NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

MDE

Labwork2 – Class 2

Introduction to Prolog

Installation and First Exercises

2024 - 2025

Objectives and Scheduling

- **Class 1:** Introduction to Prolog: Installation and Exercises
- **Class 2:** Presentation of Lab work Assignment, Problem Modelling, Dynamic Change
- **Class 3:** Lab work implementation - Graphs in Prolog
- **Class 4:** Lab work implementation - MQTT inside Prolog, Prolog HTTP Server and Integration with Java App
- **Class 5:** Lab work Finalization

Delivery date: 2025/05/26

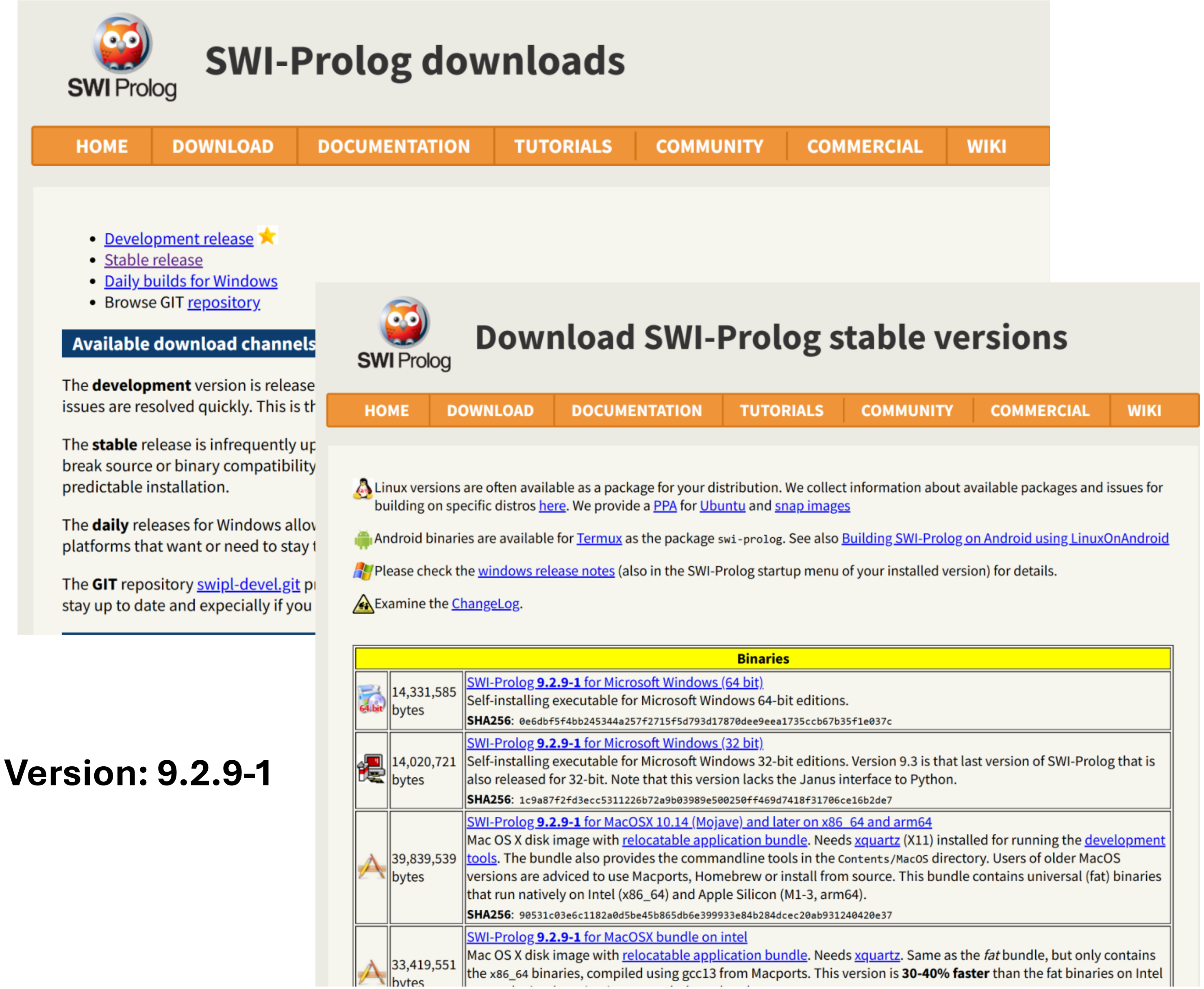
Lab2 (Class 1)

- Swi-Prolog installation

- Examples with:
  - Representation of facts and rules
  - Queries
  - Recursion

Swi-Prolog installation

https://www.swi-prolog.org/Download.html →



SWI-Prolog downloads

- Development release ★
- Stable release
- Daily builds for Windows
- Browse GIT repository

Available download channels

The **development** version is release
issues are resolved quickly. This is th

The **stable** release is infrequently up
break source or binary compatibility
predictable installation.

The **daily** releases for Windows allov
platforms that want or need to stay t

The **GIT** repository swipl-devel.git pı
stay up to date and expecially if you

Download SWI-Prolog stable versions

Linux versions are often available as a package for your distribution. We collect information about available packages and issues for building on specific distros here. We provide a PPA for Ubuntu and snap images

Android binaries are available for Termux as the package `swi-prolog`. See also Building SWI-Prolog on Android using LinuxOnAndroid

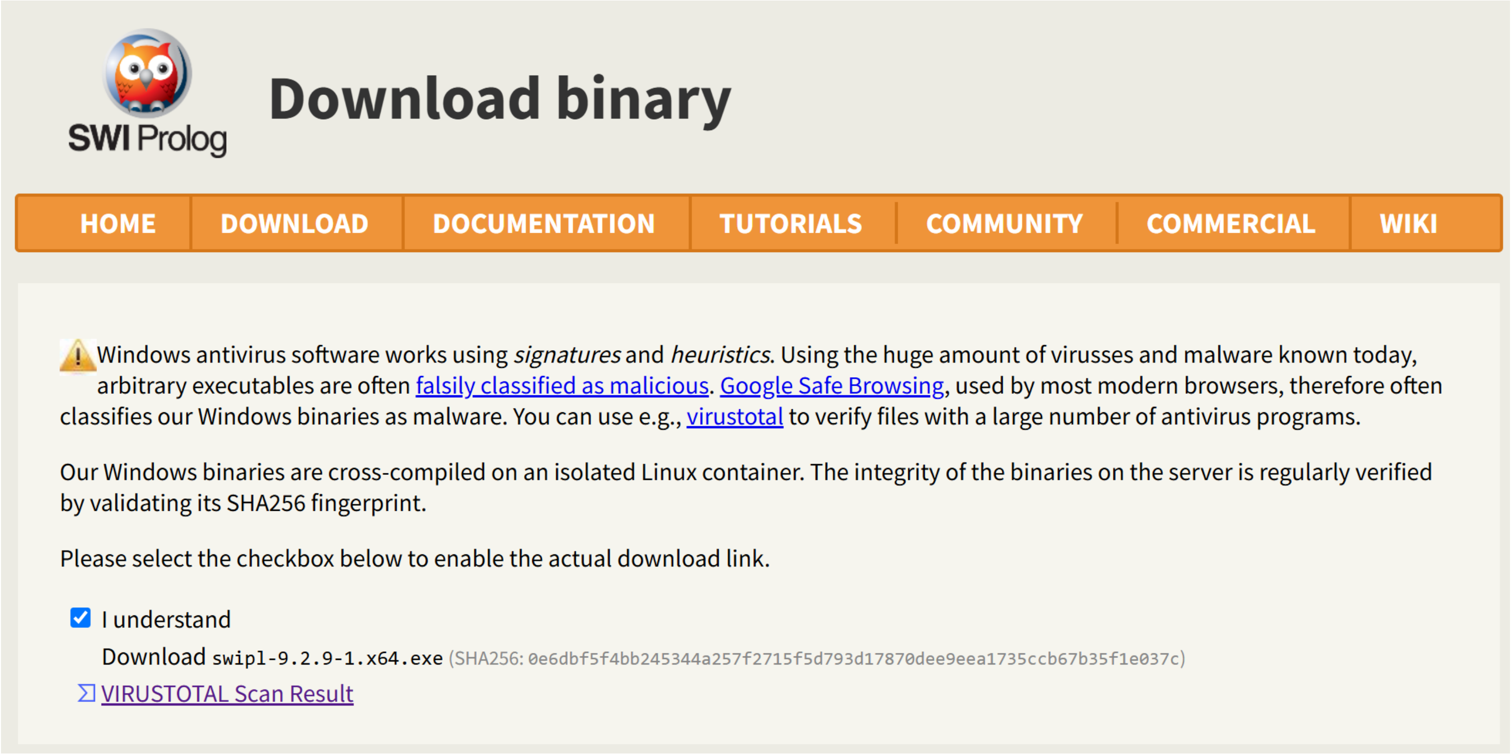
Please check the windows release notes (also in the SWI-Prolog startup menu of your installed version) for details.

Examine the ChangeLog.

		Binaries
	14,331,585 bytes	SWI-Prolog **9.2.9-1** for Microsoft Windows (64 bit) Self-installing executable for Microsoft Windows 64-bit editions. **SHA256**: `0e6dbf5f4bb245344a257f2715f5d793d17870dee9eea1735ccb67b35f1e037c`
	14,020,721 bytes	SWI-Prolog **9.2.9-1** for Microsoft Windows (32 bit) Self-installing executable for Microsoft Windows 32-bit editions. Version 9.3 is that last version of SWI-Prolog that is also released for 32-bit. Note that this version lacks the Janus interface to Python. **SHA256**: `1c9a87f2fd3ecc5311226b72a9b03989e500250ff469d7418f31706ce16b2de7`
	39,839,539 bytes	SWI-Prolog **9.2.9-1** for MacOSX 10.14 (Mojave) and later on x86_64 and arm64 Mac OS X disk image with relocatable application bundle. Needs xquartz (X11) installed for running the development tools. The bundle also provides the commandline tools in the `Contents/MacOS` directory. Users of older MacOS versions are adviced to use Macports, Homebrew or install from source. This bundle contains universal (fat) binaries that run natively on Intel (x86_64) and Apple Silicon (M1-3, arm64). **SHA256**: `90531c03e6c1182a0d5be45b865db6e399933e84b284dcec20ab931240420e37`
	33,419,551 bytes	SWI-Prolog **9.2.9-1** for MacOSX bundle on intel Mac OS X disk image with relocatable application bundle. Needs xquartz. Same as the *fat* bundle, but only contains the `x86_64` binaries, compiled using gcc13 from Macports. This version is **30-40% faster** than the fat binaries on Intel

New Stable Version: 9.2.9-1

Swi-prolog installation – ALERT!!



Download binary

⚠ Windows antivirus software works using *signatures* and *heuristics*. Using the huge amount of virusses and malware known today, arbitrary executables are often falsily classified as malicious. Google Safe Browsing, used by most modern browsers, therefore often classifies our Windows binaries as malware. You can use e.g., virustotal to verify files with a large number of antivirus programs.

Our Windows binaries are cross-compiled on an isolated Linux container. The integrity of the binaries on the server is regularly verified by validating its SHA256 fingerprint.

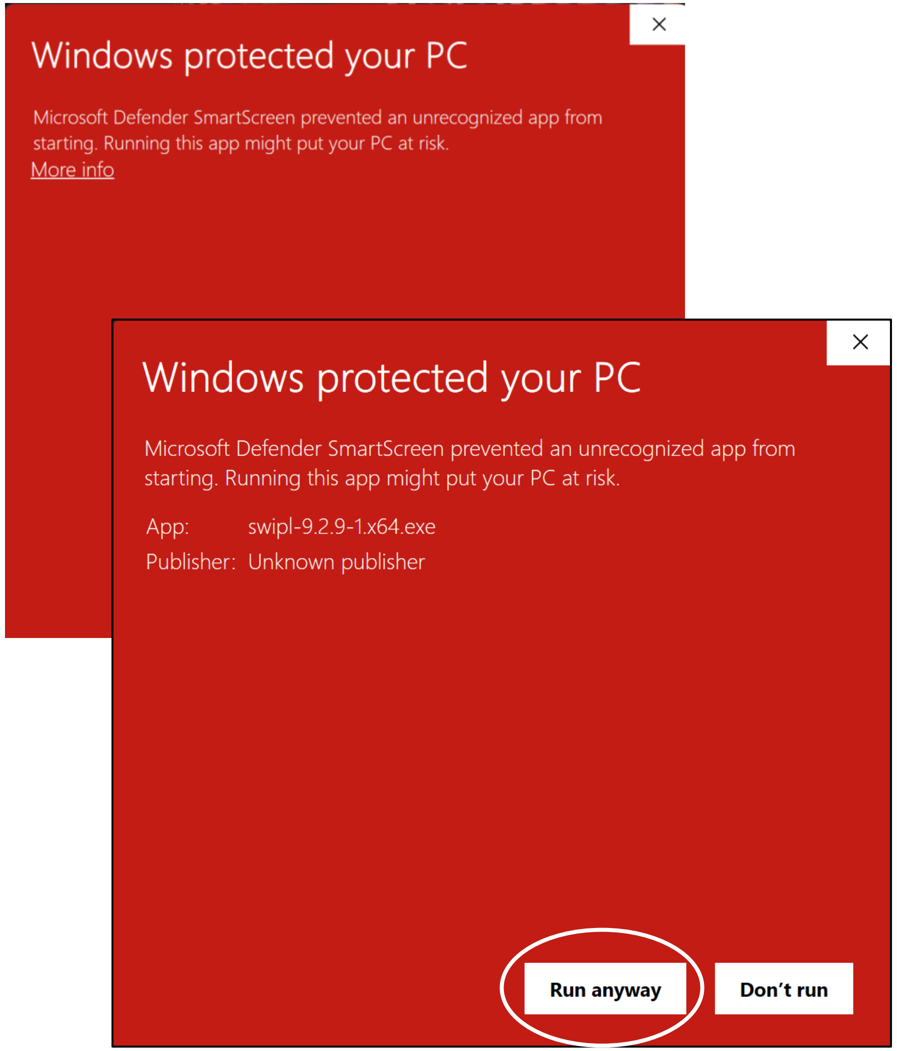
Please select the checkbox below to enable the actual download link.

☑ I understand

Download `swipl-9.2.9-1.x64.exe` (SHA256: `0e6dbf5f4bb245344a257f2715f5d793d17870dee9eea1735ccb67b35f1e037c`)

VIRUSTOTAL Scan Result

Since the downloader package is an executable file, it is recommended to check it with **VirusTotal before downloading**. This tool scans the file using around 70 antivirus engines to detect any malware or suspicious content.



Windows protected your PC

Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.

More info

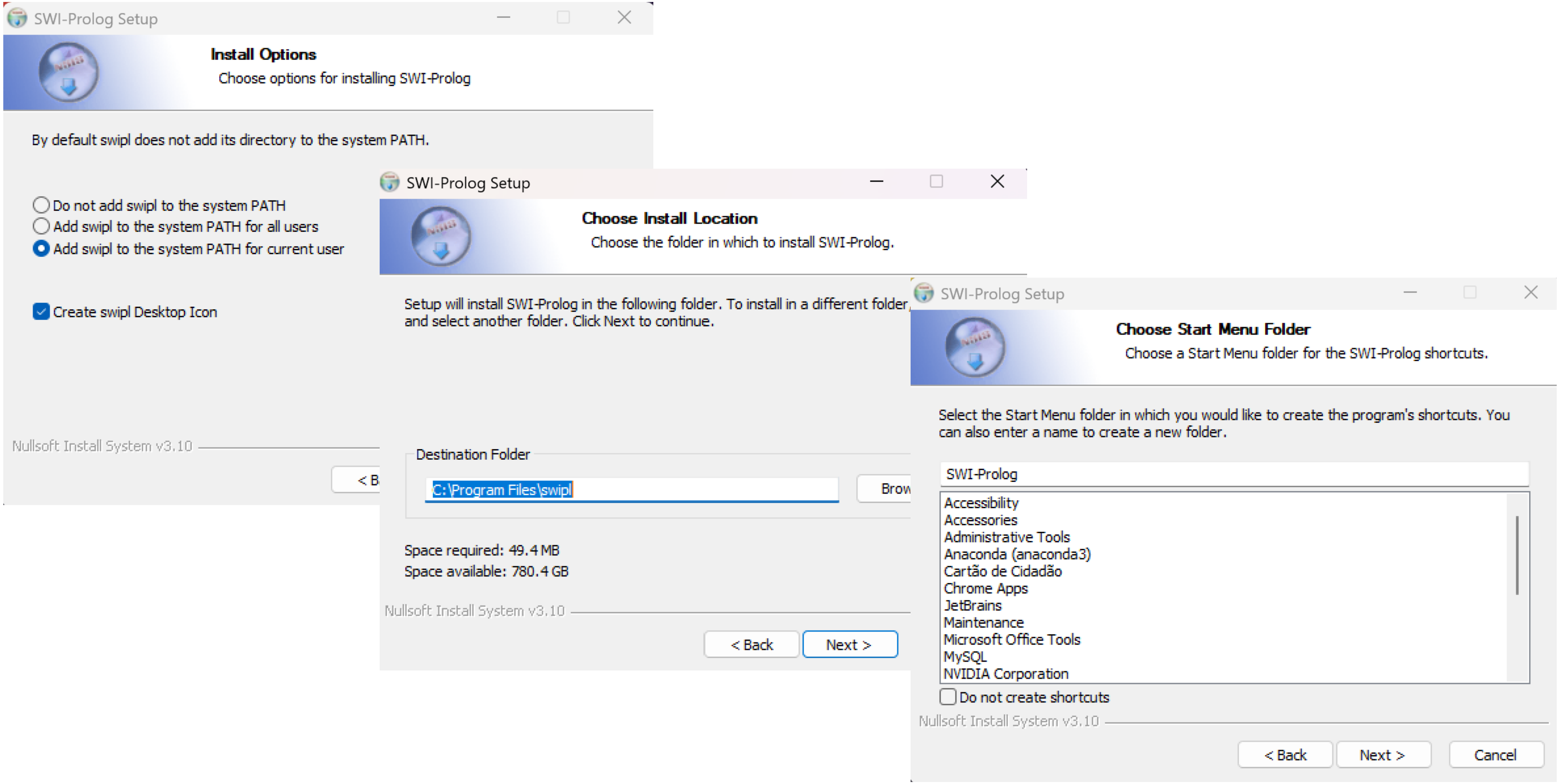
Windows protected your PC

Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.

App: swipl-9.2.9-1.x64.exe

Publisher: Unknown publisher

Swi-prolog installation – Options…



SWI-Prolog Setup

Install Options

Choose options for installing SWI-Prolog

By default swipl does not add its directory to the system PATH.

○ Do not add swipl to the system PATH
○ Add swipl to the system PATH for all users
● Add swipl to the system PATH for current user

☑ Create swipl Desktop Icon

Nullsoft Install System v3.10

SWI-Prolog Setup

Choose Install Location

Choose the folder in which to install SWI-Prolog.

Setup will install SWI-Prolog in the following folder. To install in a different folder, and select another folder. Click Next to continue.

Destination Folder

C:\Program Files\swipl

Space required: 49.4 MB
Space available: 780.4 GB

Nullsoft Install System v3.10

SWI-Prolog Setup

Choose Start Menu Folder

Choose a Start Menu folder for the SWI-Prolog shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

SWI-Prolog

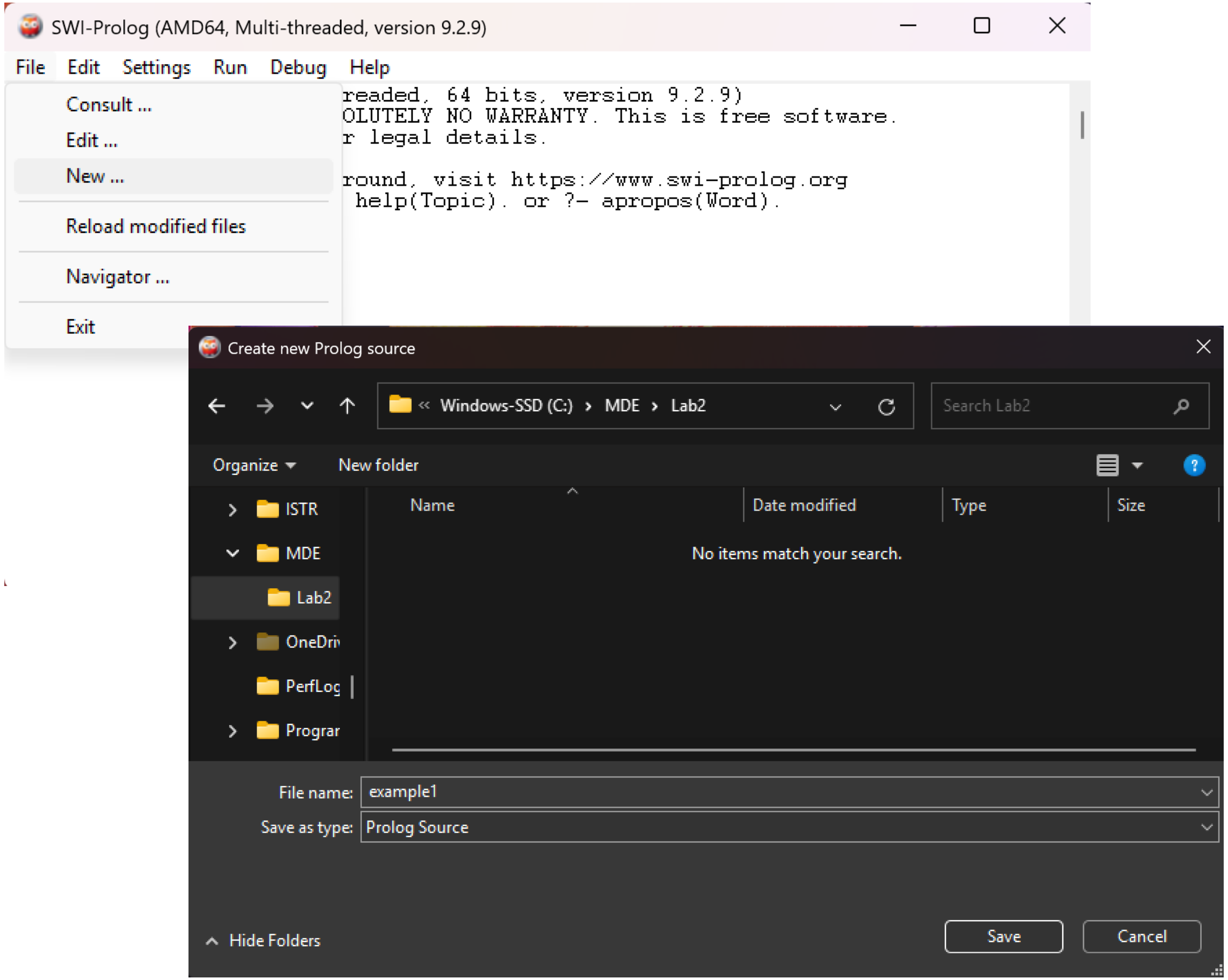
Accessibility
Accessories
Administrative Tools
Anaconda (anaconda3)
Cartão de Cidadão
Chrome Apps
JetBrains
Maintenance
Microsoft Office Tools
MySQL
NVIDIA Corporation

□ Do not create shortcuts

Nullsoft Install System v3.10

Using Swi-Prolog

- Execute “swipl-win.exe”
- Create new .pl file
  - File->new



```
readed, 64 bits, version 9.2.9)
OLUTELY NO WARRANTY. This is free software.
r legal details.

round, visit https://www.swi-prolog.org
 help(Topic). or ?- apropos(Word).
```

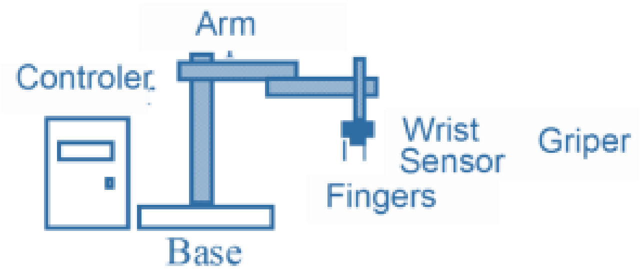
No items match your search.

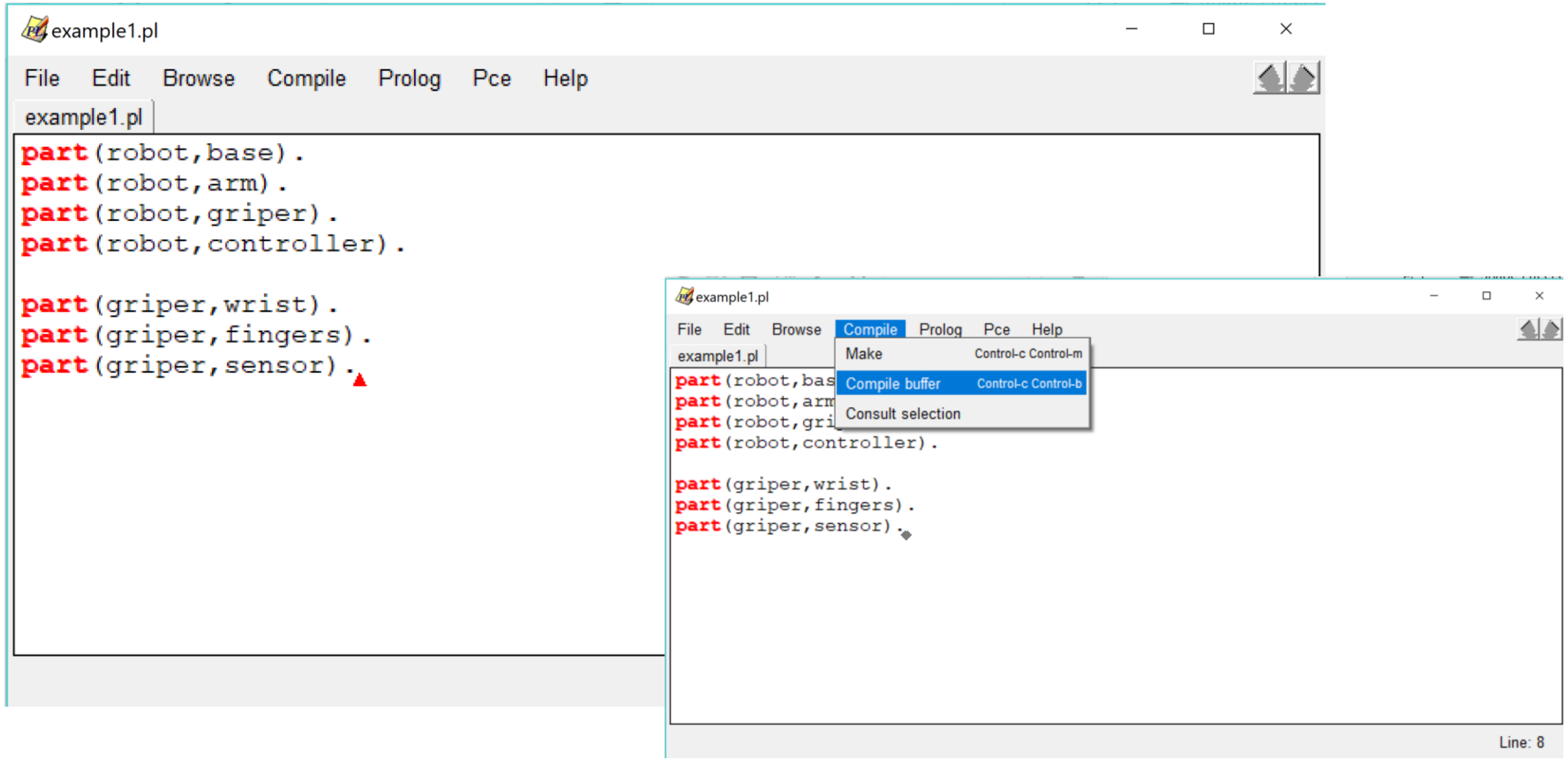
File name: example1

Save as type: Prolog Source

Representation of Facts

Example 1: Model the structure of a robot [Done in THORETICAL class]




example1.pl
File Edit Browse Compile Prolog Pce Help
example1.pl

```
part(robot,base).
part(robot,arm).
part(robot,griper).
part(robot,controller).

part(griper,wrist).
part(griper,fingers).
part(griper,sensor).
```

example1.pl
File Edit Browse Compile Prolog Pce Help
example1.pl
Make Control-c Control-m
Compile buffer Control-c Control-b
Consult selection
Line: 8

```
part(robot,bas
part(robot,arm
part(robot,gri
part(robot,controller).

part(griper,wrist).
part(griper,fingers).
part(griper,sensor).
```

Representation of Rules

FROM THEORETICAL CLASSES...

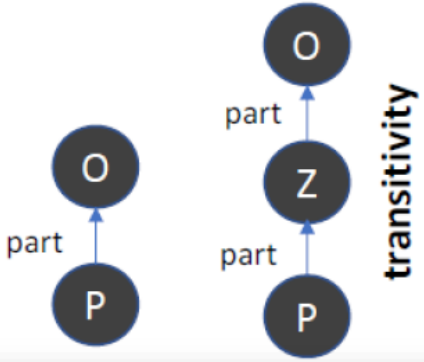
Rules:

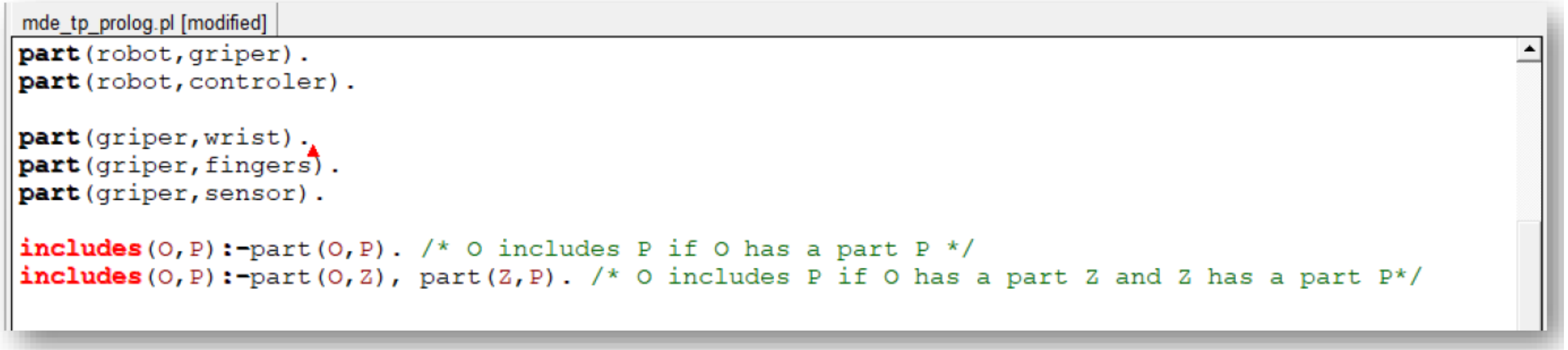
Conclusion if Condition: conclusion :- condition.

if → **:-** *and* → **,** *or* → **;** *not* → **not**(…)

includes(O,P) :- part(O,P). /* O includes P if O has a part P */

includes(O, P) :- part(O,Z), part(Z,P). /* O includes P if O has a part Z and Z has a part P*/

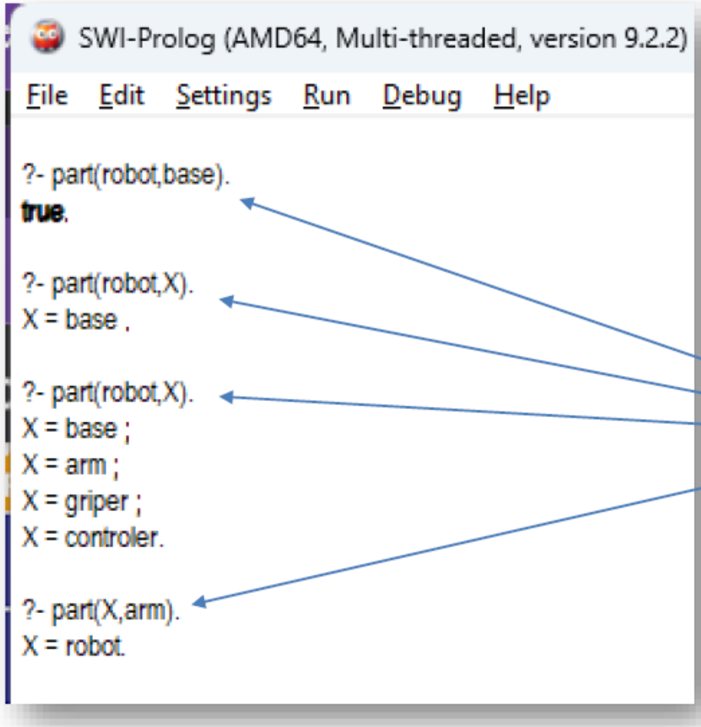




```
part(robot,griper).
part(robot,controler).

part(griper,wrist).
part(griper,fingers).
part(griper,sensor).

includes(O,P):-part(O,P). /* O includes P if O has a part P */
includes(O,P):-part(O,Z), part(Z,P). /* O includes P if O has a part Z and Z has a part P*/
```

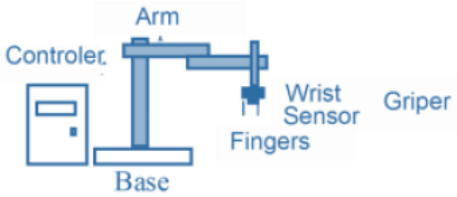
Queries

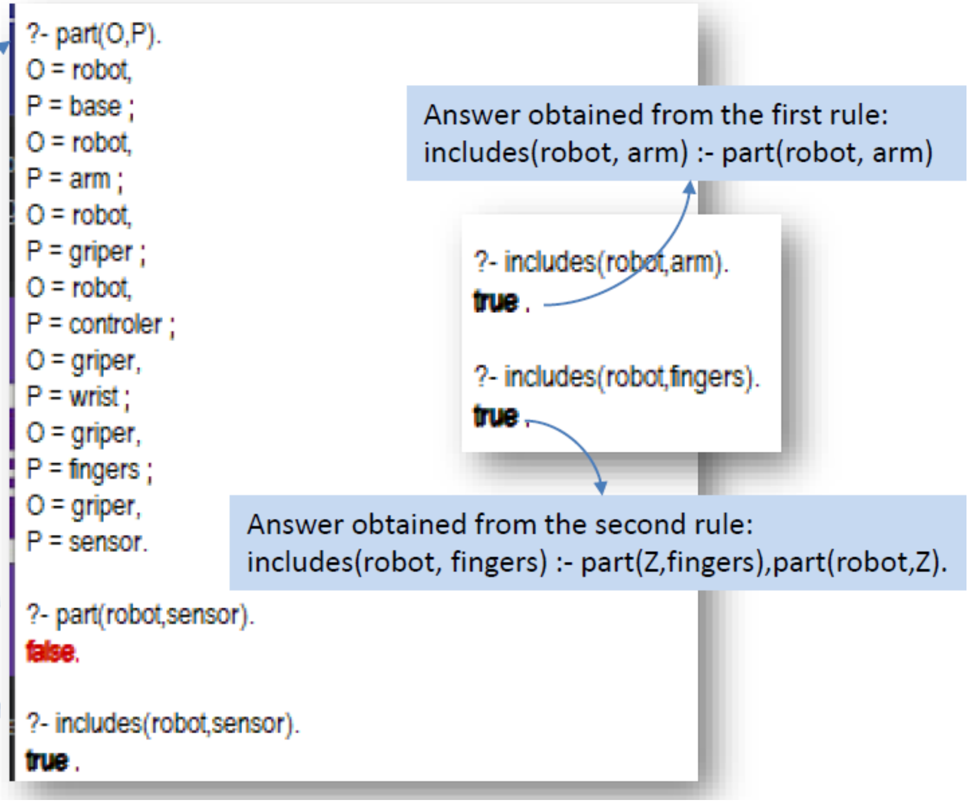
FROM THEORETICAL CLASSES…


SWI-Prolog (AMD64, Multi-threaded, version 9.2.2)
File Edit Settings Run Debug Help

```
?- part(robot,base).
true.

?- part(robot,X).
X = base .

?- part(robot,X).
X = base ;
X = arm ;
X = griper ;
X = controler.

?- part(X,arm).
X = robot.
```

Examples "?-"





```
?- part(O,P).
O = robot,
P = base ;
O = robot,
P = arm ;
O = robot,
P = griper ;
O = robot,
P = controler ;
O = griper,
P = wrist ;
O = griper,
P = fingers ;
O = griper,
P = sensor.

?- part(robot,sensor).
false.

?- includes(robot,sensor).
true .
```

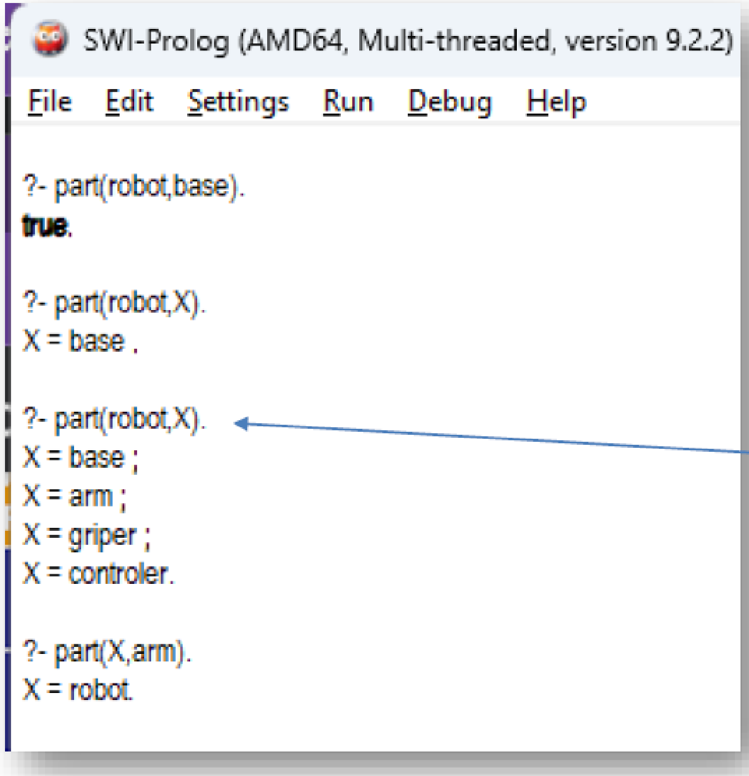
Answer obtained from the first rule: includes(robot, arm) :- part(robot, arm)

```
?- includes(robot,arm).
true .

?- includes(robot,fingers).
true .
```

Answer obtained from the second rule: includes(robot, fingers) :- part(Z,fingers),part(robot,Z).

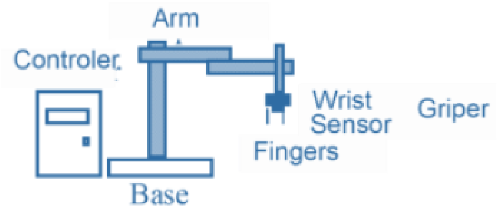
Queries

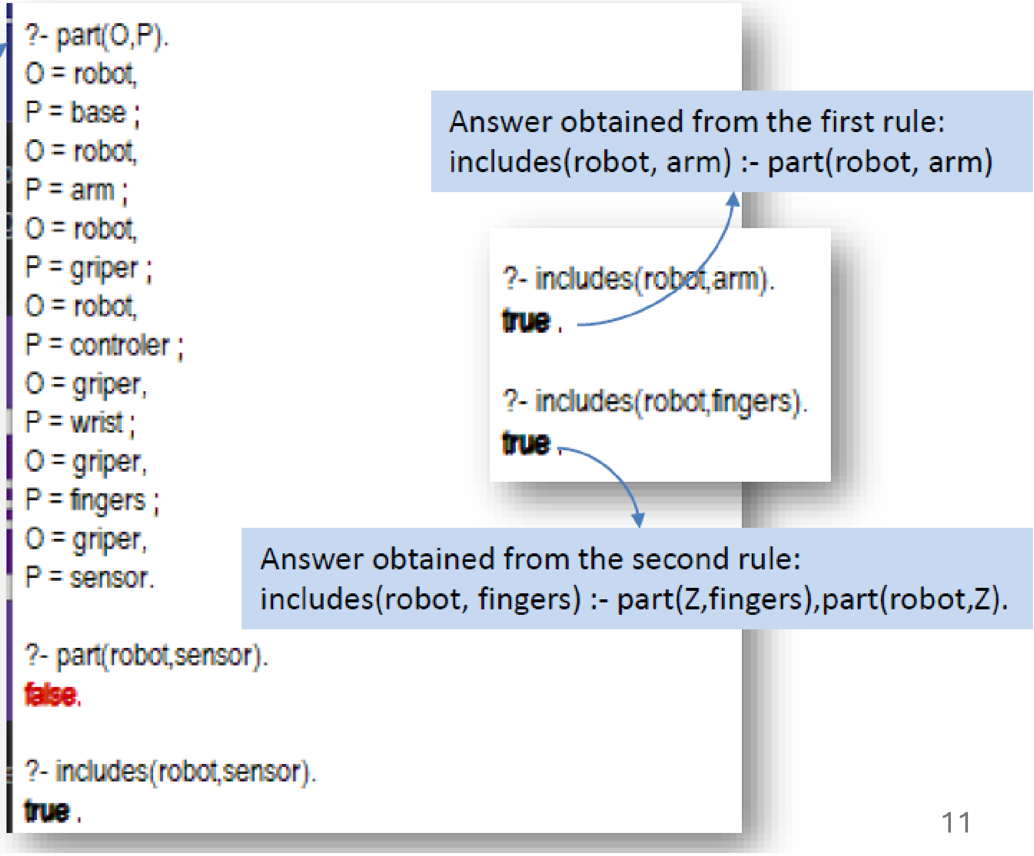
FROM THEORETICAL CLASSES...


SWI-Prolog (AMD64, Multi-threaded, version 9.2.2)
File Edit Settings Run Debug Help

```
?- part(robot,base).
true.

?- part(robot,X).
X = base .

?- part(robot,X).
X = base ;
X = arm ;
X = griper ;
X = controler.

?- part(X,arm).
X = robot.
```

Multiple results





```
?- part(O,P).
O = robot,
P = base ;
O = robot,
P = arm ;
O = robot,
P = griper ;
O = robot,
P = controler ;
O = griper,
P = wrist ;
O = griper,
P = fingers ;
O = griper,
P = sensor.

?- part(robot,sensor).
false.

?- includes(robot,sensor).
true .
```

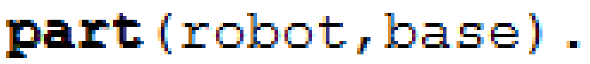
Answer obtained from the first rule:
includes(robot, arm) :- part(robot, arm)

```
?- includes(robot,arm).
true .

?- includes(robot,fingers).
true .
```

Answer obtained from the second rule:
includes(robot, fingers) :- part(Z,fingers),part(robot,Z).

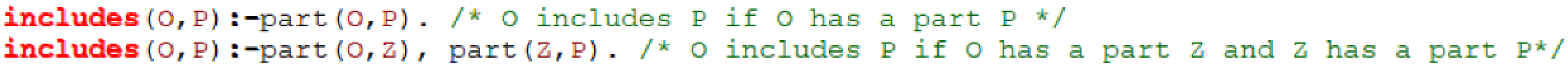
Some Notes

- Facts



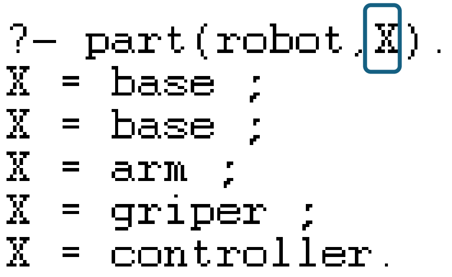
```
part(robot,base).
```

- Rules



```
includes(O,P):-part(O,P). /* O includes P if O has a part P */
includes(O,P):-part(O,Z), part(Z,P). /* O includes P if O has a part Z and Z has a part P*/
```

- Variables



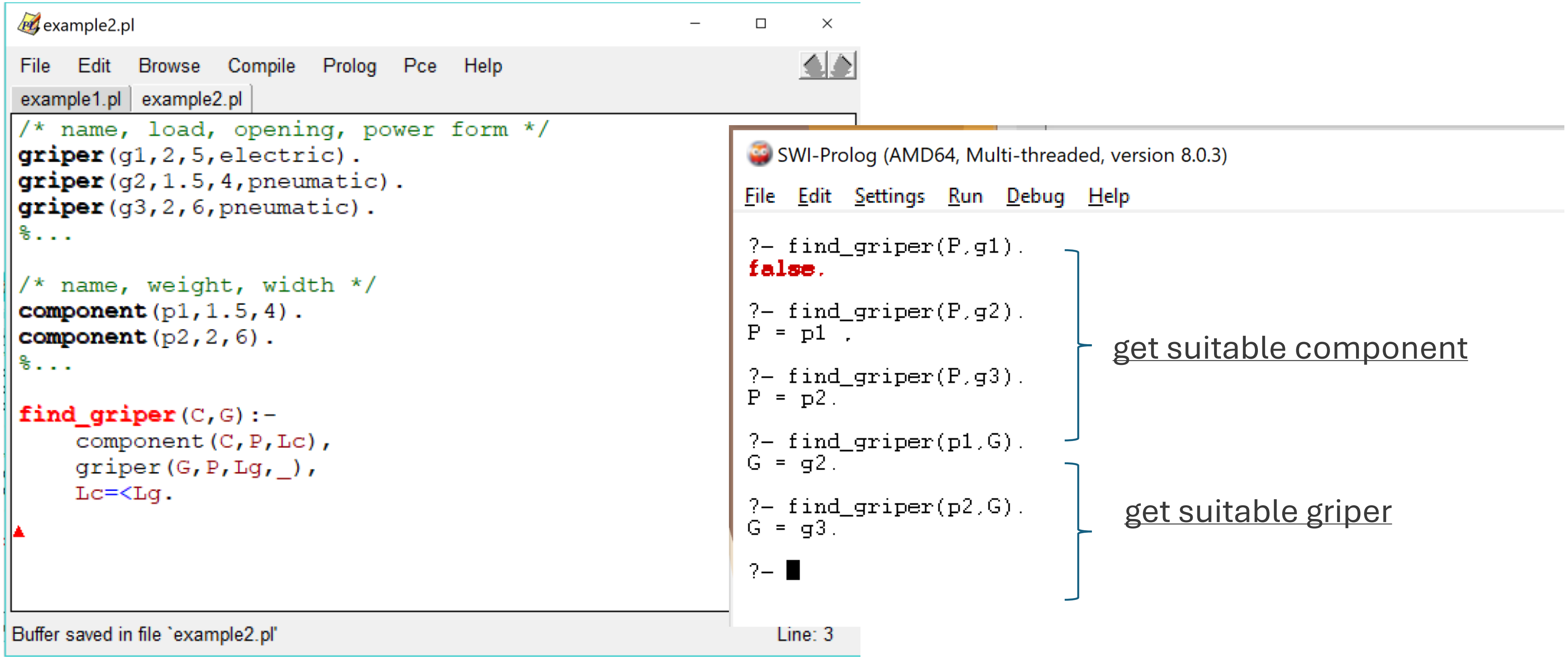
```
?- part(robot,X).
X = base ;
X = base ;
X = arm ;
X = griper ;
X = controller.
```

Capital letter

Lower case -> **Constants**

Another Example (i)

Example 2: Robot components [Done in THORETICAL class]


example2.pl
File Edit Browse Compile Prolog Pce Help
example1.pl
example2.pl
SWI-Prolog (AMD64, Multi-threaded, version 8.0.3)
File Edit Settings Run Debug Help
Line: 3

```prolog
/* name, load, opening, power form */
griper(g1,2,5,electric).
griper(g2,1.5,4,pneumatic).
griper(g3,2,6,pneumatic).
%...

/* name, weight, width */
component(p1,1.5,4).
component(p2,2,6).
%...

find_griper(C,G):-
    component(C,P,Lc),
    griper(G,P,Lg,_),
    Lc=<Lg.
```

Buffer saved in file `example2.pl'

```
?- find_griper(P,g1).
false.

?- find_griper(P,g2).
P = p1 .

?- find_griper(P,g3).
P = p2.

?- find_griper(p1,G).
G = g2.

?- find_griper(p2,G).
G = g3.

?- █
```

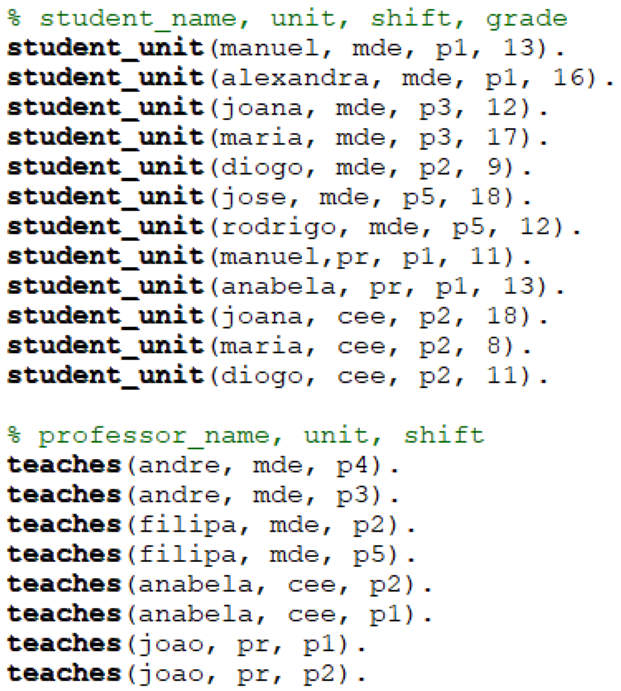
get suitable component

get suitable griper

Exercises

Exercise 1:

Consider the following predicates:



```
% student_name, unit, shift, grade
student_unit(manuel, mde, p1, 13).
student_unit(alexandra, mde, p1, 16).
student_unit(joana, mde, p3, 12).
student_unit(maria, mde, p3, 17).
student_unit(diogo, mde, p2, 9).
student_unit(jose, mde, p5, 18).
student_unit(rodrigo, mde, p5, 12).
student_unit(manuel,pr, p1, 11).
student_unit(anabela, pr, p1, 13).
student_unit(joana, cee, p2, 18).
student_unit(maria, cee, p2, 8).
student_unit(diogo, cee, p2, 11).

% professor_name, unit, shift
teaches(andre, mde, p4).
teaches(andre, mde, p3).
teaches(filipa, mde, p2).
teaches(filipa, mde, p5).
teaches(anabela, cee, p2).
teaches(anabela, cee, p1).
teaches(joao, pr, p1).
teaches(joao, pr, p2).
```

Write the corresponding rules that would answer the following queries:

a) Which students are enrolled in shift p3?

b) Which students from p5 have a grade > 14?

c) Which units is diogo enrolled in?

d) Who are the students of professor andre?

e) What are the professors of student joana?

Exercises

Exercise 2:

Given the following facts:

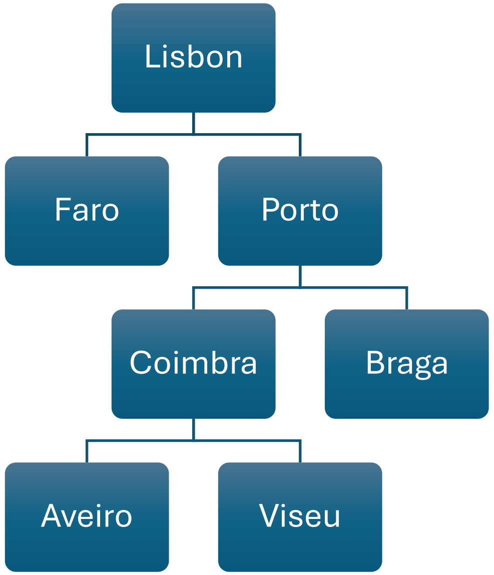
- Maria is fatter than Ana
- Ana is fatter than Luisa
- Luisa is fatter than Diana
- Diana is fatter than Sara

Write the corresponding facts and rules (using recursion) that determine that Maria is heavier that Sara.

Exercises

Exercise 3:

Consider the following road tree that connects several cities in Portugal (one direction)

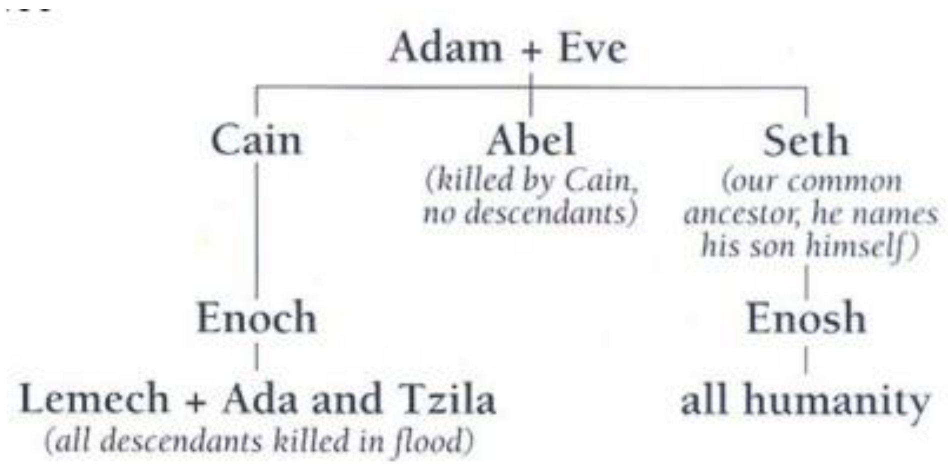


Write the corresponding facts and rules (using recursion) that would answer the following queries:

a) Can I travel from Lisbon to Viseu?
b) Can I travel from Faro to Braga?
c) How many cities I cross between Lisbon and Aveiro?

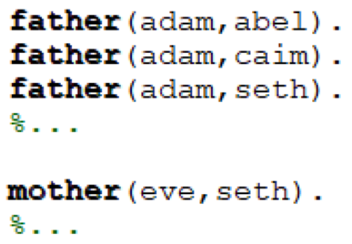
Exercises

Exercise 4:



Adam + Eve
Cain
Abel
(killed by Cain, no descendants)
Seth
(our common ancestor, he names his son himself)
Enoch
Enosh
Lemech + Ada and Tzila
(all descendants killed in flood)
all humanity

Use the predicates **father/2** and **mother/2** to represent the genealogic tree (in this example we use the Adam and Eve's genealogic tree, but you can use yours!):

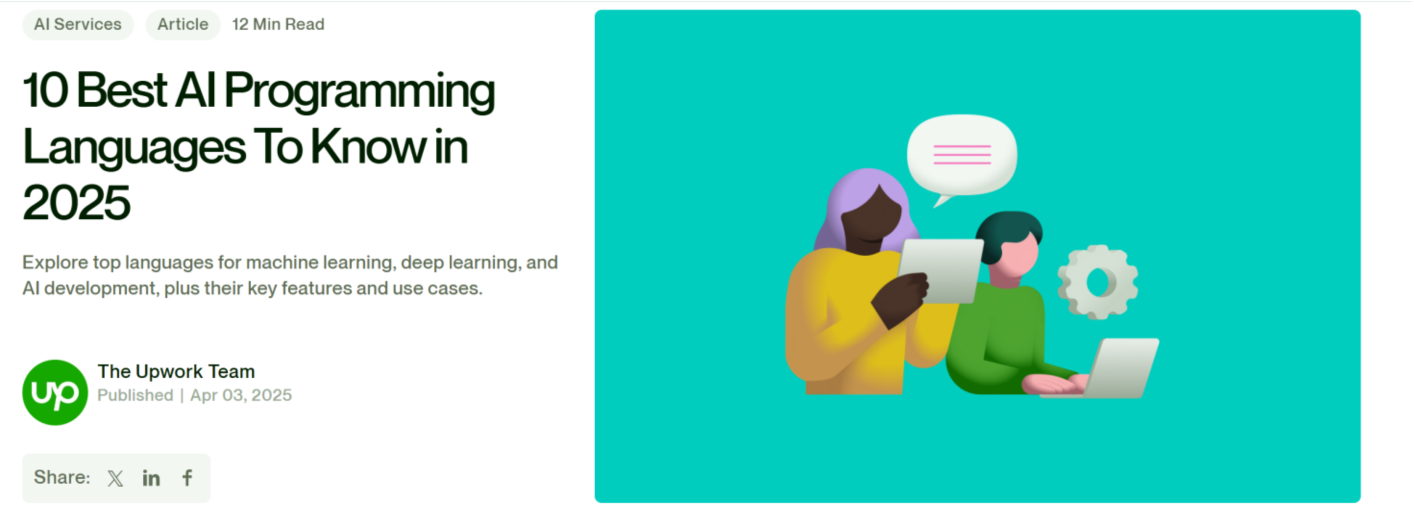


```
father(adam,abel).
father(adam,caim).
father(adam,seth).
%...

mother(eve,seth).
%...
```

Create rules to capture the following relationships:

- **son**(Father, Mother)
- **grandfather**(Grandfather, Grandson)
- **brother**(Brother1, Brother2)
- **uncle**(Uncle, Nephew)
- **cousin**(Cousin1, Cousin2)
- **ascendant**(Ascendant, Descendant)
- **descendant**(Descendant, Ascendant)

To Those Who Think Prolog is Dead… THINK AGAIN…

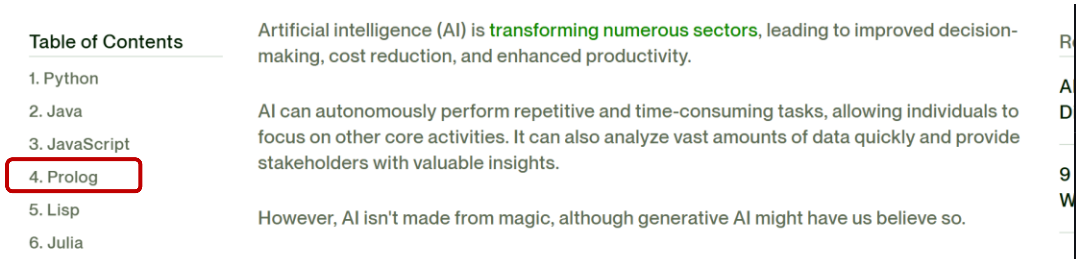


AI Services Article 12 Min Read

10 Best AI Programming Languages To Know in 2025

Explore top languages for machine learning, deep learning, and AI development, plus their key features and use cases.

The Upwork Team
Published | Apr 03, 2025

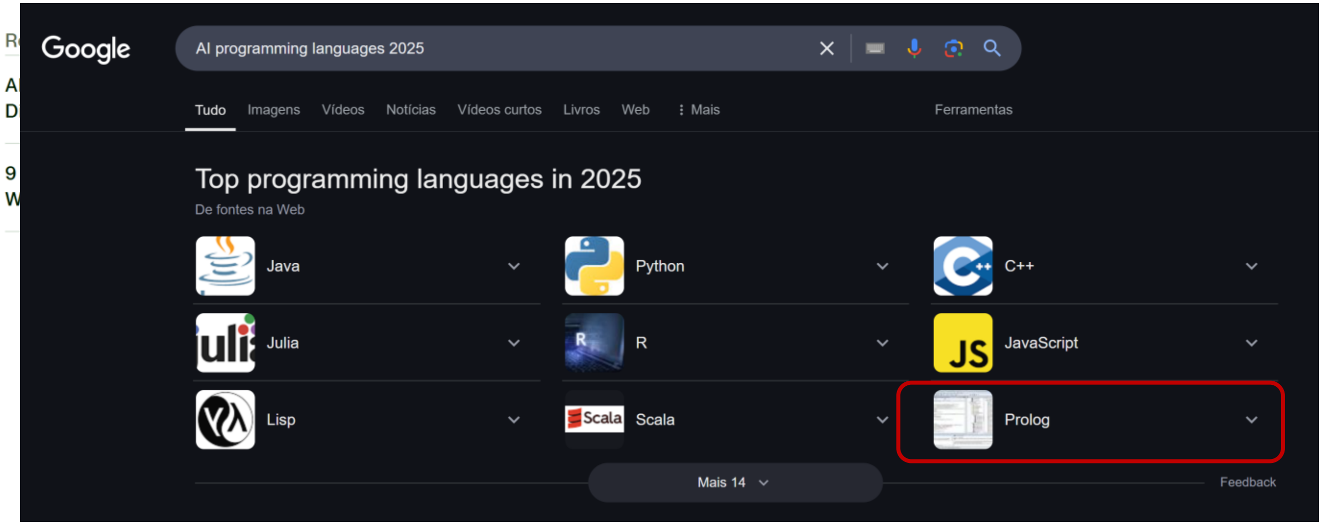


Artificial intelligence (AI) is transforming numerous sectors, leading to improved decision-making, cost reduction, and enhanced productivity.

AI can autonomously perform repetitive and time-consuming tasks, allowing individuals to focus on other core activities. It can also analyze vast amounts of data quickly and provide stakeholders with valuable insights.

However, AI isn't made from magic, although generative AI might have us believe so.

https://www.upwork.com/resources/best-ai-programming-language



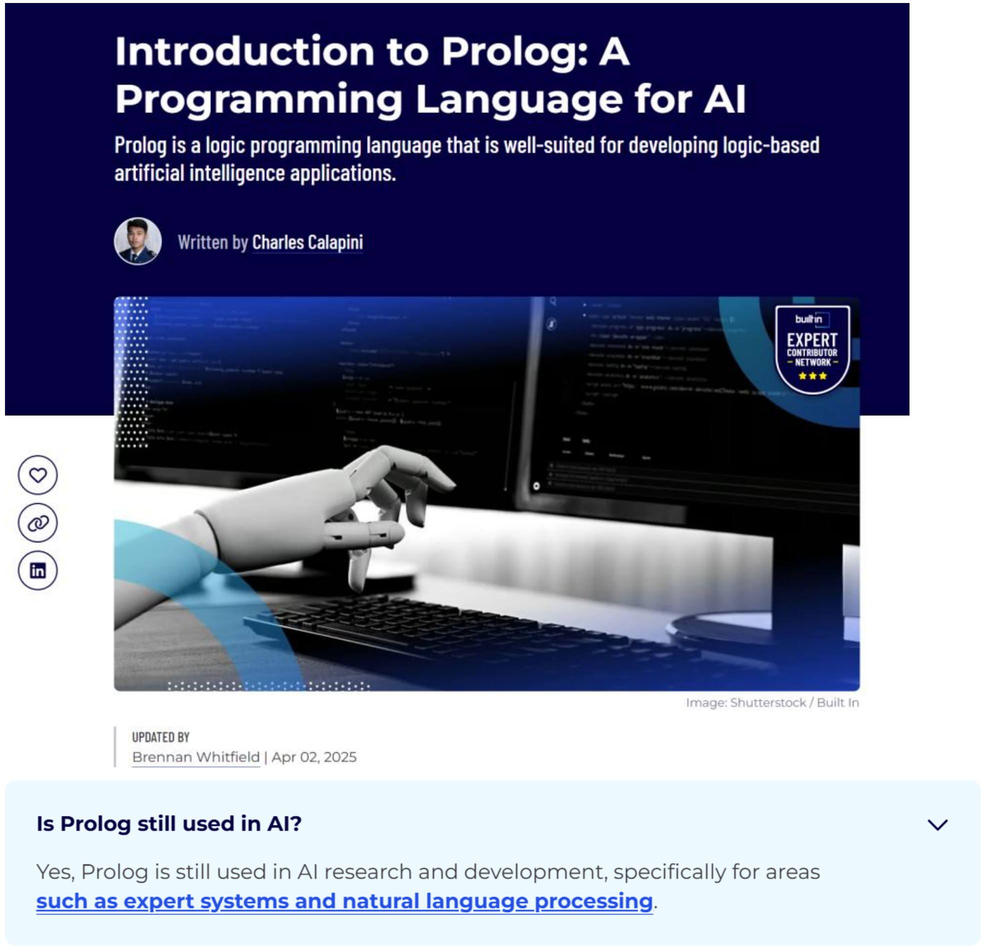
Top programming languages in 2025

De fontes na Web

Java	Python	C++
Julia	R	JavaScript
Lisp	Scala	Prolog

To Those Who Think Prolog is Dead… THINK AGAIN…

NOVA



Introduction to Prolog: A Programming Language for AI

Prolog is a logic programming language that is well-suited for developing logic-based artificial intelligence applications.

Written by Charles Calapini

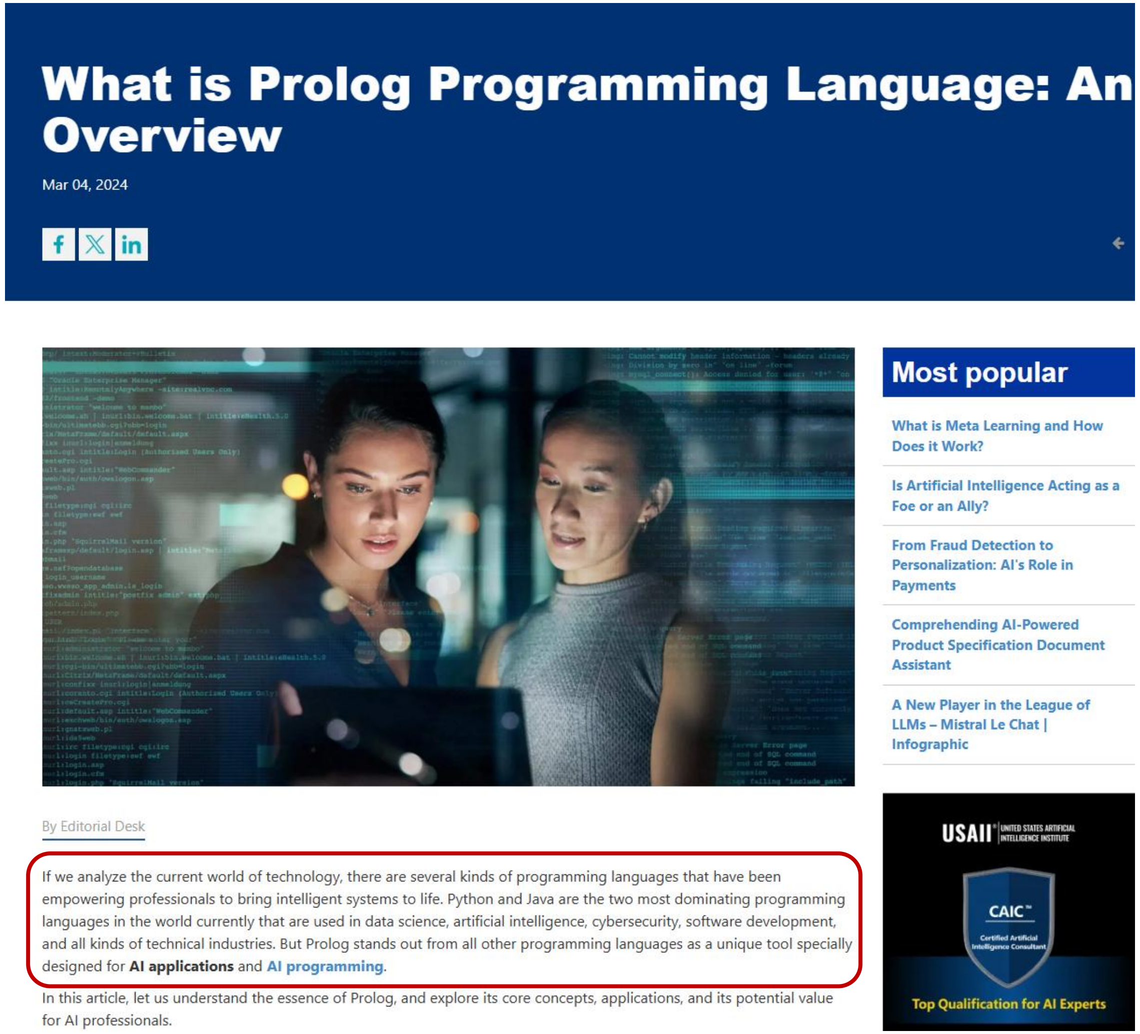
Image: Shutterstock / Built In

UPDATED BY
Brennan Whitfield | Apr 02, 2025

Is Prolog still used in AI?

Yes, Prolog is still used in AI research and development, specifically for areas such as expert systems and natural language processing.

https://builtin.com/software-engineering-perspectives/prolog



What is Prolog Programming Language: An Overview

Mar 04, 2024

Most popular

- What is Meta Learning and How Does it Work?
- Is Artificial Intelligence Acting as a Foe or an Ally?
- From Fraud Detection to Personalization: AI's Role in Payments
- Comprehending AI-Powered Product Specification Document Assistant
- A New Player in the League of LLMs – Mistral Le Chat | Infographic

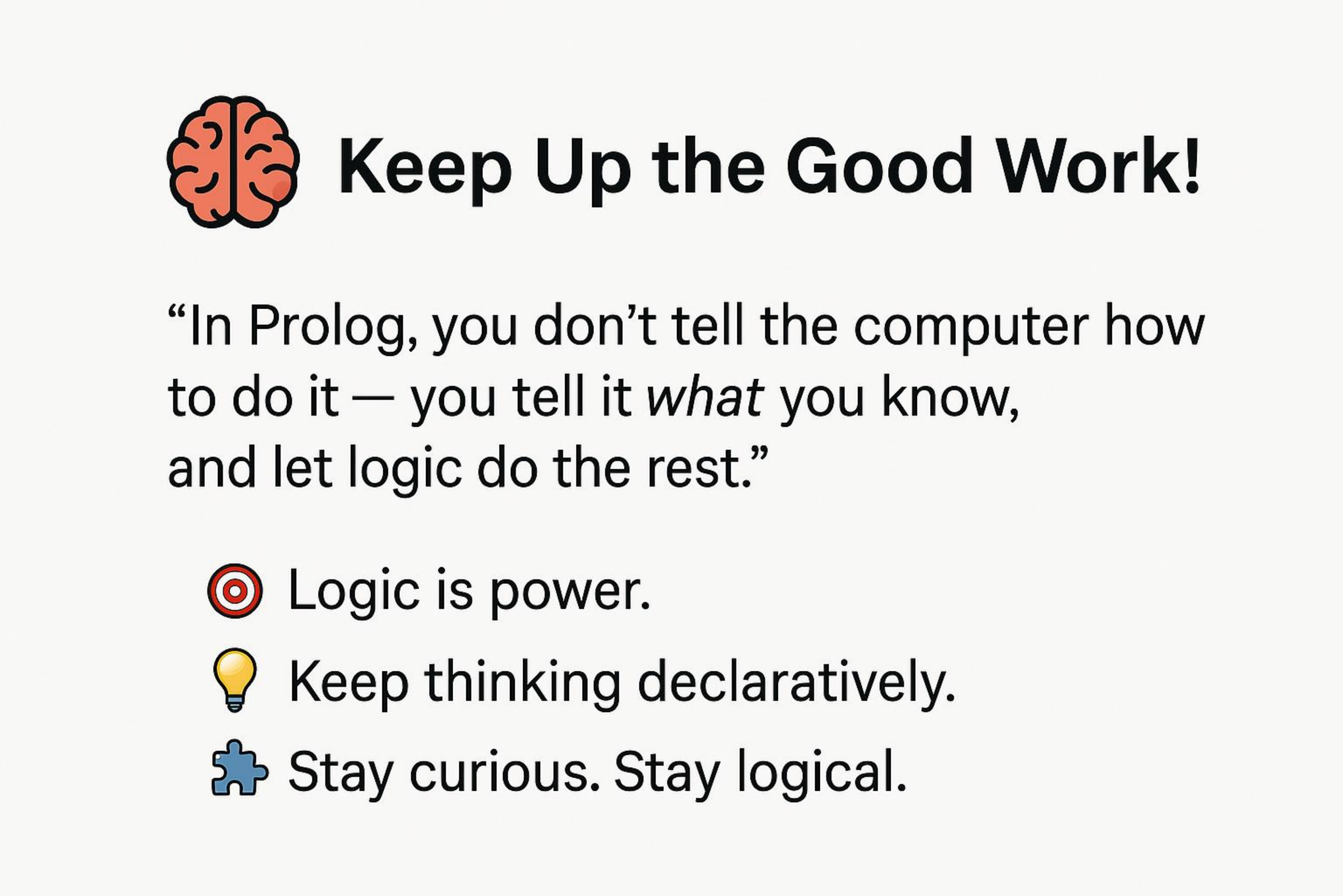
USAII® UNITED STATES ARTIFICIAL INTELLIGENCE INSTITUTE
CAIC™
Certified Artificial Intelligence Consultant
Top Qualification for AI Experts

By Editorial Desk

If we analyze the current world of technology, there are several kinds of programming languages that have been empowering professionals to bring intelligent systems to life. Python and Java are the two most dominating programming languages in the world currently that are used in data science, artificial intelligence, cybersecurity, software development, and all kinds of technical industries. But Prolog stands out from all other programming languages as a unique tool specially designed for **AI applications** and **AI programming**.

In this article, let us understand the essence of Prolog, and explore its core concepts, applications, and its potential value for AI professionals.

https://www.usaii.org/ai-insights/what-is-prolog-programming-language-an-overview



🧠 Keep Up the Good Work!

“In Prolog, you don’t tell the computer how to do it — you tell it *what* you know, and let logic do the rest.”

- 🎯 Logic is power.
- 💡 Keep thinking declaratively.
- 🧩 Stay curious. Stay logical.

Image generated with the assistance of ChatGPT (OpenAI), 2025.