520 — Principles of Programming Languages

13: Prolog II

Christian Collberg
collberg@cs.arizona.edu

Department of Computer Science
University of Arizona

Prolog Structures

- Aka, structured or compound objects
- An object with several components.
- Similar to Pascal's Record-type.
- Used to group things together.

```
functor arguments
   course(prolog, chris, mon, 11)
```

The arity of a functor is the number of arguments.

Structures – Courses

```
lectures(Lecturer, Day) :-
   course(Course, time(Day, _, _), Lecturer, _).

duration(Course, Length) :-
   course(Course, time(Day, Start, Finish), Lec, Loc),
   Length is Finish - Start.

occupied(Room, Day, Time) :-
   course(Course, time(Day, Start, Finish), Lec, Room)
   Start =< Time, Time =< Finish.
```

```
course(c231, time(mon, 4, 5), cc, plt1).
course(c231, time(wed, 10, 11), cc, plt1).
course(c231, time(thu, 4, 5), cc, plt1).
course(c363, time(mon, 11, 12), cc, slt1).
course(c363, time(thu, 11, 12), cc, slt1).

?- occupied(slt1, mon, 11).
yes

?- lectures(cc, mon).
yes
```
**Binary Trees**

```prolog
tree(Element, Left, Right)

tree(s,
    tree(b, void, void),
    tree(x,
        tree(u, void, void),
        void).
```

**Binary Trees – Counting Nodes**

```prolog
size_of_tree(Tree, Size) :-
    size_of_tree(Tree, 0, Size).

size_of_tree(void, Size, Size).

size_of_tree(tree(_, L, R), SizeIn, SizeOut) :-
    Size1 is SizeIn + 1,
    size_of_tree(L, Size1, Size2),
    size_of_tree(R, Size2, SizeOut).
```

**Binary Trees – Size**

<table>
<thead>
<tr>
<th>SizeIn</th>
<th>SizeOut</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

**Binary Search Trees**

```prolog
?- tree_member(K).

tree_member(X, tree(X,_,_)).

tree_member(X, tree(Y,Left,_)) :-
    X < Y,
    tree_member(Y, Left).

tree_member(X, tree(_,_,Right)) :-
    X > Y,
    tree_member(Y, Right).
```
**Binary Trees – Isomorphism**

Tree isomorphism:

![Binary Tree Diagram]

Two binary trees $T_1$ and $T_2$ are *isomorphic* if $T_2$ can be obtained by reordering the branches of the subtrees of $T_1$.

---

**Binary Trees – Tree Substitution**

Replace all occurrences of $X$ by $Y$.

```
subs(X, Y, void, void).
subs(X, Y, tree(X, L1, R1), tree(Y, L2, R2)) :-
    subs(X, Y, L1, L2),
    subs(X, Y, R1, R2).
subs(X, Y, tree(Z, L1, R1), tree(Z, L2, R2)) :-
    X =\= Y, subs(X, Y, L1, L2),
    subs(X, Y, R1, R2).
```

---

**Binary Trees – Isomorphism**

```
tree_iso(void, void).
tree_iso(tree(X, L1, R1), tree(X, L2, R2)) :-
    tree_iso(L1, L2), tree_iso(R1, R2).
tree_iso(tree(X, L1, R1), tree(X, L2, R2)) :-
    tree_iso(L1, R2), tree_iso(R1, L2).
```

1. Check if the roots of the current subtrees are identical;
2. Check if the subtrees are isomorphic;
3. If they are not, backtrack, swap the subtrees, and again check if they are isomorphic.

---

**Binary Trees – Tree Substitution**

```
subs(s, t, tree(s, tree(r, void, void), tree(q, tree(v, void, void)tree(s, tree(z, void, void)void))),
```

```
subs(s,t)
```

---

**Binary Trees – Tree Substitution**

```
t s
\_ /\_ \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```
Symbolic Differentiation

(1) \frac{dc}{dx} = 0
(2) \frac{d}{dx} = 1
(3) \frac{d(U^c)}{dx} = cU^{c-1}\frac{dU}{dx}
(4) \frac{d(-U)}{dx} = -\frac{dU}{dx}
(5) \frac{d(U + V)}{dx} = \frac{dU}{dx} + \frac{dV}{dx}
(6) \frac{d(U - V)}{dx} = \frac{dU}{dx} - \frac{dV}{dx}

Symbolic Differentiation

(7) \frac{d(cU)}{dx} = c\frac{dU}{dx}
(8) \frac{d(UV)}{dx} = U\frac{dV}{dx} + V\frac{dU}{dx}
(9) \frac{d(U^V)}{dx} = \frac{V\frac{dU}{dx} - U\frac{dV}{dx}}{V^2}
(10) \frac{d(\ln U)}{dx} = U^{-1}\frac{dU}{dx}
(11) \frac{d(\sin(U))}{dx} = \frac{dU}{dx}\cos(U)
(12) \frac{d(\cos(U))}{dx} = -\frac{dU}{dx}\sin(U)

Symbolic Differentiation

\text{deriv}(C, X, 0) :- \text{number}(C).
\text{deriv}(X, X, 1).
\text{deriv}(U^C, X, C * U^{L} * DU) :-
\text{number}(C), L \text{ is } C - 1, \text{deriv}(U, X, DU).
\text{deriv}(-U, X, -DU) :- \text{deriv}(U, X, DU).
\text{deriv}(U+V, X, DU + DV) :-
\text{deriv}(U, X, DU),
\text{deriv}(V, X, DV).
Symbolic Differentiation...

\[
\frac{d(U - V)}{dx} = \frac{dU}{dx} - \frac{dV}{dx}
\]
\[
\frac{d(cU)}{dx} = c \frac{dV}{dx}
\]

\[
\text{deriv}(U-V, X, DU - DV) :-
\text{deriv}(U, X, DU),
\text{deriv}(V, X, DV).
\]

\[
\frac{d(UV)}{dx} = U \frac{dV}{dx} + V \frac{dU}{dx}
\]
\[
\frac{d(U/V)}{dx} = \frac{V \frac{dU}{dx} - U \frac{dV}{dx}}{V^2}
\]

\[
\text{deriv}(U*V, X, U*DV + V*DU) :-
\text{deriv}(U, X, DU),
\text{deriv}(V, X, DV).
\]

\[
\text{deriv}(U/V, X, (V*DU - U*DV)/(V*V)) :-
\text{deriv}(U, X, DU),
\text{deriv}(V, X, DV).
\]

Symbolic Differentiation...

\[
\frac{d(ln U)}{dx} = U^{-1} \frac{dU}{dx}
\]
\[
\frac{d(sin(U))}{dx} = \frac{dU}{dx} \cos(U)
\]
\[
\frac{d(cos(U))}{dx} = -\frac{dU}{dx} \sin(U)
\]

\[
\text{deriv}(\log(U), X, DU/U) :- \text{deriv}(U, X, DU).
\]

\[
\text{deriv}(\sin(U), X, DU*\cos(U)) :- \text{deriv}(U, X, DU).
\]

\[
\text{deriv}(\cos(U), X, -DU*\sin(U)) :- \text{deriv}(U, X, DU).
\]

?- deriv(x, x, D).
D = 1

?- deriv(sin(x), x, D).
D = 1*cos(x)

?- deriv(sin(x) + cos(x), x, D).
D = 1*cos(x) + (-1*sin(x))

?- deriv(sin(x) * cos(x), x, D).
D = sin(x) * (-1*sin(x)) + cos(x) * (1*cos(x))

?- deriv(1 / x, x, D).
D = (x*0-1*1) / (x*x)
Symbolic Differentiation...

\[ D = U \cdot DV + V \cdot DU \]

\[ D = \sin(x) \cdot (-1 \cdot \sin(x)) + \cos(x) \cdot 1 \cdot \cos(x) \]

\[ DU = 1 \]
\[ DV = 1 \]

\[ DU = -DV \cdot \sin(x) \]
\[ DV = DU \cdot \cos(x) \]

\[ U = \sin(x) \]
\[ U = x \]

\[ \text{deriv}(U, x, DU) \]
\[ \text{deriv}(V, x, DV) \]

\[ \text{deriv}(\sin(x) \cdot \cos(x), x, D) \]

\[ D = (\sin(x) \cdot 0 - 1 \cdot (1 \cdot \cos(x))) + (\sin(x) \cdot \sin(x)) \]

\[ \text{deriv}(x^3, x, D). \]
\[ D = 1 \cdot 3 \cdot x^2 \]

\[ \text{deriv}(x^3 + x^2 + 1, x, D). \]
\[ D = 1 \cdot 3 \cdot x^2 + 1 \cdot 2 \cdot x^1 + 0 \]

\[ \text{deriv}(3 \cdot x^3, x, D). \]
\[ D = 3 \cdot (1 \cdot 3 \cdot x^2) + x^3 \cdot 0 \]

\[ \text{deriv}(4 \cdot x^3 + 4 \cdot x^2 + x - 1, x, D). \]
\[ D = 4 \cdot (1 \cdot 3 \cdot x^2) + x^3 \cdot 0 + (4 \cdot (1 \cdot 2 \cdot x^1) + x^2 \cdot 0) + 1 - 0 \]

Prolog So Far...

Prolog terms:

- **Atoms**: (a, 1, 3.14)
- **Structures**: guitar(ovation, 1111, 1975)

Infix expressions are abbreviations of “normal” Prolog terms:

<table>
<thead>
<tr>
<th>Infix</th>
<th>Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>a + b</td>
<td>+(a, b)</td>
</tr>
<tr>
<td>a + b* c</td>
<td>+(a, *(b, c))</td>
</tr>
</tbody>
</table>