Outline

Binary Resolution System, Non-Ground Case

Binary resolution is the following inference rule:

$$\frac{\underline{A} \vee C \quad \underline{\neg B} \vee D}{(C \vee D)mgu(A, B)} \text{ (BR)},$$

Factoring is the following inference rule:

$$\frac{\underline{A} \vee \underline{B} \vee C}{(A \vee C)mgu(A, B)} \text{ (Fact)},$$

Soundness and Completeness

$\mathbb{BR}$ is sound and complete, that is, if a set of clauses is unsatisfiable, then one can derive an empty clause from this set.

Soundness is evident since the conclusion of any inference rule is a logical consequence of its premises.

Completeness can be proved using completeness of propositional resolution and lifting.

Soundness and Completeness

$\mathbb{BR}$ is sound and complete, that is, if a set of clauses is unsatisfiable, then one can derive an empty clause from this set.

Soundness is evident since the conclusion of any inference rule is a logical consequence of its premises.

Completeness can be proved using completeness of propositional resolution and lifting.

Soundness and Completeness

$\mathbb{BR}$ is sound and complete, that is, if a set of clauses is unsatisfiable, then one can derive an empty clause from this set.

Soundness is evident since the conclusion of any inference rule is a logical consequence of its premises.

Completeness can be proved using completeness of propositional resolution and lifting.

Ordered resolution?

Binary resolution with arbitrary selection is incomplete.

To define ordered resolution one has to define ordering for non-ground clauses in a way so that they also work for their ground instances.

Ordered resolution?

Binary resolution with arbitrary selection is incomplete.

To define ordered resolution one has to define ordering for non-ground clauses in a way so that they also work for their ground instances.

A problem

Is the following set of clauses unsatisfiable?

$$p(x, a)$$
$$\neg p(b, x)?$$

Yes, since clauses denote their universal closures:

$$(\forall x)p(x, a)$$
$$(\forall x)\neg p(b, x).$$

But no rule of the resolution system is applicable to these clauses.

A problem

Is the following set of clauses unsatisfiable?

$$
\begin{array}{l}
p(x, a) \\
\neg p(b, x)?
\end{array}
$$

Yes, since clauses denote their universal closures:

$$
\begin{array}{l}
(\forall x) p(x, a) \\
(\forall x) \neg p(b, x).
\end{array}
$$

But no rule of the resolution system is applicable to these clauses.

A problem

Is the following set of clauses unsatisfiable?

$$\begin{array}{l} p(x, a) \\ \neg p(b, x)? \end{array}$$

Yes, since clauses denote their universal closures:

$$\begin{array}{l} (\forall x) p(x, a) \\ (\forall x) \neg p(b, x). \end{array}$$

But no rule of the resolution system is applicable to these clauses.

Renaming away

The domain of a substitution θ is the set of variables $\{x \mid \theta(x) \neq x\}$ is finite.
The range of θ is the set of terms $\{x\theta \mid x\theta \neq x\}$.

A substitution θ is called renaming if (three equivalent characterisations)

- the domain of θ coincides with its range.
- θ has an inverse σ (that is, $\theta \circ \sigma = \sigma \circ \theta = \{\}$).
- there exists an n such that $\theta^n = \{\}$.

A variant of a term (atom, literal, clause) t is any term obtained from t by appying a renaming.

Renaming away

The domain of a substitution θ is the set of variables $\{x \mid \theta(x) \neq x\}$ is finite.
The range of θ is the set of terms $\{x\theta \mid x\theta \neq x\}$.

A substitution θ is called renaming if (three equivalent characterisations)

- the domain of θ coincides with its range.
- θ has an inverse σ (that is, $\theta \circ \sigma = \sigma \circ \theta = \{\}$).
- there exists an n such that $\theta^n = \{\}$.

A variant of a term (atom, literal, clause) t is any term obtained from t by appying a renaming.

Renaming away

The domain of a substitution θ is the set of variables $\{x \mid \theta(x) \neq x\}$ is finite.
The range of θ is the set of terms $\{x\theta \mid x\theta \neq x\}$.

A substitution θ is called renaming if (three equivalent characterisations)

- the domain of θ coincides with its range.
- θ has an inverse σ (that is, $\theta \circ \sigma = \sigma \circ \theta = \{\}$).
- there exists an n such that $\theta^n = \{\}$.

A variant of a term (atom, literal, clause) t is any term obtained from t by appying a renaming.

Hidden rule: renaming away

Renaming E_1 away from E_2: replace E_1 by its variant E_1' so that E_1' and E_2 have no common variables.

Before applying resolution to two clauses C_1 and C_2 we should always **rename** C_1 away from C_2.

Renaming is sometimes called **standardising apart** (especially in the logic programming literature).

Hidden rule: renaming away

Renaming E_1 away from E_2: replace E_1 by its variant E_1' so that E_1' and E_2 have no common variables.

Before applying resolution to two clauses C_1 and C_2 we should always rename C_1 away from C_2.

Renaming is sometimes called standardising apart (especially in the logic programming literature).

Hidden rule: renaming away

Renaming E_1 away from E_2: replace E_1 by its variant E_1' so that E_1' and E_2 have no common variables.

Before applying resolution to two clauses C_1 and C_2 we should always rename C_1 away from C_2.

Renaming is sometimes called standardising apart (especially in the logic programming literature).

Example

(1)	$\neg p(x) \vee \neg q(y)$	input	
(2)	$\neg p(x) \vee q(y)$	input	
(3)	$p(x) \vee \neg q(y)$	input	
(4)	$p(x) \vee q(y)$	input	
(5)	$\neg p(x) \vee \neg p(y)$	BR	(1,2)
(6)	$\neg p(x)$	Fact	(5)
(7)	$p(x) \vee p(y)$	BR	(3,4)
(8)	$p(x)$	Fact	(7)
(9)	$\square$	BR	(6,8)