1

Logic Programming with Prolog

©1993 John Mylopoulos

2

Logic Programming with Prolog

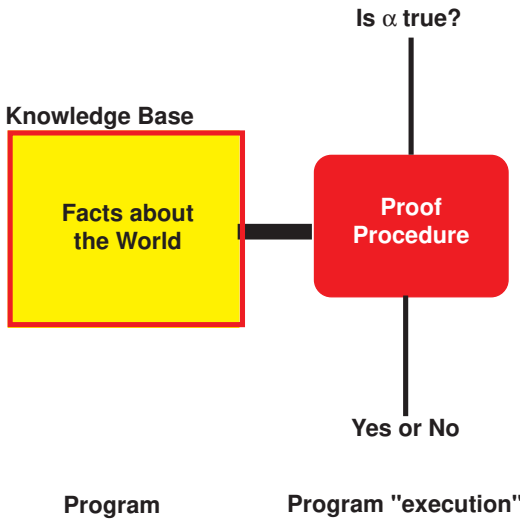
Table of Contents

1. Introduction (15 slides)
2. Proof Trees (17 slides)
3. Search (16 slides)
4. Running Prolog (7 slides)
5. List Processing (20 slides)
6. Control (15 slides)
7. Data Structures (9 slides)
8. Tools (19 slides)
9. Style (11 slides)
10. Logic (20 slides)

©1993 John Mylopoulos

3

Logic Programming



©1993 John Mylopoulos

4

Simple Deduction (Inference)

Socrates is a man
If Socrates is a man then Socrates is mortal

Socrates is mortal

More generally,

$$\frac{\alpha \quad \alpha \Rightarrow \beta}{\beta}$$ (modus ponens)

©1993 John Mylopoulos

5

Simple Deduction in Terms of Rules and Facts

Initial goal: Socrates is mortal

R1

Subgoal: Socrates is a man

Rules	**Facts**
R1: If Socrates is a man then Socrates is mortal	... ... Socrates is a man

©1993 John Mylopoulos

6

Another Example

Initial goal: Socrates is mortal

R2 R1

Socrates is a cat — Socrates is a man

R3

Socrates purrs

Rules	**Facts**
R1: If Socrates is a man then Socrates is mortal R2: If Socrates is a cat then Socrates is mortal R3: If Socrates purrs then Socrates is a cat	... ... Socrates purrs

©1993 John Mylopoulos

7

The Parking-Around-The-University Problem (generally intractable, alas!)

R1: **If** (1) You have a parking permit &
(2) This space is permit-parkable
then you can park at this space

R2: **If** (1) This space is a parking space &
(2) permit sign at this space &
(3) current date is acceptable
then this space is permit parkable

R3: **If** (1) This space is permit parkable &
(2) Current time is 7am to midnight
then you can park at this space

R4: **If** (1) Current month is December to March
then current date is acceptable

R5: **If** (1) Current month is April to November &
(2) Current day of month is ≤ 15
then current date is acceptable

©1993 John Mylopoulos

8

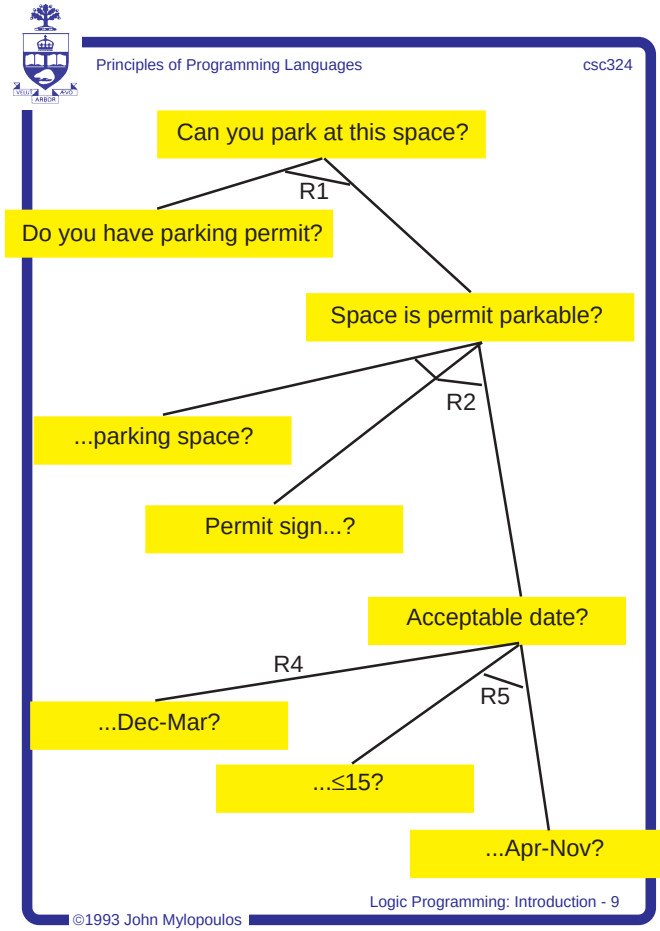
Backward Chaining: Start from Goal, Try to Reduce it to Given Facts

Goal: Can you park at this space?

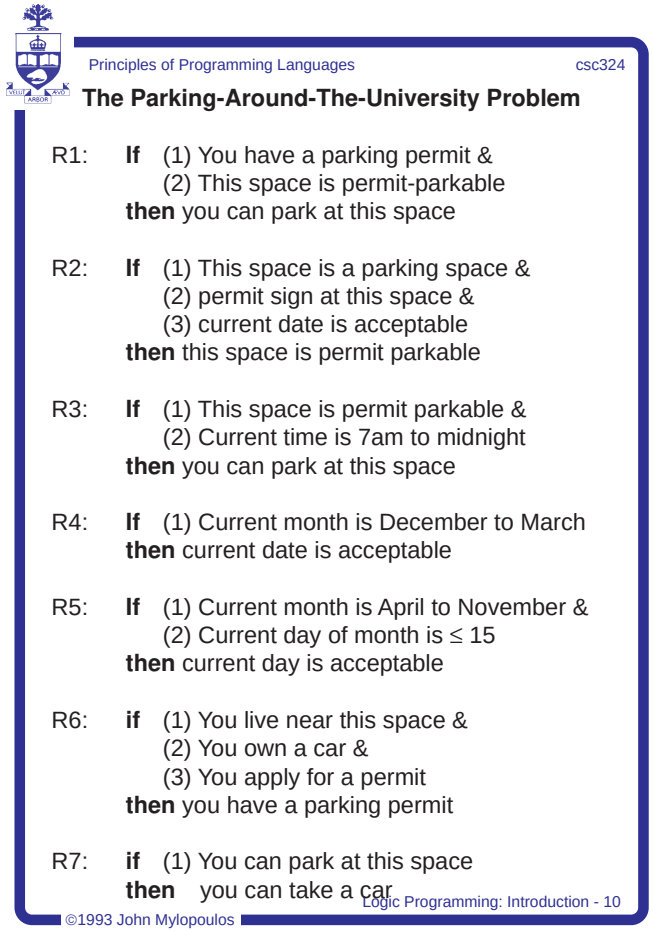
Facts: This space is a parking space
You live near the space
You have a parking permit
Current month is in April to November
Current day is Tuesday
Current day of month is ≤ 15
Permit sign at this space
You own a car
Your car is > 5 years old
Current time is 3am
You have $18.00

©1993 John Mylopoulos

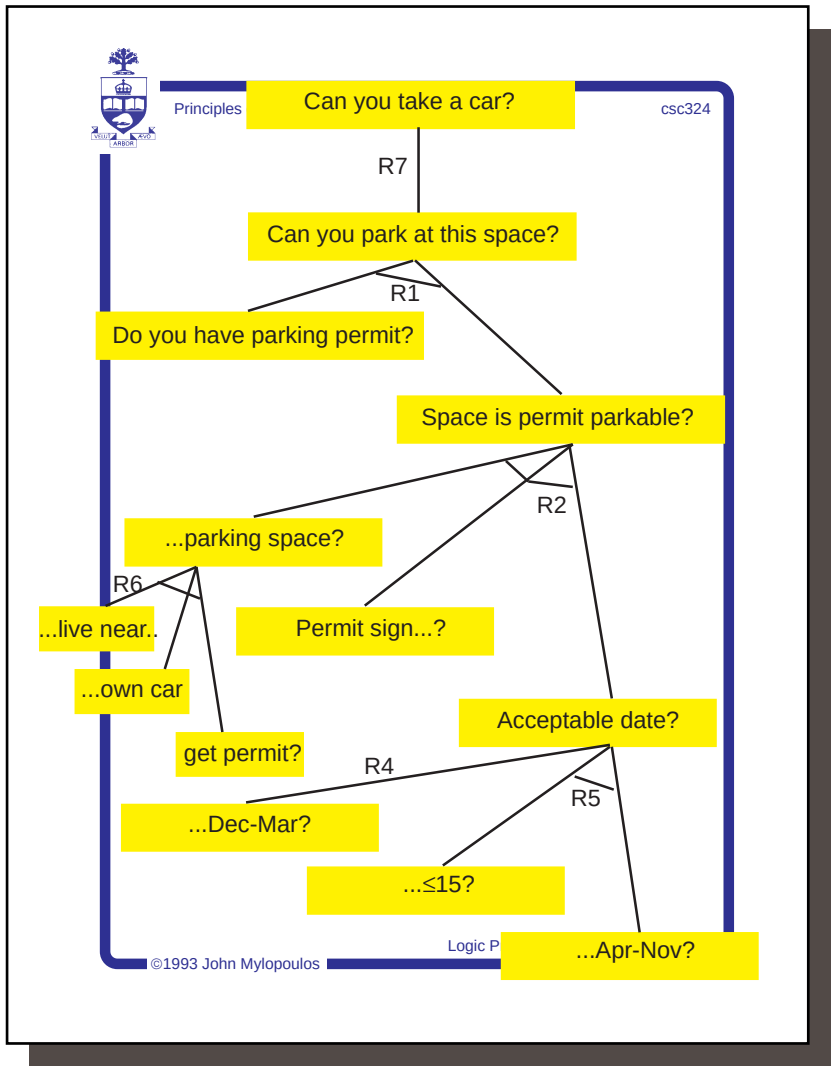
9


Principles of Programming Languages
csc324
Can you park at this space?
R1
Do you have parking permit?
Space is permit parkable?
R2
...parking space?
Permit sign...?
Acceptable date?
R4
R5
...Dec-Mar?
...≤15?
...Apr-Nov?
Logic Programming: Introduction - 9
©1993 John Mylopoulos

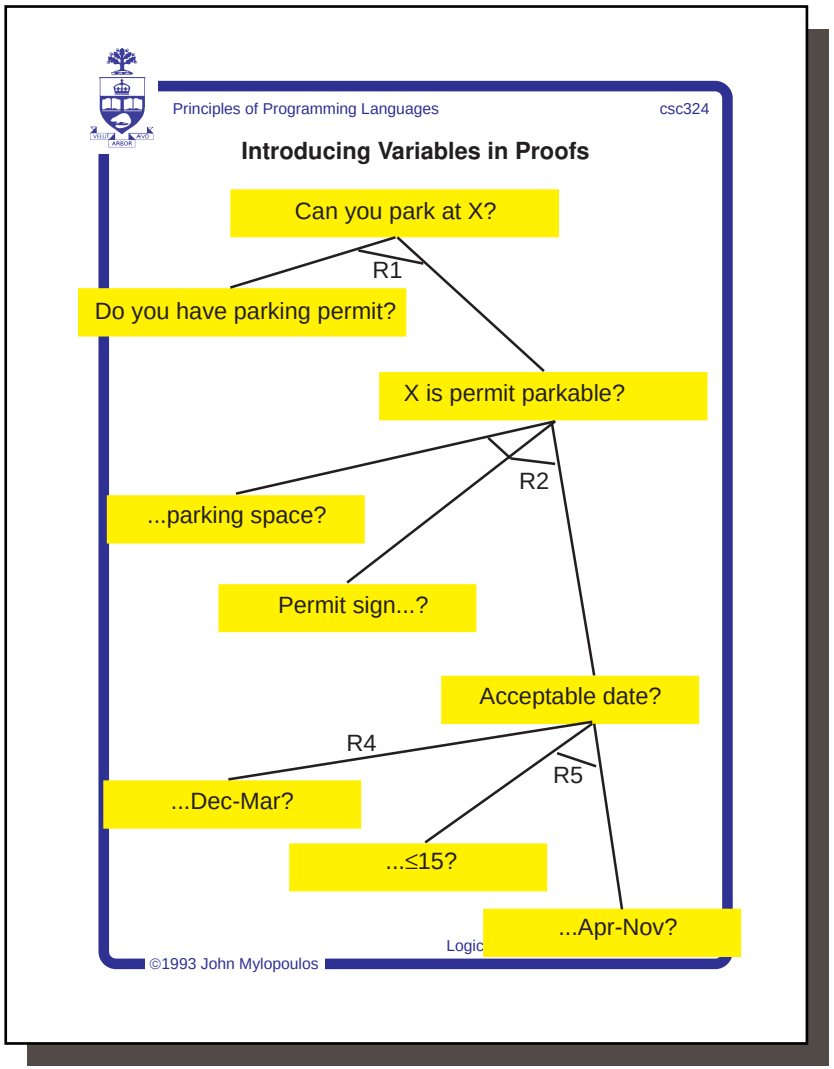
10


Principles of Programming Languages
csc324
The Parking-Around-The-University Problem
R1: If (1) You have a parking permit &
(2) This space is permit-parkable
then you can park at this space
R2: If (1) This space is a parking space &
(2) permit sign at this space &
(3) current date is acceptable
then this space is permit parkable
R3: If (1) This space is permit parkable &
(2) Current time is 7am to midnight
then you can park at this space
R4: If (1) Current month is December to March
then current date is acceptable
R5: If (1) Current month is April to November &
(2) Current day of month is ≤ 15
then current day is acceptable
R6: if (1) You live near this space &
(2) You own a car &
(3) You apply for a permit
then you have a parking permit
R7: if (1) You can park at this space
then you can take a car
Logic Programming: Introduction - 10
©1993 John Mylopoulos

11


Principles
csc324
Can you take a car?
R7
Can you park at this space?
R1
Do you have parking permit?
Space is permit parkable?
R2
...parking space?
R6
...live near..
...own car
Permit sign...?
Acceptable date?
get permit?
R4
R5
...Dec-Mar?
...≤15?
...Apr-Nov?
Logic P
©1993 John Mylopoulos

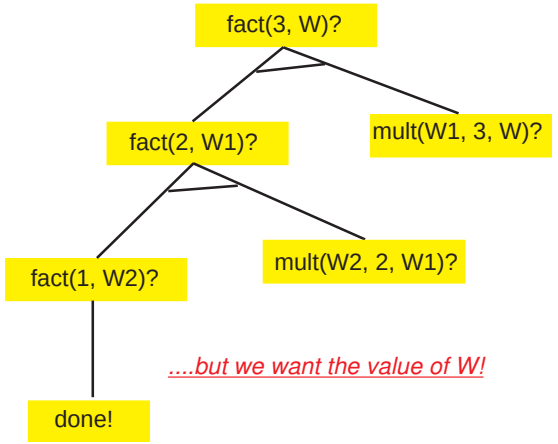
12


Principles of Programming Languages
csc324
Introducing Variables in Proofs
Can you park at X?
R1
Do you have parking permit?
X is permit parkable?
R2
...parking space?
Permit sign...?
Acceptable date?
R4
R5
...Dec-Mar?
...≤15?
...Apr-Nov?
Logic
©1993 John Mylopoulos

13

Another Example

fact(1,1).
if (1) fact(X - 1, N) &
(2) mult(N, X, M)
then fact(X, M)



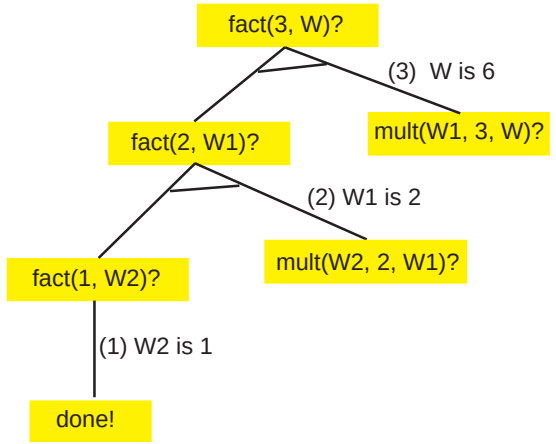
©1993 John Mylopoulos

14

Variable Bindings for the `fact` Example

fact(1,1).
if (1) fact(X - 1, N) &
(2) mult(N, X, M)
then fact(X, M)

fact(1,1).
to compute fact(X, M)
(1) compute fact(X - 1, N) &
(2) compute mult(N, X, M)



©1993 John Mylopoulos

15

PROLOG

- PROgramming in LOgic
  Sound inference engine

- User asserts true statements,
  facts -- father(george, maria)
  rules -- sibling(X, Y) if father(Z, X) & father(Z, Y)

- User poses queries
  sister(maria, myrto)?

- PROLOG provides answer {Y/N, bindings}

How do we "match" a rule or a fact?
How do we keep track of bindings?

©1993 John Mylopoulos