Resolution Method; Predicate Logic

Stepan Kuznetsov

Discrete Math Bridging Course, HSE University

Satisfiability

- We continue discussing satisfiability of Boolean formula.

Satisfiability

- We continue discussing satisfiability of Boolean formula.
- A satisfying assignment is an assignment of 0's and 1's to variables, which makes the formula true (value = 1).

Satisfiability

- We continue discussing **satisfiability** of Boolean formula.
- A satisfying assignment is an assignment of 0's and 1's to variables, which makes the formula true (value = 1).
- Satisfiability is a model example of a very general situation of **finding** (more precisely: checking for existence) an object with given properties.

Resolution Method

- Recall that **resolution method** is a method of determining whether a Boolean formula given in CNF is satisfiable.

Resolution Method

- Recall that **resolution method** is a method of determining whether a Boolean formula given in CNF is satisfiable.
- A **CNF** is a conjunction of clauses, where each clause is a disjunction of literals (e.g., $\overline{x} \vee y \vee \overline{z}$).

Resolution Method

- Recall that **resolution method** is a method of determining whether a Boolean formula given in CNF is satisfiable.
- A **CNF** is a conjunction of clauses, where each clause is a disjunction of literals (e.g., $\overline{x} \vee y \vee \overline{z}$).
- The algorithm **saturates** the CNF by adding all clauses which can be generated by the **resolution rule:**

$$\frac{A \vee p \quad B \vee \overline{p}}{A \vee B}$$

Resolution Method

- If the empty clause ($\perp$) got obtained, the CNF is not satisfiable (because the resolution rule keeps validity).

Resolution Method

- If the empty clause ($\bot$) got obtained, the CNF is not satisfiable (because the resolution rule keeps validity).
- Moreover, by **completeness theorem** this is a criterion: if the empty clause is not obtained, the CNF **is** satisfiable.

Resolution Method

- However, the non-derivability of the empty clause does not give us the satisfying assignment itself.

Resolution Method

- However, the non-derivability of the empty clause does not give us the satisfying assignment itself.
- In other words, the method solves the **decision problems** (“yes”/“no”), but not the **search problem.**

Resolution Method

- However, the non-derivability of the empty clause does not give us the satisfying assignment itself.
- In other words, the method solves the **decision problems** (“yes”/“no”), but not the **search problem.**
- If we are lucky enough, and the CNF has only one satisfying assignment, then after saturation we get **isolated** literals (like x or $\overline{y}$, for example), which dictate the desired satisfying assignment (e.g., $x = 1$ or $y = 0$).

Resolution Method

- In other cases, we can use the following consideration.

Resolution Method

- In other cases, we can use the following consideration.

Proposition

If a saturated CNF $\mathcal{S}$ includes neither $\perp$ nor $\overline{x}$ as an isolated literal, then $\mathcal{S} \wedge x$ is also satisfiable. Same for swapping x and $\overline{x}$.

Resolution Method

- In other cases, we can use the following consideration.

Proposition

If a saturated CNF $\mathcal{S}$ includes neither $\bot$ nor $\overline{x}$ as an isolated literal, then $\mathcal{S} \wedge x$ is also satisfiable. Same for swapping x and $\overline{x}$.

- In particular, if $\mathcal{S}$ is satisfiable and includes neither x nor $\overline{x}$, we can make an **arbitrary choice** for the value of x.

Resolution Method

- However, after making this arbitrary choice, we have to saturate $\mathcal{S} \wedge x$ (or $\mathcal{S} \wedge \overline{x}$) **again** before considering another variable.

Resolution Method

- However, after making this arbitrary choice, we have to saturate $\mathcal{S} \wedge x$ (or $\mathcal{S} \wedge \overline{x}$) **again** before considering another variable.
- For example, the CNF $(x \vee \overline{y}) \wedge (x \vee z)$ is saturated, but choosing $x = 0$ (adding $\overline{x}$) allows new resolutions giving $\overline{y}$ and z, and thus dictating values for all other variables.

Resolution Method

Proposition

If a saturated CNF $\mathcal{S}$ includes neither $\perp$ nor $\overline{x}$ as an isolated literal, then $\mathcal{S} \wedge x$ is also satisfiable. Same for swapping x and $\overline{x}$.

Resolution Method

Proposition

If a saturated CNF $\mathcal{S}$ includes neither $\perp$ nor $\overline{x}$ as an isolated literal, then $\mathcal{S} \wedge x$ is also satisfiable. Same for swapping x and $\overline{x}$.

- The proof of the proposition is easy.

Resolution Method

Proposition

If a saturated CNF $\mathcal{S}$ includes neither $\perp$ nor $\overline{x}$ as an isolated literal, then $\mathcal{S} \wedge x$ is also satisfiable. Same for swapping x and $\overline{x}$.

- The proof of the proposition is easy.
- Indeed, new resolutions applied when we saturate $\mathcal{S} \wedge x$, should involve x.

Resolution Method

Proposition

If a saturated CNF $\mathcal{S}$ includes neither $\bot$ nor $\overline{x}$ as an isolated literal, then $\mathcal{S} \wedge x$ is also satisfiable. Same for swapping x and $\overline{x}$.

- The proof of the proposition is easy.
- Indeed, new resolutions applied when we saturate $\mathcal{S} \wedge x$, should involve x.
- Therefore, if such a resolution generates $\bot$, there should have been $\overline{x}$ in the original $\mathcal{S}$.

Example

$$(\overline{p} \vee r \vee s), (\overline{r} \vee q), (\overline{s} \vee \overline{p} \vee z), (\overline{z} \vee t), p$$

Example

$$(\overline{p} \vee r \vee s), (\overline{r} \vee q), (\overline{s} \vee \overline{p} \vee z), (\overline{z} \vee t), p$$

$$(r \vee s), (\overline{s} \vee z), (\overline{p} \vee s \vee q), (\overline{p} \vee r \vee z), (\overline{s} \vee \overline{p} \vee t),$$

Example

$$(\overline{p} \vee r \vee s), (\overline{r} \vee q), (\overline{s} \vee \overline{p} \vee z), (\overline{z} \vee t), p$$

$$(r \vee s), (\overline{s} \vee z), (\overline{p} \vee s \vee q), (\overline{p} \vee r \vee z), (\overline{s} \vee \overline{p} \vee t),$$
$$(r \vee z), (\overline{p} \vee q \vee z), (s \vee q), (\overline{s} \vee t), (r \vee \overline{p} \vee t),$$

Example

$$(\overline{p} \vee r \vee s), (\overline{r} \vee q), (\overline{s} \vee \overline{p} \vee z), (\overline{z} \vee t), p$$

$$(r \vee s), (\overline{s} \vee z), (\overline{p} \vee s \vee q), (\overline{p} \vee r \vee z), (\overline{s} \vee \overline{p} \vee t),$$
$$(r \vee z), (\overline{p} \vee q \vee z), (s \vee q), (\overline{s} \vee t), (r \vee \overline{p} \vee t),$$
$$(q \vee t), (z \vee q), (r \vee t)$$

Example

$$(\overline{p} \vee r \vee s), (\overline{r} \vee q), (\overline{s} \vee \overline{p} \vee z), (\overline{z} \vee t), p$$

$$(r \vee s), (\overline{s} \vee z), (\overline{p} \vee s \vee q), (\overline{p} \vee r \vee z), (\overline{s} \vee \overline{p} \vee t),$$
$$(r \vee z), (\overline{p} \vee q \vee z), (s \vee q), (\overline{s} \vee t), (r \vee \overline{p} \vee t),$$
$$(q \vee t), (z \vee q), (r \vee t)$$

Example

$$(\overline{p} \vee r \vee s), (\overline{r} \vee q), (\overline{s} \vee \overline{p} \vee z), (\overline{z} \vee t), {\color{blue} p}$$

$$(r \vee s), (\overline{s} \vee z), (\overline{p} \vee s \vee q), (\overline{p} \vee r \vee z), (\overline{s} \vee \overline{p} \vee t),$$
$$(r \vee z), (\overline{p} \vee q \vee z), (s \vee q), (\overline{s} \vee t), (r \vee \overline{p} \vee t),$$
$$(q \vee t), (z \vee q), (r \vee t)$$

${\color{red} s}$

Example

$$(\overline{p} \vee r \vee s), (\overline{r} \vee q), (\overline{s} \vee \overline{p} \vee z), (\overline{z} \vee t), p$$

$$(r \vee s), (\overline{s} \vee z), (\overline{p} \vee s \vee q), (\overline{p} \vee r \vee z), (\overline{s} \vee \overline{p} \vee t),$$
$$(r \vee z), (\overline{p} \vee q \vee z), (s \vee q), (\overline{s} \vee t), (r \vee \overline{p} \vee t),$$
$$(q \vee t), (z \vee q), (r \vee t)$$

$$s, z, t, (\overline{p} \vee z), (\overline{p} \vee t)$$

$$\ldots$$

Resolution for 2-CNF

- If clauses include at least 3 literals, resolution can lead to growth:

$$\frac{x \vee \overline{y} \vee p \quad z \vee w \vee \overline{p}}{x \vee \overline{y} \vee z \vee w}$$

Resolution for 2-CNF

- If clauses include at least 3 literals, resolution can lead to growth:

$$\frac{x \vee \overline{y} \vee p \quad z \vee w \vee \overline{p}}{x \vee \overline{y} \vee z \vee w}$$

- This makes saturation a potentially exponential procedure.

Resolution for 2-CNF

- If clauses include at least 3 literals, resolution can lead to growth:

$$\frac{x \vee \overline{y} \vee p \quad z \vee w \vee \overline{p}}{x \vee \overline{y} \vee z \vee w}$$

- This makes saturation a potentially exponential procedure.
- However, for 2-CNF (each clause includes no more than 2 literals) the clauses do not grow:

$$\frac{x \vee p \quad \overline{z} \vee \overline{p}}{x \vee \overline{z}}$$

Resolution for 2-CNF

- Thus, the total number of possible clauses does not exceed $4n^2 + 2n + 1$, where n is the number of variables.

Resolution for 2-CNF

- Thus, the total number of possible clauses does not exceed $4n^2 + 2n + 1$, where n is the number of variables.
- This makes the saturation process **polynomial.**

Resolution for 2-CNF

- Thus, the total number of possible clauses does not exceed $4n^2 + 2n + 1$, where n is the number of variables.
- This makes the saturation process **polynomial.**
- This can be organized as follows: take each clause from the list, starting from the second one, and try to resolve it against eariler ones. Does it give a new clause?

Resolution for 2-CNF

- Thus, the total number of possible clauses does not exceed $4n^2 + 2n + 1$, where n is the number of variables.
- This makes the saturation process **polynomial.**
- This can be organized as follows: take each clause from the list, starting from the second one, and try to resolve it against eariler ones. Does it give a new clause?
- New clauses are added to the bottom of the list.

Resolution: Completeness Proof

Theorem

If one cannot obtain the empty clause by applying resolutions, starting from the given CNF, then the CNF is satisfiable.

Resolution: Completeness Proof

Theorem

If one cannot obtain the empty clause by applying resolutions, starting from the given CNF, then the CNF is satisfiable.

- We prove this theorem using **induction** on the number of variables.

Resolution: Completeness Proof

Theorem

If one cannot obtain the empty clause by applying resolutions, starting from the given CNF, then the CNF is satisfiable.

- We prove this theorem using **induction** on the number of variables.
- That is, we establish it for zero variables (trivial) and then validate the **step** from n to $n+1$ variables.

Resolution: Completeness Proof

- Zero variables: the only possible clause is $\bot$, therefore, our CNF is empty.

Resolution: Completeness Proof

- Zero variables: the only possible clause is $\perp$, therefore, our CNF is empty.
- From n to $n+1$. Let the extra variable be $p_{n+1} = q$ and let $\mathcal{S}$ denote the saturation of our CNF.

Resolution: Completeness Proof

- Zero variables: the only possible clause is $\bot$, therefore, our CNF is empty.
- From n to $n+1$. Let the extra variable be $p_{n+1} = q$ and let $\mathcal{S}$ denote the saturation of our CNF.
- Take all clauses which do not include $\overline{q}$, and remove q out of them. This gives $\mathcal{S}^+$.

Resolution: Completeness Proof

- Zero variables: the only possible clause is $\bot$, therefore, our CNF is empty.
- From n to $n+1$. Let the extra variable be $p_{n+1} = q$ and let $\mathcal{S}$ denote the saturation of our CNF.
- Take all clauses which do not include $\overline{q}$, and remove q out of them. This gives $\mathcal{S}^+$.
- Dually, take clauses without q and remove $\overline{q}$. This gives $\mathcal{S}^-$.

Resolution: Completeness Proof

- Both $\mathcal{S}^+$ and $\mathcal{S}^-$ are saturated.

Resolution: Completeness Proof

- Both $\mathcal{S}^+$ and $\mathcal{S}^-$ are saturated.
  - Indeed, any new resolution in $\mathcal{S}^+$ or in $\mathcal{S}^-$ would induce a resolution in $\mathcal{S}$.

Resolution: Completeness Proof

- Both $\mathcal{S}^+$ and $\mathcal{S}^-$ are saturated.
  - Indeed, any new resolution in $\mathcal{S}^+$ or in $\mathcal{S}^-$ would induce a resolution in $\mathcal{S}$.
- Let us show that **at least one** of $\mathcal{S}^+$ and $\mathcal{S}^-$ is satisfiable.

Resolution: Completeness Proof

- Both $\mathcal{S}^+$ and $\mathcal{S}^-$ are saturated.
  - Indeed, any new resolution in $\mathcal{S}^+$ or in $\mathcal{S}^-$ would induce a resolution in $\mathcal{S}$.
- Let us show that **at least one** of $\mathcal{S}^+$ and $\mathcal{S}^-$ is satisfiable.
  - Suppose, both $\mathcal{S}^+$ and $\mathcal{S}^-$ include $\perp$.

Resolution: Completeness Proof

- Both $\mathcal{S}^+$ and $\mathcal{S}^-$ are saturated.
  - Indeed, any new resolution in $\mathcal{S}^+$ or in $\mathcal{S}^-$ would induce a resolution in $\mathcal{S}$.
- Let us show that **at least one** of $\mathcal{S}^+$ and $\mathcal{S}^-$ is satisfiable.
  - Suppose, both $\mathcal{S}^+$ and $\mathcal{S}^-$ include $\perp$.
  - Then $\mathcal{S}$ includes both q and $\overline{q}$, and therefore $\perp$. Contradiction.

Resolution: Completeness Proof

- Both $\mathcal{S}^+$ and $\mathcal{S}^-$ are saturated.
  - Indeed, any new resolution in $\mathcal{S}^+$ or in $\mathcal{S}^-$ would induce a resolution in $\mathcal{S}$.
- Let us show that **at least one** of $\mathcal{S}^+$ and $\mathcal{S}^-$ is satisfiable.
  - Suppose, both $\mathcal{S}^+$ and $\mathcal{S}^-$ include $\perp$.
  - Then $\mathcal{S}$ includes both q and $\overline{q}$, and therefore $\perp$. Contradiction.
  - Since $\mathcal{S}^+$ and $\mathcal{S}^-$ use only $p_1, \dots, p_n$, we already know our theorem for them.

Resolution: Completeness Proof

- Both $\mathcal{S}^+$ and $\mathcal{S}^-$ are saturated.
    - Indeed, any new resolution in $\mathcal{S}^+$ or in $\mathcal{S}^-$ would induce a resolution in $\mathcal{S}$.
- Let us show that **at least one** of $\mathcal{S}^+$ and $\mathcal{S}^-$ is satisfiable.
    - Suppose, both $\mathcal{S}^+$ and $\mathcal{S}^-$ include $\bot$.
    - Then $\mathcal{S}$ includes both q and $\overline{q}$, and therefore $\bot$. Contradiction.
    - Since $\mathcal{S}^+$ and $\mathcal{S}^-$ use only $p_1, \dots, p_n$, we already know our theorem for them.
    - The one which does not include $\bot$ is satisfiable.

Resolution: Completeness Proof

- If $\mathcal{S}^+$ is satisfiable, take the satisfying assignment and let $q = 0$.

Resolution: Completeness Proof

- If $\mathcal{S}^+$ is satisfiable, take the satisfying assignment and let $q = 0$.
- Clauses without $\overline{q}$ are already satisfied via $\mathcal{S}^+$.

Resolution: Completeness Proof

- If $\mathcal{S}^+$ is satisfiable, take the satisfying assignment and let $q = 0$.
- Clauses without $\overline{q}$ are already satisfied via $\mathcal{S}^+$.
- Clauses with $\overline{q}$ are satisfied by $\overline{q} = 1$.

Resolution: Completeness Proof

- If $\mathcal{S}^+$ is satisfiable, take the satisfying assignment and let $q = 0$.
- Clauses without $\overline{q}$ are already satisfied via $\mathcal{S}^+$.
- Clauses with $\overline{q}$ are satisfied by $\overline{q} = 1$.
- Dually, if $\mathcal{S}^-$ is satisfiable, take $q = 1$.

Beyond Propositional: Predicate Logic

- Of course, Boolean (*propositional*) logic is too weak for many situations.

Beyond Propositional: Predicate Logic

- Of course, Boolean (*propositional*) logic is too weak for many situations.
- In order to allow richer expressive capabilities, more powerful logical languages were introduced.

Beyond Propositional: Predicate Logic

- Of course, Boolean (*propositional*) logic is too weak for many situations.
- In order to allow richer expressive capabilities, more powerful logical languages were introduced.
- One of those is **first-order predicate logic,** which is usually used to formalize mathematics.

Predicate Logic

- In predicate logic, we have **individual variables** which range over a domain.

Predicate Logic

- In predicate logic, we have **individual variables** which range over a domain.
- **Atomic formulae** are of the form $P(x, y, z, ...)$, where P is a **predicate symbol**.

Predicate Logic

- In predicate logic, we have **individual variables** which range over a domain.
- **Atomic formulae** are of the form $P(x, y, z, ...)$, where P is a **predicate symbol**.
- E.g., a two-argument P denotes a **binary relation** (say, $x < y$, written as $< (x, y)$).'

Predicate Logic

- In predicate logic, we have **individual variables** which range over a domain.
- **Atomic formulae** are of the form $P(x, y, z, ...)$, where P is a **predicate symbol**.
- E.g., a two-argument P denotes a **binary relation** (say, $x < y$, written as $< (x, y)$).‘
- Besides propositional operations ($\rightarrow$, $\vee$, $\wedge$, $\neg$), there are **quantifiers** $\forall$ (forall) and $\exists$ (exists).

Predicate Logic: Example

$$\forall x \forall y (R(x, y) \to \exists z (R(x, z) \land R(z, y)))$$

Predicate Logic: Example

$$\forall x \forall y (x < y \rightarrow \exists z (x < z \land z < y))$$

Predicate Logic: Example

$$\forall x \forall y (x < y \rightarrow \exists z (x < z \land z < y))$$

- This formula expresses the *density* of the order.

Predicate Logic: Example

$$\forall x \forall y (x < y \to \exists z (x < z \land z < y))$$

- This formula expresses the *density* of the order.
- Its truth depends on the interpretation: e.g., it is true on $\mathbb{Q}$ (rational numbers), but false on $\mathbb{Z}$ (integers).

Predicate Logic: Example

$$\forall x \forall y (x < y \to \exists z (x < z \land z < y))$$

- This formula expresses the *density* of the order.
- Its truth depends on the interpretation: e.g., it is true on $\mathbb{Q}$ (rational numbers), but false on $\mathbb{Z}$ (integers).
- So, it is **satisfiable,** but not **universally true.**

Predicate Logic: Example

$$\forall x \forall y (x < y \rightarrow \exists z (x < z \land z < y))$$

- This formula expresses the *density* of the order.
- Its truth depends on the interpretation: e.g., it is true on $\mathbb{Q}$ (rational numbers), but false on $\mathbb{Z}$ (integers).
- So, it is **satisfiable,** but not **universally true.**
    - Again, universal truth and satisfiability are dual: A is universally true iff $\neg A$ is not satisfiable.

Predicate Logic: Example

- Other desired properties of $<$: transitivity, antisymmetry, linearity, are also expressible by first-order formulae (see exercises).

Predicate Logic: Example

- Other desired properties of $<$: transitivity, antisymmetry, linearity, are also expressible by first-order formulae (see exercises).
- Thus, one may write a formula which states that $<$ is a dense linear order and has at least two elements.

Predicate Logic: Example

- Other desired properties of $<$: transitivity, antisymmetry, linearity, are also expressible by first-order formulae (see exercises).
- Thus, one may write a formula which states that $<$ is a dense linear order and has at least two elements.
- Any such structure is necessarily **infinite,** thus, one cannot reduce checking satisfiability (or universal truth) to finite structures.

Example: Paradox of Material Implication

- Moving to first-order logic allows resolving one of the so-called *paradoxes of material implication*.

Example: Paradox of Material Implication

- Moving to first-order logic allows resolving one of the so-called *paradoxes of material implication*.
- *“If I’m in London, I’m in England. If I’m in Paris, I’m in France. Therefore, if I’m in London, I’m in France, or if I’m in Paris, I’m in England”.*

Example: Paradox of Material Implication

- Moving to first-order logic allows resolving one of the so-called *paradoxes of material implication*.
- *"If I'm in London, I'm in England. If I'm in Paris, I'm in France. Therefore, if I'm in London, I'm in France, or if I'm in Paris, I'm in England".*
- A naive formulation in Boolean logic is a tautology:

$$((L \to E) \wedge (P \to F)) \to ((L \to F) \vee (P \to E)).$$

Example: Paradox of Material Implication

- The informal argument, however is obviously invalid.

Example: Paradox of Material Implication

- The informal argument, however is obviously invalid.
- A more accurate formulation in predicate logic adds a dependency on the moment of time t: *"If I'm in London, I'm in England"* is a universal statement, $\forall t\,(L(t) \to E(t))$.

Example: Paradox of Material Implication

- The informal argument, however is obviously invalid.
- A more accurate formulation in predicate logic adds a dependency on the moment of time t: *"If I'm in London, I'm in England"* is a universal statement, $\forall t\,(L(t) \to E(t))$.
- The resulting first-order formula is not universally true:

$$(\forall t\,(L(t) \to E(t)) \land \forall t\,(P(t) \to F(t))) \to (\forall t\,(L(t) \to F(t)) \lor \forall t\,(P(t) \to E(t)))$$

Algorithmic Issues

- Satisfiability in predicate logic (unlike Boolean logic) is **algorithmically undecidable.**

Algorithmic Issues

- Satisfiability in predicate logic (unlike Boolean logic) is **algorithmically undecidable.**
  - This means that there is theoretically no algorithm for solving it, even without any time constraints.

Algorithmic Issues

- Satisfiability in predicate logic (unlike Boolean logic) is **algorithmically undecidable.**
    - This means that there is theoretically no algorithm for solving it, even without any time constraints.
- This motivates studying **decidable fragments** of predicate logic, where we restrict its expressivity in order to gain decidability.

Algorithmic Issues

- Satisfiability in predicate logic (unlike Boolean logic) is **algorithmically undecidable.**
  - This means that there is theoretically no algorithm for solving it, even without any time constraints.
- This motivates studying **decidable fragments** of predicate logic, where we restrict its expressivity in order to gain decidability.
  - Toy example: predicate logic with only unary predicates.

Algorithmic Issues

- Indeed, if we have only unary predicates, $P_1, ..., P_n$, then for a given element a they can have only 2^n possible values.

Algorithmic Issues

- Indeed, if we have only unary predicates, P_1, ..., P_n, then for a given element a they can have only 2^n possible values.
- Elements on which all P_i have the same value, may be identified.

Algorithmic Issues

- Indeed, if we have only unary predicates, P_1, ..., P_n, then for a given element a they can have only 2^n possible values.
- Elements on which all P_i have the same value, may be identified.
- Thus, now we have finite search over all possible interpretations, as we have had in Boolean logic.

Predicate Calculus

- How one understands that a first-order formula is universally true, if checking it by definition requires infinite time?

Predicate Calculus

- How one understands that a first-order formula is universally true, if checking it by definition requires infinite time?
- Universally true formulae can be **proved** as theorems in the **predicate calculus**.

Predicate Calculus

- How one understands that a first-order formula is universally true, if checking it by definition requires infinite time?
- Universally true formulae can be **proved** as theorems in the **predicate calculus**.
- The classical predicate calculus is obtained from Boolean logic by adding axioms and rules for quantifiers.

Predicate Calculus

1. All Boolean tautologies, where arbitrary formulae can be substituted.
2. Quantifier axioms:

$$(\forall x\, A(x)) \to A(t)$$
$$A(t) \to \exists x\, A(x)$$

   (Here the substitution of t for x should be correct.)
3. Rules of inference:

$$\frac{A \quad A \to B}{B} \qquad \frac{A(x)}{\forall x\, A(x)}$$

Predicate Calculus

- **Gödel's completeness theorem:** a formula A can be derived from a set of axioms Γ iff A is true under any interpretation where so is Γ.

Predicate Calculus

- **Gödel's completeness theorem:** a formula A can be derived from a set of axioms Γ iff A is true under any interpretation where so is Γ.
- Thus, if something **is** a theorem, one can find this out by searching over possible proofs.

Predicate Calculus

- **Gödel's completeness theorem:** a formula A can be derived from a set of axioms Γ iff A is true under any interpretation where so is Γ.
- Thus, if something **is** a theorem, one can find this out by searching over possible proofs.
- However, if A is **not** a theorem, it does not mean that $\neg A$ is. Thus, falsifying a formula can be a non-trivial task.

Decidable Fragments

- More interesting examples include **description logics** used in **formal ontologies** (used in OWL, SNOMED CT etc).

Decidable Fragments

- More interesting examples include **description logics** used in **formal ontologies** (used in OWL, SNOMED CT etc).
- These systems are between propositional and predicate logics and are used in **knowledge representation.**

Decidable Fragments

- More interesting examples include **description logics** used in **formal ontologies** (used in OWL, SNOMED CT etc).
- These systems are between propositional and predicate logics and are used in **knowledge representation.**
- Knowledge bases extend relational databases by a richer, logically enhanced language of queries. (This requires, obviously, fast algorithms.)