Escola Politécnica da USP - Depto. de Enga. Mecatrônica

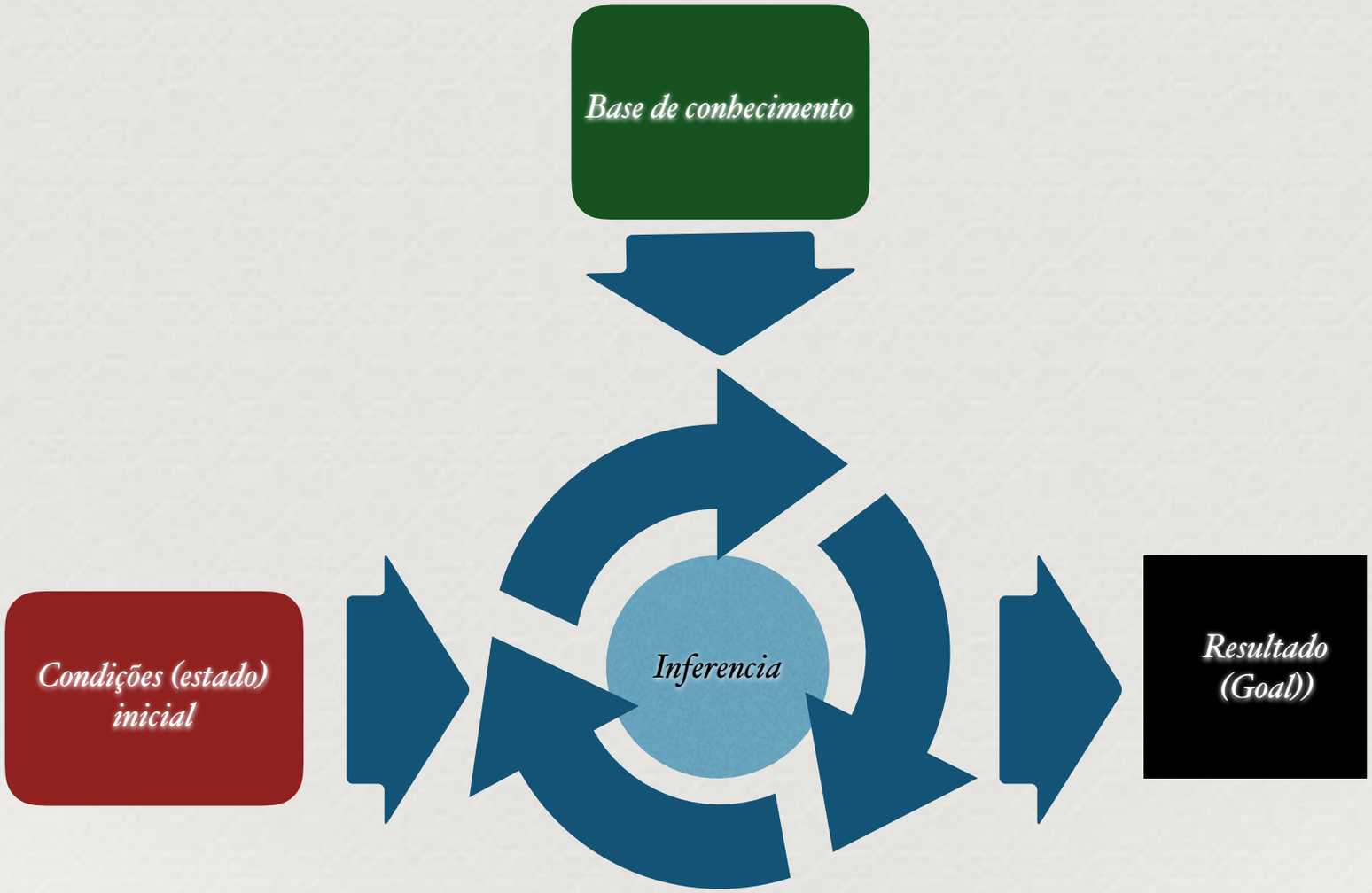
PMR-3510 Inteligência Artificial

Aula 4 - Resolução de problemas por máquinas e por humanos

Prof. José Reinaldo Silva

reinaldo@usp.br

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



7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

Como "resolver problemas automaticamente"

Podemos utilizar duas grande abordagens para resolver problemas automaticamente:

Problema

1. achar uma abordagem geral que leva do estado inicial ao estado final;
2. testar esta abordagem em alguns casos (sem levar em conta o tempo para chegar à solução);
3. checar se a abordagem é completa, isto é, resolve todos os casos ou há casos especiais onde o problema "não converge";
4. preparar a implementação do revolvedor (estrutura de dados e base de conhecimento, regras de dedução);

Solução

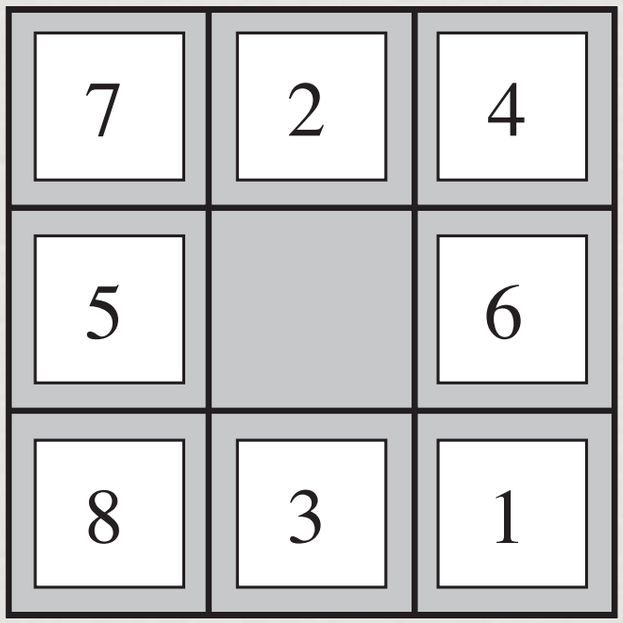
Achar um método geral de resolução de problemas

Estrutura de um "resolvedor automático de problemas"

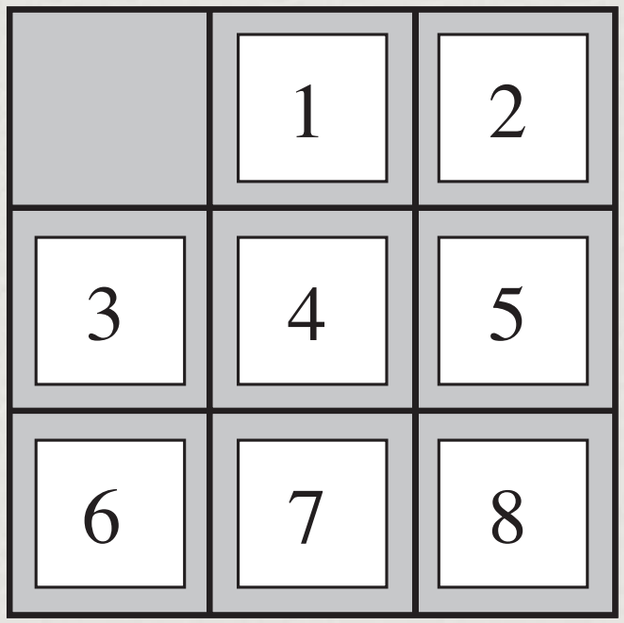
Para dotar uma máquina da capacidade de resolver problemas (ou uma classe de problemas) é preciso ter uma estrutura com os seguintes atributos:

1. uma descrição clara do "estado inicial" ou seja das condições iniciais do problema a ser resolvido;
2. uma descrição clara do objetivo ou "estado final", de modo que seja possível saber quando (e se) o problema foi resolvido;
3. em cada estágio do processo de solução saber quais os próximos estados que podem ser atingidos;
4. poder escolher um (ou o melhor) caminho entre os estados acima;
5. saber que operadores (ou passos) aplicar para fazer a "transição" para um próximo estado;
6. discernir se estamos convergindo para a solução.

Uma solução específica para este problema... ?



Start State



Goal State

Uma possível solução...

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

1. o problema consiste em receber uma configuração genérica de "tiles" (pastilha) - o estado inicial - e colocar os tiles numerados em ordem crescente (o estado final);
2. portanto uma possível solução específica é posicionar o tile maior e assim sucessivamente até ordenar todos;

1. achar uma abordagem geral que leva do estado inicial ao estado final; ✓
2. testar esta abordagem em alguns casos (sem levar em conta o tempo para chegar à solução); ✓
3. checar se a abordagem é completa, isto é, resolve todos os casos ou há casos especiais onde o problema "não converge";
4. preparar a implementação do revolvedor (estrutura de dados e base de conhecimento, regras de dedução);

Discutir uma primeira abordagem...

... antes porém vamos deixar claro a hierarquia de "métodos" para resolução de problemas que vamos abordar nesta e nas próximas aulas:

Paradigma de resolução: estado/transição

método geral de resolução (STRIPS)

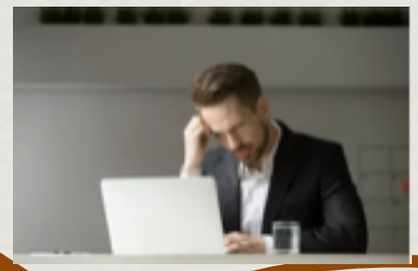
Solução específica para os problemas

abstração

Generalizando problemas e sua resolução...

Segundo Newell e Simon, no seu artigo "Human Problem Solving: The State of the Theory in 1970", se alguém se depara com uma "magica" da retirada de um coelho da cartola, pode reagir de duas maneiras distintas:

Aceitação

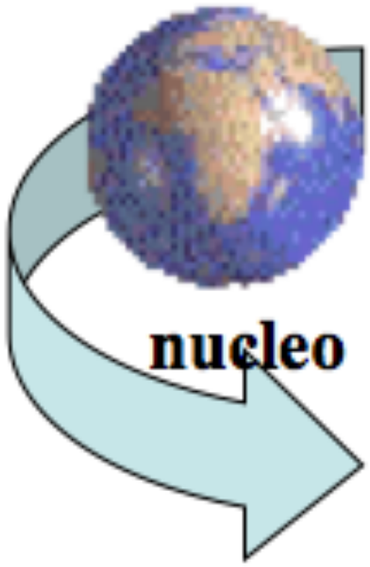


Curiosidade

Como se resolve um problema cientificamente ?

Elementos básicos de uma teoria

Imre Lakatos



heuristicas

Cinturão protetor

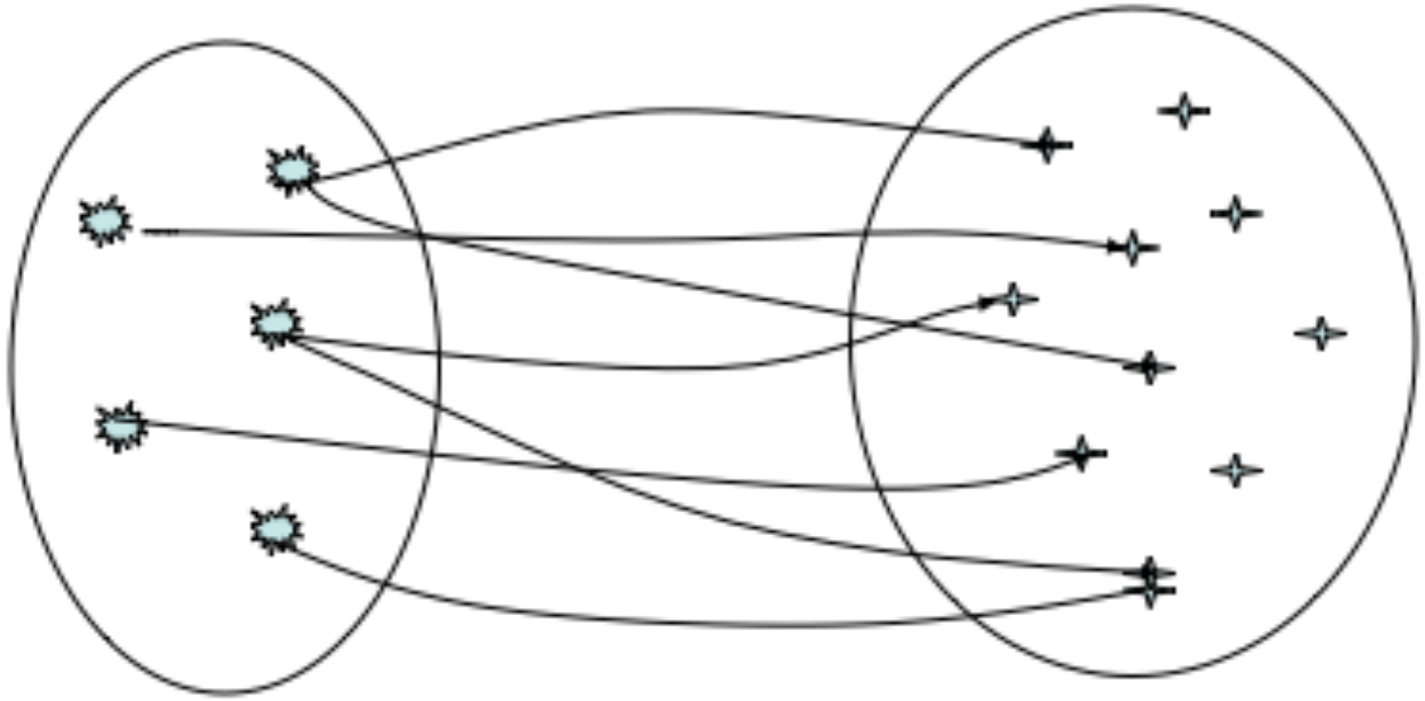
Racionalidade é a resposta

Allen Newel

Herbert Simon

Por volta de 1965 Herbert Simon e Allen Newell desenvolveram em Carnegie Mellon a teoria geral da racionalidade e propuseram que uma parte limitada dela fosse capturada em um sistema computacional capaz de "resolver problemas".

Porque o problema parece tão complicado?



Requisito de omnisciência

Portanto, relacionar problemas e soluções não é trivial

Uma maneira de buscar soluções para o problema, usando a hipótese do "mundo fechado" é trabalhar com uma base de conhecimento baseado em fatos e regras e tentar inferir, dada uma pergunta, qual a "resposta correta", como vimos fazendo nos exemplos práticos anteriores.

Modus Ponens

Modus Tollens

Modus Ponens

A base do Modus Ponens é a expressão condicional A implica B ou A → B. Relações causa-efeito podem então ser representadas, tais como:

i) se o interruptor da sala for desligado as lâmpadas se apagarão.
ii) um aluno desligou o interruptor;
iii) as lâmpadas estão agora apagadas

```
status_key(ligado).
status_key(desligado).
agente_op(aluno).
agente_op(funcionario).
agente_op(professor).
interruptor(X):- status_key(X).
desliga_interruptor(X):- agente_op(X).
lampadas(apagadas):- X=ligado, interruptor(X), desliga_interruptor(aluno).
```

	P	Q	$P \rightarrow Q$
1	V	V	V
2	V	F	F
3	F	V	V
4	F	F	V

A → B também pode ser escrito como *if* A *then* B onde A é chamado antecedente e B o consequente

((A → B) AND B) → A

lampadas(apagadas) :- ligado(interruptor), desliga_interruptor(X).

if :

swish.swi-prolog.org

SWISH File Edit Examples Help

Search

Program

```
1  status_key(ligado).
2  status_key(desligado).
3
4  agente_op(aluno).
5  agente_op(funcionario).
6  agente_op(professor).
7
8  interruptor(X):- status_key(X).
9
10 desliga_interruptor(X):- agente_op(X).
11
12 lampadas(apagadas):- X=ligado, interruptor(X), desliga_interruptor(aluno).
```

lampadas(apagadas).

true

?-

```
lampadas(apagadas).
```

Examples History Solutions table results Run!

Modus Tollens

A base do Modus Tollens também é a expressão condicional A implica B ou A → B.

Relações causa-efeito podem então ser representadas, tais como:

i) se o interruptor da sala for desligado as lâmpadas se apagarão.
ii) ninguém desligou o interruptor;
iii) as lâmpadas estão acesas

```
status_key(ligado).
status_key(desligado).
agente_op(aluno).
agente_op(funcionario).
agente_op(professor).
interruptor(X):- status_key(X).
desliga_interruptor(X):- agente_op(X).
not(lampadas(apagadas)):- X=ligado, interruptor(X), not(desliga_interruptor(aluno)).
```

	P	Q	$P \rightarrow Q$
1	V	V	V
2	V	F	F
3	F	V	V
4	F	F	V

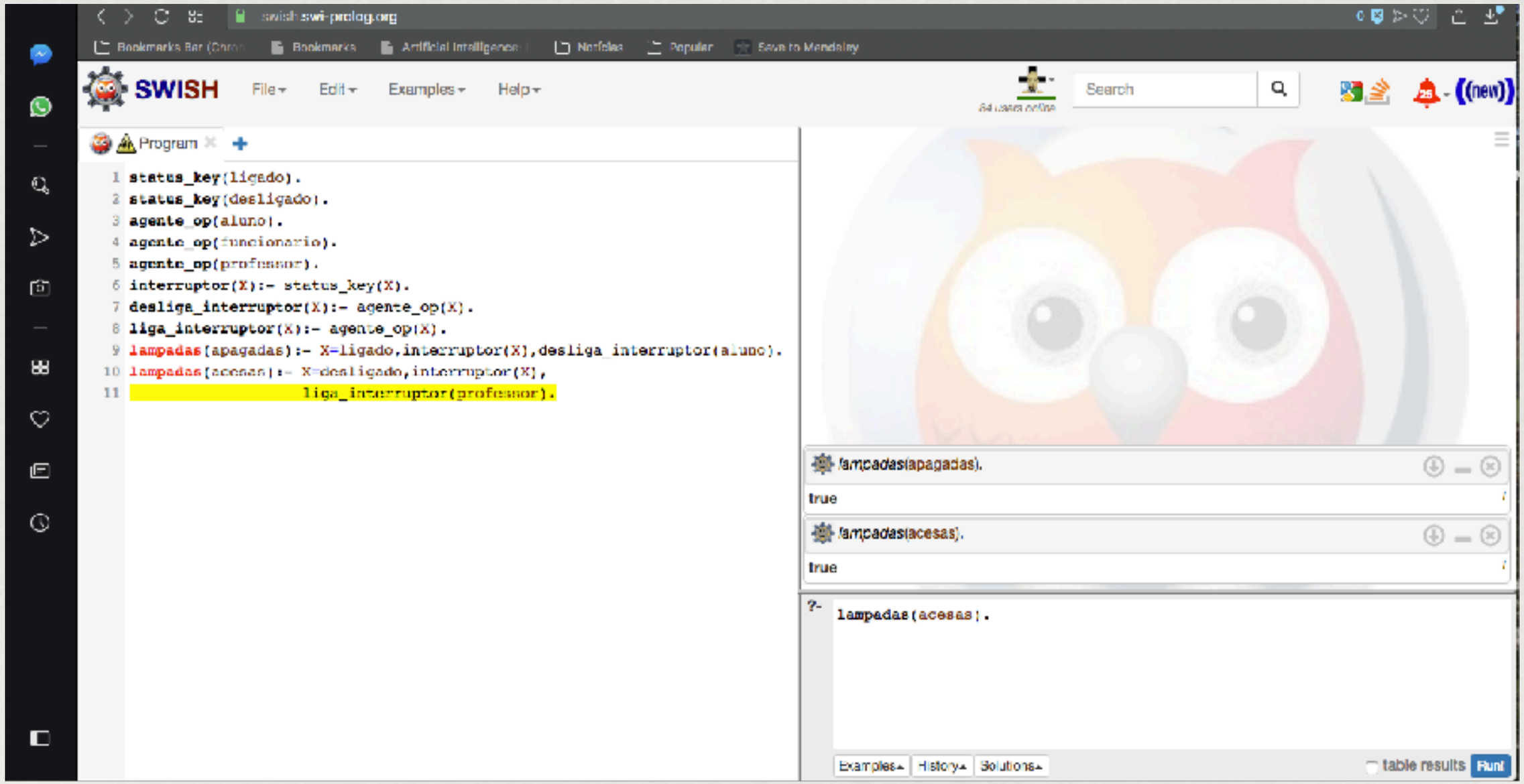
((A → B) AND not(B)) → not(A)

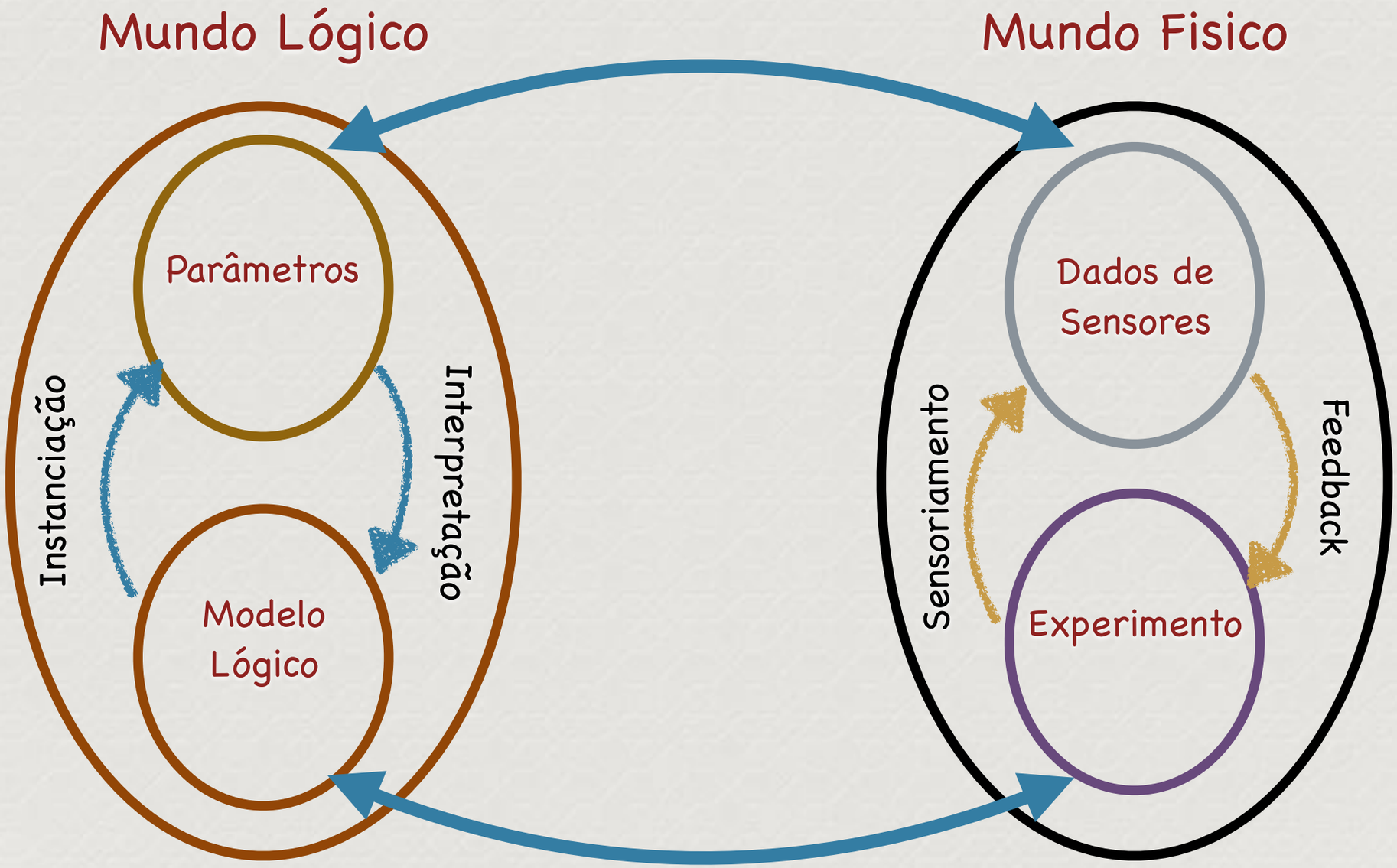
not(lampadas(apagadas)) :- ligado(interruptor),not(desliga_interruptor(X)).

Isso significa que precisaremos de uma representação de conhecimento que vá além de fatos e regras diretas. Precisaremos também definir predicados, funções e operadores, sua "aridade", e conjunto-verdade. Voltaremos a este assunto na aula que vem.

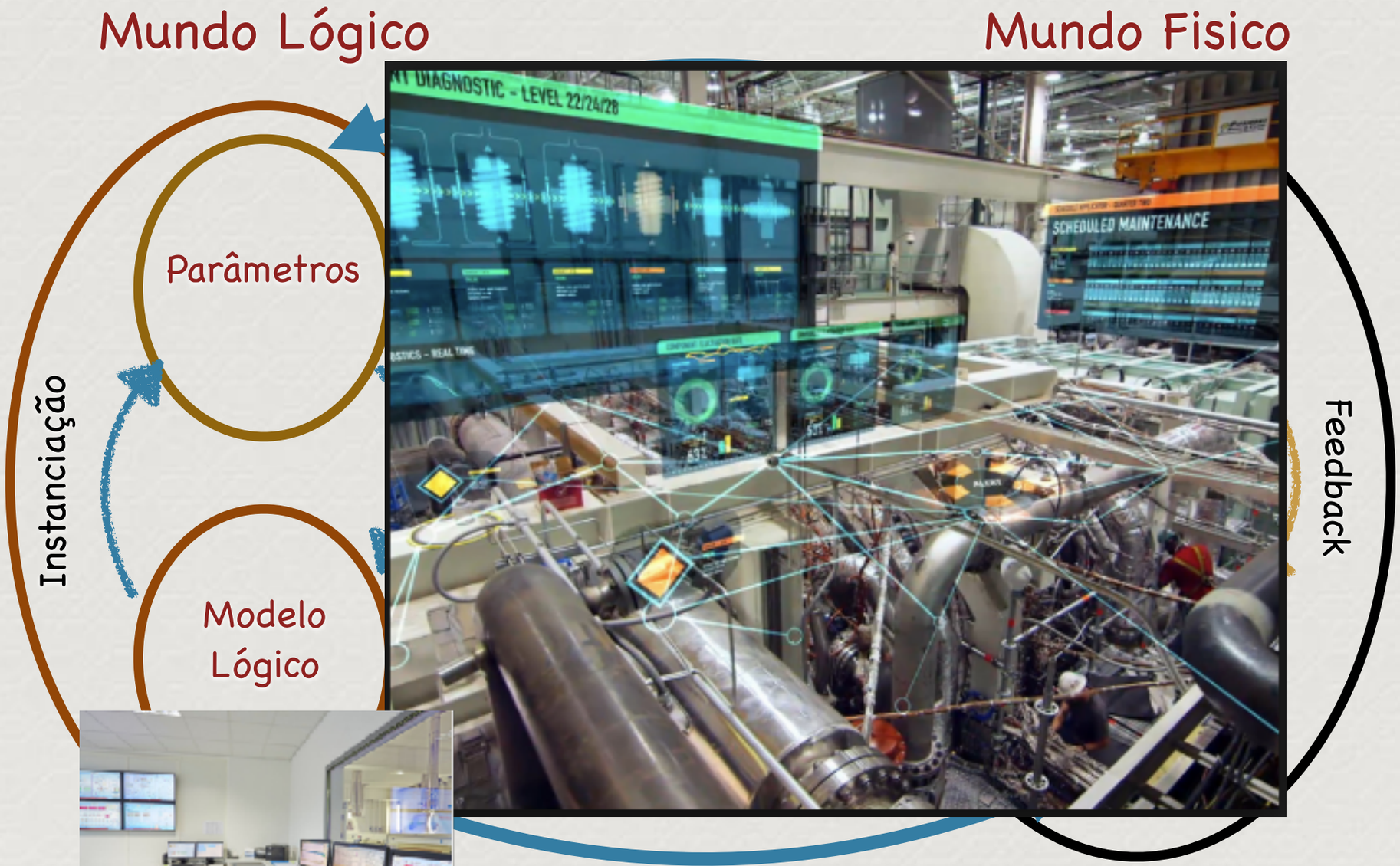
Será que este programa está correto... e completo?

	P	Q	$P \rightarrow Q$
1	V	V	V
2	V	F	F
3	F	V	V
4	F	F	V


swish.swi-prolog.org
SWISH
File
Edit
Examples
Help
Search
Program
1 status_key(ligado).
2 status_key(desligado).
3 agente_op(aluno).
4 agente_op(funcionario).
5 agente_op(professor).
6 interruptor(X):- status_key(X).
7 desliga_interruptor(X):- agente_op(X).
8 liga_interruptor(X):- agente_op(X).
9 lampadas(apagadas):- X=ligado,interruptor(X),desliga_interruptor(aluno).
10 lampadas(acesas):- X=desligado,interruptor(X),
11 liga_interruptor(professor).
lampadas(apagadas).
true
lampadas(acesas).
true
?-
lampadas(acesas).
Examples
History
Solutions
table results
Run!

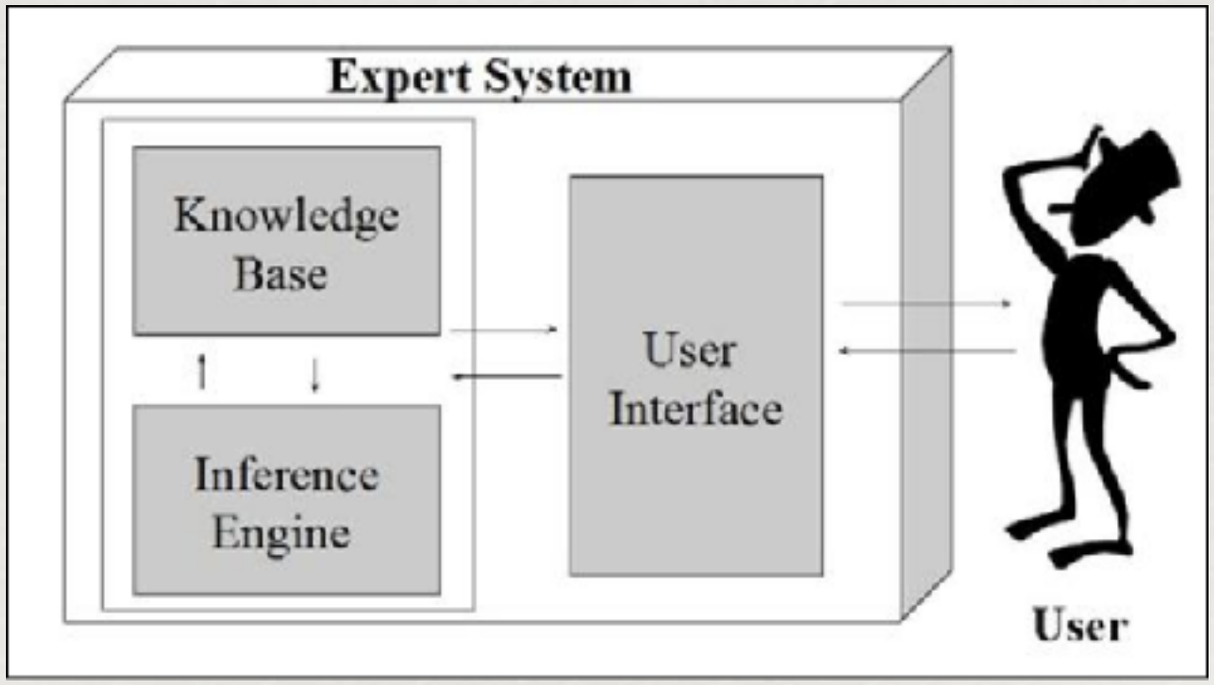


Virtualização em IA

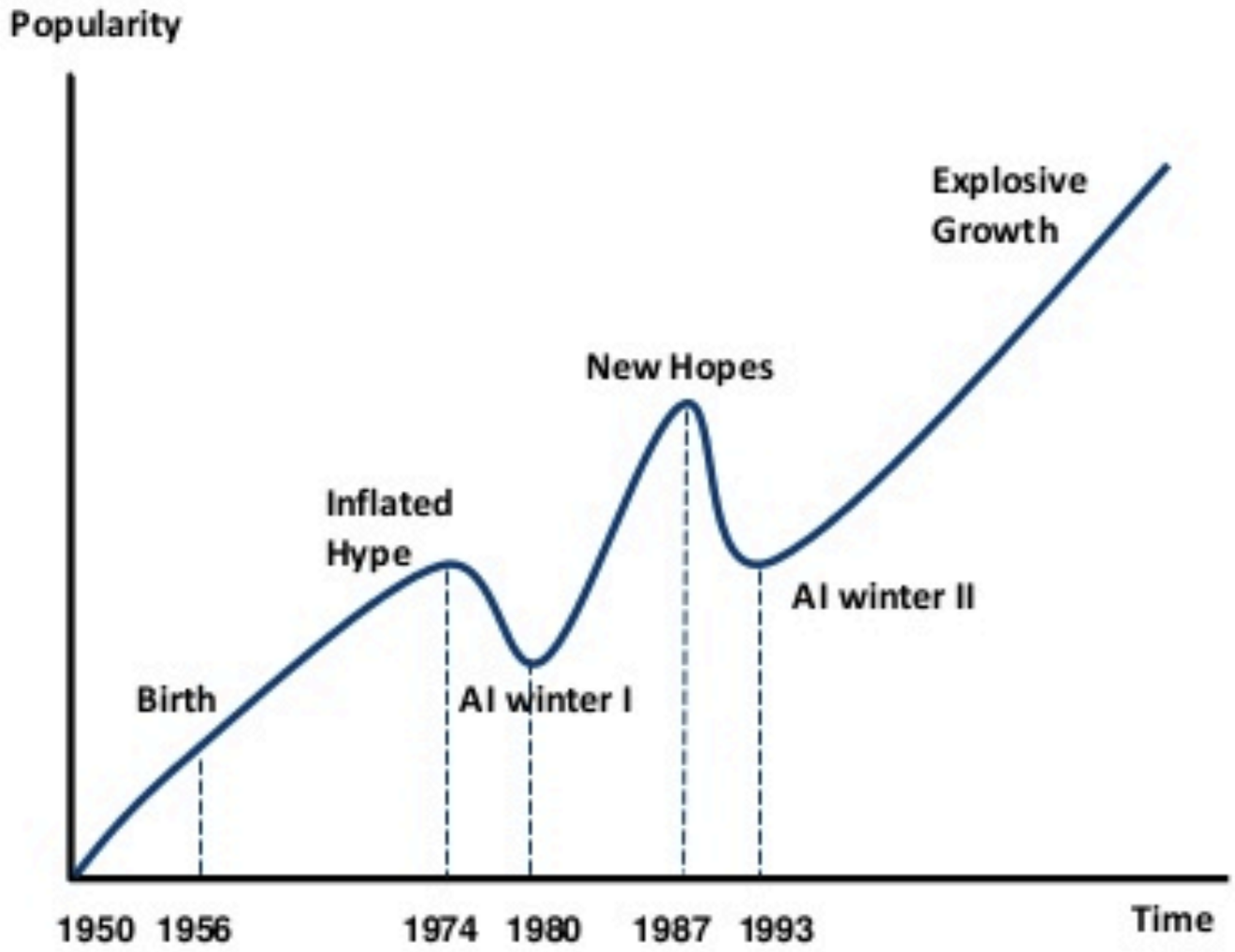


Virtualização em IA

Edward Feigenbaum

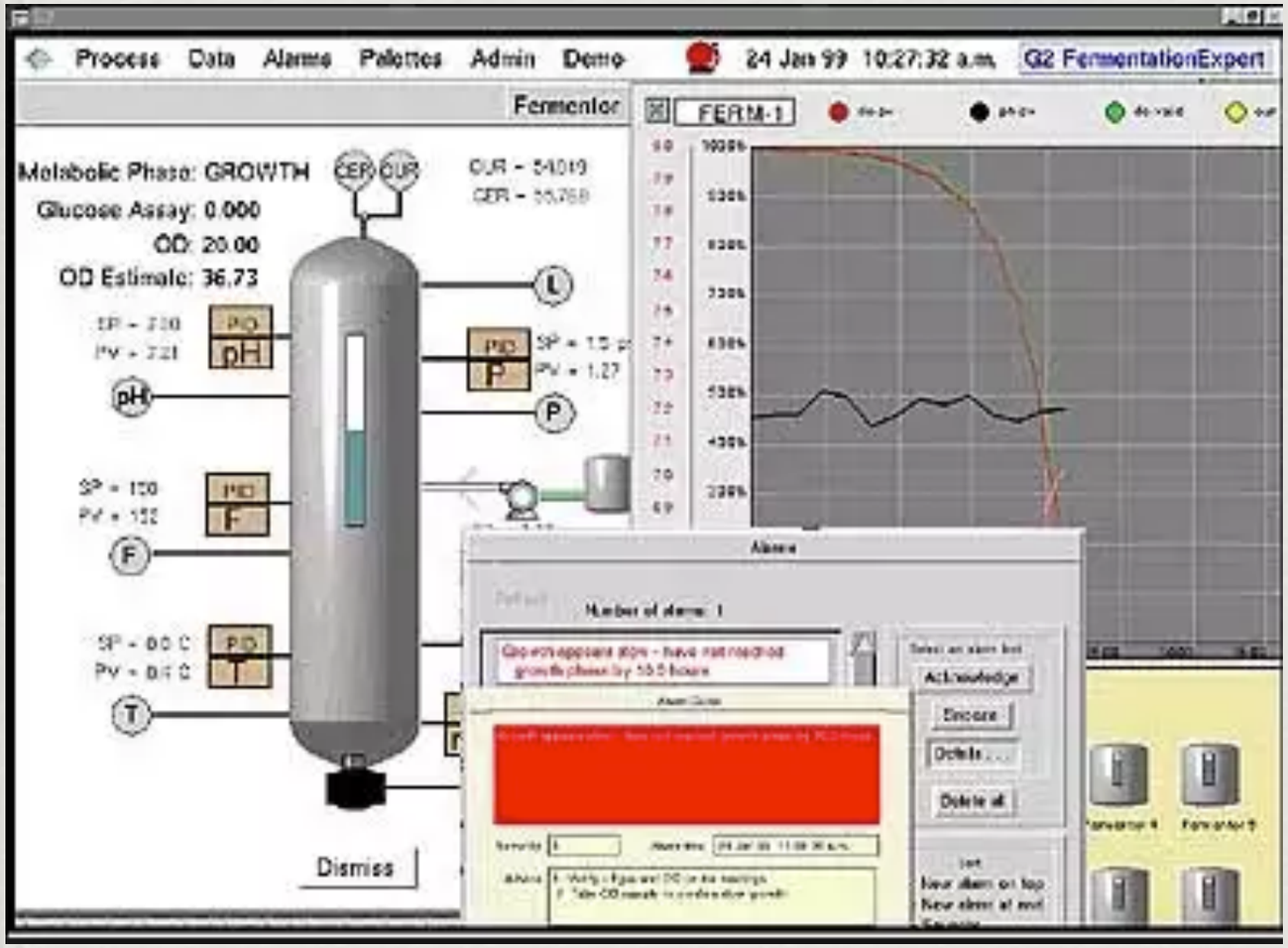


AI HAS A LONG HISTORY OF BEING "THE NEXT BIG THING"…



Timeline of AI Development
▪ **1950s-1960s**: First AI boom - the age of reasoning, prototype AI developed
▪ **1970s**: AI winter I
▪ **1980s-1990s**: Second AI boom: the age of Knowledge representation (appearance of expert systems capable of reproducing human decision-making)
▪ **1990s**: AI winter II
▪ **1997**: Deep Blue beats Gary Kasparov
▪ **2006**: University of Toronto develops Deep Learning
▪ **2011**: IBM's Watson won Jeopardy
▪ **2016**: Go software based on Deep Learning beats world's champions

Source: Literature review, Gao Feng analysis


Process
Data
Alarms
Palettes
Admin
Demo
24 Jan 99 10:27:32 a.m.
G2 FermentationExpert
Fermentor
FERM-1
Metabolic Phase: GROWTH
Glucose Assay: 0.000
OD: 20.00
OD Estimate: 36.73
Alarms
Acknowledge
Dismiss

gensym

Até agora ninguém tomou iniciativa de organizar os grupos de trabalho. Não teremos tempo de fazer os exercícios se não começarmos logo. Portanto a partir de agora o exercício é unificado (o jogo de tiles) e daqui a um mês cada equipe deverá rodar o seu algoritmo e competir no tempo de resolução. Como deve ter dificuldade na organização dos grupos farei isso via o sistema e-disciplinas.

Até a próxima aula!