Lecture 8

Revision of Forward Chaining (from lecture 7)

- You start with facts that you know are true.
- You combine the facts in **logical fashion** to infer new correct statements.

Oggie says that this is really not very often how things work in reality. He says you can have intuitions about some really quite abstract things, you normally come up with a theorem before you really know with certainty that its true (before proving it).

The bad thing about forward chaining is that you are proving a lot of things that you may not be interested in.

Forward chaining is:

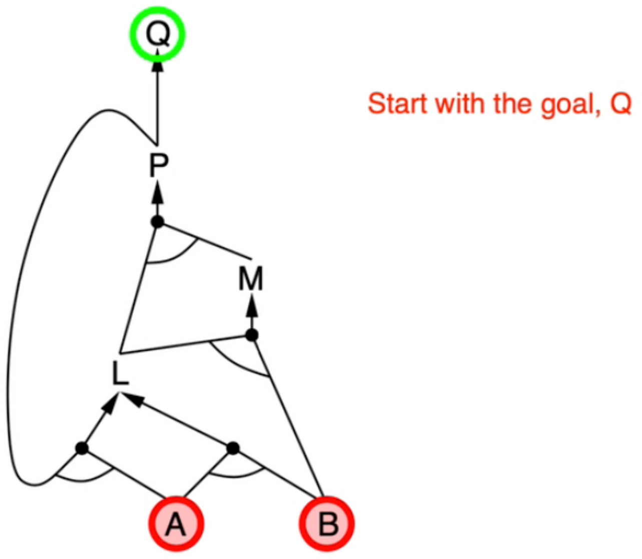
- sound (complete)
- linear in the size of the knowledge base
- Similar to human reasoning (Oggie debates that)
- Not very goal directed. Not driven by some sort of specific question.

Backward Chaining

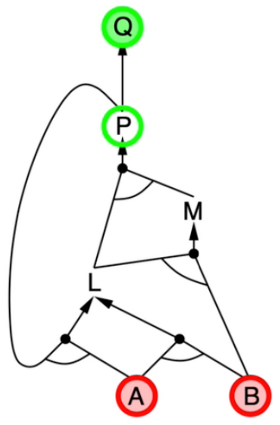
There is a converse algorithm called Backward Chaining, which tries to prove a certain query. You start from the query, look at what the conditions for it being true are, look at whether you can prove them and so on until you go again to the facts.

How to do Backward Chaining

1. Start with a query - *Q*

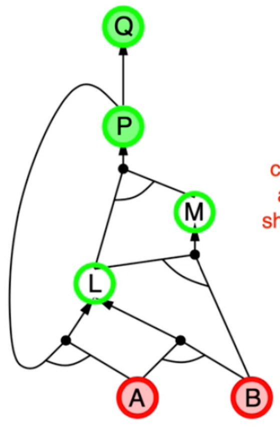


2. You can see that *Q* is entailed by *P*. You have to prove *P*.



If we can show that
P holds, then Q
must also hold

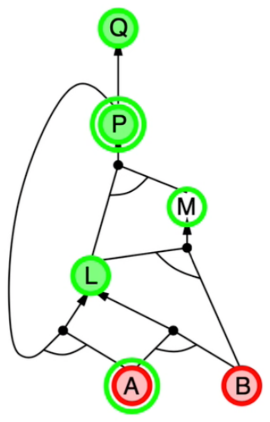
3. Is *P* true? Then you can see that *P* is a conjunction of *M* & *L*. Therefore you need to show that these are also true (both of them).



P is implied by
conjuctions involving L
and M so we need to
show that these are true

You keep going down this way.

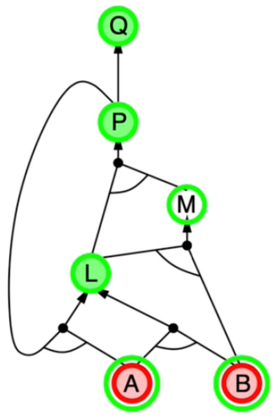
4. *L* is implied by the conjuction of *A* and *B.*



L is implied by the
conjunction A and B

Try A first

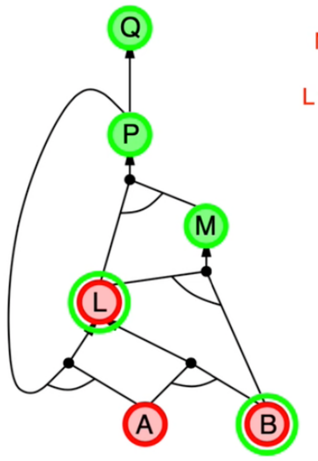
5. You know that *L* is true because *A* & *B* are facts (given to be True by the problem/spec).



Chain L to B

We now have L true, but still need to consider M

6. Go backwards to the tree and try show M is true. M is implied by L conjunction B. L was shown to be true in previous step. B is a fact. Therefore, M is True.



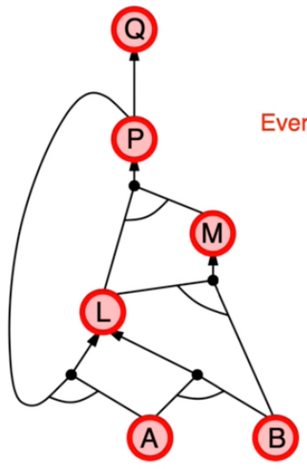
M is implied by L ∧ B

L was shown to be true earlier in BC

B is a fact

So M is true

7. Go backwards again, you have proven everything you needed for P to be True. Therefore, P is true, and Q is also True.



Everything in this example is now a known fact

Backward Chaining is a lot more goal directed. You start from the goal and work backwards from it. Complexity of BC is much lower than linear in size of KB. This is because that depends very much on the structure of your knowledge base. The relation between different facts and different proposition in your knowledge base.

Resolution

Conjuctive normal form – a very convenient way of expressing knowledge. Its conjuctive of normal forms where this normal forms are disjucnctions of literals.

You basically bracket things in the form:
(something or not_something) AND (something or not_something or not_something)

E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$.

You have a simple resolution inference.

$$\frac{P_{1,3} \vee P_{2,2}, \qquad \neg P_{2,2}}{P_{1,3}}$$

Imagine that you have a disjunction of two propositions P1,3 and P2,2 and then imagine that you learn that NOT P2,2 is true. This creates a clash there.

If you know that P2,2 is not true, then you know that P1,3 must be true (if the statement holds).

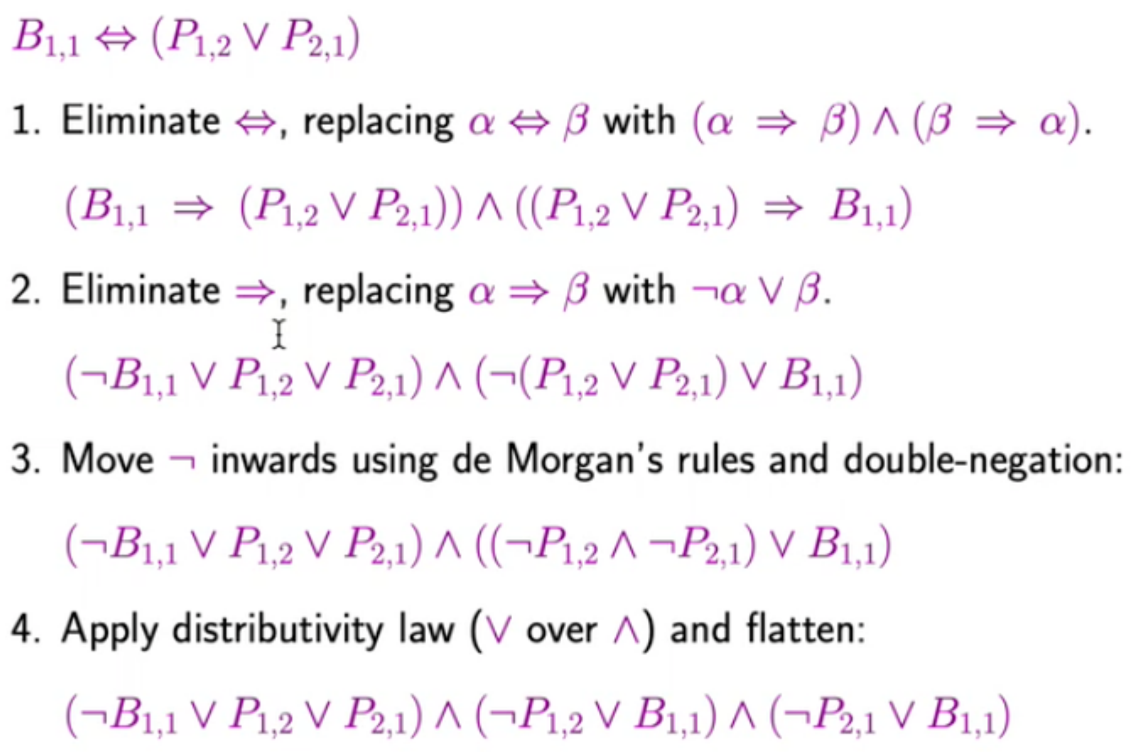
The proposition under the line is the one deduced via the resolution.

Conversion to CNF

How to end up with a conjunctive normal form? There are some very simple rules.

Remember (as he said in previous lecture) that you can basically eliminate the implication and bidirectional implications entirely. You don't need them, you just use them for convenience. They can be experessed using just conjunctions, disjunctions and negatives.

He shows the example below.



$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.

   $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.

   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

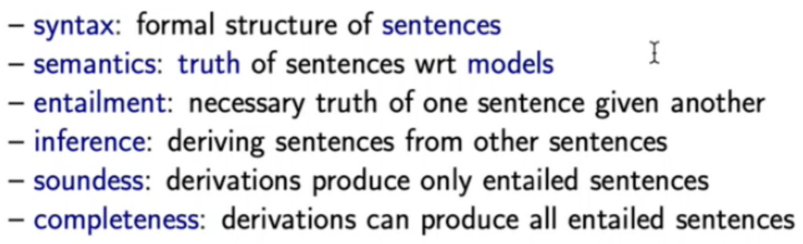
3. Move $\neg$ inwards using de Morgan's rules and double-negation:

   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

4. Apply distributivity law ($\vee$ over $\wedge$) and flatten:

   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

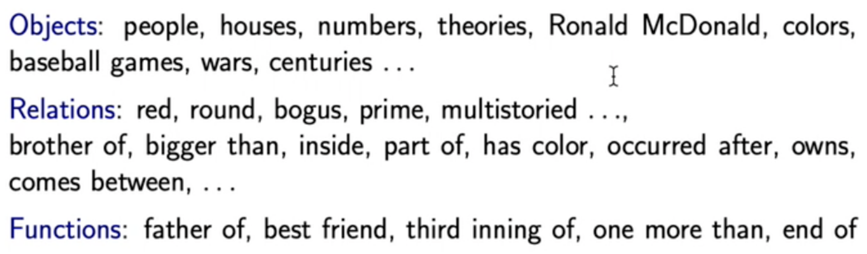
Oggie says that we should be familiar with these terms:



– syntax: formal structure of sentences
– semantics: truth of sentences wrt models
– entailment: necessary truth of one sentence given another
– inference: deriving sentences from other sentences
– soundess: derivations produce only entailed sentences
– completeness: derivations can produce all entailed sentences

First-order Logic

More powerful. First-order Logic (FOL) assumes that there are some **objects** and **relations** over those objects, as well as some **functions**.



Objects: people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries . . .

Relations: red, round, bogus, prime, multistoried . . .,
brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, . . .

Functions: father of, best friend, third inning of, one more than, end of

In FOL you have constants, predicates, functions, variables, connectives, equality and quantifiers.

Constants	$KingJohn,\ 2,\ UCB, \ldots$
Predicates	$Brother,\ >, \ldots$
Functions	$Sqrt,\ LeftLegOf, \ldots$
Variables	$x,\ y,\ a,\ b, \ldots$
Connectives	$\wedge \quad \vee \quad \neg \quad \Rightarrow \quad \Leftrightarrow$
Equality	$=$
Quantifiers	$\forall \exists$

Quantifiers are important. They are used to say

1. Something applies to all - ∀
2. Something exists - ∃

Atomic Sentences

Atomic sentence can be predicates that apply to certain terms, expressed in some kind of relationship between these terms e.g.:

$$predicate(term_1, \ldots, term_n)$$

Or it can be equality between terms e.g.:

$$term_1 = term_2$$

A term can be some kind of function of other terms, or a constant, or a variable.

$$\text{Term} = function(term_1, \ldots, term_n) \text{ or } constant \text{ or } variable$$

E.g., $Brother(KingJohn, RichardTheLionheart)$

$> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))$

Complex sentences

Can be made from atomic sentences using connectives.

$$\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2$$

E.g. $Sibling(KingJohn, Richard) \Rightarrow Sibling(Richard, KingJohn)$

$>(1,2) \vee \leq(1,2)$

$>(1,2) \wedge \neg >(1,2)$

Truth in FOL

Some atomic sentence where you have some predicate or some terms is true if and only if the objects refered to by these terms is in relation with the object which is referred to by the predicate.

This is basically saying, a predicate expresses a certain kind of relationship and that sentence is true if and only if these objects are in this kind of relationship that you are tying to express using a predicate.

Consider the interpretation in which
$Richard \rightarrow$ Richard the Lionheart
$John \rightarrow$ the evil King John
$Brother \rightarrow$ the brotherhood relation

Under this interpretation, $Brother(Richard, John)$ is true just in case Richard the Lionheart and the evil King John are in the brotherhood relation in the model

Brother(Richard, John) is true because these two guys are in brotherhood relationship.

Universal Quantification - ∀

Forall.

How to read it:

$$\forall x \;\; At(x, Berkeley) \;\Rightarrow\; Smart(x)$$

For every *x* (whatever value *x* can take), function at determines if person *x* is at *Berkeley* that implies that the person *x* is smart.

A common mistake to avoid:
Using ∧ as the main connective with ∀. This ends up saying a completely different thing.

$$\forall x \;\; At(x, Berkeley) \land Smart(x)$$

Everyone is at Berkeley AND everyone is smart.

Remember to use => with ∀, to avoid this mistake.

Existential Quantification - ∃

There is one *x* for some possible object at the model for which this is true.

$$\exists x \;\; At(x, Stanford) \land Smart(x)$$

E.g.: someone at Stanford is smart.

Another common mistake to avoid:
Using => as the main connective with ∃.

$$\exists x \;\; At(x, Stanford) \;\Rightarrow\; Smart(x)$$

is true if there is anyone who is not at Stanford!

Common connective with ∃ is ∧.

Quantifier duality:
Each quantifier can be expressed using the other.

$$\forall x \;\; Likes(x, IceCream) \qquad \neg\exists x \;\; \neg Likes(x, IceCream)$$

$$\exists x \;\; Likes(x, Broccoli) \qquad \neg\forall x \;\; \neg Likes(x, Broccoli)$$

Everyone likes ice cream == there does not exist someone who does not like ice cream.

Oggie suggests to practice with first order logic sentences.

Equality

One term is equal to another term is under their given interepretation, if and only if these two terms refer to the same object.

Father(John) = Henry

Interacting with FOL knowledge base

An agent can tell something to the knowledge base – basically giving it some kind of a fact.

Suppose a wumpus-world agent is using an FOL KB
and perceives a smell and a breeze (but no glitter) at $t = 5$:

$Tell(KB, Percept([Smell, Breeze, None], 5))$
$Ask(KB, \exists a \;\; Action(a, 5))$

I.e., does KB entail any particular actions at $t = 5$?

Answer: $Yes, \; \{a/Shoot\}$ $\quad\leftarrow$ substitution (binding list)

Wumpus perceives a smell and breeze at *t=5*. It tells it to the KB, and then asks what kind of action (*a*) they can perform at *t=5*.

If KB entails any particular actions at *t=5*, it returns them and then we can substitute *a* with *Shoot.*

$$\forall s, b, t \;\; Percept([s, b, Glitter], t) \Rightarrow AtGold(t)$$

For every smell (*s*), breeze (*b*), and time (*t*), if you percept Glitter, then there is gold at time *t.* We don't care about smell and breeze, all we care about is that if there is glitter, there is gold at *t*.

Lecture 9

In FOL, Quantifiers can only be applied to objects.

You cannot say that a quantifier applies for a predicate.

You can represent **plans** as a sequence of certain actions [a1, a2,…, an]

PlanResult(p,s) is the result of executing p in s.

Universal Instantiation

Applies for Universal Quantifiers. Taking a specific instance of this general statement.

$$\frac{\forall v \;\; \alpha}{\textsc{Subst}(\{v/g\}, \alpha)}$$

If something applies to all *v*, then I can take a specific whatever *v* and substitute it at the bottom part and things will hold.

Existential Instantiation

Applies for Existential Quantifiers. Like a dummy variable (not used anywhere else).

For any sentence *a,* variable *v*, and constant symbol *k*
That does not appear elsewhere in the knowledge base:

$$\frac{\exists v \;\; \alpha}{\textsc{Subst}(\{v/k\}, \alpha)}$$

Put specific thing inside the dummy variable.

E.g., $\exists x \;\; Crown(x) \wedge OnHead(x, John)$ yields

$$Crown(C_1) \wedge OnHead(C_1, John)$$

There exists a variable *x* that Crown(x) and OnHead(x, John) returns true, then we can eliminate the existential quanitifier by replacing it with a constant (that does not exist elsewhere). In this case C1.

Use different names for different variables.

Oggie repeatedly says he finds these things very boring.