FORWARD and BACKWARD CHAINING

A question which arises once we have built a knowledge base and we know an inference rule to make inferences from it concerns the direction of inference.

Forward Chaining: returns a substitution or *false*

Precond: KB and a query *q*

Uses local variable: *new*

repeat until new is empty

new ← {}

given the KB apply one inference step: **p'**

if **p' "equiv to" q**, stop

else augment KB by **p'**

return ***false***

The crime example:

The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American. Therefore West is a criminal.

Use the following notation:

1. a(x) : "x is American"
2. w(x) : "x is a weapon"
3. s(x,y,z) : "x sells y to z"
4. h(x) : "x is hostile to America"
5. c(x) : "x is a criminal"
6. e(x,y) : "x is an enemy to y"
7. m(x) : "x is a missile"
8. owns(x,y) : "x owns y"

objects/constants: America, West, Nono, mi (mi stands for a particular missile or collection of missiles)

^: and

- : implication

With this notation the above text becomes:

1. **a(x) ^ w(y) ^ s(x,y,z) ^ h(z) → c(x)** : "*....it is a crime for an American to sell weapons to hostile nations*"
2. x, o(Nono,x) ^ m(x) : "*...country Nono has some missiles*"

   Replace this by the following two definite clauses
   a) o(Nono,mi)
   b) m(mi)

3. e(Nono, America) :" *... The country Nono, an enemy of America...*"
4. **m(x) → w(x)** additional knowledge
5. **e(x, America) → h(x)** additional knowledge: a country enemy of America is hostile to America
6. **m(x) ^ o(Nono,x) →s(West, x, Nono)** "*...all of its missiles were sold to it by Colonel West,...*"
7. a(West) "*...who is American..*"

INFERENCE (FORWARD-CHAINING)

I. Iteration I

- Make substitution **{x/mi} in rule** 6 which becomes
  m(mi) ^ o(Nono,mi) →s(West, mi, Nono); with a) and b) above we infer **s(West, mi, Nono)**
  we add

8. **s(West, mi, Nono)**
   to the knowledge base

- Make substitution **{x/mi} in rule** 4 which becomes
  m(mi) → w(mi); with b) and Modus Ponens infer **w(mi)**
  add

9. **w(mi)**

to KB

- Make substitution **{x/Nono} in rule 5** which becomes
  e(Nono, America) → h(Nono); with clause 3 we infer **h(Nono)**
  Add

10. **h(Nono)**
to KB

II. Iteration II

Make substitution {x/West, y/mi, z/Nono} in rule 1 which becomes

11. **a(West) ^ w(mi) ^ s(West, mi, Nono) ^ h(Nono) → c(West)**

From 11 with **8, 9, 10** (conjunction introduction + Modus Ponens) infer **c(West)** which is added to KB and is the desired conclusion.

It is easy to see that given a KB forward chaining may find many other conclusions before hitting on the desired one.
It is mostly used when given a KB we add a fact and want to find all the consequences of this addition.
In it Modus Ponens is used in the forward way: that is from premises to conclusion.

Backward Chaining

Conclusions are generated in a much more controlled manner: given a desired conclusion we search the knowledge base for one (or more) implication sentences that can be used to prove it. This generates other, intermediate conclusions (these are the premises of the implication sentence) to be proved. Modus ponens is used here in a **backward way**, that is, **from conclusion** such that the premises necessary to prove it are generated, etc.

Examples:

The given KB is:

1.1 h(x) → m(x)

1.2 c(x) → m(x)

1.3 p(x) → m(x)

2. h(y) ∧ o(x,y) → h(x)

3. h(Bluebeard)

4. pa(Bluebeard, Charlie)

5. pa(x, y) → o(y, x)

6. m(x) → pa(y, x)

Let us suppose that we add the fact that "Oink is a pig", that is

7. pig(Oink)
is added as a clause

What are the effects of adding this?

FORWARD CHAINING:

From **1.3, 7** with the substitution **{x/Oink}** and MP (or, we can say due to GMP) it follows:
m(Oink), that is "*Oink is a mammal"*.

We add this to the knowledge base to obtain

8. m(Oink)

Now from **6, 8,** with the substitution **{x/Oink}** and MP, (or GMP) it follows
pa(y, Oink)
that is, that "*Oink has a parent*". We add this to the knowledge base

9. pa(y, Oink)

Finally, from **9, 5,** after standarzing apart these sentences

pa(y1, Oink)
pa(x, y2) → o(y2, x)

with the substitution **{ x/y1, y2/Oink}** we can infer

10. o(Oink, y1)

that is, that "*Oink is someone's offspring*".

BACKWARD CHAINING:

Let us now see the backward chaining proof for **o(Oink, y)**

Find an implication with conclusions **o(_ , _)** : **pa(x, y) → o(y, x)**

Note the conflicting use of y, so standardize apart the two clauses (sentences):

o(Oink, y1)

pa(x, y2) → o(y2, x)

With the substitution **{y2/Oink, x/y1}**

we generate the new desired conclusion

pa(y1, Oink)

Look for an implication with conclusion **pa(_ , _) : m(x) → pa(y, x)**

With the substitution **{ y/y1, x/Oink}**

we generate the new desired conclusion

m(Oink)

Look for an implication with conclusion **m(_) : p(x) → m(x)**

(I skept here the steps of selecting h(x) → m(x) , c(x) → m(x), which would fail)

With the substitution **{x/Oink}**
we generate the new desired conclusion

p(Oink)

Look for an implication with conclusion **p(_):**

with the substitution **{Oink/Oink}** select **p(Oink)**

(which is really the implication **True → p(Oink)**)