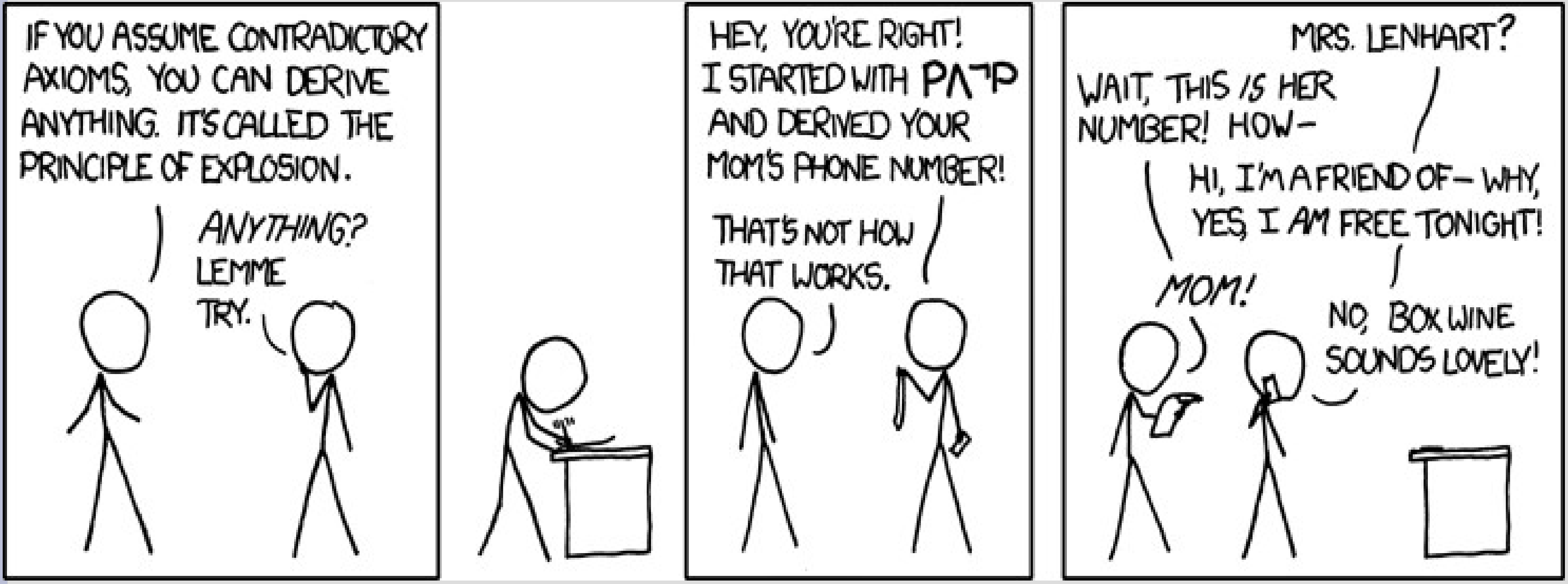
Using first order logic (Ch. 9)



Announcements

Writing 2 graded (was last Thurs but I forgot to announce)
-Regrade deadline: Dec. 5

Writing 4 due on Sunday (need to decide project)

Resolution in FO logic

You try it!

1. Use logical equivalence to remove implies
2. Move logical negation next to relations
3. Standardize variables
4. Generalize existential quantifiers
5. Drop universal quantifiers
6. Distribute ORs over ANDs

Convert this to CNF:

$$\forall x \ A(x) \iff \forall y \ B(x,y)$$

Resolution in FO logic

$\forall x \ A(x) \iff \forall y \ B(x,y)$

1. $\left(\forall x \ A(x) \Rightarrow \forall y \ B(x,y)\right) \wedge \left(\forall x \ \forall y \ B(x,y) \Rightarrow A(x)\right)$
1. $\left(\forall x \ \neg A(x) \vee \forall y \ B(x,y)\right) \wedge \left(\forall x \ \boxed{\neg \forall y \ B(x,y)} \vee A(x)\right)$
2. $\left(\forall x \ \neg A(x) \vee \forall y \ B(x,y)\right) \wedge \left(\forall x \ \exists y \ \neg B(x,y) \vee A(x)\right)$
3. (nothing to do)
4. $\left(\forall x \ \neg A(x) \vee \forall y \ B(x,y)\right) \wedge \left(\forall x \ \neg B(x,Y(x)) \vee A(x)\right)$
5. $\left(\neg A(x) \vee B(x,y)\right) \wedge \left(\neg B(x,Y(x)) \vee A(x)\right)$
6. (nothing to do)

The negation goes where show in the blue box, because y is localized to one side, while not x

Resolution in FO logic

Resolution is <u>refutation-complete</u> in first-order logic (due to it being semi-decidable)

So using resolution we can tell if: “a entails b”

But we cannot tell if: “a does not entail b”

Resolution recap:
PL: complete, can do “entails” and “not entail”
FOL: refutation-complete, only does “entails”

Resolution in FO logic

Consider this KB: $A(Dog) \vee A(Cat)$

$\neg A(Dog)$

$\forall x\ A(x) \Rightarrow B(x)$

If we ask: B(Cat)?

$A(Dog) \vee A(Cat)$

$\neg A(Dog)$

$\forall x\ \neg A(x) \vee B(x)$

$\neg B(Cat)$

$\neg A(Cat)$

unify {x/Cat}

$\neg A(Dog)$

Contradiction!

KB enails B(Cat)

Resolution in FO logic

The last example worked correctly as it identified entailment

However, it has trouble giving us answers to existentials: Ask “exists x, A(x)”?

$A(Dog) \vee A(Cat)$

$\neg A(Dog)$

$\forall x \; \neg A(x) \vee B(x)$

$\forall x \neg A(x)$

unify {x/Cat}

$A(Dog)$

contra-diction

unify {x/Dog}

This only tells us (2 unify): A(Cat) OR A(Dog)

Resolution in FO logic

Thus, resolution in first-order logic will always tell you if a sentence is entailed

However, it might not be able to tell you for what values it is satisfiable

Similar to the semi-decidable nature of FO logic, resolution is complete if entailment can be found in a finite number of inferences (or “resolves”)

Resolution and equality

Once again, I have avoided equality as it is not much fun to deal with

Two ways to deal with this are:
1. Add rules of equality to KB
2. De/Para-modulation (i.e. more substituting)

Both can increase the complexity of the KB or inference by a large amount, so it is better to just avoid equality if possible

Resolution and equality

There are three basic rules of equality:

1. reflexive: $\forall x \; x = x$
2. symmetric: $\forall x, y \; x = y \Rightarrow y = x$
3. transitive: $\forall x, y, z \; x = y \land y = z \Rightarrow x = z$

Then **for each relation/function** we have to add an explicit statement:

Relations (1 var): $\forall x, y \; x = y \Rightarrow A(x) \iff A(y)$

Functions (2 vars): (=> instead of iff)

$$\forall a, b, x, y \; a = x \land b = y \Rightarrow F(a, b) = F(x, y)$$

Resolution and equality

Consider this KB: $A(x) \vee B(x, F(x))$

$\forall x, y \; x = y \Rightarrow B(x, y)$

Would need to be converted into:

$$\forall x \; x = x$$
$$\forall x, y \; x = y \Rightarrow y = x$$
$$\forall x, y, z \; x = y \wedge y = z \Rightarrow x = z$$
$$\forall a, x \; a = x \Rightarrow [A(a) \iff A(x)]$$
$$\forall a, b, x, y \; a = x \wedge b = y \Rightarrow [B(a, b) \iff B(x, y)]$$
$$\forall a, x \; a = x \Rightarrow F(a) = F(x)$$
$$A(x) \vee B(x, F(x))$$
$$\forall x, y \; x = y \Rightarrow B(x, y)$$

Resolution and equality

Consider this KB: $A(x) \lor B(x, F(x))$

$\forall x, y \; x = y \Rightarrow B(x, y)$

Basically, you convert = into a relationship

$$\forall x \; Eq(x, x)$$
$$\forall x, y \; Eq(x, y) \Rightarrow Eq(y, x)$$
$$\forall x, y, z \; Eq(x, y) \land Eq(y, z) \Rightarrow Eq(x, z)$$
$$\forall a, x \; Eq(a, x) \Rightarrow [A(a) \iff A(x)]$$
$$\forall a, b, x, y \; Eq(a, x) \land Eq(b, y) \Rightarrow [B(a, b) \iff B(x, y)]$$
$$\forall a, x \; Eq(a, x) \Rightarrow Eq(F(a), F(x))$$
$$A(x) \lor B(x, F(x))$$
$$\forall x, y \; Eq(x, y) \Rightarrow B(x, y)$$

Resolution and equality

The second option doubles the available inferences instead of doubling the KB

We allow paramodulation, in addition to the normal resolution rule

Paramodulation is essentially substituting with a sentence that contains an equals, while also applying resolution to combine (and ensures there is no conflict in the KB)

Resolution and equality

Consider this KB:

$$A(x) \lor B(F(x, Cat)) \lor C(x, Cat)$$

$$[F(Dog, y) = G(y)] \lor D(y)$$

We can then unify {x/Dog, y/Cat} and get:

$$A(Dog) \lor B(F(Dog, Cat)) \lor C(Dog, Cat)$$

$$[F(Dog, Cat) = G(Cat)] \lor D(Cat)$$

Which we can infer:

$$A(Dog) \lor B(G(Cat)) \lor C(Dog, Cat) \lor D(Cat)$$

1. Like resolution you combine sentences
2. Valid substitutions if necessary

Resolution and equality

Consider this KB:

$$A(x) \vee B(F(x, Cat)) \vee C(x, Cat)$$

$$[F(Dog, y) = G(y)] \vee D(y)$$

We can then unify {x/Dog, y/Cat} and get:

$$A(Dog) \vee B(\boxed{F(Dog, Cat)}) \vee C(Dog, Cat)$$

$$[\boxed{F(Dog, Cat)} = \boxed{G(Cat)}] \vee D(Cat)$$

Which we can infer:

$$A(Dog) \vee B(\boxed{G(Cat)}) \vee C(Dog, Cat) \vee D(Cat)$$

1. Like resolution you combine sentences
2. Valid substitutions if necessary

Resolution efficiency

Four (brief) ways to speed up resolution:

1. Subsumption
2. Unit preference
3. Support set
4. Input resolution

1. and 2. are general and do not effect the completeness of resolution

3. and 4. can limit resolvability

Resolution efficiency

Subsumption is to remove any sentences that are fully expressed by another sentence

Consider this KB: $\forall x \ A(x)$
$A(Cat)$

The first sentence is more general and the second is not adding anything

We could simply reduce the KB to: $\forall x \ A(x)$
(and keep th same meaning)

Resolution efficiency

Unit preference is to always apply a clause containing one literal before any others

Since we want to end up with an empty clause for a contradiction, this will shrink the size of the original clause

one literal

For example: $\left(A(x) \vee B(x) \vee C(x)\right) \wedge \left(\neg A(x)\right)$

... will resolve to: $\left(B(x) \vee C(x)\right)$

Resolution efficiency

A Support set is artificially restricting the KB and removing (what you think are) irrelevant clauses

The set of clauses you use can be based on the query, so if we have this KB:

$$A(x) \Rightarrow B(x)$$
$$B(x) \Rightarrow C(x)$$
$$\exists x \; A(x)$$

Then we ask: $\exists x$ B(x)?

We can see the middle sentence is worthless, so we can solve it just with the first and third

Resolution efficiency

If the support set contains no equalities, there will be a large efficiency increase

However, if the support set does not contain an important sentence you can reach an incorrect conclusion (about entailment)

Even without equality, eliminating a portion of the KB can give large speed ups (as inference is NP-hard, i.e. exponential)

Resolution efficiency

Input resolution starts with a single sentence, and only tries to apply resolution to that sentence (and the resulting sentences)

The resolution of this earlier example is one:

$$A(Dog) \lor A(Cat)$$
$$\neg A(Dog)$$
$$\forall x \; \neg A(x) \lor B(x)$$
$$\boxed{\neg B(Cat)}$$

$$\boxed{\neg A(Cat)}$$

$$\boxed{A(Dog)}$$

Contradiction!

The blue line is involved in all resolutions