



CS 362: Intelligent Systems

Slide 1

UNIT

Introduction to Prolog



Introduction to Prolog



Slide 2

- Teaching Material

- Learn Prolog Now!

- <http://lpn.swi-prolog.org/lpnpag.php?pageid=top>

- SWI Prolog interpreter

- <http://www.swi-prolog.org/>

- Knowledge bases code available at

- <http://l3s.de/~gaugaz/ki/prolog/>



Lecture 1



Slide 3

Aim of this lecture

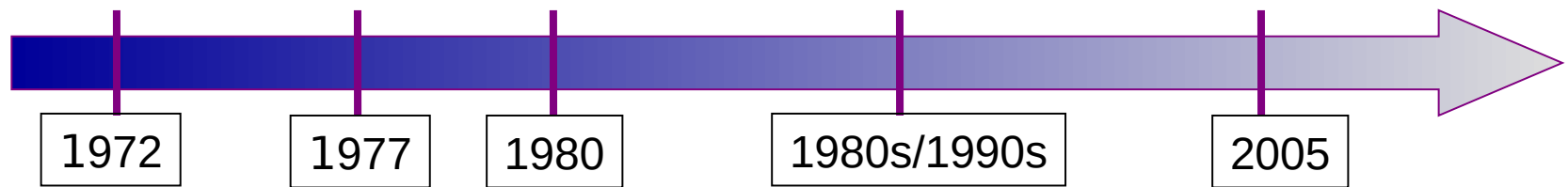
- Give some simple examples of Prolog programs
- Discuss the three basic constructs in Prolog:
 - Facts
 - Rules
 - Queries
- Introduce other concepts, such as
 - the role of logic
 - unification with the help of variables
- Begin the systematic study of Prolog by defining terms, atoms, and variables



History of Prolog

Slide 4

first Prolog interpreter by
Colmerauer and Roussel

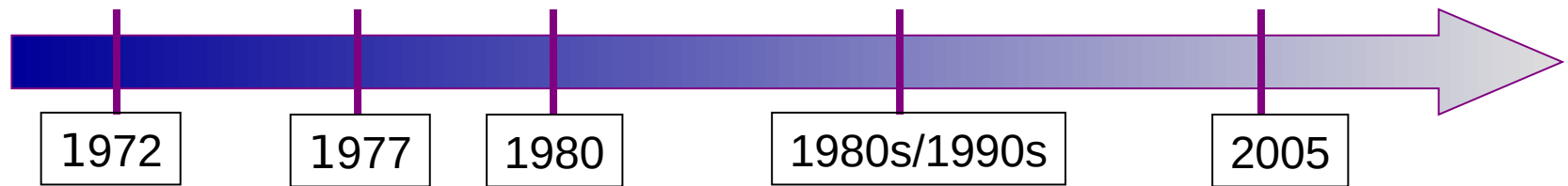




History of Prolog

Slide 5

implementation of DEC10
compiler by Warren

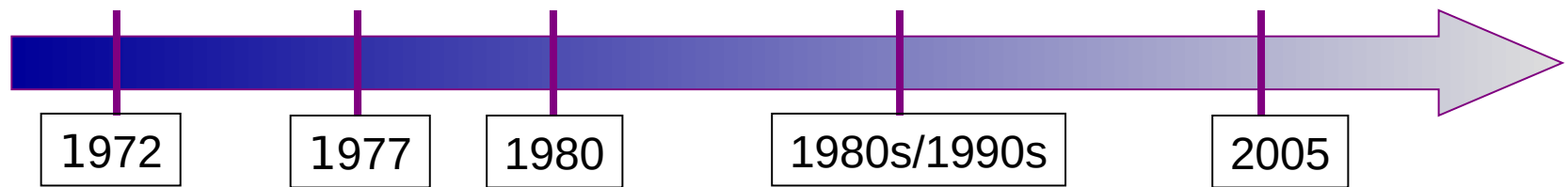




History of Prolog

Slide 6

Definite Clause Grammars
implementation by Pereira and
Warren

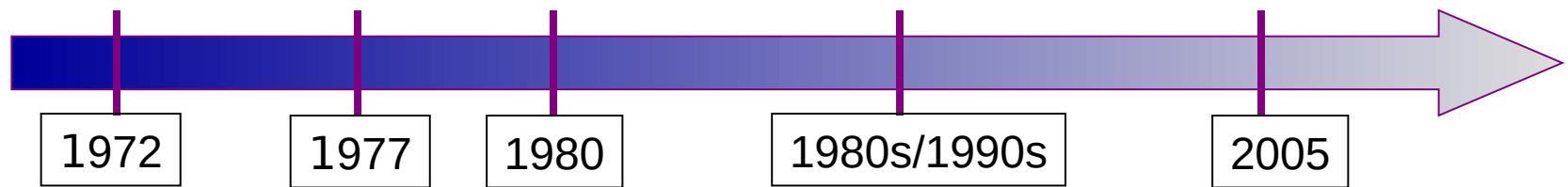




History of Prolog

Slide 7

Prolog grows in popularity especially in Europe and Japan

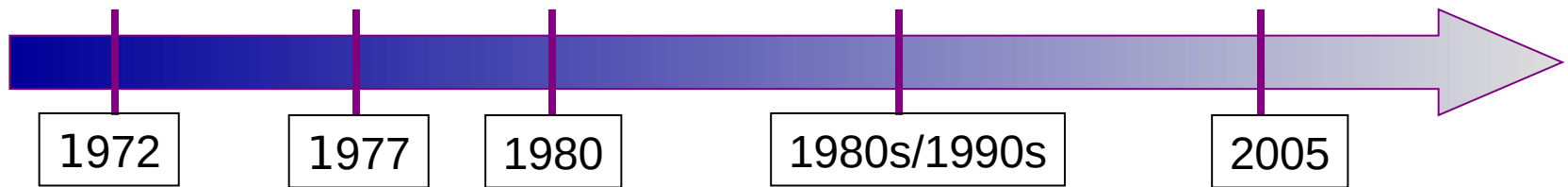




History of Prolog

Slide 8

Prolog used to program natural language interface in International Space Station by NASA





Prolog



Slide 9

- "Programming with Logic"
- Declarative
- Very different from other (procedural) programming languages
- Good for knowledge-rich tasks

Basic idea of Prolog

- Describe the situation of interest
- Ask a question
- Prolog logically deduces new facts about the situation we described
- Prolog gives us its deductions back as answers



Consequences



Slide
10

- **Think declaratively, not procedurally**
 - Challenging
 - Requires a different mindset
- **High-level language**
 - Not as efficient as, say, C
 - Good for rapid prototyping
 - Useful in many AI applications



Knowledge Base 1



Slide
11

woman(mia).

woman(jody).

woman(yolanda).

playsAirGuitar(jody).

party.



Knowledge Base 1



Slide
12

woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
party.

?-



Knowledge Base 1



Slide
12

woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
party.

?- woman(mia).



Knowledge Base 1



Slide
14

woman(mia).

woman(jody).

woman(yolanda).

playsAirGuitar(jody).

party.

?- woman(mia).

yes

?-



Knowledge Base 1



Slide
15

woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
party.

?- woman(mia).
yes
?- playsAirGuitar(jody).



Knowledge Base 1



Slide
16

woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
party.

?- woman(mia).
yes
?- playsAirGuitar(jody).
yes
?-



Knowledge Base 1



Slide
17

woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
party.

?- woman(mia).
yes
?- playsAirGuitar(jody).
yes
?- playsAirGuitar(mia).
no



Knowledge Base 1



Slide
18

woman(mia).

woman(jody).

woman(yolanda).

playsAirGuitar(jody).

party.

?- tattoed(jody).



Knowledge Base 1



Slide
10

woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
party.

?- tattoed(jody).
no
?-



Knowledge Base 1



Slide
20

woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
party.

?- tattoed(jody).
ERROR: predicate tattoed/1 not defined.
?-



Knowledge Base 1



Slide
21

woman(mia).

woman(jody).

woman(yolanda).

playsAirGuitar(jody).

party.

?- party.



Knowledge Base 1



Slide
22

woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
party.

?- party.
yes
?-



Knowledge Base 1



Slide
22

woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
party.

?- rockConcert.



Knowledge Base 1



Slide
24

woman(mia).

woman(jody).

woman(yolanda).

playsAirGuitar(jody).

party.

?- rockConcert.

no

?-



Knowledge Base 2



Slide
25

happy(yolanda).

listens2music(mia).

listens2music(yolanda):- happy(yolanda).

playsAirGuitar(mia):- listens2music(mia).

playsAirGuitar(yolanda):- listens2music(yolanda).



Knowledge Base 2



Slide
26

fact

happy(yolanda).

listens2music(mia).

listens2music(yolanda):- happy(yolanda).

playsAirGuitar(mia):- listens2music(mia).

playsAirGuitar(yolanda):- listens2music(yolanda).



Knowledge Base 2



Slide
27

happy(yolanda).

fact

listens2music(mia).

fact

listens2music(yolanda):- happy(yolanda).

playsAirGuitar(mia):- listens2music(mia).

playsAirGuitar(yolanda):- listens2music(yolanda).



Knowledge Base 2



Slide
28

happy(yolanda).

fact

listens2music(mia).

fact

listens2music(yolanda):- happy(yolanda).

rule

playsAirGuitar(mia):- listens2music(mia).

playsAirGuitar(yolanda):- listens2music(yolanda).



Knowledge Base 2



Slide
20

happy(yolanda).

fact

listens2music(mia).

fact

listens2music(yolanda):- happy(yolanda).

rule

playsAirGuitar(mia):- listens2music(mia).

rule

playsAirGuitar(yolanda):- listens2music(yolanda).



Knowledge Base 2



Slide
20

happy(yolanda).

fact

listens2music(mia).

fact

listens2music(yolanda):- happy(yolanda).

rule

playsAirGuitar(mia):- listens2music(mia).

rule

playsAirGuitar(yolanda):- listens2music(yolanda).

rule



Knowledge Base 2



Slide
21

happy(yolanda).

listens2music(mia).

listens2music(yolanda):- happy(yolanda).

playsAirGuitar(mia):- listens2music(mia).

playsAirGuitar(yolanda):- listens2music(yolanda).

head

body



Knowledge Base 2



Slide
22

happy(yolanda).

listens2music(mia).

listens2music(yolanda):- happy(yolanda).

playsAirGuitar(mia):- listens2music(mia).

playsAirGuitar(yolanda):- listens2music(yolanda).

?-



Knowledge Base 2



Slide
22

happy(yolanda).

listens2music(mia).

listens2music(yolanda):- happy(yolanda).

playsAirGuitar(mia):- listens2music(mia).

playsAirGuitar(yolanda):- listens2music(yolanda).

?- playsAirGuitar(mia).

yes

?-



Knowledge Base 2



Slide
24

happy(yolanda).

listens2music(mia).

listens2music(yolanda):- happy(yolanda).

playsAirGuitar(mia):- listens2music(mia).

playsAirGuitar(yolanda):- listens2music(yolanda).

?- playsAirGuitar(mia).

yes

?- playsAirGuitar(yolanda).

yes



Clauses



Slide
25

happy(yolanda).

listens2music(mia).

listens2music(yolanda):- happy(yolanda).

playsAirGuitar(mia):- listens2music(mia).

playsAirGuitar(yolanda):- listens2music(yolanda).

*There are five clauses in this knowledge base:
two facts, and three rules.*

The end of a clause is marked with a full stop.



Predicates



Slide
26

happy(yolanda).

listens2music(mia).

listens2music(yolanda):- happy(yolanda).

playsAirGuitar(mia):- listens2music(mia).

playsAirGuitar(yolanda):- listens2music(yolanda).

*There are three **predicates**
in this knowledge base:
happy, listens2music, and playsAirGuitar*



Knowledge Base 3



Slide
27

happy(vincent).

listens2music(butch).

playsAirGuitar(vincent):- listens2music(vincent), happy(vincent).

playsAirGuitar(butch):- happy(butch).

playsAirGuitar(butch):- listens2music(butch).



Expressing Conjunction



Slide
28

happy(vincent).

listens2music(butch).

playsAirGuitar(vincent):- listens2music(vincent), happy(vincent).

playsAirGuitar(butch):- happy(butch).

playsAirGuitar(butch):- listens2music(butch).

The comma “,” expresses conjunction in Prolog



Knowledge Base 3



Slide
20

happy(vincent).

listens2music(butch).

playsAirGuitar(vincent):- listens2music(vincent), happy(vincent).

playsAirGuitar(butch):- happy(butch).

playsAirGuitar(butch):- listens2music(butch).

?- playsAirGuitar(vincent).

no

?-



Knowledge Base 3



Slide
40

happy(vincent).

listens2music(butch).

playsAirGuitar(vincent):- listens2music(vincent), happy(vincent).

playsAirGuitar(butch):- happy(butch).

playsAirGuitar(butch):- listens2music(butch).

?- playsAirGuitar(butch).

yes

?-



Expressing Disjunction



Slide
41

```
happy(vincent).  
listens2music(butch).  
playsAirGuitar(vincent):- listens2music(vincent), happy(vincent).  
playsAirGuitar(butch):- happy(butch).  
playsAirGuitar(butch):- listens2music(butch).
```

```
happy(vincent).  
listens2music(butch).  
playsAirGuitar(vincent):- listens2music(vincent), happy(vincent).  
playsAirGuitar(butch):- happy(butch); listens2music(butch).
```



Prolog and Logic



Slide
42

- Clearly Prolog has something to do with logic
- Operators
 - Implication :-
 - Conjunction ,
 - Disjunction ;
- Use of modus ponens
- Negation



Knowledge Base 4



Slide
42

woman(mia).

woman(jody).

woman(yolanda).

loves(vincent, mia).

loves(marsellus, mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).



Prolog Variables



Slide
44

woman(mia).

woman(jody).

woman(yolanda).

loves(vincent, mia).

loves(marsellus, mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

?- woman(X).



Variable Instantiation



Slide
45

woman(mia).

woman(jody).

woman(yolanda).

loves(vincent, mia).

loves(marsellus, mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

?- woman(X).

X=mia



Asking Alternatives



Slide
46

woman(mia).

woman(jody).

woman(yolanda).

loves(vincent, mia).

loves(marsellus, mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

?- woman(X).

X=mia;



Asking Alternatives



Slide
47

woman(mia).

woman(jody).

woman(yolanda).

loves(vincent, mia).

loves(marsellus, mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

?- woman(X).

X=mia;

X=jody



Asking Alternatives



Slide
48

woman(mia).

woman(jody).

woman(yolanda).

loves(vincent, mia).

loves(marsellus, mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

?- woman(X).

X=mia;

X=jody;

X=yolanda



Asking Alternatives



Slide
49

woman(mia).

woman(jody).

woman(yolanda).

loves(vincent, mia).

loves(marsellus, mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

?- woman(X).

X=mia;

X=jody;

X=yolanda;

no



Knowledge Base 4



Slide
50

woman(mia).

woman(jody).

woman(yolanda).

loves(vincent, mia).

loves(marsellus, mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

?- loves(marsellus,X), woman(X).



Knowledge Base 4



Slide
51

woman(mia).

woman(jody).

woman(yolanda).

loves(vincent, mia).

loves(marsellus, mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

?- loves(marsellus,X), woman(X).

X=mia

yes

?-



Knowledge Base 4



Slide
52

woman(mia).

woman(jody).

woman(yolanda).

loves(vincent, mia).

loves(marsellus, mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

?- loves(pumpkin,X), woman(X).



Knowledge Base 4



Slide
52

woman(mia).

woman(jody).

woman(yolanda).

loves(vincent, mia).

loves(marsellus, mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

?- loves(pumpkin,X), woman(X).

no

?-



Knowledge Base 5



Slide
54

loves(vincent,mia).

loves(marsellus,mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

jealous(X,Y):- loves(X,Z), loves(Y,Z).



Knowledge Base 5



Slide
55

loves(vincent,mia).

loves(marsellus,mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

jealous(X,Y):- loves(X,Z), loves(Y,Z).

?- jealous(marsellus,W).



Knowledge Base 5



Slide
56

loves(vincent,mia).

loves(marsellus,mia).

loves(pumpkin, honey_bunny).

loves(honey_bunny, pumpkin).

jealous(X,Y):- loves(X,Z), loves(Y,Z).

?- jealous(marsellus,W).

W=vincent

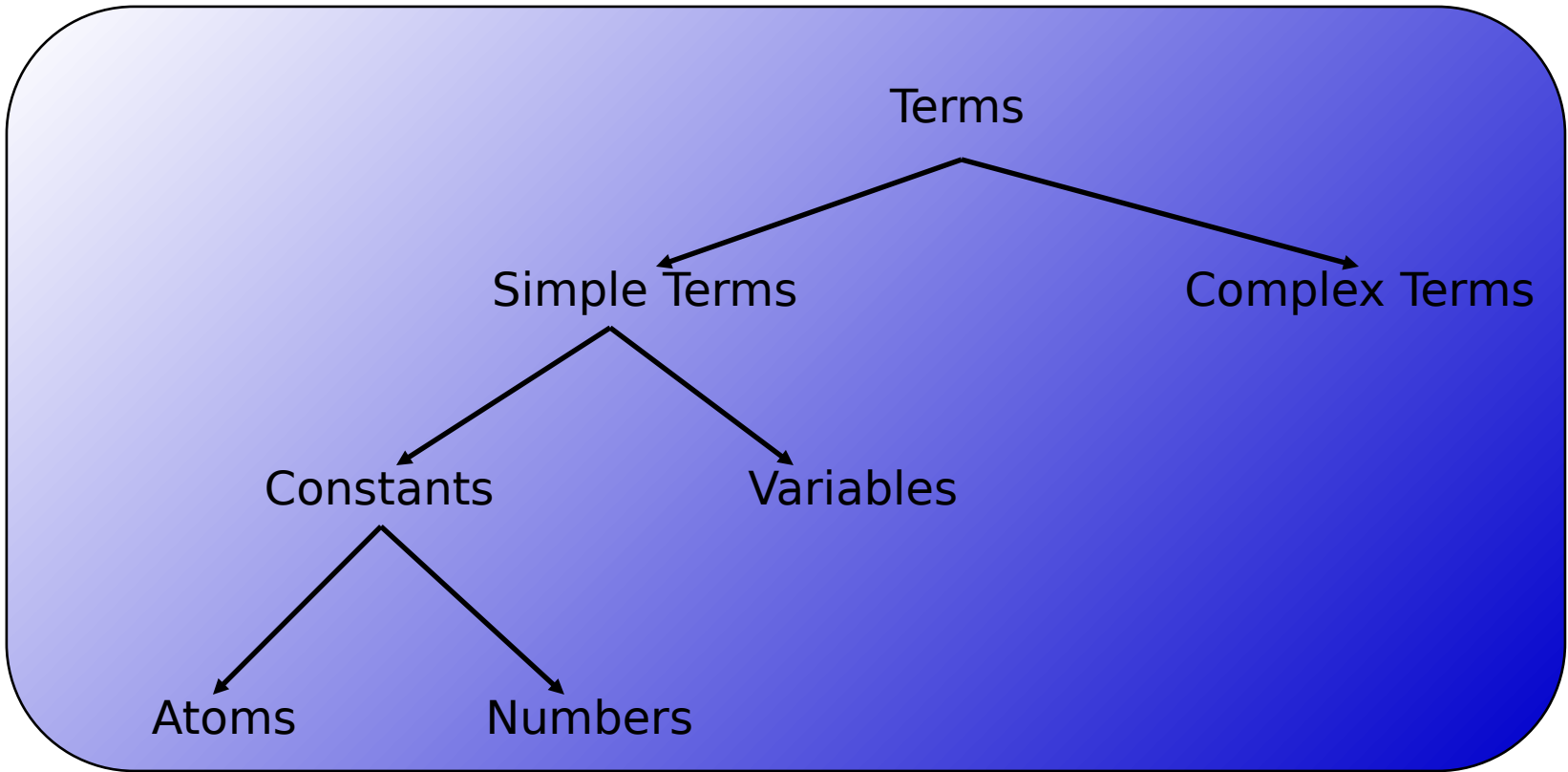
?-



Prolog Syntax

Slide 57

- What exactly are facts, rules and queries built out of?





Prolog Syntax



Slide
58

Atoms

- A sequence of characters of upper-case letters, lower-case letters, digits, or underscore, starting with a lowercase letter
✂ *Examples:* butch, big_kahuna_burger, playGuitar
- An arbitrary sequence of characters enclosed in single quotes
✂ *Examples:* 'Vincent', 'Five dollar shake', '@\$%'
- A sequence of special characters
✂ *Examples:* : , ; . :-



Prolog Syntax

Slide
50

Numbers

- Integers: 12, -34, 22342
- Floats: 34573.3234

Variables

- A sequence of characters of upper-case letters, lower-case letters, digits, or underscore, starting with either an uppercase letter or an underscore
- Examples:

X, Y, Variable, Vincent, _tag



Prolog Syntax

Slide
60

Complex Terms

- Atoms, numbers and variables are building blocks for complex terms
- Complex terms are built out of a functor directly followed by a sequence of arguments
- Arguments are put in round brackets, separated by commas
- The functor must be an atom

Examples of complex terms

- Examples we have seen before:
 - `playsAirGuitar(jody)`
 - `loves(vincent, mia)`
 - `jealous(marsellus, W)`
- Complex terms inside complex terms:
 - `hide(X, father(father(father(butch))))`



Prolog Syntax

Slide
61

Arity

- The number of arguments a complex term has is called its arity
- Examples:
 - woman(mia) is a term with arity 1
 - loves(vincent,mia) has arity 2
 - father(father(butch)) arity 1

Arity is important

- In Prolog you can define two predicates with the same functor but with different arity
- Prolog would treat this as two different predicates
- In Prolog documentation arity of a predicate is usually indicated with the suffix `"/"` followed by a number to indicate the arity



Example of Arity



Slide
62

```
happy(yolanda).  
listens2music(mia).  
listens2music(yolanda):- happy(yolanda).  
playsAirGuitar(mia):- listens2music(mia).  
playsAirGuitar(yolanda):- listens2music(yolanda).
```

- **This knowledge base defines**
 - **happy/1**
 - **listens2music/1**
 - **playsAirGuitar/1**



Exercises

Slide
62

Learn Prolog Now!

Chapter 1

by Patrick Blackburn, Johan Bos, and Kristina Striegnitz

[LPNI Home](#)

[Free Online Version](#)

[Paperback English](#)

[Paperback Français](#)

[Teaching Prolog](#)

[Prolog Implementations](#)

[Prolog Manuals](#)

[Prolog Links](#)

[Thanks!](#)

[Contact us](#)

[\[next \]](#) [\[prev \]](#) [\[prev-tail \]](#) [\[tail \]](#) [\[up \]](#)

Chapter 1 Facts, Rules, and Queries

This chapter has two main goals:

1. To give some simple examples of Prolog programs. This will introduce us to the three basic constructs in Prolog: facts, rules, and queries. It will also introduce us to a number of other themes, like the role of logic in Prolog, and the idea of performing unification with the aid of variables.
2. To begin the systematic study of Prolog by defining terms, atoms, variables and other syntactic concepts.

[1.1 Some Simple Examples](#)

[Knowledge Base 1](#)

[Knowledge Base 2](#)

[Knowledge Base 3](#)

[Knowledge Base 4](#)

[Knowledge Base 5](#)

[1.2 Prolog Syntax](#)

[Atoms](#)

[Numbers](#)

[Variables](#)

[Complex terms](#)

[1.3 Exercises](#)

[1.4 Practical Session](#)

[\[next \]](#) [\[prev \]](#) [\[prev-tail \]](#) [\[front \]](#) [\[up \]](#)



Summary of this lecture



Slide
64

- Simple examples of Prolog programs
- Introduced three basic constructs in Prolog:
 - Facts
 - Rules
 - Queries
- Discussed other concepts, such as
 - the role of logic
 - unification with the help of variables
- Definition of Prolog constructs:
terms, atoms, and variables