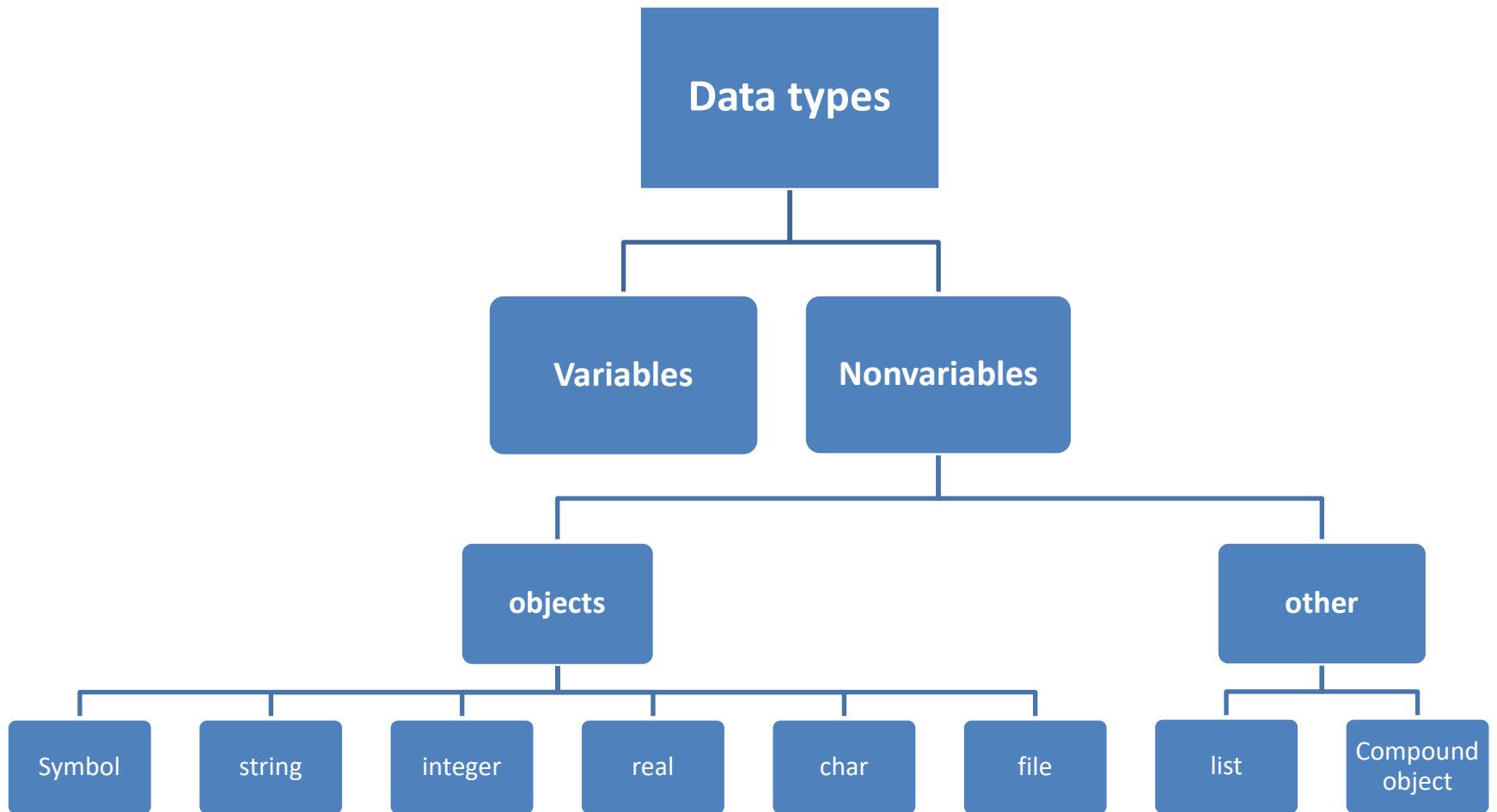


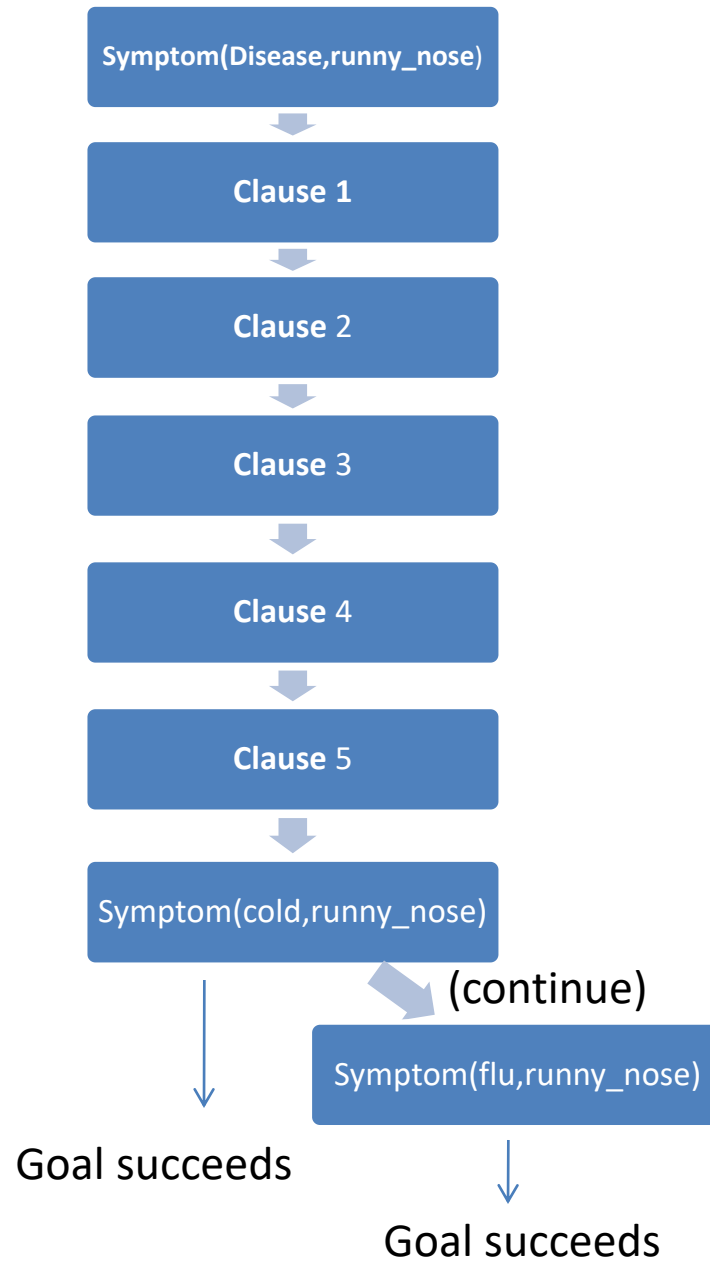
Prolog variables

- Variables can be used in clauses or goal to specify an unknown quantity
- Variable name must
 - begin with uppercase letter
 - may be from 1 to 250 characters long
 - Remaining characters may be uppercase, lowercase, digits or underlining characters.
 - Name should be meaningful
 - Ex.
 - Age
 - Disease
 - Patient
 - John_Smith



- Goal: symptom(Disease,runny_nose)
- Disease is a variable
- So the program respond
 - Goal: symptom(Disease,runny_nose)
Disease=cold
Disease=flu
2 solutions
Goal:

Execution Flow of Prolog Program



Bound and Free variable

- If a variable has a value at a particular time, it is said to be **bound or instantiated**.
- If a variable does not have a value at a particular time, it is said to be **free or uninstantiated**.
- In example disease is a Free variable
- Disease become bound to cold after goal succeeds
- For a goal succeed all the variables in goal must become bound

Anonymous Variables

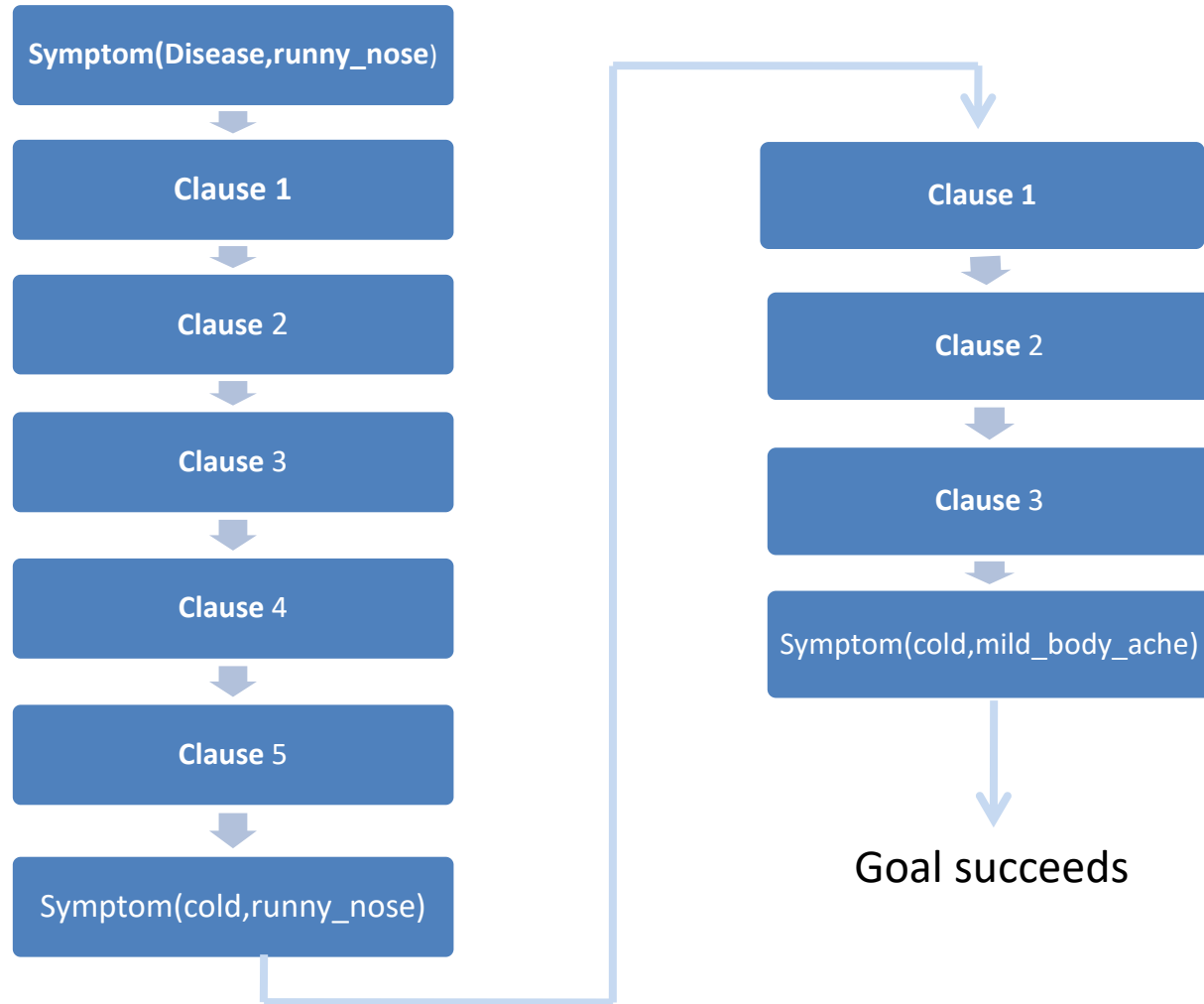
- value of one or more arguments can be ignored when determining a goal's failure or success.
- This is accomplished to express the argument as a underline
- Ex.
 - Goal: symptom(_,chills)
 - True
 - Goal:
 - here there is no variable and there is no binding

- Underscore is also used in a clause to express fact that the clause is true for all argument values
- Ex. likes(ram,_).
 - Means that ram likes anything
- Goal: likes(ram,cricket) is succeed
- In a clause the anonymous variable stands for all values
- In a goal it is specified if at least one value corresponds to it

Compound goals

- Ex. Patient has runny_nose and mild_body_ache this can be expressed as
- **Goal:symptom(Disease,mild_body_ache) and symptom(Disease,runny_nose)**
 - Prolog returns
Disease=cold
1 solution
Goal:
- Goal is compound all specified conditions must succeed for goal to succeed.
- Works from left to right in proving compound goal

Execution with a compound goal



Backtracking

- If any condition in the chain fails ,prolog backtrack the previous condition ,tries to prove it again with another variable binding ,and then moves forward again to see if failed condition will succeed with new binding.
- Prolog moves relentlessly forward and backward through the condition ,trying every available binding in an attempt to get the goal to succeed in as many ways as possible

Variables rules

- Variable name must
 - begin with uppercase letter
 - may be from 1 to 250 characters long
 - Remaining characters may be uppercase, lowercase, digits or underlining characters.
- It is either bound or free at any time
- Binding applies only to a specific clause. Variable declare in another clause consider as a different variable
- In goals ,including compound goals ,same binding applies whether the same variable name is used.
- Once a variable is bound to an object, it is typed the same as the object