s1 galaxy htc salsa

s2 galaxy
s-14000
http://www.desitvforum.net/forum/bigg-boss-season-5/172864-bigg-boss-season-5-episode-1-2nd-october-2011-video-watch-online.html

forward reasonig
Forward chaining is one of the two main methods of reasoning when using inference rules (in artificial intelligence) and can be described logically as repeated application of modus ponens. Forward chaining is a popular implementation strategy for expert systems, business and production rule systems. The opposite of forward chaining is backward chaining.

Forward chaining starts with the available data and uses inference rules to extract more data (from an end user for example) until a goal is reached. An inference engine using forward chaining searches the inference rules until it finds one where the antecedent (If clause) is known to be true. When found it can conclude, or infer, the consequent (Then clause), resulting in the addition of new information to its data.

Inference engines will iterate through this process until a goal is reached.

For example, suppose that the goal is to conclude the color of a pet named Fritz, given that he croaks and eats flies, and that the rule base contains the following four rules:

1. If X croaks and eats flies - Then X is a frog
2. If X chirps and sings - Then X is a canary
3. If X is a frog - Then X is green
4. If X is a canary - Then X is yellow

Backward chaining (or backward reasoning) is an inference method that can be described (in lay terms) as working backward from the goal(s).
It is used in automated theorem provers, proof assistants and other artificial intelligence applications, but it has also been observed in primates.

In game theory, its application to (simpler) subgames in order to find a solution to the game is is called backward induction. In chess, it's called retrograde analysis, and it is used to generate tablebases for chess endgames for computer chess.

Backward chaining is implemented in logic programming by SLD resolution. Both rules are based on the modus ponens inference rule. It is one of the two most commonly used methods of reasoning with inference rules and logical implications
the other is forward chaining. Backward chaining systems usually employ a depth-first search strategy,

Farword Versus Backword Reasoning.
FORWARD VERSUS BACKWARD REASONING

(Search Direction)

A search procedure must find a path between initial and goal states. There are t
wo directions in which a search process could proceed.

(1) Reason forward from the initial states: Being form the root of the search tr
ee. General the next level of the tree by finding all the rules whose left sides
match the root node, and use their right sides to generate the siblings. Repeat
the process until a configuration that matches the goal state is generated.

(2) Reason forward from the goal state(s): Begin building a search tree starting
with the goal configuration(s) at the root. Generate the next level of the tree
by finding all the rules whose right sides match with the root node. Use the le
ft sides of the rules to generate the new nodes. Continue until a node that matc
hes the start state is generated. This method of chaining backward from the desi
red final state is called goal directed reasoning or back tracing.

Selection of forward reasoning or backward reasoning depends on which direction
offers less branching factor and justifies its reasoning process to the user. Mo
st of the search techniques can be used to search either forward or backward. On
e exception is the means-ends analysis technique which proceeds by reducing diff
erences between current and goal states, sometimes reasoning forward and sometim
es backward.

The following are the factors which determine the choice of direction for a part
icular problem.

1. Are there more possible start states on goal states? We would like to move fr
om the smaller set of states to the larger set of states.

2. In which direction is the branching factor (that is, their average number of
nodes that can be reached directly from a single node) greater ? we would lime t
o proceed in the direction with the lower branching factor.

3. Will the program be asked to justify its reasoning process to a user ? If so,
it is important to proceed in the direction that corresponds more closely with
the way the user will think.

4. What kind of event is going to trigger a problem-solving episode? If it is
the arrival of a new factor, forward reasoning makes sends. If it is a query
to which a response is desired, backward reasoning is more natural.

Backward chaining (a la Prolog) is more like finding what initial conditions for
m a path to your goal. At a very basic level it is a backward search from your g
oal to
find conditions that will fulfil it.

Backward chaining is used for interrogative applications (finding items that ful
fil certain criteria) - one commercial example of a backward chaining applicatio
n might
be finding which insurance policies are covered by a particular reinsurance con
tract.

Forward chaining (a la CLIPS) matches conditions and then generates inferences f

rom those conditions. These conditions can in turn match other rules. Basically, this takes a set of initial conditions and then draws all inferences it can from those conditions.

The inferences (if asserted) can also be actions or events that can trigger exte rnal actions. This is useful in event driven systems, as the rule sets can be configured to (for example) initiate a workflow or some other action. This type of rule engine is the most commonly used in commercial applications.

Event driven systems are a common application of forward chaining rule engines. One example of a forward chaining application might be a telecoms plan provisioning engine (typically used for administering mobile phone plans). Enter ing a particular user with a particular plan will trigger a range of items to be

set up in various phone switches, billing systems, financials, CRM systems etc. s