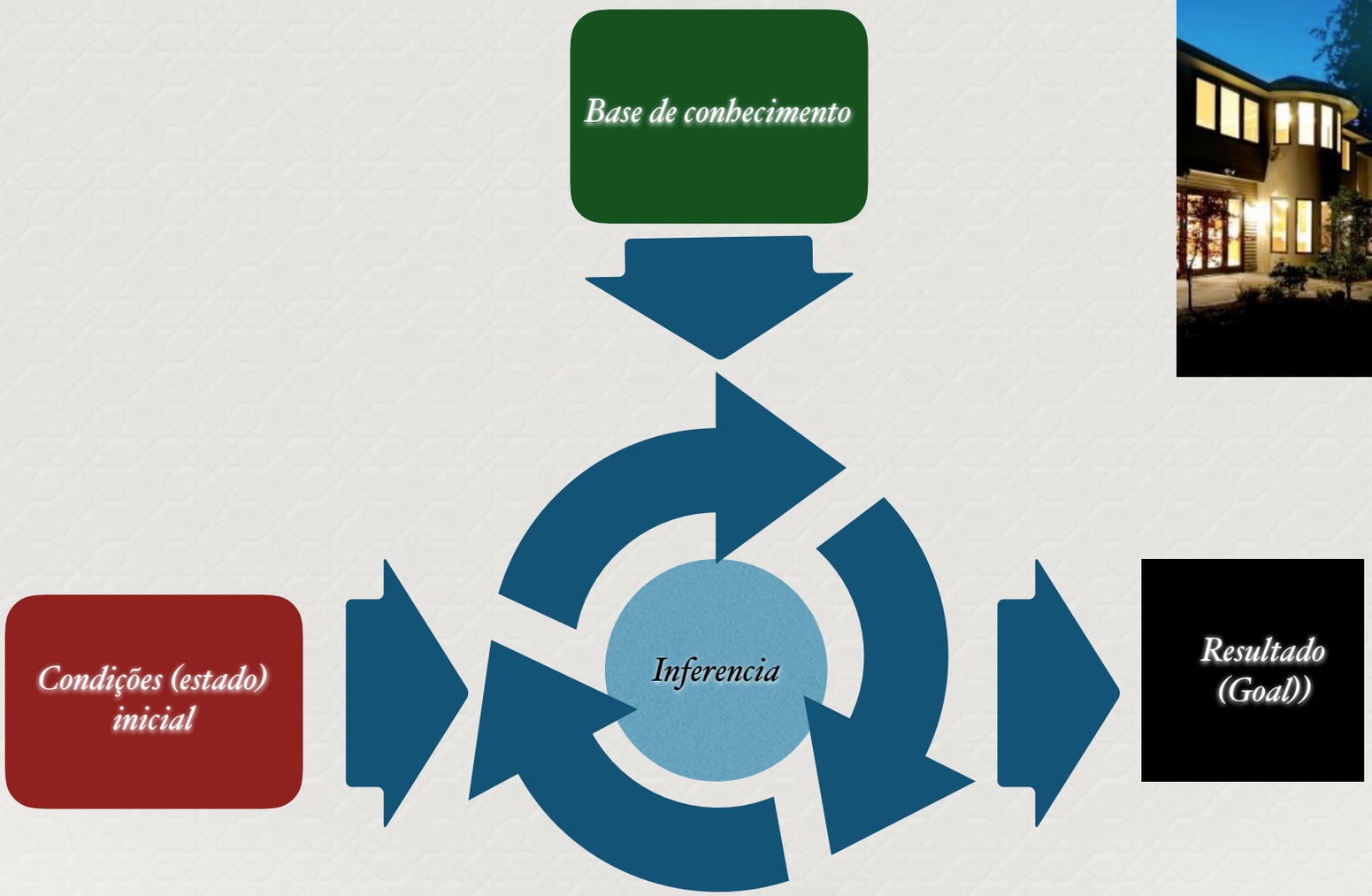
```
mae(maria, paulo).
mae(maria, carla).
mae(susana, jose).
mae(vania, mara).
mae(carla, antonio).

pai(flavio, jose).
pai(flavio, beatriz).

corintiana(maria)

irmao(X, Y) :- mae(Z, X), mae(Z, Y), dif(X, Y).

irmao(X, Y) :- pai(Z, X), pai(Z, Y), dif(X, Y).

tio(X, Y) :- irmao(X, Z), pai(Z, Y).
tio(X, Y) :- irmao(X, Z), mae(Z, Y).
```

swish.swi-prolog.org



Podemos continuar elaborando regras cada vez mais sofisticadas, deixando mais claro a diferença entre os "fatos" mais identificados com os "dados" de um programa convencional, e regras que permitem fazer relacionamentos mais complicados entre fatos e regras. Começa portanto a ficar mais claro o que significa de fato a "inteligência" computável por trás da inteligência artificial.

Em uma primeira abordagem, gostaríamos de ter "agentes inteligentes" capazes de "resolver problemas". O que significa isso?



Look Ahead

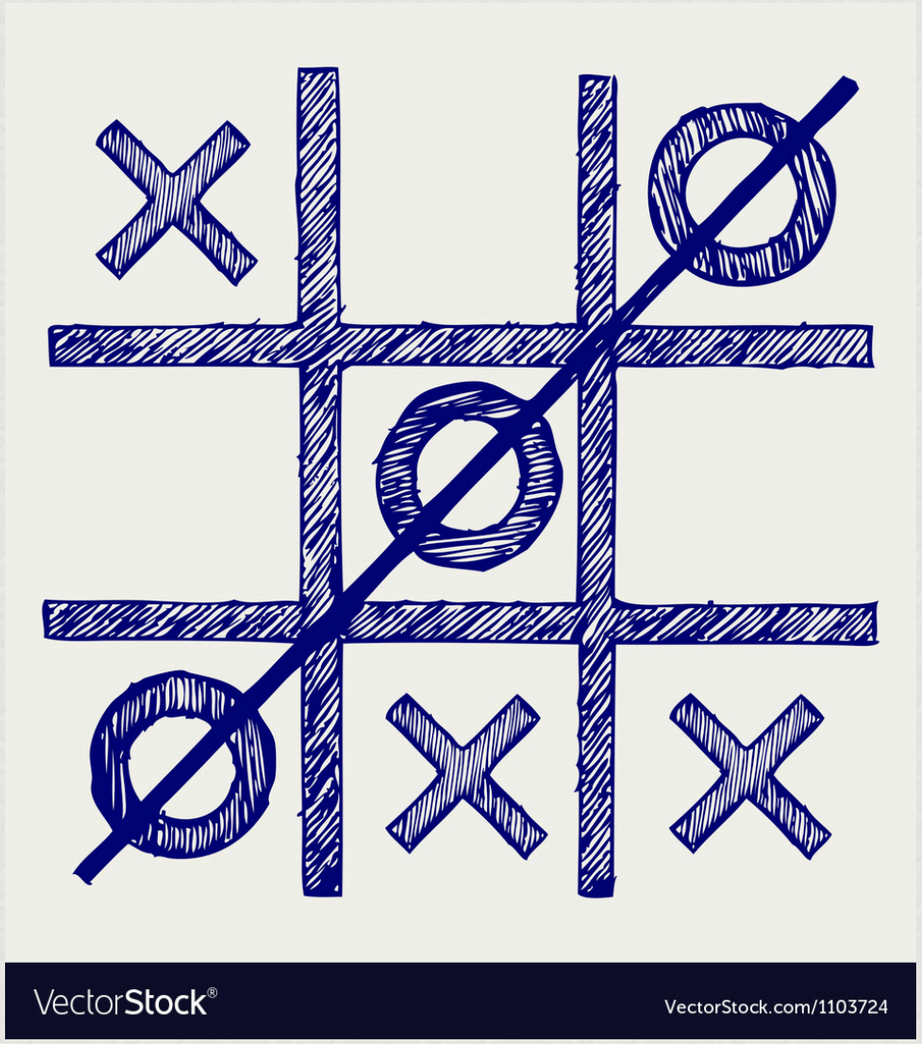
7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

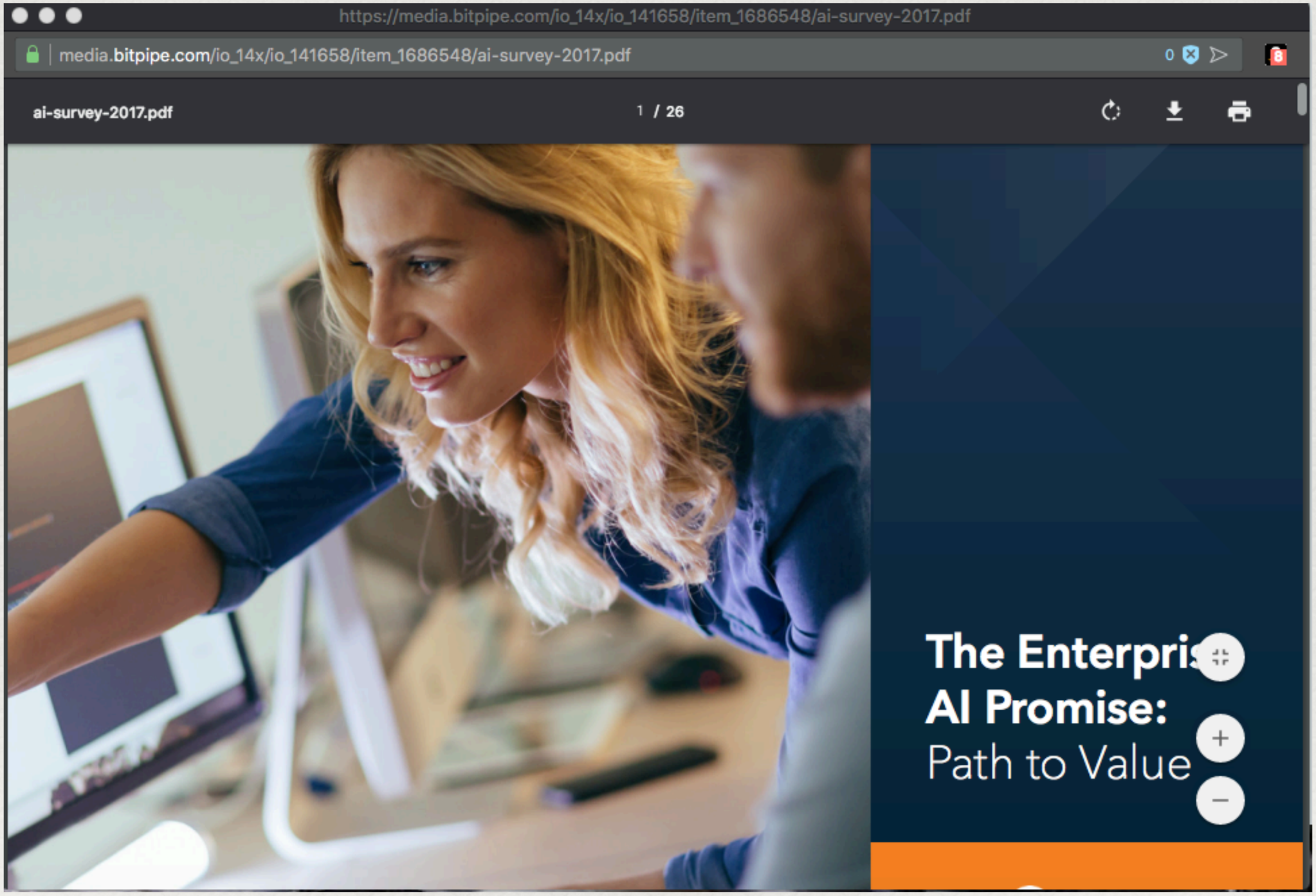
Goal State

Tic-tac-toe (o jogo da velha)



Exercício

Cada um deve, individualmente, exercitar fazer uma "árvore genealógica" dos parentes mais próximos tendo como relação de base (fatos) as relações pai, mãe, e regras para definição de tio, tia, primo, sobrinho, sobrinha, avô, avó, etc. Usar o swish-SWI para implementar o programa. Dúvidas sobre programação em Prolog deve ser retiradas no tutorial ou na bibliografia auxiliar (Cloksin & Mellish).


https://media.bitpipe.com/io_14x/io_141658/item_1686548/ai-survey-2017.pdf
media.bitpipe.com/io_14x/io_141658/item_1686548/ai-survey-2017.pdf
ai-survey-2017.pdf
1 / 26
The Enterpris
AI Promise:
Path to Value

Até a próxima aula!