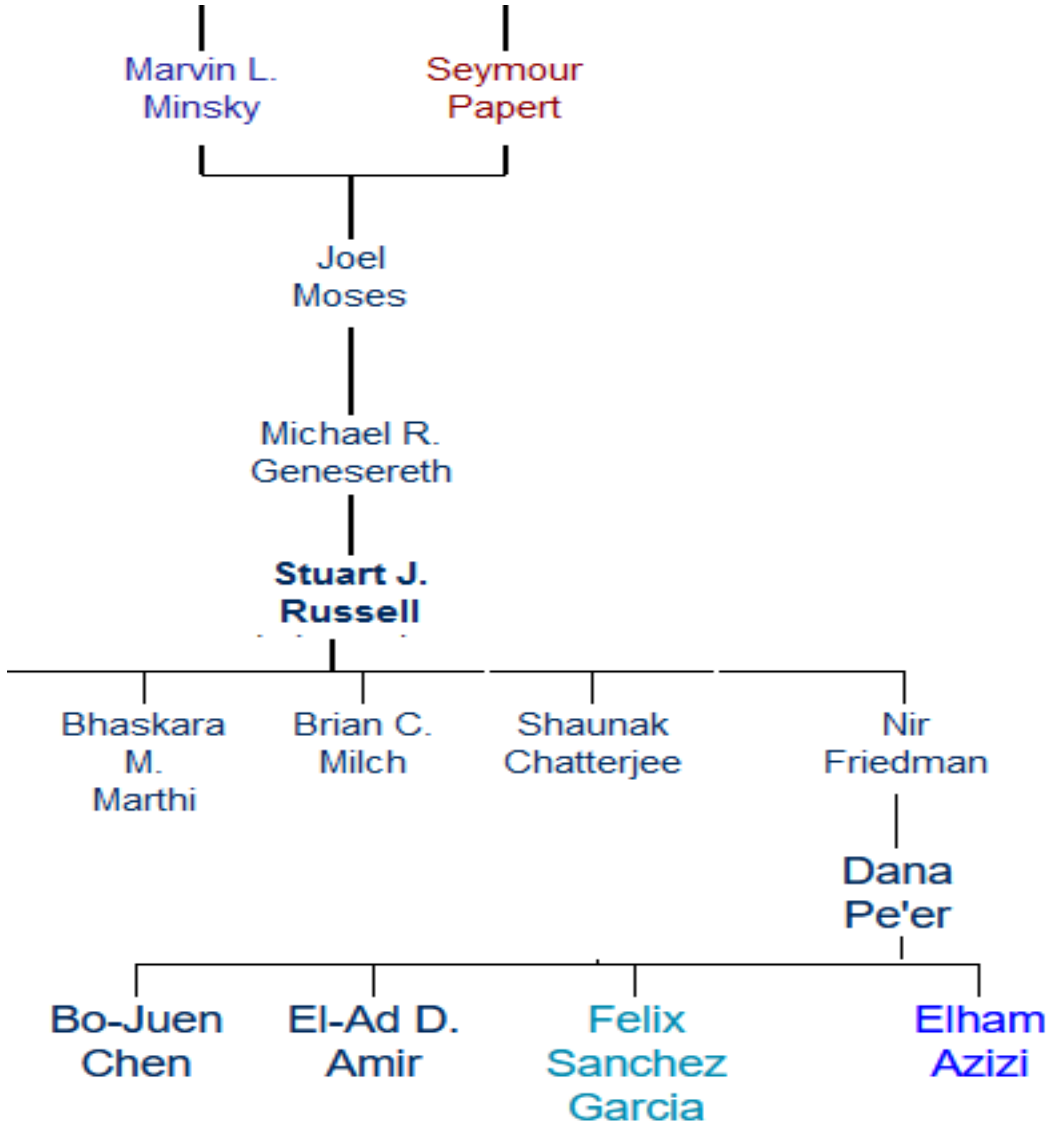
Logic Programming: Prolog

10/02/2025

Koustav Rudra

Prolog Facts



- advisor(minsky, moses).
- advisor(papert, moses).
- advisor(moses, genesereth).
- advisor(genesereth, russell).
- advisor(russell, bhaskara).
- advisor(russell, milch).
- advisor(russell, shaunak).
- advisor(russell, friedman).
- advisor(friedman, dana).
- advisor(dana, felix).
- advisor(dana, chen).
- advisor(dana, amir).
- advisor(dana, azizi).

- male(felix).
- female(dana).

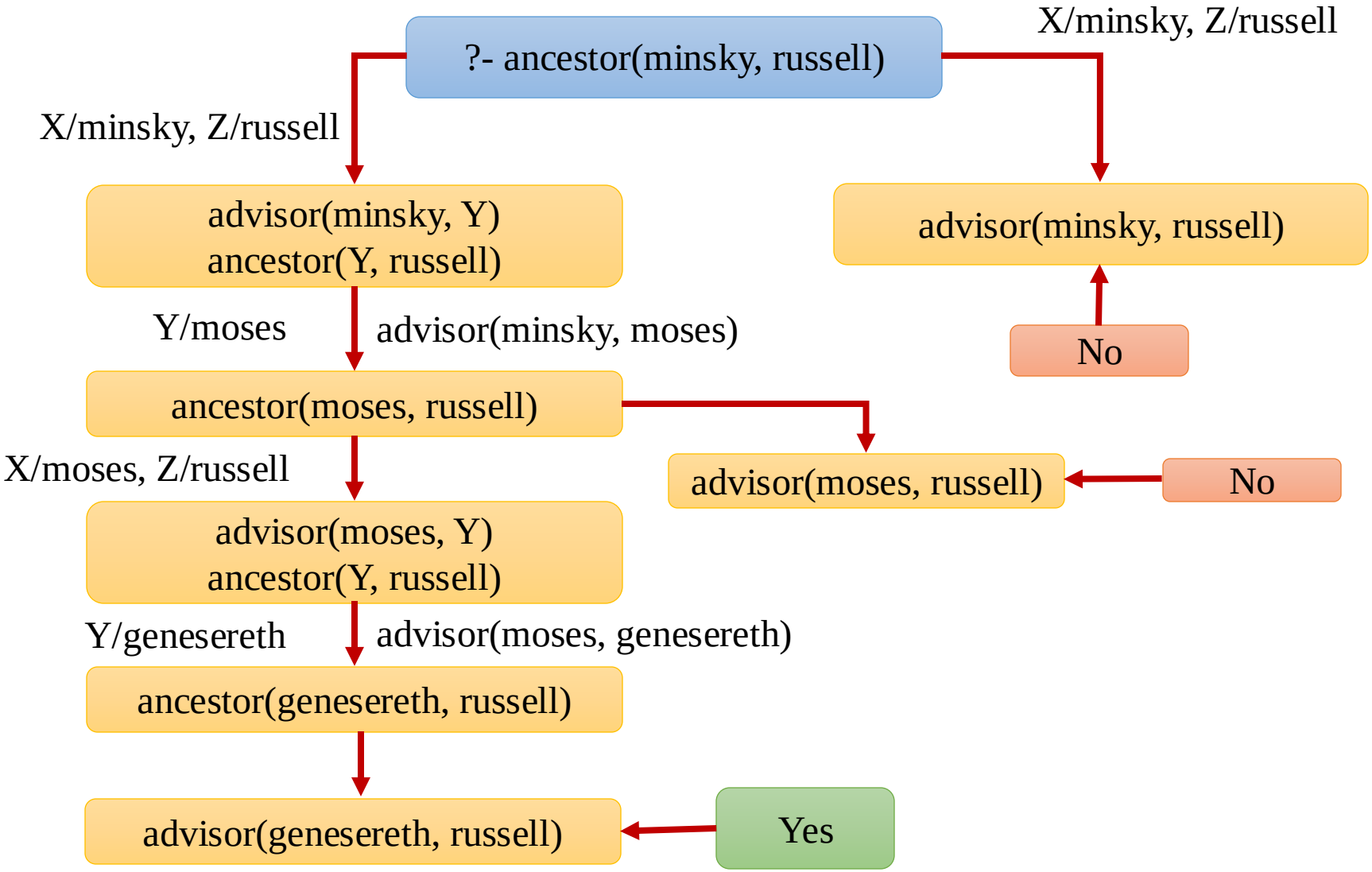
How Prolog answers?

ancestor(X, Z) :- advisor(X, Z)

ancestor(X, Z) :- advisor(X, Y),
ancestor(Y, Z)

?- ancestor(minsky, russell)

advisor(minsky, moses).
advisor(papert, moses).
advisor(moses, genesereth).
advisor(genesereth, russell).



Reordering of Clauses

Original

```
ancestor1(X,Z) :-
advisor(X, Z).

ancestor1(X,Z) :-
advisor(X, Y),
ancestor1(Y, Z).
```

Clause swap

```
ancestor2(X,Z) :-
advisor(X, Y),
ancestor2(Y, Z).

ancestor2(X,Z) :-
advisor(X, Z).
```

Goal swap

```
ancestor3(X,Z) :-
advisor(X, Z).

ancestor3(X,Z) :-
ancestor3(Y, Z),
advisor(X, Y).
```

Clause and Goal swap

```
ancestor4(X,Z) :-
ancestor4(Y, Z),
advisor(X, Y).

ancestor4(X,Z) :-
advisor(X, Z).
```

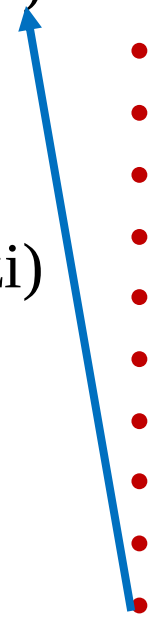
Reordering of Clauses: Original

```
ancestor1(X,Z) :-
advisor(X, Z).

ancestor1(X,Z) :-
advisor(X, Y),
ancestor1(Y, Z).
```

Original

- Call ancestor1(dana, azizi)
- Call advisor(dana, azizi)
- Exit advisor(dana, azizi)
- Exit ancestor1(dana, azizi)



- advisor(minsky, moses).
- advisor(papert, moses).
- advisor(moses, genesereth).
- advisor(genesereth, russell).
- advisor(russell, bhaskara).
- advisor(russell, milch).
- advisor(russell, shaunak).
- advisor(russell, friedman).
- advisor(friedman, dana).
- advisor(dana, felix).
- advisor(dana, chen).
- advisor(dana, amir).
- advisor(dana, azizi).

- male(felix).
- female(dana).

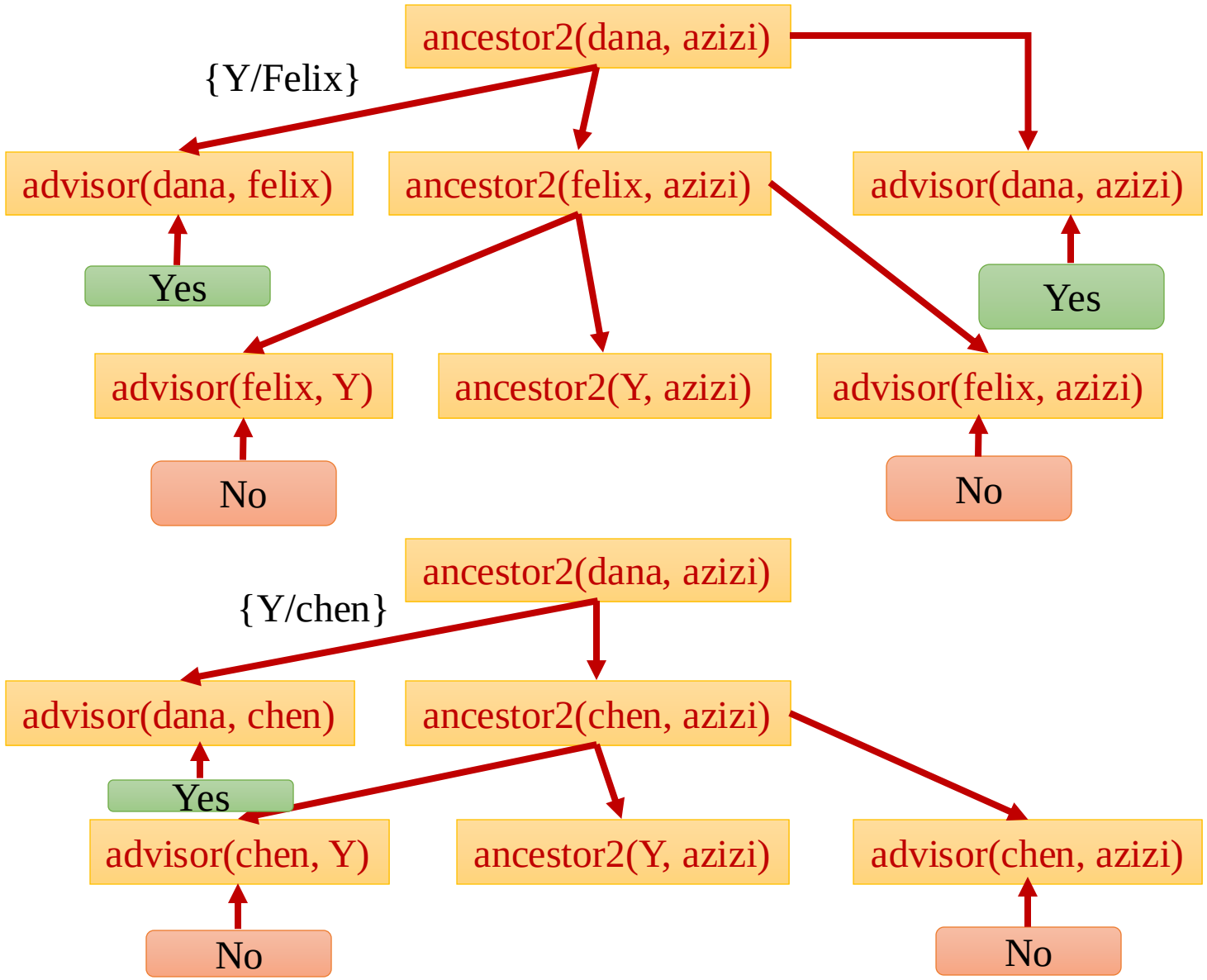
Reordering of Clauses: Clause Swap

Clause swap

```
ancestor2(X,Z) :-
advisor(X, Y),
ancestor2(Y, Z).

ancestor2(X,Z) :-
advisor(X, Z).
```

- advisor(dana, felix).
- advisor(dana, chen).
- advisor(dana, amir).
- advisor(dana, azizi).



Reordering of Clauses: Goal Swap

```
ancestor3(X,Z) :-
advisor(X, Z).

ancestor3(X,Z) :-
ancestor3(Y, Z),
advisor(X, Y).
```

```
ancestor4(X,Z) :-
ancestor4(Y, Z),
advisor(X, Y).

ancestor4(X,Z) :-
advisor(X, Z).
```

- `?- ancestor3(bhaskara, felix)`
- Infinite Loop

Takeaways from Ordering

- Try simplest idea first (practical heuristics in problem solving)
  - ancestor1 being the simplest, ancestor4 being the most complex
- Check your clause ordering to avoid infinite recursion
- Procedural aspect is also important along with declarative

Family Tree Example

- offspring(Y, X) :- parent(X, Y)
- mother(X, Y) :- parent(X, Y), female(X)
- grandparent(X, Z) :- parent(X, Y), parent(Y, Z)
- sister(X, Y) :- parent(Z, X), parent(Z, Y), female(X), different(X, Y)
- predecessor(X, Z) :- parent(X, Z)
- predecessor(X, Z) :- parent(X, Y), predecessor(Y, Z)

Prolog: Data Structure

- Lists
  - Lists of anything, symbolic lists
- Lists can be written as:
  - [item1, item2, …]
  - [Head|Tail]
    - Head is the first element in the list, remaining is the tail (list)
  - [item1, item2, …|Others]
    - Head consists of several items, followed by the tail which is other items [list]
- [a, b, c] = [a|[b,c]] = [a,b|[c]] = [a,b,c|[]]

- Items can be list as well
  - [[a,b], c, [d, [e,f]]]
  - The head of the above list is list [a,b]

Thank You