Knowledge Representation
Chapter 4. Predicate Calculus

Pedro Cabalar

Dept. Computer Science
University of Corunna, SPAIN

February 15, 2017

Predicate Calculus

- Predicate Calculus, also known as First Order Logic, allows us to talk about individuals and their relations.
- The term "First Order" means we can quantify over individuals using $\forall$ (for all) and $\exists$ (exists). Example: $\forall x(Man(x) \to Mortal(x))$.
- Second Order Logic would allow quantifying over relations. Example: $\forall P \forall x \forall y(P(x, y) \to P(y, x))$. This says: "any binary relation is symmetric."
- Higher Order Logics allow quantifying over relations on relations, functions on relations, etc.

1. Syntax and Semantics
2. Some properties
3. Resolution
4. Undecidability and Expressiveness
5. Some usual extensions
    - Equality
    - Arithmetics

Syntax

- The signature is slightly more complicated $\Sigma = \langle \mathcal{C}, \mathcal{F}, \mathcal{R}, \delta \rangle$:
  - $\mathcal{C}$ set of constants $C, D, E \ldots$
  - $\mathcal{F}$ set of functions $f, g, h \ldots$
  - $\mathcal{R}$ set of relations or predicates $P, Q, R, \ldots$
  - $\delta : \mathcal{F} \cup \mathcal{R} \rightarrow \mathbf{Z}^+$ degree or arity of each function or predicate.

- Example:
  - $\mathcal{C} = \{Mon, Tue, Wed, Thu, Fri, Sat, Sun\}$
  - $\mathcal{F} = \{next, min\}$
  - $\mathcal{P} = \{Lower, Workday\}$
  - $\delta(next) = 1$, $\delta(min) = 2$, $\delta(Lower) = 2$, $\delta(Workday) = 1$.

- A constant C can also be seen as a function with $\delta(C) = 0$.

Syntax

- Formulae are formed combining the signature with $(\;) \; , \; \neg \; \wedge \; \vee \; \rightarrow \equiv \; \exists \, \forall$ plus variables $x, y, z, \ldots$
- We define a term as:
  - a constant C
  - a variable x
  - an expression $f(t_1, \ldots, t_n)$ with $\delta(f) = n$ and being each t_i a term in its turn.
- An atom is an expression $P(t_1, \ldots, t_n)$ with $P \in \mathcal{R}$, $\delta(P) = n$ being t_i terms.

Syntax

- A well formed formula (wff) is defined as:
  - Any atom $P(t_1, \ldots, t_n)$
  - $\neg\alpha, \alpha \wedge \beta, \alpha \vee \beta, \alpha \rightarrow \beta, \alpha \equiv \beta$
  - $\forall x \alpha$, $\exists x \alpha$

  were α, β are wff's

- A variable x is free (resp. bound) in a formula when:
  - in an atom: every variable is free
  - in $\neg\alpha, \alpha \wedge \beta, \alpha \vee \beta, \alpha \rightarrow \beta, \alpha \equiv \beta$: x is free (resp. bound) iff it is so in α or in β.
  - in $\exists x \alpha$, $\forall x \alpha$: x bound; y free (resp. bound) iff it is so in α.

- Formulae without free variables are called sentences or closed formule.

Syntax

- Substitution $\alpha[x/t]$: replace each free occurrence of x in α by term t.
- If t contains free variables, these cannot become accidentally bound in $\alpha[x/t]$. We should previously replace the homonym bound variable in α by a fresh symbol.
- Example: let α be $P(x) \wedge \forall y\ Q(x,y)$:
  - $\alpha[x/f(c)] = P(f(c)) \wedge \forall y\ Q(f(c),y)$
  - $\alpha[y/f(x)] = \alpha$
  - $\alpha[x/f(x)] = P(f(x)) \wedge \forall y\ Q(f(x),y)$
  - $\alpha[x/f(y)] =$ not directly: $P(f(y)) \wedge \forall y\ Q(f(y),y)$
    First replace y in α by a new variable name:
    $P(x) \wedge \forall z\ Q(x,z)$ and then replace $P(f(y)) \wedge \forall z\ Q(f(y),z)$

Semantics

- Interpretation I (for signature Σ) contains:
  - Domain or universe D: it's just a set $D \neq \emptyset$
  - for each $C \in \mathcal{C}$ the interpretation defines a $C_I \in D$.
  - for each $f \in \mathcal{F}$ with $\delta(f) = n$ the interpretation defines a mapping $f_I : D^n \rightarrow D$.
  - for each $P \in \mathcal{R}$ with $\delta(P) = n$ the interpretation defines a relation $P_I : D^n \rightarrow \{F, T\}$.

- An assignment σ (for an interpretation I) is a function that assigns, to each variable x, an element from the universe $\sigma(x) \in D$.

- The expression $\sigma[x \leftarrow d]$ denotes an assignment where $\sigma[x \leftarrow d](x) = d$ and $\sigma[x \leftarrow d](y) = \sigma(y)$

Semantics

- The valuation $v_{I,\sigma}(\alpha)$ of a formula or term α w.r.t interpretation I and assignment σ is defined as:
  - $v_{I,\sigma}(C) = C_I$ for any constant C
  - $v_{I,\sigma}(x) = \sigma x$ for any variable x
  - $v_{I,\sigma}(f(t_1, \ldots, t_n)) = f_I(v_{I,\sigma}(t_1), \ldots, v_{I,\sigma}(t_n))$
  - $v_{I,\sigma}(P(t_1, \ldots, t_n)) = T$ iff $\langle v_{I,\sigma}(t_1), \ldots, v_{I,\sigma}(t_n)) \rangle \in P_I$
  - $\neg \ \wedge \ \vee \ \rightarrow \equiv$ as in propositional case
  - $v_{I,\sigma}(\forall x \ \alpha) = T$ iff for all $d \in D$, $v_{I,\sigma[x \leftarrow d]}(\alpha) = T$
  - $v_{I,\sigma}(\exists x \ \alpha) = T$ off for some $d \in D$, $v_{I,\sigma[x \leftarrow d]}(\alpha) = T$

Semantics

- Example 1. $\forall x\ Lower(Mon, x)$. Let's take I where:
  - $D = \{1, 2, 3, 4, 5, 6, 7\}$
  - $Mon_I = 1, Tue_I = 2, Wed_I = 3, Thu_I = 4, Fri_I = 5, Sat_I = 6, Sun_I = 7$
  - $min_I(d, e)$ is the minimum of d, e
  - $next_I(d) = (d \mod 7) + 1$
  - $Lower_I(d, e) = T$ iff $d \leq e$
  - $Workday_I = \{\langle 6 \rangle, \langle 7 \rangle\}$
  - Thus $v(\sigma_I, \forall x\ Lower(Mon, x))$ iff $\langle 1, 1 \rangle \in Lower_I$ and $\langle 1, 2 \rangle \in Lower_I$ and ... y $\langle 1, 7 \rangle \in Lower_I$

Semantics

- Example 2. $\forall x\ Lower(Mon, x)$. Take now I such that:
  - D ASCII codes of capital letters $Mon, Tue, Wed, tHu, Fri, Sat, sUn$
  - $Mon_I = 77, Tue_I = 84, Wed_I = 87, Thu_I = 72, Fri_I = 70, Sat_I = 83, Sun_I = 85$
  - $min_I(d, e)$ is the minimum of d, e
  - $next_I(d) = d$
  - $Lower_I(d, e) = T$ iff $d \leq e$
  - $Workday_I = \{83, 85\}$
- Now the formula is false: $Mon_I = 77$ is not lower than $Fri_I = 70$.
- Notice that: we could have interpretations where $Mon_I = Tue_I$ for instance.
- Notice that: D can be infinite (e.g. integers).

1 Syntax and Semantics

2 Some properties

3 Resolution

4 Undecidability and Expressiveness

5 Some usual extensions

- Equality
- Arithmetics

Properties of quantifiers

The next formulae are tautologies:

- De Morgan:

$$\neg\forall x\ A(x) \equiv \exists x\ \neg A(x)$$
$$\neg\exists x\ A(x) \equiv \forall x\ \neg A(x)$$

- The universal quantifier is stronger $\forall x\ A(x) \rightarrow \exists x\ A(x)$
- Nesting quantifiers:

$$\forall x \forall y\ A(x,y) \equiv \forall y \forall x A(x,y)$$
$$\exists x \exists y\ A(x,y) \equiv \exists y \exists x A(x,y)$$
$$\exists x \forall y\ A(x,y) \rightarrow \forall y \exists x A(x,y)$$

Warning: in the general case $\forall x \exists y\ A(x,y) \not\equiv \exists\ y \forall x\ A(x,y)$

Properties of quantifiers

- Disjunction

$$\begin{array}{rcl} \exists x\ A(x) \vee \exists x\ B(x) & \equiv & \exists x \big(A(x) \vee B(x)\big) \\ \forall x\ A(x) \vee \forall x\ B(x) & \rightarrow & \forall x \big(A(x) \vee B(x)\big) \\ \exists x\ A(x) \vee C & \equiv & \exists x \big(A(x) \vee C\big) \\ \forall x\ A(x) \vee C & \equiv & \forall x \big(A(x) \vee C\big) \end{array}$$

with x not free in C.

- Conjunction

$$\begin{array}{rcl} \forall x\ A(x) \wedge \forall x\ B(x) & \equiv & \forall x \big(A(x) \wedge B(x)\big) \\ \exists x \big(A(x) \wedge B(x)\big) & \rightarrow & \exists x\ A(x) \wedge \exists x\ B(x) \\ \exists x\ A(x) \wedge C & \equiv & \exists x \big(A(x) \wedge C\big) \\ \forall x\ A(x) \wedge C & \equiv & \forall x \big(A(x) \wedge C\big) \end{array}$$

with x not free in C.

Properties of quantifiers

- Implication

$$\begin{array}{rcl}
\forall x\ (C \to A(x)) & \equiv & C \to \forall x\ A(x) \\
\forall x\ (A(x) \to C) & \equiv & \exists x\ A(x) \to C \\
\exists x\ (A(x) \to B(x)) & \equiv & \forall x A(x) \to \exists x\ B(x) \\
\forall x\ (A(x) \to B(x)) & \to & (\forall x\ A(x) \to \forall x\ B(x)) \\
\forall x\ (A(x) \to B(x)) & \to & (\forall x\ A(x) \to \exists x\ B(x)) \\
\forall x\ (A(x) \to B(x)) & \to & (\exists x\ A(x) \to \exists x\ B(x)) \\
(\exists x\ B(x) \to \forall x\ A(x)) & \to & \forall x\ (A(x) \to B(x))
\end{array}$$

with x not free in C.

Properties of quantifiers

- Equivalence

$$\begin{aligned}\forall x\ (A(x) \equiv B(x)) &\rightarrow (\forall x\ A(x) \equiv \forall x\ B(x))\\ \forall x\ (A(x) \equiv B(x)) &\rightarrow (\exists x\ A(x) \equiv \exists x\ B(x))\end{aligned}$$

- Substitutions:

$$\begin{aligned}A(c) &\rightarrow \exists x\ A(x)\\ \forall x\ A(x) &\rightarrow A(c)\end{aligned}$$

  where $A(c)$ denotes $A(x)[x/c]$

- Generalisation rule

$$\frac{A}{\forall x\ A}$$

From natural to formal language . . .

- Universal quantifier: warning! it usually involves an implicit implication. Example: “Every wild horse eats grass”
  $\forall x\ (\mathit{WildHorse}(x) \rightarrow \mathit{EatsGrass}(x))$
- However, its existential version is an implicit conjunction: “Some wild horse eats grass”
  $\exists x\ (\mathit{WildHorse}(x) \wedge \mathit{EatsGrass}(x))$
- Alternative ways for saying $\forall$:
  - “Any/Every/Each even number is a multiple of two”
  - “Even numbers are multiple of two”
  - “An even number is a multiple of two”

From natural to formal language . . .

- Other ways of saying $\exists$:
  - “Some even number is a multiple of three”
  - “There is a/some even number multiple of three”
  - “At least one even number is a multiple of three”
- Be careful when combining negations:
  - “All integers are not even”
  - “All integeres are non-even”
  - “Not all integers are even”

1. Syntax and Semantics

2. Some properties

3. Resolution

4. Undecidability and Expressiveness

5. Some usual extensions
   - Equality
   - Arithmetics

Resolution

- Prenex Normal Form: no quantifier in the scope of Boolean connectives. That is, all quantifiers to the left.
- In classical First-Order Logic, we can always reduce to Prenex (in intuitionistic logic, no).
- To get Prenex NF: first reduce to negative normal form as in propositional case plus:

$$\begin{aligned} \neg\exists x\ \alpha &\equiv \forall x\ \neg\alpha \\ \neg\forall x\ \alpha &\equiv \exists x\ \neg\alpha \end{aligned}$$

Prenex Normal Form

- then, rename variables to have a different one in each quantifier. Finally, apply:

$$\begin{aligned}
\forall x\ \alpha(x) \wedge \beta &\equiv \forall x(\alpha(x) \wedge \beta)\\
\forall x\ \alpha(x) \vee \beta &\equiv \forall x(\alpha(x) \vee \beta)\\
\exists x\ \alpha(x) \wedge \beta &\equiv \exists x(\alpha(x) \wedge \beta)\\
\exists x\ \alpha(x) \vee \beta &\equiv \exists x(\alpha(x) \vee \beta)
\end{aligned}$$

  with x not free in β.

Prenex Normal Form

- Example:

$$
\begin{array}{rl}
 & (P(y) \vee \exists x\ Q(x)) \rightarrow \forall y(R(y) \wedge \exists x\ A(x,y)) \\
\equiv & \neg P(y) \wedge \forall x\ \neg Q(x) \vee \forall y(R(y) \wedge \exists x\ A(x,y) \\
\equiv & \neg P(y) \wedge \forall x\ \neg Q(x) \vee \forall z(R(z) \wedge \exists w\ A(w,z) \\
\equiv & \forall x \forall z \exists w\ \underbrace{(\neg P(y) \wedge \neg Q(x) \vee (R(z) \wedge A(w,z))}_{\text{matrix}}
\end{array}
$$

Skolemization

- Remove existential quantifiers from Prenex NF in favour of Skolem functions.
- Each $\exists x$ is removed by replacing x in the matrix by $f(y_1, \ldots, y_n)$ where:
  - f is a new function that didn't occur in the formula
  - $y_1, \ldots, y_n$ are all the universally quantified variables to the left of $\exists x$
  - If $n = 0$ the Skolem function is 0-ary, that is, we use a new constant c.
- In the previous example:

$$\begin{array}{rl} & \forall x \forall z \exists w (\neg P(y) \wedge \neg Q(x) \vee (R(z) \wedge A(w, z)) \\ \text{becomes} & \forall x \forall z (\neg P(y) \wedge \neg Q(x) \vee (R(z) \wedge A(f(x, z), z)) \end{array}$$

Skolemization

- Warning: α and its skolemization are not equivalent. However: α is satisfiable iff its skolemization is satisfiable.
- A more intuitive example: every person has a mother.

$$\begin{array}{rl} & \forall x(P(x) \rightarrow \exists y\ M(y,x)) \\ \equiv & \forall x \exists y(P(x) \rightarrow M(y,x)) \\ \text{becomes} & \forall x(P(x) \rightarrow M(mother(x),x)) \end{array}$$

Herbrand Models

- We take formula α, reduce it to Prenex NF, skolemize and the reduce the matrix to CNF.
- But then, how many models can we consider? Since we are just interested in satisfiability, we can just restrict to Herbrand models.
- Keypoint: in a Herbrand model, each constant C is just interpreted as $C_I = C$.

  And the same holds for any ground term

  $v_{I,\sigma}(f(t_1, \ldots, t_n)) = f(t_1, \ldots, t_n)$

Herbrand Models

- The Herbrand Base is defined as the set of all ground (no vars.) we can form. It can be understood as a propositional signature.
- A formula α is satisfiable if and only if the (possibly infinite) set of clauses we can form from its Skolemization is satisfiable (in the propositional sense).
- However: this set is usually infinite (it suffices with having one function symbol). Trying to generate it for applying resolution does not make sense. Example:

$$\left[P(f(x)) \vee \neg Q(x)\right] \wedge Q(y) \wedge \neg P(z)$$

Most General Unifier (Robinson)

- Given a set of expressions E, we obtain a disagreement pair $\langle E_1, E_2 \rangle$ searching from left to right the first different symbol and taking its corresponding subexpression.
- For instance, given $P(f(x), y)$ y $P(f(g(a, z)), f(z))$ a disagreement pair would be $\langle x, g(a, z) \rangle$.
- If 2 literals can be unified, there exists a most general unifier that can be computed:

```
$\sigma := [\,]$;
while $|E| > 1$ {
   $D :=$ disagreement pair $E$;
   if $D$ contains some $x$ and a term $t$ not containing $x$ {
      $E := E[x/t]$;
      $\sigma := \sigma \cdot [x/t]$;  }
   else return 'non-unifiable';
}
```

1. Syntax and Semantics
2. Some properties
3. Resolution
4. **Undecidability and Expressiveness**
5. Some usual extensions
   - Equality
   - Arithmetics

Soundness, Completeness and Undecidability

- Predicate calculus is sound and complete (Gödel): $\Gamma \models \alpha$ iff $\Gamma \vdash \alpha$.
- However, validity in predicate calculus is undecidable (Church). That is, given φ there is no program that decides (i.e., answers 'yes' or 'no') whether $\models \varphi$ in a finite number of steps.
- Consequence 1: satisfiability is also undecidable, since φ satisfiable iff not $\models \neg\varphi$.
- Consequence 2: provability $\vdash \varphi$ is also undecidable, since $\vdash \varphi$ iff $\models \varphi$.
- Still, some fragments of Predicate Calculus are known to be decidable.

Some decidable fragments of FOL

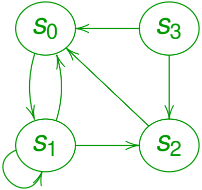
- Monadic predicate calculus (only 1-ary predicates)
- The class with prefix $\exists^*\forall^*$
- The class with prefix $\exists^*\forall\exists^*$
- The class with prefix $\exists^*\forall\forall\exists^*$ (no equality axioms)
- The class with two variables at most (Description Logics)
- Guarded Predicate Calculus:

$$\exists\overline{y}(\ \alpha(\overline{x},\overline{y}) \wedge \varphi(\overline{x},\overline{y}\ )$$
$$\forall\overline{y}(\ \alpha(\overline{x},\overline{y}) \rightarrow \varphi(\overline{x},\overline{y}\ )$$

where α atomic and including all the free variables of φ.

Expressiveness

- Example: let G be a graph with vertices S and edges E. For instance S could represent states $\{s_0, s_1, s_2, s_3\}$ and E transitions among them like in:



- Decision problem REACH (Graph reachability): given two vertices $u, v \in V$, can we find a finite path from u to v in G?
- Since REACH is a decision problem, perhaps we can try to represent it as FOL-satisfiability of some formula $\varphi_{REACH}(u, v)$.

Expressiveness

- We use predicate $R(x, y)$ to represent edges and free variables u, v to represent the nodes to check.
- Given any graph G, we have its corresponding model $I(G)$. We look for a formula $\varphi_{REACH}(u, v)$ such that G has a finite path from u to v iff $I(G) \models \varphi_{REACH}(u, v)$.
- Trying to encode reachability as a formula …

$$\begin{aligned}\varphi_{REACH}(u, v) \overset{def}{=} u = v \quad & \vee \quad \exists x(R(u, x) \wedge R(x, v)) \\ & \vee \quad \exists x_1 \exists x_2 (R(u, x_1) \wedge R(x_1, x_2) \wedge R(x_2, v)) \\ & \vee \quad \ldots\end{aligned}$$

But this is not a well-formed formula! (infinite disjunction)
Can we find an equivalent well-founded formula? NO

Expressiveness

- First, two important properties:

Theorem (Compactness Theorem)

Let Γ be a set of sentences. If all finite subsets of Γ are satisfiable, then Γ is satisfiable.

Theorem (Löwenheim-Skolem Theorem)

If Γ has a model then it has a model with a countable domain.

Countable domain means: $|D| = |S|$ for some subset S of natural numbers (including the whole set too).

Expressiveness

Theorem

There is no FOL-formula $\varphi_{REACH}(u, v)$ depending on R, u, v such that there is a finite path from u to v in G iff $I(G) \models \varphi_{REACH}(u, v)$.

Proof.

Assume $\varphi_{REACH}(u, v)$ exists. Take 2 constants c, d and define ψ_i="G has a path of length i from c to d", that is:

$$\psi_0 \overset{def}{=} (c = d)$$

$$\psi_1 \overset{def}{=} R(c, d)$$

$$\psi_n \overset{def}{=} \exists x_1 \dots \exists x_n (R(c, x_1) \wedge R(x1, x_2) \cdots \wedge R(x_{n-1}, d) \quad \text{for any } n > 1$$

The (infinite) theory $\Gamma(c, d) = \{\neg\psi_i \mid i \geq 0\}$ would mean there is no path of any length between c and d. □

Proof (continued).

Then $\Gamma' \overset{def}{=} \Gamma(c, d) \cup \{\varphi_{REACH}(c, d)\}$ is unsatisfiable - no graph G can satisfy both. Consider now any finite subset $\Gamma'' \subset \Gamma'$:

1. If $\varphi_{REACH}(c, d) \notin \Gamma''$, Γ'' says that there are not paths of some given lengths. This is satisfiable (for instance, take G with no path at all).
2. If $\varphi_{REACH}(c, d) \in \Gamma''$, Γ'' says that there are not paths of some given lengths, but there is some finite path. This is also satisfiable (for instance, pick any n such that $\varphi_n \notin \Gamma''$).

But then, by Compactness Theorem, Γ' is satisfiable, and we reach a contradiction. □

1 Syntax and Semantics

2 Some properties

3 Resolution

4 Undecidability and Expressiveness

5 Some usual extensions
- Equality
- Arithmetics

FOL with equality

- *FOL*$_=$: We have an (infix) binary predicate '$=$' whose meaning is fixed by the axiom schemata:

$$\begin{aligned} x = x & \\ x = y & \rightarrow f(\overline{z}, x, \overline{z'}) = f(\overline{z}, y, \overline{z'}) \\ x = y \wedge \varphi(x) & \rightarrow \varphi(y) \end{aligned}$$

  for any variables x, y, tuples of variables $\overline{z}, \overline{z'}$, function symbol f and any formula φ.

- Symmetry and transitivity can be proved from the axioms above:

$$\begin{aligned} x = y & \rightarrow y = x \\ x = y \wedge y = z & \rightarrow x = z \end{aligned}$$

Dedekind/Peano axioms

- We use $FOL_=$ and we have one constant 0, a unary function s (successor) and two (infix) binary functions $+$ and $\cdot$.
- Each natural number n is represented by n nested applications of s to 0. Example: 5 is written $s(s(s(s(s(0)))))$ or just $s^5(0)$.
- Peano Arithmetics (PA) axioms: universal closure of

$$\begin{array}{rcl}
\neg(0 & = & s(x)) \\
s(x) = s(y) \ \rightarrow \ x & = & y \\
x + 0 & = & x \\
x + s(y) & = & s(x + y) \\
x \cdot 0 & = & 0 \\
x \cdot s(y) & = & x \cdot y + x
\end{array}$$

plus the induction schema ...

Dedekind/Peano axioms

- Induction schema: contains a countably infinite set of axioms:

$$\begin{aligned}\forall \overline{y}\ (\quad & \varphi(0,\overline{y}) \wedge \\ & \forall x\ (\varphi(x,\overline{y}) \rightarrow \varphi(s(x),\overline{y})\ ) \\ & \rightarrow \forall x\ \varphi(x,\overline{y})\ )\end{aligned}$$

  for any formula $\varphi(x,\overline{y})$ with free variables x and (tuple) $\overline{y}$.

- Induction has a simpler encoding in second order logic:

$$\forall P\ (\ P(0)\ \wedge\ \forall x\ (P(x) \rightarrow P(s(x))\ )\ \rightarrow \forall x\ P(x)\ )$$

Gödel's first incompleteness theorem

- First incompleteness theorem: there is no *recursive* set of axioms for arithmetics that is both consistent and complete.
- By *recursive* we mean that it can be infinite, but effectively generated (for instance, by a computer program). Otherwise, we could take the trivial axiomatisation = all the valid formulas!
- It follows that there are valid formulas that are unprovable (in fact, there are infinitely many of them).
- The theorem can also be stated as: for a recursive, consistent set of axioms for arithmetics there are sentences such that neither φ nor $\neg\varphi$ has a proof.

Gödel's first incompleteness theorem

- The proof is ingenious and simple! It relies on the idea of Gödel's numbering.
- Assign an arbitrary but unique positive integer to each syntactic symbol 'σ' denoted $\#(\sigma)$. Example

$$\#(0)=1,\;\; \#(s)=1,\;\; \#(\neg)=2,\;\; \#(\wedge)=3,$$
$$\#(\forall)=4,\;\; \#(x)=5,\;\; \ldots$$

- A formula φ is just a sequence of symbols $\sigma_1 \ldots \sigma_n$. We assign a number $\lceil\varphi\rceil$ defined as:

$$\lceil\varphi\rceil \overset{def}{=} 2^{\#(\sigma_1)} \cdot 3^{\#(\sigma_2)} \cdot 5^{\#(\sigma_3)} \cdot \quad \ldots \quad \cdot p_n^{\#(\sigma_n)}$$

where p_n is the n-th prime number.

Gödel's first incompleteness theorem

- Fixing a separator symbol, we can also assign a Gödel's number to sequences of formulas, that is, proofs.
- The trick is that numbers referring to formulas and proofs can be used as arguments of other formulas in their turn.
- Gödel constructed a formula $Prov(\ulcorner\varphi\urcorner)$ such that it is true if formula φ is provable in PA for any formula φ. He showed that, for some φ:

$$PA \quad \vdash \quad \varphi \leftrightarrow \neg Prov(\ulcorner\varphi\urcorner)$$

- But then:
  1. If $PA \vdash \varphi$ we conclude $PA \vdash \neg Prov(\ulcorner\varphi\urcorner)$ so φ is not provable in PA (inconsistency).
  2. If $PA \vdash \neg\varphi$ we conclude $PA \vdash Prov(\ulcorner\varphi\urcorner)$ so φ is provable in PA but the system would be inconsistent.

  So, we can't prove nor refute φ.