Proof by Resolution

Simon Dixon
Queen Mary University of London

Objective

- Learn about general-purpose theorem proving in predicate calculus

The Problem

- Given
  - a knowledge base, KB (a set of sentences) and an interpretation, I
- Prove
  - a sentence, S (under the same interpretation, I)
- Formally
  - Show that KB ⊧ S
    - KB *entails* S
    - S *follows from* KB

The Toolbox

- Modus ponens
  - Given $\{ P \rightarrow Q, P \} \subset KB$, is Q true?
  - Yes: $\{ P \rightarrow Q, P \} \models \{ Q \}$
- Modus Tollens
  - Given $\{ P \rightarrow Q, \neg Q \} \subset KB$, is P true?
  - No: $\{ P \rightarrow Q, \neg Q \} \models \{ \neg P \}$
- We can form arbitrarily long "chains" of inference to prove a sentence
- We can reason
  - forwards from what we know to what we want to prove
  - backwards from what we want to prove to what we know
    - backwards is generally more efficient: no search branches leading off-topic

Theorem Proving

- A theorem proving process involves choosing and applying such rules until the desired sentence is shown to be entailed
- It’s called a *proof* because the rules used are known, a priori, to be *sound* (i.e., correct)

Theorem Proving

- A theorem proving process involves choosing and applying such rules until the desired sentence is shown to be entailed
- It's called a *proof* because the rules used are known, a priori, to be *sound* (i.e., correct)
- However, the choice of rule is hard, because you can't know whether a particular rule chosen will help lead toward a solution, in a long proof
    - e.g., Modus Ponens is incomplete
    - therefore, each time we use it, we also have to consider other possibilities
    - therefore, each time we use it, we create a set of alternative choices

Modus Ponens is incomplete

- Consider these rules:
  - If it is raining (R), I will carry an umbrella (U)
  - If it is not raining (¬R), I will carry an umbrella (U)

Modus Ponens is incomplete

- Consider these rules:
  - If it is raining (R), I will carry an umbrella (U)
  - If it is not raining (¬R), I will carry an umbrella (U)
- It is easy to conclude (as a human) that I always carry an umbrella
  - $\{ R \rightarrow U, \neg R \rightarrow U \} \models \{ U \}$
  - but this isn't provable using modus ponens alone
    - we'd need the law of excluded middle: $R \vee \neg R$
- However, there is a more general rule, that is *complete*

Resolution

- Unit resolution
  - $\{ P \lor Q, \neg Q \} \models \{ P \}$

$$(P \lor Q) \land \neg Q \rightarrow P$$

$$\frac{P \lor Q, \qquad \neg Q}{P}$$

Resolution

- Unit resolution
  - $\{ P \lor Q, \neg Q \} \models \{ P \}$

$$(P \lor Q) \land \neg Q \rightarrow P$$

$$\frac{P \lor Q, \qquad \neg Q}{P}$$

- Generalised resolution
  - $\{ P \lor Q, R \lor \neg Q \} \models \{ P \lor R \}$
  - in general: P, R can be arbitrarily long disjunctions

$$(P \lor Q) \land (R \lor \neg Q) \rightarrow (P \lor R)$$

$$\frac{P \lor Q, \qquad R \lor \neg Q}{P \lor R}$$

Resolution

- Unit resolution
  - { P ∨ Q, ¬Q } ⊧ { P }

$$(P \lor Q) \land \neg Q \to P$$

$$\frac{P \lor Q, \qquad \neg Q}{P}$$

- Generalised resolution
  - { P ∨ Q, R ∨ ¬Q } ⊧ { P ∨ R }
  - in general: P, R can be arbitrarily long disjunctions

$$(P \lor Q) \land (R \lor \neg Q) \to (P \lor R)$$

$$\frac{P \lor Q, \qquad R \lor \neg Q}{P \lor R}$$

- Example: Umbrella again

$$\dfrac{\textit{resolution}\ \dfrac{\textit{defn}\ R \to U}{\neg R \lor U} \qquad \dfrac{\dfrac{\textit{defn}\ \neg R \to U}{\neg\ \textit{elim}\ \neg\neg R \lor U}}{R \lor U}}{\dfrac{\lor\ \textit{elim}\ U \lor U}{U}}$$

Resolution refutation

- In order to prove a sentence, S, this simple rule can be used as follows:
  - add the negation of S to the KB
  - see if this leads to a contradiction

Resolution refutation

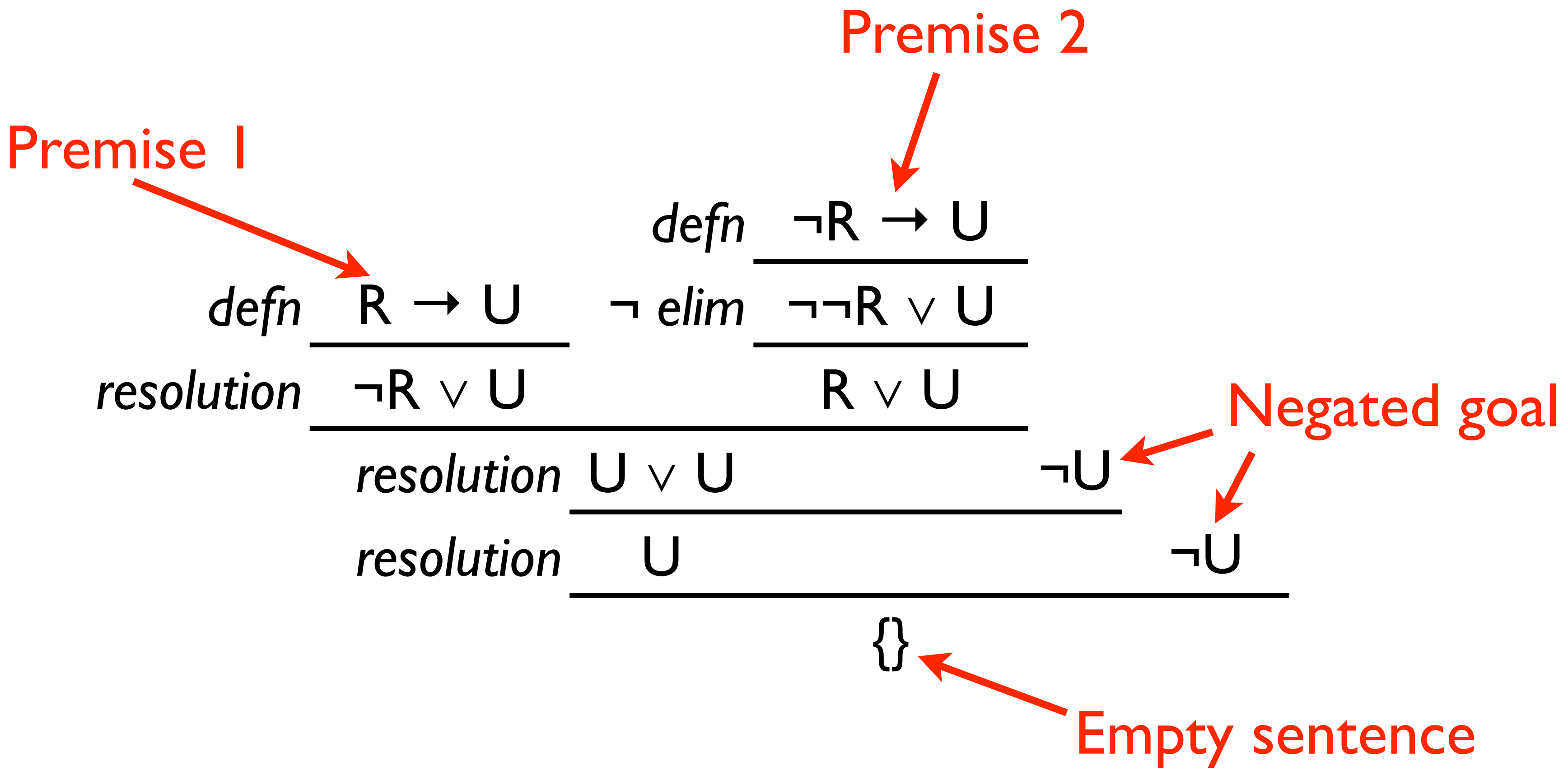
- In order to prove a sentence, S, this simple rule can be used as follows:
  - add the negation of S to the KB
  - see if this leads to a contradiction
- This idea applies the law of the excluded middle ($S \vee \neg S$)
  - if $\neg S$ is inconsistent with KB, then $KB \models S$

Resolution refutation

- In order to prove a sentence, S, this simple rule can be used as follows:
  - add the negation of S to the KB
  - see if this leads to a contradiction
- This idea applies the law of the excluded middle (S ∨ ¬S)
  - if ¬S is inconsistent with KB, then KB ⊧ S
- This is called *resolution refutation*
  - this is the basis of the "Logic Programming" language, Prolog

Resolution Refutation

- Notation
  - note that the premises are brought in when needed, not all at the top



Premise 2

Premise 1

defn ¬R → U

defn R → U   *¬ elim* ¬¬R ∨ U

resolution ¬R ∨ U   R ∨ U

resolution U ∨ U   ¬U

resolution U   ¬U

{}

Negated goal

Empty sentence

Clausal Form

- Resolution is a single, simple, sound, complete rule
  - But we had to do some manipulation first, to get the sentences into a form in which we could use it
- This is *conjunctive* (or *clausal*) *normal form (CNF)*
- Putting FOPC sentences into clausal form is a mechanical procedure that can be done without search

Conjunctive Normal Form in FOPC

1. Rewrite →: $A \rightarrow B \Rightarrow \neg A \vee B$

Conjunctive Normal Form in FOPC

1. Rewrite →: $A \rightarrow B \Rightarrow \neg A \vee B$

2. Minimise the scope of negations using logical definitions given before

- $\neg\exists x.A(x) \Rightarrow \forall x.\neg A(x)$
- $\neg\forall x.A(x) \Rightarrow \exists x.\neg A(x)$
- $\neg(A \vee B) \Rightarrow \neg A \wedge \neg B$ (De Morgan's laws)
- $\neg(A \wedge B) \Rightarrow \neg A \vee \neg B$ (De Morgan's laws)

- Note that in this case, these definitions are *unidirectional*, so there is no need for searching through alternatives

Conjunctive Normal Form in FOPC

1. Rewrite →: $A \rightarrow B \Rightarrow \neg A \vee B$

2. Minimise the scope of negations using logical definitions given before

   - $\neg\exists x.A(x) \Rightarrow \forall x.\neg A(x)$
   - $\neg\forall x.A(x) \Rightarrow \exists x.\neg A(x)$
   - $\neg(A \vee B) \Rightarrow \neg A \wedge \neg B$ (De Morgan's laws)
   - $\neg(A \wedge B) \Rightarrow \neg A \vee \neg B$ (De Morgan's laws)

   ‣ Note that in this case, these definitions are *unidirectional*, so there is no need for searching through alternatives

3. Rewrite remaining double negations: $\neg\neg A \Rightarrow A$

Conjunctive Normal Form in FOPC

4. Standardise variables apart

- rename all quantified variables so that each quantifier is associated with a different name, regardless of its scope

Conjunctive Normal Form in FOPC

4. Standardise variables apart

- rename all quantified variables so that each quantifier is associated with a different name, regardless of its scope

5. *Skolemise* all existential quantifiers

- A Skolem constant (or function) is a made-up name for an object that must exist (even though we don't know what it is)
  - $\exists x.P(x) \Rightarrow P(A)$ where A is an arbitrary object in the allowable substitutions of x
  - Use a different arbitrary object for each quantifier
  - If the existentially quantified variable is in the scope of a universally quantified variable, it is replaced with a function of the universally quantified variable: $\forall x.\exists y.P(x, y) \Rightarrow \forall x.P(x, F(x))$

Conjunctive Normal Form in FOPC

4. Standardise variables apart

   - rename all quantified variables so that each quantifier is associated with a different name, regardless of its scope

5. *Skolemise* all existential quantifiers

   - A Skolem constant (or function) is a made-up name for an object that must exist (even though we don't know what it is)
     - $\exists x.P(x) \Rightarrow P(A)$ where A is an arbitrary object in the allowable substitutions of x
     - Use a different arbitrary object for each quantifier
     - If the existentially quantified variable is in the scope of a universally quantified variable, it is replaced with a function of the universally quantified variable: $\forall x.\exists y.P(x, y) \Rightarrow \forall x.P(x, F(x))$

6. Drop all universal quantifiers

   - At this point, all variables are universally quantified, because we Skolemised the existentials, so we no longer need to say so explicitly

Conjunctive Normal Form in FOPC

7. Convert the sentence into *conjunctive normal form*
    - a sentence in CNF is a conjunction of disjunctions of atomic sentences
        - recall that the resolution rule works on disjunctions
    - rewrite the sentence using logical rules
        - distributivity of $\wedge$ and $\vee$

Conjunctive Normal Form in FOPC

7. Convert the sentence into *conjunctive normal form*
    - a sentence in CNF is a conjunction of disjunctions of atomic sentences
        - recall that the resolution rule works on disjunctions
    - rewrite the sentence using logical rules
        - distributivity of $\wedge$ and $\vee$

8. Split the top-level conjunction up, to make a set of disjunctions

Conjunctive Normal Form in FOPC

7. Convert the sentence into *conjunctive normal form*
   - a sentence in CNF is a conjunction of disjunctions of atomic sentences
     - recall that the resolution rule works on disjunctions
   - rewrite the sentence using logical rules
     - distributivity of $\wedge$ and $\vee$

8. Split the top-level conjunction up, to make a set of disjunctions

9. Standardise the variables apart again, w.r.t. clauses
   - so that there is no overlap between variables in different clauses

Conjunctive Normal Form in FOPC

7. Convert the sentence into *conjunctive normal form*
   - a sentence in CNF is a conjunction of disjunctions of atomic sentences
     - recall that the resolution rule works on disjunctions
   - rewrite the sentence using logical rules
     - distributivity of ∧ and ∨

8. Split the top-level conjunction up, to make a set of disjunctions

9. Standardise the variables apart again, w.r.t. clauses
   - so that there is no overlap between variables in different clauses

- Use the resulting set of clauses as KB in a resolution proof

First Order Term Unification

- As a result of Skolemising and removing universal quantifiers, we can introduce a new procedure, *unification*, for assigning values to variables
- This is related to ∀ instantiation in the standard FOPC
  - using the relevant constants and functions to instantiate the variables
- There is a deterministic algorithm for unification
- The idea is to use literals of a given predicate which contain information about the values of variables to deduce the values of variables in other literals of the same predicate
  - e.g. Father(x, y) and Father(John, Jim) are unified to Father(John, Jim)
  - with a *unifier* (or substitution set) of { John/x, Jim/y }
    - notation: a/x means "a replaces x"

First Order Term Unification

- To unify two terms (or literals):
  1. if either is a variable, let it be identical to the other, and add the resulting pair to the unifier; otherwise...
  2. compare their functors (the outermost predicate or function symbol); if they do not match, then fail; otherwise...
  3. for each pair of respective arguments, unify the two arguments using this procedure, and combine the unifiers for each argument pair.
- The resulting list of substitutions is called the Most General Unifier (MGU)
- Note: in step 1, the expression substituted for a variable is **not** allowed to contain the variable it substitutes (e.g. P(x) and P(f(y,x)) do not unify, since the substitution f(y,x)/x is not allowed)

First Order Term Unification

- Notation
  - We sometimes write a *Term* followed by a *Unifier* to mean "The result of applying this *unifier* to this *term*"
    - P(x, y) { A/x, B/y } which evaluates to P(A, B)
- Successive application of unifiers
  - We can write *Term Unifier1 Unifier2* to mean "The result of applying these unifiers, one at a time, to Term"
    - P(x, y) { A/x }{ B/y } which evaluates to P(A, B)
- Composition of unifiers
  - We can combine unifiers, so long as there are no contradictory assignments
    - { A/x } { B/y } combine to give { A/x, B/y }
    - { A/x } { B/x } do not combine, because x would have to take 2 different values at once

Unification examples

- P(x, y) unified with P(A, B) gives P(A, B) unifier { A/x, B/y }
- P(x, y) unified with Q(A, B) gives ??
- P(F(x)) unified with P(F(A)) gives ??
- P(F(x), x, u, u) unified with P(F(y), z, z, A) gives ??

Unification examples: solutions

- P(x, y) unified with P(A, B) gives P(A, B) unifier { A/x, B/y }
- P(x, y) unified with Q(A, B) gives no unifier (P ≠ Q)
- P(F(x)) unified with P(F(A)) gives P(F(A)) unifier { A/x }
- P(F(x), x, u, u) unified with P(F(y), z, z, A) gives

  P(F(A), A, A, A) unifier { x/y, x/z, x/u, A/x }

  - note that we don't need to write down all the different permutations
    - this is enough to say that they're all the same

Resolution example

- Some rules
  - All people who are graduating are happy. All happy people smile. Jane is graduating.
- and a question
  - Is Jane smiling?

Resolution example

- Some rules
  - All people who are graduating are happy. All happy people smile. Jane is graduating.
- and a question
  - Is Jane smiling?
- First convert to predicate logic
- Premise
  - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \wedge \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \wedge \text{Graduating}(\text{Jane})$
- Goal
  - Smiling(Jane)

Resolution example

- Convert to CNF
  - Premise
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \wedge \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \wedge \text{Graduating}(\text{Jane})$

Resolution example

- Convert to CNF
  - Premise
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \land \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \land \text{Graduating}(\text{Jane})$

1. Rewrite →
   - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall x.(\neg\text{Happy}(x) \lor \text{Smiling}(x)) \land \text{Graduating}(\text{Jane})$

Resolution example

- Convert to CNF
  - Premise
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \wedge \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \wedge \text{Graduating}(\text{Jane})$

1. Rewrite →
    - $\forall x.(\neg\text{Graduating}(x) \vee \text{Happy}(x)) \wedge \forall x.(\neg\text{Happy}(x) \vee \text{Smiling}(x)) \wedge \text{Graduating}(\text{Jane})$

2. Reduce scope of negations: all minimal, so nothing to do

Resolution example

- Convert to CNF
  - Premise
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \wedge \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \wedge \text{Graduating}(\text{ Jane })$

1. Rewrite →
    - $\forall x.(\neg\text{Graduating}(x) \vee \text{Happy}(x)) \wedge \forall x.(\neg\text{Happy}(x) \vee \text{Smiling}(x)) \wedge \text{Graduating}(\text{ Jane })$
2. Reduce scope of negations: all minimal, so nothing to do
3. Rewrite double negations: no double negations, so nothing to do

Resolution example

- Convert to CNF
  - Premise
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \wedge \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \wedge \text{Graduating}(\text{ Jane })$

1. Rewrite →
   - $\forall x.(\neg\text{Graduating}(x) \vee \text{Happy}(x)) \wedge \forall x.(\neg\text{Happy}(x) \vee \text{Smiling}(x)) \wedge \text{Graduating}(\text{ Jane })$
2. Reduce scope of negations: all minimal, so nothing to do
3. Rewrite double negations: no double negations, so nothing to do
4. Standardise variables apart
   - $\forall x.(\neg\text{Graduating}(x) \vee \text{Happy}(x)) \wedge \forall y.(\neg\text{Happy}(y) \vee \text{Smiling}(y)) \wedge \text{Graduating}(\text{ Jane })$

Resolution example

5. Skolemise existentials

- no existentials, so nothing to do

Resolution example

5. Skolemise existentials

- no existentials, so nothing to do

6. Drop all universal quantifiers

- ▮ $(\neg \mathrm{Graduating}(x) \lor \mathrm{Happy}(x)) \land$ ▮ $(\neg \mathrm{Happy}(y) \lor \mathrm{Smiling}(y)) \land \mathrm{Graduating}(\mathit{Jane})$

Resolution example

Queen Mary
University of London

5. Skolemise existentials

- no existentials, so nothing to do

6. Drop all universal quantifiers

- ▮ (¬Graduating(x) ∨ Happy(x)) ∧ ▮(¬Happy(y) ∨ Smiling(y)) ∧ Graduating(Jane)

7. Convert to CNF

- already in CNF

Resolution example

5. Skolemise existentials

- no existentials, so nothing to do

6. Drop all universal quantifiers

- (¬Graduating(x) ∨ Happy(x)) ∧ (¬Happy(y) ∨ Smiling(y)) ∧ Graduating(Jane)

7. Convert to CNF

- already in CNF

8. Separate into disjunctive clauses

- ¬Graduating(x) ∨ Happy(x)
- ¬Happy(y) ∨ Smiling(y)
- Graduating(Jane)

Resolution example

5. Skolemise existentials

- no existentials, so nothing to do

6. Drop all universal quantifiers

- (¬Graduating(x) ∨ Happy(x)) ∧ (¬Happy(y) ∨ Smiling(y)) ∧ Graduating(Jane)

7. Convert to CNF

- already in CNF

8. Separate into disjunctive clauses

- ¬Graduating(x) ∨ Happy(x)
- ¬Happy(y) ∨ Smiling(y)
- Graduating(Jane)

9. Standardise variables apart between clauses

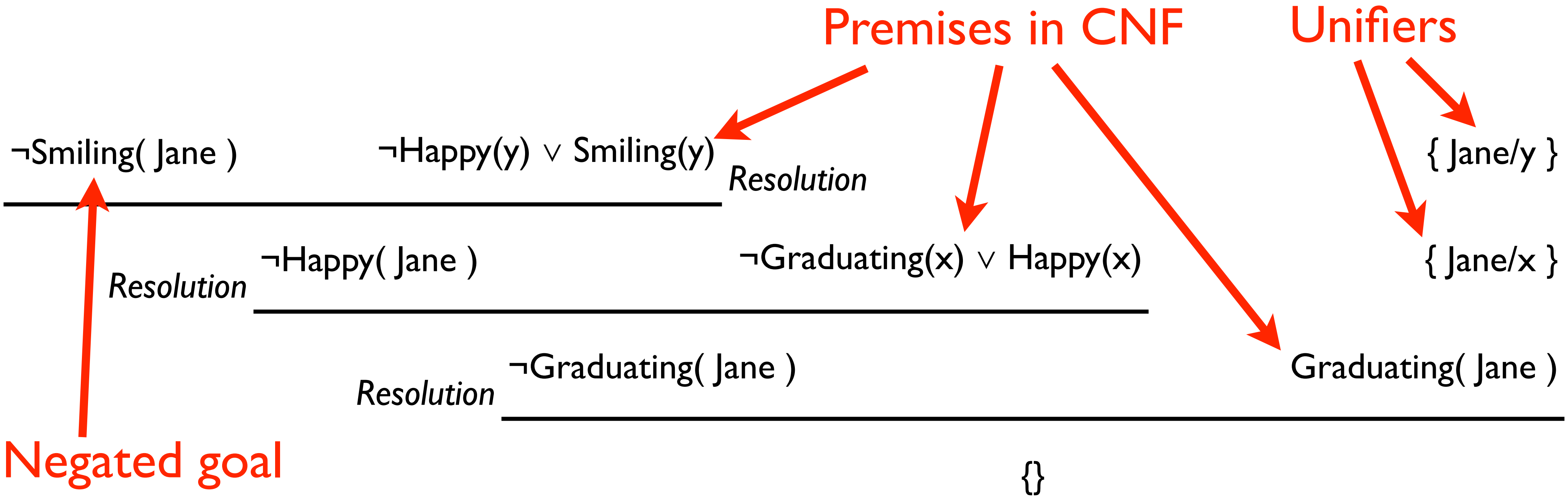
- no change: they're already named apart

Resolution example with unification

- Now, the only rules we need are
  - unification
  - factoring: unifying literals of the same polarity within a clause
  - resolution

Resolution example with unification

- Now, the only rules we need are
  - unification
  - factoring: unifying literals of the same polarity within a clause
  - resolution



Premises in CNF

Unifiers

¬Smiling(Jane)   ¬Happy(y) ∨ Smiling(y)   *Resolution*   { Jane/y }

Resolution   ¬Happy(Jane)   ¬Graduating(x) ∨ Happy(x)   { Jane/x }

Resolution   ¬Graduating(Jane)   Graduating(Jane)

{}

Negated goal

Example with existential quantification

- Some rules
  - All people who are graduating are happy. All happy people smile. Someone is graduating.
- and a question
  - Is anyone smiling?

Example with existential quantification

- Some rules
  - All people who are graduating are happy. All happy people smile. Someone is graduating.
  - and a question
    - Is anyone smiling?
- First convert to predicate logic
  - Premise
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \land \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \land \exists x.\text{Graduating}(x)$
  - Goal
    - $\exists x.\text{Smiling}(x)$

Example with existential quantification

- Conversion to CNF

Example with existential quantification

- Conversion to CNF
  - This time I’ve added the negated goal to the premises, instead of later
    - $\forall x.(\mathrm{Graduating}(x) \rightarrow \mathrm{Happy}(x)) \wedge \forall x.(\mathrm{Happy}(x) \rightarrow \mathrm{Smiling}(x)) \wedge \exists x.\mathrm{Graduating}(x) \wedge \neg\exists x.\mathrm{Smiling}(x)$

Example with existential quantification

- Conversion to CNF
  - This time I've added the negated goal to the premises, instead of later
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \land \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \land \exists x.\text{Graduating}(x) \land \neg\exists x.\text{Smiling}(x)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall x.(\neg\text{Happy}(x) \lor \text{Smiling}(x)) \land \exists x.\text{Graduating}(x) \land \neg\exists x.\text{Smiling}(x)$

Example with existential quantification

- Conversion to CNF
  - This time I've added the negated goal to the premises, instead of later
    - $\forall x.(\mathrm{Graduating}(x) \rightarrow \mathrm{Happy}(x)) \wedge \forall x.(\mathrm{Happy}(x) \rightarrow \mathrm{Smiling}(x)) \wedge \exists x.\mathrm{Graduating}(x) \wedge \neg\exists x.\mathrm{Smiling}(x)$
    - $\forall x.(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge \forall x.(\neg\mathrm{Happy}(x) \vee \mathrm{Smiling}(x)) \wedge \exists x.\mathrm{Graduating}(x) \wedge \neg\exists x.\mathrm{Smiling}(x)$
    - $\forall x.(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge \forall x.(\neg\mathrm{Happy}(x) \vee \mathrm{Smiling}(x)) \wedge \exists x.\mathrm{Graduating}(x) \wedge \forall x.\neg\mathrm{Smiling}(x)$

Example with existential quantification

- Conversion to CNF
  - This time I've added the negated goal to the premises, instead of later
    - $\forall x.(Graduating(x) \rightarrow Happy(x)) \wedge \forall x.(Happy(x) \rightarrow Smiling(x)) \wedge \exists x.Graduating(x) \wedge \neg\exists x.Smiling(x)$
    - $\forall x.(\neg Graduating(x) \vee Happy(x)) \wedge \forall x.(\neg Happy(x) \vee Smiling(x)) \wedge \exists x.Graduating(x) \wedge \neg\exists x.Smiling(x)$
    - $\forall x.(\neg Graduating(x) \vee Happy(x)) \wedge \forall x.(\neg Happy(x) \vee Smiling(x)) \wedge \exists x.Graduating(x) \wedge \forall x.\neg Smiling(x)$
    - $\forall x.(\neg Graduating(x) \vee Happy(x)) \wedge \forall y.(\neg Happy(y) \vee Smiling(y)) \wedge \exists z.Graduating(z) \wedge \forall u.\neg Smiling(u)$

Example with existential quantification

- Conversion to CNF
  - This time I've added the negated goal to the premises, instead of later
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \land \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \land \exists x.\text{Graduating}(x) \land \neg\exists x.\text{Smiling}(x)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall x.(\neg\text{Happy}(x) \lor \text{Smiling}(x)) \land \exists x.\text{Graduating}(x) \land \neg\exists x.\text{Smiling}(x)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall x.(\neg\text{Happy}(x) \lor \text{Smiling}(x)) \land \exists x.\text{Graduating}(x) \land \forall x.\neg\text{Smiling}(x)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall y.(\neg\text{Happy}(y) \lor \text{Smiling}(y)) \land \exists z.\text{Graduating}(z) \land \forall u.\neg\text{Smiling}(u)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall y.(\neg\text{Happy}(y) \lor \text{Smiling}(y)) \land \text{Graduating}(\ \text{Someone}\ ) \land \forall u.\neg\text{Smiling}(u)$

Example with existential quantification

- Conversion to CNF
  - This time I've added the negated goal to the premises, instead of later
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \land \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \land \exists x.\text{Graduating}(x) \land \neg\exists x.\text{Smiling}(x)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall x.(\neg\text{Happy}(x) \lor \text{Smiling}(x)) \land \exists x.\text{Graduating}(x) \land \neg\exists x.\text{Smiling}(x)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall x.(\neg\text{Happy}(x) \lor \text{Smiling}(x)) \land \exists x.\text{Graduating}(x) \land \forall x.\neg\text{Smiling}(x)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall y.(\neg\text{Happy}(y) \lor \text{Smiling}(y)) \land \exists z.\text{Graduating}(z) \land \forall u.\neg\text{Smiling}(u)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall y.(\neg\text{Happy}(y) \lor \text{Smiling}(y)) \land \text{Graduating}(\ \text{Someone}\ ) \land \forall u.\neg\text{Smiling}(u)$
    - $(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land (\neg\text{Happy}(y) \lor \text{Smiling}(y)) \land \text{Graduating}(\ \text{Someone}\ ) \land \neg\text{Smiling}(u)$

Example with existential quantification

- Conversion to CNF
  - This time I've added the negated goal to the premises, instead of later
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \land \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \land \exists x.\text{Graduating}(x) \land \neg\exists x.\text{Smiling}(x)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall x.(\neg\text{Happy}(x) \lor \text{Smiling}(x)) \land \exists x.\text{Graduating}(x) \land \neg\exists x.\text{Smiling}(x)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall x.(\neg\text{Happy}(x) \lor \text{Smiling}(x)) \land \exists x.\text{Graduating}(x) \land \forall x.\neg\text{Smiling}(x)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall y.(\neg\text{Happy}(y) \lor \text{Smiling}(y)) \land \exists z.\text{Graduating}(z) \land \forall u.\neg\text{Smiling}(u)$
    - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall y.(\neg\text{Happy}(y) \lor \text{Smiling}(y)) \land \text{Graduating}(\text{ Someone }) \land \forall u.\neg\text{Smiling}(u)$
    - $(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land (\neg\text{Happy}(y) \lor \text{Smiling}(y)) \land \text{Graduating}(\text{ Someone }) \land \neg\text{Smiling}(u)$
    - $\{\ \neg\text{Graduating}(x) \lor \text{Happy}(x),\ \neg\text{Happy}(y) \lor \text{Smiling}(y),\ \text{Graduating}(\text{ Someone }),\ \neg\text{Smiling}(u)\ \}$

Example with existential quantification

- Note that the proof is exactly the same as with the constant "Jane"
  - the only difference is that the variable u gets passed around instead

Example with existential quantification

- Note that the proof is exactly the same as with the constant "Jane"
  - the only difference is that the variable u gets passed around instead

$$\frac{\neg\text{Smiling}(u) \qquad \neg\text{Happy}(y) \lor \text{Smiling}(y)}{\neg\text{Happy}(u)}\ \textit{Resolution} \qquad \{\ u/y\ \}$$

$$\textit{Resolution}\ \frac{\neg\text{Happy}(u) \qquad \neg\text{Graduating}(x) \lor \text{Happy}(x)}{\neg\text{Graduating}(u)} \qquad \{\ u/x\ \}$$

$$\textit{Contradiction}\ \frac{\neg\text{Graduating}(u) \qquad \text{Graduating}(\ \text{Someone}\ )}{\{\}} \qquad \{\ \text{Someone}/u\ \}$$

Example with universal quantification

- Some rules
    - All people who are graduating are happy. All happy people smile. Everyone is graduating.
- and a question
    - Is everyone smiling?

Example with universal quantification

- Some rules
  - All people who are graduating are happy. All happy people smile. Everyone is graduating.
  - and a question
    - Is everyone smiling?
- First convert to predicate logic
  - Premise
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \wedge \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \wedge \forall x.\text{Graduating}(x)$
  - Goal
    - $\forall x.\text{Smiling}(x)$

Example with universal quantification

- Conversion to CNF

Example with universal quantification

- Conversion to CNF
  - $\forall x.(\mathrm{Graduating}(x) \rightarrow \mathrm{Happy}(x)) \wedge \forall x.(\mathrm{Happy}(x) \rightarrow \mathrm{Smiling}(x)) \wedge \forall x.\mathrm{Graduating}(x) \wedge \neg\forall x.\mathrm{Smiling}(x)$

Example with universal quantification

- Conversion to CNF
  - $\forall x.(\mathrm{Graduating}(x) \rightarrow \mathrm{Happy}(x)) \wedge \forall x.(\mathrm{Happy}(x) \rightarrow \mathrm{Smiling}(x)) \wedge \forall x.\mathrm{Graduating}(x) \wedge \neg\forall x.\mathrm{Smiling}(x)$
  - $\forall x.(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge \forall x.(\neg\mathrm{Happy}(x) \vee \mathrm{Smiling}(x)) \wedge \forall x.\mathrm{Graduating}(x) \wedge \neg\forall x.\mathrm{Smiling}(x)$

Example with universal quantification

- Conversion to CNF
  - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \wedge \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \wedge \forall x.\text{Graduating}(x) \wedge \neg\forall x.\text{Smiling}(x)$
  - $\forall x.(\neg\text{Graduating}(x) \vee \text{Happy}(x)) \wedge \forall x.(\neg\text{Happy}(x) \vee \text{Smiling}(x)) \wedge \forall x.\text{Graduating}(x) \wedge \neg\forall x.\text{Smiling}(x)$
  - $\forall x.(\neg\text{Graduating}(x) \vee \text{Happy}(x)) \wedge \forall x.(\neg\text{Happy}(x) \vee \text{Smiling}(x)) \wedge \forall x.\text{Graduating}(x) \wedge \exists x.\neg\text{Smiling}(x)$

Example with universal quantification

- Conversion to CNF
  - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \wedge \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \wedge \forall x.\text{Graduating}(x) \wedge \neg\forall x.\text{Smiling}(x)$
  - $\forall x.(\neg\text{Graduating}(x) \vee \text{Happy}(x)) \wedge \forall x.(\neg\text{Happy}(x) \vee \text{Smiling}(x)) \wedge \forall x.\text{Graduating}(x) \wedge \neg\forall x.\text{Smiling}(x)$
  - $\forall x.(\neg\text{Graduating}(x) \vee \text{Happy}(x)) \wedge \forall x.(\neg\text{Happy}(x) \vee \text{Smiling}(x)) \wedge \forall x.\text{Graduating}(x) \wedge \exists x.\neg\text{Smiling}(x)$
  - $\forall x.(\neg\text{Graduating}(x) \vee \text{Happy}(x)) \wedge \forall y.(\neg\text{Happy}(y) \vee \text{Smiling}(y)) \wedge \forall z.\text{Graduating}(z) \wedge \exists u.\neg\text{Smiling}(u)$

Example with universal quantification

- Conversion to CNF
  - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \land \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \land \forall x.\text{Graduating}(x) \land \neg\forall x.\text{Smiling}(x)$
  - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall x.(\neg\text{Happy}(x) \lor \text{Smiling}(x)) \land \forall x.\text{Graduating}(x) \land \neg\forall x.\text{Smiling}(x)$
  - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall x.(\neg\text{Happy}(x) \lor \text{Smiling}(x)) \land \forall x.\text{Graduating}(x) \land \exists x.\neg\text{Smiling}(x)$
  - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall y.(\neg\text{Happy}(y) \lor \text{Smiling}(y)) \land \forall z.\text{Graduating}(z) \land \exists u.\neg\text{Smiling}(u)$
  - $\forall x.(\neg\text{Graduating}(x) \lor \text{Happy}(x)) \land \forall y.(\neg\text{Happy}(y) \lor \text{Smiling}(y)) \land \forall z.\ \text{Graduating}(z) \land \neg\text{Smiling}(\ \text{Someone}\ )$

Example with universal quantification

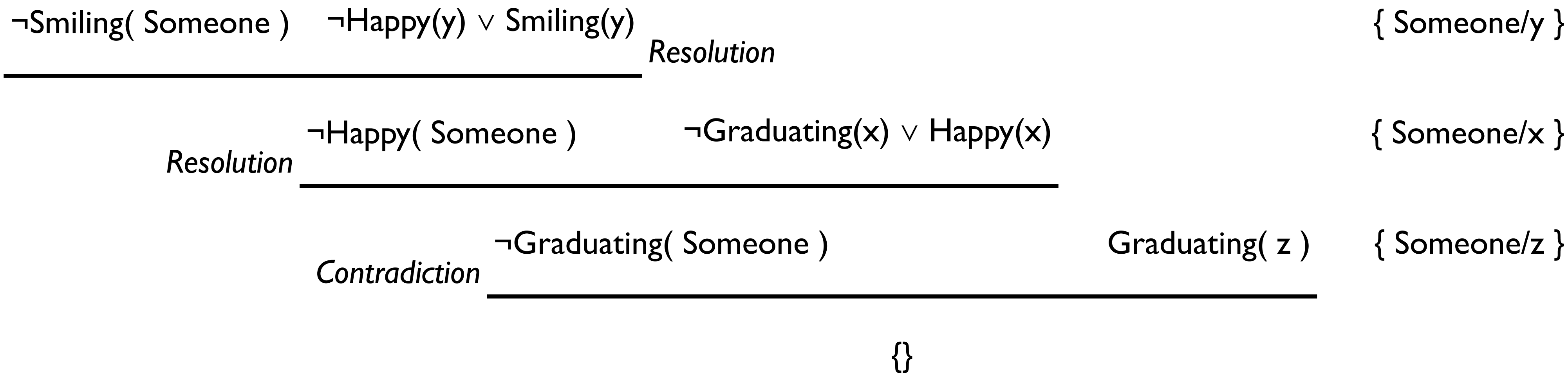
- Conversion to CNF
  - $\forall x.(\mathrm{Graduating}(x) \rightarrow \mathrm{Happy}(x)) \wedge \forall x.(\mathrm{Happy}(x) \rightarrow \mathrm{Smiling}(x)) \wedge \forall x.\mathrm{Graduating}(x) \wedge \neg\forall x.\mathrm{Smiling}(x)$
  - $\forall x.(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge \forall x.(\neg\mathrm{Happy}(x) \vee \mathrm{Smiling}(x)) \wedge \forall x.\mathrm{Graduating}(x) \wedge \neg\forall x.\mathrm{Smiling}(x)$
  - $\forall x.(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge \forall x.(\neg\mathrm{Happy}(x) \vee \mathrm{Smiling}(x)) \wedge \forall x.\mathrm{Graduating}(x) \wedge \exists x.\neg\mathrm{Smiling}(x)$
  - $\forall x.(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge \forall y.(\neg\mathrm{Happy}(y) \vee \mathrm{Smiling}(y)) \wedge \forall z.\mathrm{Graduating}(z) \wedge \exists u.\neg\mathrm{Smiling}(u)$
  - $\forall x.(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge \forall y.(\neg\mathrm{Happy}(y) \vee \mathrm{Smiling}(y)) \wedge \forall z.\ \mathrm{Graduating}(z) \wedge \neg\mathrm{Smiling}(\ \mathrm{Someone}\ )$
  - $(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge (\neg\mathrm{Happy}(y) \vee \mathrm{Smiling}(y)) \wedge \mathrm{Graduating}(z) \wedge \neg\mathrm{Smiling}(\ \mathrm{Someone}\ )$

Example with universal quantification

- Conversion to CNF
  - $\forall x.(\mathrm{Graduating}(x) \rightarrow \mathrm{Happy}(x)) \wedge \forall x.(\mathrm{Happy}(x) \rightarrow \mathrm{Smiling}(x)) \wedge \forall x.\mathrm{Graduating}(x) \wedge \neg\forall x.\mathrm{Smiling}(x)$
  - $\forall x.(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge \forall x.(\neg\mathrm{Happy}(x) \vee \mathrm{Smiling}(x)) \wedge \forall x.\mathrm{Graduating}(x) \wedge \neg\forall x.\mathrm{Smiling}(x)$
  - $\forall x.(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge \forall x.(\neg\mathrm{Happy}(x) \vee \mathrm{Smiling}(x)) \wedge \forall x.\mathrm{Graduating}(x) \wedge \exists x.\neg\mathrm{Smiling}(x)$
  - $\forall x.(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge \forall y.(\neg\mathrm{Happy}(y) \vee \mathrm{Smiling}(y)) \wedge \forall z.\mathrm{Graduating}(z) \wedge \exists u.\neg\mathrm{Smiling}(u)$
  - $\forall x.(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge \forall y.(\neg\mathrm{Happy}(y) \vee \mathrm{Smiling}(y)) \wedge \forall z.\ \mathrm{Graduating}(z) \wedge \neg\mathrm{Smiling}(\ \mathrm{Someone}\ )$
  - $(\neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x)) \wedge (\neg\mathrm{Happy}(y) \vee \mathrm{Smiling}(y)) \wedge \mathrm{Graduating}(z) \wedge \neg\mathrm{Smiling}(\ \mathrm{Someone}\ )$
  - $\{\ \neg\mathrm{Graduating}(x) \vee \mathrm{Happy}(x),\ \neg\mathrm{Happy}(y) \vee \mathrm{Smiling}(y),\ \mathrm{Graduating}(z),\ \neg\mathrm{Smiling}(\ \mathrm{Someone}\ )\ \}$

Example with universal quantification

- Note that the proof is exactly the same as with the constant Jane
  - the difference is that the Skolem constant Someone gets passed around instead



¬Smiling(Someone)   ¬Happy(y) ∨ Smiling(y)   { Someone/y }

Resolution

¬Happy(Someone)   ¬Graduating(x) ∨ Happy(x)   { Someone/x }

Resolution

¬Graduating(Someone)   Graduating(z)   { Someone/z }

Contradiction

{}

Alternative proof

- Here, we use general resolution first – but the effect is the same

$$\dfrac{\neg\text{Graduating}(x) \vee \text{Happy}(x) \qquad \neg\text{Happy}(y) \vee \text{Smiling}(y)}{\neg\text{Graduating}(x) \vee \text{Smiling}(x)}\ \textit{Resolution} \qquad \{\ x/y\ \}$$

$$\textit{Resolution}\ \dfrac{\neg\text{Graduating}(x) \vee \text{Smiling}(x) \qquad \text{Graduating}(\ z\ )}{\text{Smiling}(x)} \qquad \{\ x/z\ \}$$

$$\textit{Contradiction}\ \dfrac{\text{Smiling}(x) \qquad \neg\text{Smiling}(\ \text{Someone}\ )}{\{\}} \qquad \{\ \text{Someone}/x\ \}$$

A (slightly) more realistic example

- In this example, not all of the quantifiers match up neatly
  - Some rules and a question
    - All people who are graduating are happy. All happy people smile. Someone is graduating.
    - Is everyone smiling?

A (slightly) more realistic example

- In this example, not all of the quantifiers match up neatly
  - Some rules and a question
    - All people who are graduating are happy. All happy people smile. Someone is graduating.
    - Is everyone smiling?
- First convert to predicate logic
    - $\forall x.(\mathrm{Graduating}(x) \rightarrow \mathrm{Happy}(x)) \wedge \forall x.(\mathrm{Happy}(x) \rightarrow \mathrm{Smiling}(x)) \wedge \exists x.\mathrm{Graduating}(x)$
    - $\forall x.\mathrm{Smiling}(x)$

A (slightly) more realistic example

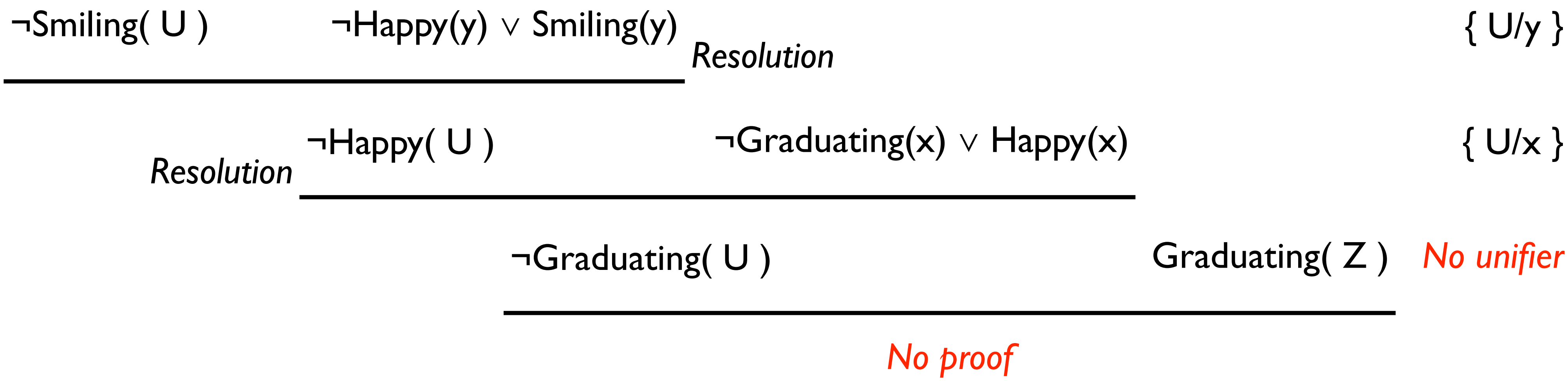
- In this example, not all of the quantifiers match up neatly
    - Some rules and a question
        - All people who are graduating are happy. All happy people smile. Someone is graduating.
        - Is everyone smiling?
- First convert to predicate logic
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \wedge \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \wedge \exists x.\text{Graduating}(x)$
    - $\forall x.\text{Smiling}(x)$

Skolem constants

- Resolution-ready form
    - $\{ \neg\text{Graduating}(x) \vee \text{Happy}(x), \neg\text{Happy}(y) \vee \text{Smiling}(y), \text{Graduating}(Z), \neg\text{Smiling}(U) \}$

A (slightly) more realistic example

- We can’t infer that everyone is smiling from the knowledge that one person is graduating



¬Smiling(U)   ¬Happy(y) ∨ Smiling(y)   *Resolution*   { U/y }

Resolution   ¬Happy(U)   ¬Graduating(x) ∨ Happy(x)   { U/x }

¬Graduating(U)   Graduating(Z)   *No unifier*

No proof

The other way round

- In this example, not all of the quantifiers match up neatly
  - Some rules and a question
    - All people who are graduating are happy. All happy people smile. Everyone is graduating.
    - Is anyone smiling?

The other way round

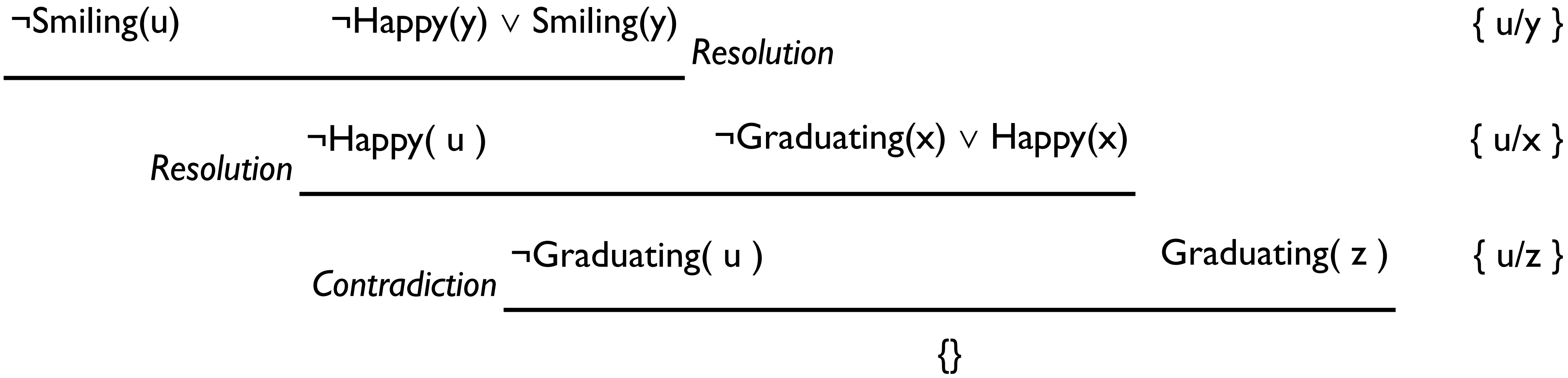
- In this example, not all of the quantifiers match up neatly
  - Some rules and a question
    - All people who are graduating are happy. All happy people smile. Everyone is graduating.
    - Is anyone smiling?
- First convert to predicate logic
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \wedge \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \wedge \forall x.\text{Graduating}(x)$
    - $\exists x.\text{Smiling}(x)$

The other way round

- In this example, not all of the quantifiers match up neatly
  - Some rules and a question
    - All people who are graduating are happy. All happy people smile. Everyone is graduating.
    - Is anyone smiling?
- First convert to predicate logic
    - $\forall x.(\text{Graduating}(x) \rightarrow \text{Happy}(x)) \wedge \forall x.(\text{Happy}(x) \rightarrow \text{Smiling}(x)) \wedge \forall x.\text{Graduating}(x)$
    - $\exists x.\text{Smiling}(x)$
- Resolution-ready form
    - $\{ \neg\text{Graduating}(x) \vee \text{Happy}(x),\ \neg\text{Happy}(y) \vee \text{Smiling}(y),\ \text{Graduating}(z),\ \neg\text{Smiling}(u) \}$

The other way round

- We can infer that one is person is smiling from the knowledge that everyone is graduating



¬Smiling(u)   ¬Happy(y) ∨ Smiling(y)   *Resolution*   { u/y }

Resolution   ¬Happy(u)   ¬Graduating(x) ∨ Happy(x)   { u/x }

Contradiction   ¬Graduating(u)   Graduating(z)   { u/z }

{}

Summary

- Resolution (with unification and factoring) is a simple and powerful rule to perform inference in logical databases
- It is sound and complete for propositional calculus
- It is sound and "refutation complete" for first order predicate calculus
  - a set of sentences is unsatisfiable iff there exists a resolution derivation of the empty clause
  - if the set of sentences is satisfiable, the computation might not terminate

Past **Exam**ple Questions

- 3(c) Express the following first order formula in clausal form:

  isNumber(0) $\wedge$ $\forall x$ isNumber(x) $\rightarrow$ $\exists y$ Greater(y, x) [3 marks]

- 3(d) Using the predicate S(x, y) meaning "x shaves y", express the following sentence in first order logic:

  There is a person who shaves only every person who does not shave themselves.

  Then convert the logical expression into clausal form and use resolution and factoring to show that the sentence is inconsistent. (Note: factoring reduces two identical literals within a clause to one; if two literals in a clause are unifiable, their most general unifier is applied to the whole clause and then the duplicate literal is deleted.) [10 marks]