Logic Programming (PLP 11, CTM 9.1)

Terms, Resolution, Unification, Search, Backtracking (Prolog)

Relational Computation Model (Oz)

Carlos Varela

Rennselaer Polytechnic Institute

November 11, 2016

Prolog Terms

- Constants

```
rpi
troy
```

- Variables

```
University
City
```

- Predicates

```
located_at(rpi,troy)
pair(a, pair(b,c))
```

Can be nested.

Resolution

- To derive new statements, Robinson's resolution principle says that if two Horn clauses:

$$H_1 \Leftarrow B_{11}, B_{12}, \ldots, B_{1m}$$
$$H_2 \Leftarrow B_{21}, B_{22}, \ldots, B_{2n}$$

are such that $\mathbf{H_1}$ matches $\mathbf{B_{2i}}$, then we can replace $\mathbf{B_{2i}}$ with $\mathbf{B_{11}, B_{12}, \ldots, B_{1m}}$:

$$H_2 \Leftarrow B_{21}, B_{22}, \ldots, B_{2(i-1)}, \underline{B_{11}, B_{12}, \ldots, B_{1m}}, B_{2(i+1)} \ldots, B_{2n}$$

- For example:

$$\frac{\begin{array}{l} C \Leftarrow A,B \\ E \Leftarrow C,D \end{array}}{E \Leftarrow A,B,D}$$

Resolution Example

```
father(X,Y) :- parent(X,Y), male(X).
grandfather(X,Y) :- father(X,Z), parent(Z,Y).
```

```
grandfather(X,Y) :-
        parent(X,Z), male(X), parent(Z,Y).
```

`:-` is Prolog's notation (syntax) for $\Leftarrow$.

Unification

- During *resolution*, free variables acquire values through *unification* with expressions in matching terms.
- For example:

```
male(carlos).
parent(carlos, tatiana).
parent(carlos, catalina).
father(X,Y) :- parent(X,Y), male(X).
```

```
father(carlos, tatiana).
father(carlos, catalina).
```

Unification Process

- A constant unifies only with itself.

- Two predicates unify if and only if they have
  - the same *functor*,
  - the same number of *arguments*, and
  - the corresponding arguments *unify*.

- A variable unifies with anything.
  - If the other thing has a *value*, then the variable is *instantiated*.
  - If it is an *uninstantiated variable*, then the two variables are *associated*.

Backtracking

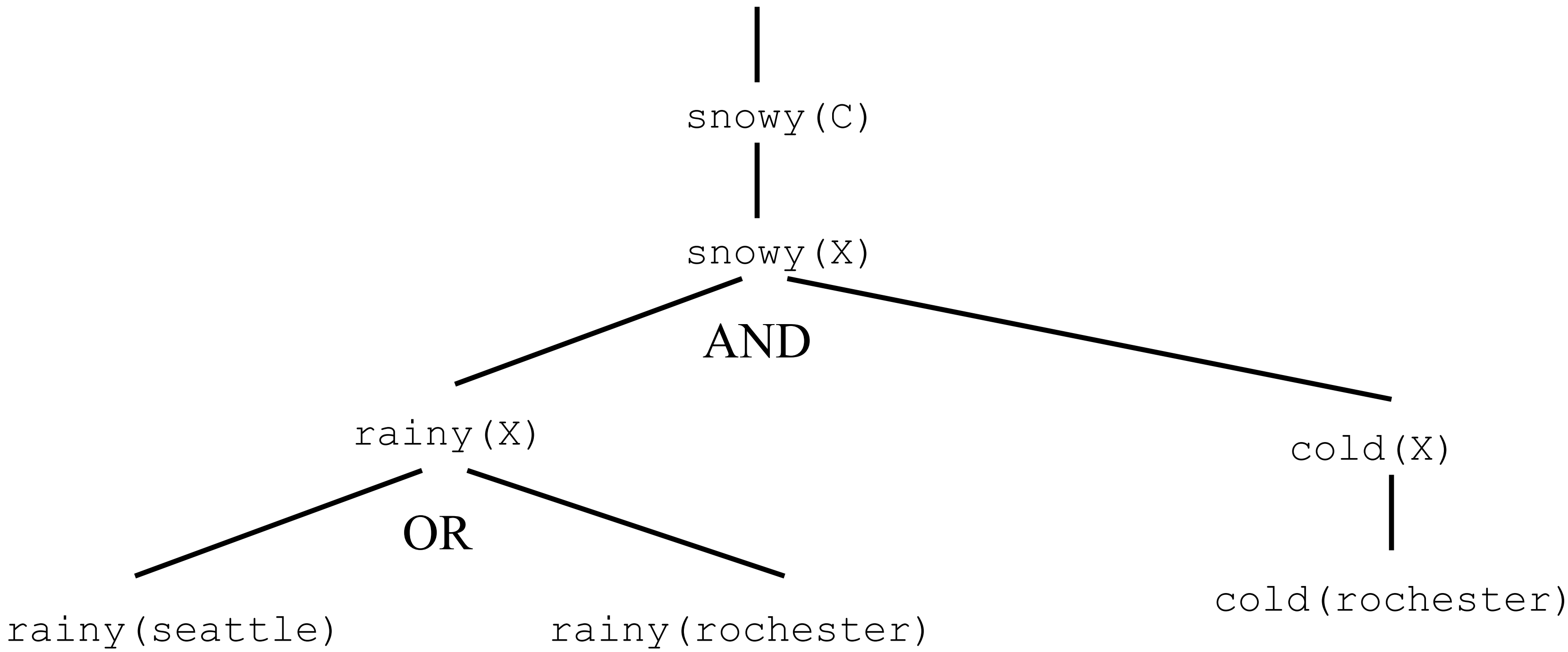
- *Forward chaining* goes from axioms forward into goals.
- *Backward chaining* starts from goals and works backwards to prove them with existing axioms.

Backtracking example

```
rainy(seattle).
rainy(rochester).
cold(rochester).
snowy(X) :- rainy(X), cold(X).
```

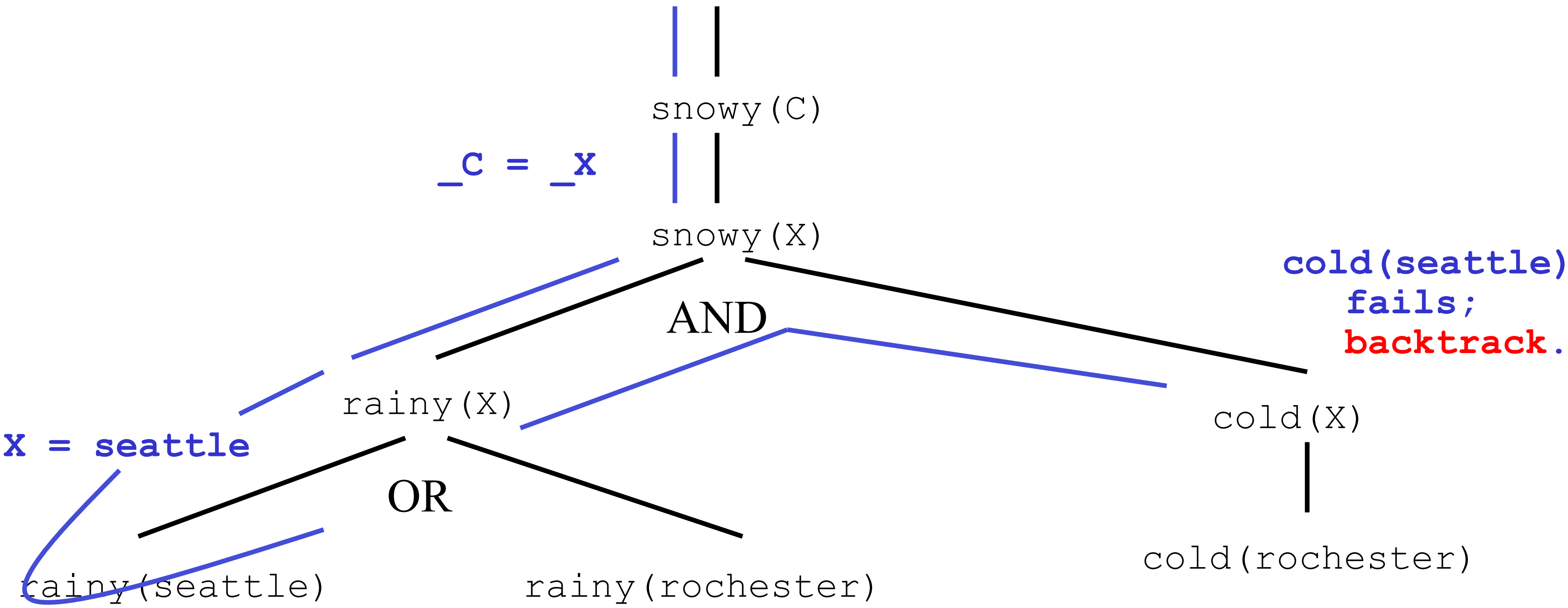
Backtracking example

```
rainy(seattle).
rainy(rochester).
cold(rochester).
snowy(X) :- rainy(X), cold(X).
```



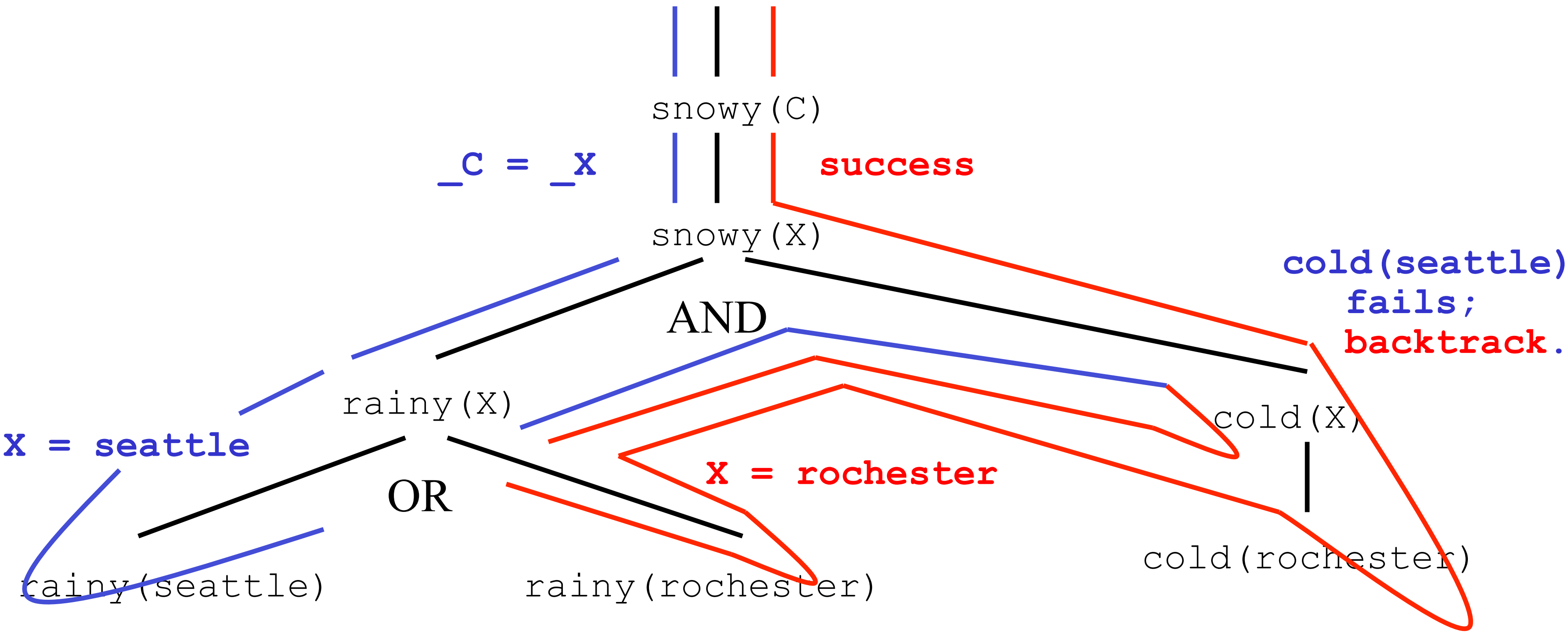
Backtracking example

```
rainy(seattle).
rainy(rochester).
cold(rochester).
snowy(X) :- rainy(X), cold(X).
```



Backtracking example

```
rainy(seattle).
rainy(rochester).
cold(rochester).
snowy(X) :- rainy(X), cold(X).
```



Relational computation model (Oz)

The following defines the syntax of a statement, ⟨s⟩ denotes a statement

⟨s⟩	::=	skip	*empty statement*
	|	⟨x⟩ = ⟨y⟩	*variable-variable binding*
	|	⟨x⟩ = ⟨v⟩	*variable-value binding*
	|	$\langle s_1 \rangle$ $\langle s_2 \rangle$	*sequential composition*
	|	local ⟨x⟩ in $\langle s_1 \rangle$ end	*declaration*
	|	proc {⟨x⟩ $\langle y_1 \rangle$ … $\langle y_n \rangle$ } $\langle s_1 \rangle$ end	*procedure introduction*
	|	if ⟨x⟩ then $\langle s_1 \rangle$ else $\langle s_2 \rangle$ end	*conditional*
	|	{ ⟨x⟩ $\langle y_1 \rangle$ … $\langle y_n \rangle$ }	*procedure application*
	|	case ⟨x⟩ of ⟨pattern⟩ then $\langle s_1 \rangle$ else $\langle s_2 \rangle$ end	*pattern matching*
	|	**choice $\langle s_1 \rangle$ [] … [] $\langle s_n \rangle$ end**	***choice***
	|	**fail**	***failure***

Relational Computation Model

- Declarative model (purely functional) is extended with *relations*.
- The choice statement groups a set of alternatives.
  - Execution of choice statement chooses one alternative.
  - Semantics is to rollback and try other alternatives if a failure is subsequently encountered.
- The fail statement indicates that the current alternative is wrong.
  - A fail is implicit upon trying to bind incompatible values, e.g., 3=4. This is in contrast to raising an exception (as in the declarative model).

Search tree and procedure

- The search tree is produced by creating a new branch at each *choice point*.
- When fail is executed, execution « backs up » or backtracks to the most recent choice statement, which picks the next alternative (left to right).
- Each path in the tree can correspond to no solution (« fail »), or to a solution (« succeed »).
- A search procedure returns a lazy list of all solutions, ordered according to a depth-first search strategy.

Rainy/Snowy Example

```
fun {Rainy}
   choice
     seattle [] rochester
   end
end

fun {Cold}
   rochester
end

proc {Snowy X}
   {Rainy X}
   {Cold X}
end
```

```
{Browse
   {Search.base.all
        proc {$ C} {Rainy C} end}}


{Browse {Search.base.all Snowy}}
```

Exercises

76. Download SWI Prolog and Mozart 1.4.0 and install them in your laptop.
77. Execute the "`snowy(City)`" example. In Prolog, use "tracing" to follow backtracking step by step.
78. Create a knowledge base with facts about your family members using predicates and constants. Create rules using variables to define the following relationships: `brother, sister, uncle, aunt, nephew, niece, grandfather, grandmother,`etc. Query your Prolog/Oz program for family relationships.