Lecture 12

Resolution for first-order logic

Unification, resolution

Introduction to Logic for Computer Science

Prof Hongseok Yang
KAIST

These slides are minor variants of those made by Prof Worrell and Dr Haase for their logic course at Oxford.

Drawbacks of ground resolution

Ground resolution good for showing semi-decidability, but bad for practical purposes.

Requires “looking ahead” to see which ground terms will be needed.

Want to instantiate ground terms “by need.”

Topic of this lecture

First-order-logic version of resolution.

Uses so called unification.

Supports by-need grounding.

Forms basis of programming language Prolog.

Substitution

Used to replace variables by σ-terms.

More generally, a substitution is a function θ mapping σ-terms to σ-terms such that

$$c\theta = c,$$
$$f(t_1, \ldots, t_k)\theta = f(t_1\theta, \ldots, t_k\theta).$$

Substitution

Used to replace variables by σ-terms.

More generally, a substitution is a function θ mapping σ-terms to σ-terms such that

$$c\theta = c,$$
$$f(t_1, \ldots, t_k)\theta = f(t_1\theta, \ldots, t_k\theta).$$

Ex1: Show that if θ and θ' work the same for variables, they are equal.

Substitution

Used to replace variables by σ-terms.

More generally, a substitution is a function θ mapping σ-terms to σ-terms such that

$$c\theta = c,$$
$$f(t_1, \ldots, t_k)\theta = f(t_1\theta, \ldots, t_k\theta).$$

Ex1: Show that if θ and θ' work the same for variables, they are equal.

Denote by $\theta \cdot \theta'$ substitution first performing θ and then θ'.

Substitution

Used to replace variables by σ-terms.

More generally, a substitution is a function θ mapping σ-terms to σ-terms such that

$$c\theta = c,$$
$$f(t_1, \ldots, t_k)\theta = f(t_1\theta, \ldots, t_k\theta).$$

Ex1: Show that if θ and θ' work the same for variables, they are equal.

Denote by $\theta \cdot \theta'$ substitution first performing θ and then θ'.

Ex2: Extend θ to arbitrary formulas in first-order logic.

Ex3: Let $\theta = [f(y)/x]$, $\theta' = [g(c, z)/y]$, and $P(x, c)$ be an atomic formula. Compute the following three:

$$P(x, c)\theta, \qquad \theta \cdot \theta', \qquad P(x, c)(\theta \cdot \theta').$$

Ex4: Do you think $F(\theta \cdot \theta') = (F\theta)\theta'$ always?

Unifier and most general unifier

For a set of literals $D = \{L_1, \ldots, L_k\}$, define $D\theta := \{L_1\theta, \ldots, L_k\theta\}$.

θ **unifies** D if $D\theta = \{L\}$ for some literal L.

Unifier and most general unifier

For a set of literals $D = \{L_1, \ldots, L_k\}$, define $D\theta := \{L_1\theta, \ldots, L_k\theta\}$.

θ **unifies** D if $D\theta = \{L\}$ for some literal L.

Example

We have that $\theta = [f(a)/x][a/y]$ unifies $\{P(x), P(f(y))\}$ since

$$\{P(x)\theta, P(f(y))\theta\} = \{P(f(a)), P(f(a))\} = \{P(f(a))\},$$

but $\theta' = [f(y)/x]$ is also unifier.

Unifier and most general unifier

For a set of literals $D = \{L_1, \ldots, L_k\}$, define $D\theta := \{L_1\theta, \ldots, L_k\theta\}$.

θ **unifies** D if $D\theta = \{L\}$ for some literal L.

Example

We have that $\theta = [f(a)/x][a/y]$ unifies $\{P(x), P(f(y))\}$ since

$$\{P(x)\theta, P(f(y))\theta\} = \{P(f(a)), P(f(a))\} = \{P(f(a))\},$$

but $\theta' = [f(y)/x]$ is also unifier. Note that $\theta = \theta' \cdot [a/y]$.

Unifier and most general unifier

For a set of literals $D = \{L_1, \dots, L_k\}$, define $D\theta := \{L_1\theta, \dots, L_k\theta\}$.

θ **unifies** D if $D\theta = \{L\}$ for some literal L.

Example

We have that $\theta = [f(a)/x][a/y]$ unifies $\{P(x), P(f(y))\}$ since

$$\{P(x)\theta, P(f(y))\theta\} = \{P(f(a)), P(f(a))\} = \{P(f(a))\},$$

but $\theta' = [f(y)/x]$ is also unifier. Note that $\theta = \theta' \cdot [a/y]$.

Definition

We call θ a **most general unifier (mgu)** of D if θ is a unifier and for all other unifiers θ' there is a substitution θ'' such that $\theta' = \theta \cdot \theta''$.

Unifier and most general unifier

For a set of literals $D = \{L_1, \ldots, L_k\}$, define $D\theta := \{L_1\theta, \ldots, L_k\theta\}$.

θ **unifies** D if $D\theta = \{L\}$ for some literal L.

Example

We have that $\theta = [f(a)/x][a/y]$ unifies $\{P(x), P(f(y))\}$ since

$$\{P(x)\theta, P(f(y))\theta\} = \{P(f(a)), P(f(a))\} = \{P(f(a))\},$$

but $\theta' = [f(y)/x]$ is also unifier. Note that $\theta = \theta' \cdot [a/y]$.

Definition

We call θ a **most general unifier (mgu)** of D if θ is a unifier and for all other unifiers θ' there is a substitution θ'' such that $\theta' = \theta \cdot \theta''$.

Ex1: Is an mgu unique?

Ex2: When doesn't it exist? Find examples.

Ex3: Assume that D is unifiable. Does an mgu exist in this case?

Theorem (Unification Theorem)

A unifiable set of literals D has a most general unifier.

Theorem (Unification Theorem)

A unifiable set of literals D has a most general unifier.

Proof by an algorithm.

Theorem (Unification Theorem)

A unifiable set of literals D has a most general unifier.

Proof by an algorithm.

Unification Algorithm

Input: Set of literals D
Output: Either a most general unifier θ of D or “fail”

$\theta :=$ identity substitution
while θ is not a unifier of D **do**
  pick two distinct literals in $D\theta$ and
    find the left-most positions at which they differ
  if one of the corresponding sub-terms is variable x and
    the other term t does not contain x
  then $\theta := \theta \cdot [t/x]$ **else** output “fail” and halt
od
output θ

Theorem (Unification Theorem)

A unifiable set of literals D has a most general unifier.

Proof by an algorithm.

Unification Algorithm

Input: Set of literals D
Output: Either a most general unifier θ of D or "fail"

$\theta :=$ identity substitution
while θ is not a unifier of D **do**
  pick two distinct literals in $D\theta$ and
    find the left-most positions at which they differ
  if one of the corresponding sub-terms is variable x and
    the other term t does not contain x
  then $\theta := \theta \cdot [t/x]$ **else** output "fail" and halt
od
output θ

Ex1: Run the algo. for $\{P(x), P(f(y))\}$ and $\{P(x, y), P(f(z), x)\}$.

Ex2: Why correct? Why terminate? Loop invariant?

Proof of unification theorem

Termination: if we do not halt, a variable x gets replaced by a term in which x does not occur.

Proof of unification theorem

Termination: if we do not halt, a variable x gets replaced by a term in which x does not occur. Ex: Why does this imply termination?

Proof of unification theorem

Termination: if we do not halt, a variable *x* gets replaced by a term in which *x* does not occur. Ex: Why does this imply termination?

Loop invariant: for any unifier θ' of *D*, have $\theta' = \theta \cdot \theta'$.

- Clearly holds before entering loop.

Proof of unification theorem

Termination: if we do not halt, a variable *x* gets replaced by a term in which *x* does not occur. Ex: Why does this imply termination?

Loop invariant: for any unifier θ' of *D*, have $\theta' = \theta \cdot \theta'$.

- Clearly holds before entering loop.
- Suppose the algo. finds *x* and *t* in $D\theta$.

Proof of unification theorem

Termination: if we do not halt, a variable *x* gets replaced by a term in which *x* does not occur. Ex: Why does this imply termination?

Loop invariant: for any unifier θ' of *D*, have $\theta' = \theta \cdot \theta'$.

- Clearly holds before entering loop.
- Suppose the algo. finds *x* and *t* in $D\theta$. Ex: Complete the rest.

Proof of unification theorem

Termination: if we do not halt, a variable x gets replaced by a term in which x does not occur. Ex: Why does this imply termination?

Loop invariant: for any unifier θ' of D, have $\theta' = \theta \cdot \theta'$.

- Clearly holds before entering loop.
- Suppose the algo. finds x and t in $D\theta$. Ex: Complete the rest.
- Since θ' unifies D and $\theta' = \theta \cdot \theta'$ by ind. hypo., we have $t\theta' = x\theta'$.

Proof of unification theorem

Termination: if we do not halt, a variable x gets replaced by a term in which x does not occur. Ex: Why does this imply termination?

Loop invariant: for any unifier θ' of D, have $\theta' = \theta \cdot \theta'$.

- Clearly holds before entering loop.
- Suppose the algo. finds x and t in $D\theta$. Ex: Complete the rest.
- Since θ' unifies D and $\theta' = \theta \cdot \theta'$ by ind. hypo., we have $t\theta' = x\theta'$.
- This implies $\theta' = [t/x] \cdot \theta'$.

Proof of unification theorem

Termination: if we do not halt, a variable x gets replaced by a term in which x does not occur. Ex: Why does this imply termination?

Loop invariant: for any unifier θ' of D, have $\theta' = \theta \cdot \theta'$.

- Clearly holds before entering loop.
- Suppose the algo. finds x and t in $D\theta$. Ex: Complete the rest.
- Since θ' unifies D and $\theta' = \theta \cdot \theta'$ by ind. hypo., we have $t\theta' = x\theta'$.
- This implies $\theta' = [t/x] \cdot \theta'$.
- Thus, $\theta' = \theta \cdot \theta' = \theta \cdot ([t/x] \cdot \theta') = (\theta \cdot [t/x]) \cdot \theta'$.

Proof of unification theorem

Termination: if we do not halt, a variable x gets replaced by a term in which x does not occur. Ex: Why does this imply termination?

Loop invariant: for any unifier θ' of D, have $\theta' = \theta \cdot \theta'$.

- Clearly holds before entering loop.
- Suppose the algo. finds x and t in $D\theta$. Ex: Complete the rest.
- Since θ' unifies D and $\theta' = \theta \cdot \theta'$ by ind. hypo., we have $t\theta' = x\theta'$.
- This implies $\theta' = [t/x] \cdot \theta'$.
- Thus, $\theta' = \theta \cdot \theta' = \theta \cdot ([t/x] \cdot \theta') = (\theta \cdot [t/x]) \cdot \theta'$.

Ex: Use the invariant and prove the theorem.

Proof of unification theorem

Termination: if we do not halt, a variable x gets replaced by a term in which x does not occur. Ex: Why does this imply termination?

Loop invariant: for any unifier θ' of D, have $\theta' = \theta \cdot \theta'$.

- Clearly holds before entering loop.
- Suppose the algo. finds x and t in $D\theta$. Ex: Complete the rest.
- Since θ' unifies D and $\theta' = \theta \cdot \theta'$ by ind. hypo., we have $t\theta' = x\theta'$.
- This implies $\theta' = [t/x] \cdot \theta'$.
- Thus, $\theta' = \theta \cdot \theta' = \theta \cdot ([t/x] \cdot \theta') = (\theta \cdot [t/x]) \cdot \theta'$.

Ex: Use the invariant and prove the theorem.

At abnormal termination, the loop inv. implies that D is not unifiable.

At normal termination, θ is a unifier. The loop inv. implies θ is an mgu.

First-order-logic resolution

For a set of literals D, $\overline{D}$ denotes the complements of all literals in D.

First-order-logic resolution

For a set of literals D, $\overline{D}$ denotes the complements of all literals in D.

Definition (First-order-logic resolution)

Let C_1, C_2 be clauses with no variables in common.

R is a **resolvent** of C_1 and C_2 if there are $D_1 \subseteq C_1$ and $D_2 \subseteq C_2$ s.t.

- $D_1 \cup \overline{D_2}$ has an mgu θ, and
- $R = (C_1\theta \setminus \{L\}) \cup (C_2\theta \setminus \{\overline{L}\})$ with $L = D_1\theta$ and $\overline{L} = D_2\theta$.

First-order-logic resolution

For a set of literals D, $\overline{D}$ denotes the complements of all literals in D.

Definition (First-order-logic resolution)

Let C_1, C_2 be clauses with no variables in common.

R is a **resolvent** of C_1 and C_2 if there are $D_1 \subseteq C_1$ and $D_2 \subseteq C_2$ s.t.

- $D_1 \cup \overline{D_2}$ has an mgu θ, and
- $R = (C_1\theta \setminus \{L\}) \cup (C_2\theta \setminus \{\overline{L}\})$ with $L = D_1\theta$ and $\overline{L} = D_2\theta$.

If C_1, C_2 have variables in common, R is a **resolvent** if there are renamings θ_1, θ_2 s.t.

- $C_1\theta_1, C_2\theta_2$ have no variables in common, and
- R is a resolvent of $C_1\theta_1$ and $C_2\theta_2$.

Example

Ex: Given the signature with constant symbol e, unary function symbols f and g, and ternary predicate symbol P, compute a resolvent of

$$C_1 = \{\neg P(f(e), x, f(g(e)))\} \text{ and } C_2 = \{\neg P(x, y, z), P(f(x), y, f(z))\}.$$

Example

Ex: Given the signature with constant symbol e, unary function symbols f and g, and ternary predicate symbol P, compute a resolvent of

$$C_1 = \{\neg P(f(e), x, f(g(e)))\} \text{ and } C_2 = \{\neg P(x, y, z), P(f(x), y, f(z))\}.$$

$$\begin{tikzcd}
\{\neg P(f(e), x, f(g(e)))\} \arrow[d, dash, "{[u/x]}"] & & \\
\{\neg P(f(e), u, f(g(e)))\} \arrow[dr, dash] & {[e/x][u/y][g(e)/z]} & \{\neg P(x, y, z),\ P(f(x), y, f(z))\} \arrow[dl, dash] \\
& \{\neg P(e, u, g(e))\} &
\end{tikzcd}$$

Figure: First-order-logic resolution example.

Resolution derivation

Use resolution in order to derive a clause C from a set of clauses F.

A **derivation** of C is a sequence of clauses $C_1, \ldots, C_m$ such that

- $C = C_m$; and
- each C_i is either a clause from F possibly with variable renaming or obtained from resolution of C_j and C_k for some $j, k < i$.

$\mathrm{Res}^*(F)$ is the set of all clauses derivable from F.

Putting it all together

Let

$$F_1 = \forall x\, A(e, x, x),$$
$$F_2 = \forall x \forall y \forall z\, (\neg A(x, y, z) \vee A(s(x), y, s(z))),$$
$$F_3 = \forall x \exists y\, A(s(s(e)), x, y).$$

Ex: Prove that $F_1 \wedge F_2 \models F_3$, i.e. $F_1 \wedge F_2 \wedge \neg F_3$ is unsatisfiable. Use Skolemisation and first-order-logic resolution.

Putting it all together

$$
\begin{aligned}
F_1 &= \forall x\, A(e, x, x),\\
F_2 &= \forall x \forall y \forall z\, (\neg A(x, y, z) \vee A(s(x), y, s(z))),\\
F_3 &= \forall x \exists y\, A(s(s(e)), x, y).
\end{aligned}
$$

Step 1: Skolemise each F_i separately.

$$\neg F_3 = \exists y \forall z\, \neg A(s(s(e)), y, z) \rightsquigarrow G_3 := \forall z\, \neg A(s(s(e)), c, z)$$

Step 2: Use resolution to derive the empty clause.

1. $\{\neg A(s(s(e)), c, z_1)\}$	clause of G_3
2. $\{\neg A(x_2, y_2, z_2), A(s(x_2), y_2, s(z_2))\}$	clause of F_2
3. $\{\neg A(s(e), c, z_3)\}$	1,2 Res. with sub. $[s(e)/x_2][c/y_2][s(z_2)/z_1][z_3/z_2]$
4. $\{\neg A(e, c, z_4)\}$	2,3 Res. with sub. $[e/x_2][c/y_2][s(z_2)/z_3][z_4/z_2]$
5. $\{A(e, y_5, y_5)\}$	clause of F_1
6. $\Box$	4,5 Res. with sub. $[c/y_5][c/z_4]$

Lemma (Resolution Lemma)

Let $F = \forall x_1 \ldots \forall x_n G$ be a closed formula in Skolem form, with G quantifier-free and CNF. Let R be a resolvent of two clauses in G. Then $F \equiv \forall^*(G \cup \{R\})$.[a]

---

[a] For a first-order formula G', we write $\forall^* G'$ for $\forall y_1 \ldots \forall y_n\, G'$ where $y_1, \ldots, y_n$ are all the free variables of G'.

Lemma (Resolution Lemma)

Let $F = \forall x_1 \ldots \forall x_n G$ be a closed formula in Skolem form, with G quantifier-free and CNF. Let R be a resolvent of two clauses in G. Then $F \equiv \forall^*(G \cup \{R\})$.[a]

[a] For a first-order formula G', we write $\forall^* G'$ for $\forall y_1 \ldots \forall y_n\, G'$ where $y_1, \ldots, y_n$ are all the free variables of G'.

Ex: Prove the lemma.

Lemma (Resolution Lemma)

Let $F = \forall x_1 \ldots \forall x_n G$ be a closed formula in Skolem form, with G quantifier-free and CNF. Let R be a resolvent of two clauses in G. Then $F \equiv \forall^*(G \cup \{R\})$. [a]

[a] For a first-order formula G', we write $\forall^* G'$ for $\forall y_1 \ldots \forall y_n\, G'$ where $y_1, \ldots, y_n$ are all the free variables of G'.

Ex: Prove the lemma.

Proof.

Clearly $\forall^*(G \cup \{R\}) \models F$.

Lemma (Resolution Lemma)

Let $F = \forall x_1 \ldots \forall x_n G$ be a closed formula in Skolem form, with G quantifier-free and CNF. Let R be a resolvent of two clauses in G. Then $F \equiv \forall^*(G \cup \{R\})$.[a]

[a] For a first-order formula G', we write $\forall^* G'$ for $\forall y_1 \ldots \forall y_n\, G'$ where $y_1, \ldots, y_n$ are all the free variables of G'.

Ex: Prove the lemma.

Proof.

Clearly $\forall^*(G \cup \{R\}) \models F$.

For converse direction, show that $F \models R$. Suppose R is a resolvent of clauses $C_1, C_2 \in G$, with $R = (C_1\theta \setminus \{L\}) \cup (C_2\theta' \setminus \{\overline{L}\})$ for some substitutions θ, θ' and complementary literals $L \in C_1\theta$ and $\overline{L} \in C_2\theta'$.

Lemma (Resolution Lemma)

Let $F = \forall x_1 \ldots \forall x_n G$ be a closed formula in Skolem form, with G quantifier-free and CNF. Let R be a resolvent of two clauses in G. Then $F \equiv \forall^*(G \cup \{R\})$.[a]

[a] For a first-order formula G', we write $\forall^* G'$ for $\forall y_1 \ldots \forall y_n\, G'$ where $y_1, \ldots, y_n$ are all the free variables of G'.

Ex: Prove the lemma.

Proof.

Clearly $\forall^*(G \cup \{R\}) \models F$.

For converse direction, show that $F \models R$. Suppose R is a resolvent of clauses $C_1, C_2 \in G$, with $R = (C_1\theta \setminus \{L\}) \cup (C_2\theta' \setminus \{\overline{L}\})$ for some substitutions θ, θ' and complementary literals $L \in C_1\theta$ and $\overline{L} \in C_2\theta'$.

Let $\mathcal{A}$ be an assignment that satisfies $F = \forall^* G$. Since $C_1, C_2 \in G$, by the Translation Lemma $\mathcal{A} \models C_1\theta$ and $\mathcal{A} \models C_2\theta'$.

Lemma (Resolution Lemma)

Let $F = \forall x_1 \ldots \forall x_n G$ be a closed formula in Skolem form, with G quantifier-free and CNF. Let R be a resolvent of two clauses in G. Then $F \equiv \forall^*(G \cup \{R\})$.[a]

[a] For a first-order formula G', we write $\forall^* G'$ for $\forall y_1 \ldots \forall y_n\, G'$ where $y_1, \ldots, y_n$ are all the free variables of G'.

Ex: Prove the lemma.

Proof.

Clearly $\forall^*(G \cup \{R\}) \models F$.

For converse direction, show that $F \models R$. Suppose R is a resolvent of clauses $C_1, C_2 \in G$, with $R = (C_1\theta \setminus \{L\}) \cup (C_2\theta' \setminus \{\overline{L}\})$ for some substitutions θ, θ' and complementary literals $L \in C_1\theta$ and $\overline{L} \in C_2\theta'$.

Let $\mathcal{A}$ be an assignment that satisfies $F = \forall^* G$. Since $C_1, C_2 \in G$, by the Translation Lemma $\mathcal{A} \models C_1\theta$ and $\mathcal{A} \models C_2\theta'$.

Moreover, since $\mathcal{A}$ satisfies at most one of the complementary literals L and $\overline{L}$, $\mathcal{A}$ satisfies at least one of $C_1\theta \setminus \{L\}$ and $C_2\theta' \setminus \{\overline{L}\}$.

Lemma (Resolution Lemma)

Let $F = \forall x_1 \ldots \forall x_n G$ be a closed formula in Skolem form, with G quantifier-free and CNF. Let R be a resolvent of two clauses in G. Then $F \equiv \forall^*(G \cup \{R\})$.[a]

[a] For a first-order formula G', we write $\forall^* G'$ for $\forall y_1 \ldots \forall y_n\, G'$ where $y_1, \ldots, y_n$ are all the free variables of G'.

Ex: Prove the lemma.

Proof.

Clearly $\forall^*(G \cup \{R\}) \models F$.

For converse direction, show that $F \models R$. Suppose R is a resolvent of clauses $C_1, C_2 \in G$, with $R = (C_1\theta \setminus \{L\}) \cup (C_2\theta' \setminus \{\overline{L}\})$ for some substitutions θ, θ' and complementary literals $L \in C_1\theta$ and $\overline{L} \in C_2\theta'$.

Let $\mathcal{A}$ be an assignment that satisfies $F = \forall^* G$. Since $C_1, C_2 \in G$, by the Translation Lemma $\mathcal{A} \models C_1\theta$ and $\mathcal{A} \models C_2\theta'$.

Moreover, since $\mathcal{A}$ satisfies at most one of the complementary literals L and $\overline{L}$, $\mathcal{A}$ satisfies at least one of $C_1\theta \setminus \{L\}$ and $C_2\theta' \setminus \{\overline{L}\}$.

Thus, $\mathcal{A}$ satisfies R, as required. $\square$

Soundness

Lemma (Resolution Lemma)

Let $F = \forall x_1 \dots \forall x_n G$ be a closed formula in Skolem form, with G quantifier-free and CNF. Let R be a resolvent of two clauses in G. Then $F \equiv \forall^*(G \cup \{R\})$.[a]

[a] For a first-order formula G', we write $\forall^* G'$ for $\forall y_1 \dots \forall y_n\, G'$ where $y_1, \dots, y_n$ are all the free variables of G'.

Theorem (Soundness)

Let $F = \forall x_1 \dots \forall x_n G$ be a closed formula in Skolem form with G quantifier-free and CNF. If there is a resolution derivation of $\Box$ from G, then F is unsatisfiable.

Ex: Prove the theorem using the Resolution Lemma.

Completeness

Theorem (Completeness)

Let $F = \forall x_1 \ldots \forall x_n G$ be a closed formula in Skolem form with G quantifier-free and CNF. If F is unsatisfiable, there is a resolution derivation of $\Box$ from G.

Completeness

Theorem (Completeness)

Let $F = \forall x_1 \ldots \forall x_n G$ be a closed formula in Skolem form with G quantifier-free and CNF. If F is unsatisfiable, there is a resolution derivation of $\square$ from G.

Ex1: Prove the Completeness Theorem.

Completeness

Theorem (Completeness)

Let $F = \forall x_1 \ldots \forall x_n G$ be a closed formula in Skolem form with G quantifier-free and CNF. If F is unsatisfiable, there is a resolution derivation of $\Box$ from G.

Ex1: Prove the Completeness Theorem.

Hint: Use the Lifting Lemma below.

Lemma (Lifting Lemma)

Let C_1 and C_2 be clauses with respective ground instances D_1 and D_2. Suppose that R is a propositional resolvent of D_1 and D_2. Then, C_1 and C_2 have a first-order-logic resolvent R' such that R is a ground instance of R'.

Completeness

Theorem (Completeness)

Let $F = \forall x_1 \ldots \forall x_n G$ be a closed formula in Skolem form with G quantifier-free and CNF. If F is unsatisfiable, there is a resolution derivation of $\square$ from G.

Ex1: Prove the Completeness Theorem.

Hint: Use the Lifting Lemma below.

Lemma (Lifting Lemma)

Let C_1 and C_2 be clauses with respective ground instances D_1 and D_2. Suppose that R is a propositional resolvent of D_1 and D_2. Then, C_1 and C_2 have a first-order-logic resolvent R' such that R is a ground instance of R'.

Ex2: Prove the Lifting Lemma.