Godel's Incompleteness Theorem

Exposition by William Gasarch—U of MD

Peano Arithmetic

Peano Arithmetic

Peano Arithmetic is a finite set of axioms and rules of inference.

Peano Arithmetic

Peano Arithmetic is a finite set of axioms and rules of inference. **Language** $=, S, 0$. $S(x) = x + 1$. $+, \times, \div, -, <, >$ can be defined in terms of S. Negative numbers can be defined. We also have symbols $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$ (they range over $\mathbb{N}$).

Peano Arithmetic

Peano Arithmetic is a finite set of axioms and rules of inference.
Language $=, S, 0$. $S(x) = x + 1$. $+, \times, \div, -, <, >$ can be defined in terms of S. Negative numbers can be defined. We also have symbols $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$ (they range over $\mathbb{N}$).
Axioms

Peano Arithmetic

Peano Arithmetic is a finite set of axioms and rules of inference.
Language $=, S, 0$. $S(x) = x + 1$. $+, \times, \div, -, <, >$ can be defined in terms of S. Negative numbers can be defined. We also have symbols $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$ (they range over $\mathbb{N}$).
Axioms

1) $=$ is symmetric, reflexive, transitive.

Peano Arithmetic

Peano Arithmetic is a finite set of axioms and rules of inference.
Language $=, S, 0$. $S(x) = x + 1$. $+, \times, \div, -, <, >$ can be defined in terms of S. Negative numbers can be defined. We also have symbols $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$ (they range over $\mathbb{N}$).

Axioms

1) $=$ is symmetric, reflexive, transitive.

2) 0 is a natural number. For all n, $S(n)$ is a natural number.

Peano Arithmetic

Peano Arithmetic is a finite set of axioms and rules of inference.
Language $=, S, 0$. $S(x) = x + 1$. $+, \times, \div, -, <, >$ can be defined in terms of S. Negative numbers can be defined. We also have symbols $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$ (they range over $\mathbb{N}$).
Axioms

1) $=$ is symmetric, reflexive, transitive.

2) 0 is a natural number. For all n, $S(n)$ is a natural number.

3) S is 1-1.

Peano Arithmetic

Peano Arithmetic is a finite set of axioms and rules of inference.
Language $=, S, 0$. $S(x) = x + 1$. $+, \times, \div, -, <, >$ can be defined in terms of S. Negative numbers can be defined. We also have symbols $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$ (they range over $\mathbb{N}$).
Axioms

1) $=$ is symmetric, reflexive, transitive.

2) 0 is a natural number. For all n, $S(n)$ is a natural number.

3) S is 1-1.

4) There is no n such that $S(n) = 0$.

Peano Arithmetic

Peano Arithmetic is a finite set of axioms and rules of inference.
Language $=, S, 0$. $S(x) = x + 1$. $+, \times, \div, -, <, >$ can be defined in terms of S. Negative numbers can be defined. We also have symbols $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$ (they range over $\mathbb{N}$).

Axioms

1) $=$ is symmetric, reflexive, transitive.

2) 0 is a natural number. For all n, $S(n)$ is a natural number.

3) S is 1-1.

4) There is no n such that $S(n) = 0$.

5) Let $\phi(x)$ be a formula. If $\phi(0)$ is true and $(\forall n)[\phi(n) \implies \phi(n+1)]$ then $\forall n)[\phi(n)]$.

Peano Arithmetic

Peano Arithmetic is a finite set of axioms and rules of inference.
Language $=, S, 0$. $S(x) = x + 1$. $+, \times, \div, -, <, >$ can be defined in terms of S. Negative numbers can be defined. We also have symbols $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$ (they range over $\mathbb{N}$).

Axioms

1) $=$ is symmetric, reflexive, transitive.

2) 0 is a natural number. For all n, $S(n)$ is a natural number.

3) S is 1-1.

4) There is no n such that $S(n) = 0$.

5) Let $\phi(x)$ be a formula. If $\phi(0)$ is true and $(\forall n)[\phi(n) \implies \phi(n+1)]$ then $\forall n)[\phi(n)]$.

6) The usual logical rules of inference.

Peano and Decidability

Lemma The following is **decidable:**

Peano and Decidability

Lemma The following is **decidable:**

Given a formula ϕ which you are promised is provable in PA, find the proof in PA of ϕ.

Peano and Decidability

Lemma The following is **decidable:**

Given a formula ϕ which you are promised is provable in PA, find the proof in PA of ϕ.

PROMISE ALGORITHM

Peano and Decidability

Lemma The following is **decidable:**

Given a formula ϕ which you are promised is provable in PA, find the proof in PA of ϕ.

PROMISE ALGORITHM

1) Input ϕ

Peano and Decidability

Lemma The following is **decidable:**

Given a formula ϕ which you are promised is provable in PA, find the proof in PA of ϕ.

PROMISE ALGORITHM

1) Input ϕ

2) For $n = 1$ to ∞

Peano and Decidability

Lemma The following is **decidable:**

Given a formula ϕ which you are promised is provable in PA, find the proof in PA of ϕ.

PROMISE ALGORITHM

1) Input ϕ

2) For $n = 1$ to ∞

   For all strings x of length n

   (There are a finite number of x.)

Peano and Decidability

Lemma The following is **decidable:**

Given a formula ϕ which you are promised is provable in PA, find the proof in PA of ϕ.

PROMISE ALGORITHM

1) Input ϕ

2) For $n = 1$ to ∞

For all strings x of length n

(There are a finite number of x.)

Determine if x encodes a proof of ϕ in PA.

If it does then output x and STOP.

Peano and Decidability

Lemma The following is **decidable:**

Given a formula ϕ which you are promised is provable in PA, find the proof in PA of ϕ.

PROMISE ALGORITHM

1) Input ϕ

2) For $n = 1$ to ∞

    For all strings x of length n

    (There are a finite number of x.)

        Determine if x encodes a proof of ϕ in PA.

        If it does then output x and STOP.

END OF ALGORITHM

Peano and Decidability

Lemma The following is **decidable:**

Given a formula ϕ which you are promised is provable in PA, find the proof in PA of ϕ.

PROMISE ALGORITHM

1) Input ϕ
2) For $n = 1$ to ∞
   For all strings x of length n
   (There are a finite number of x.)
      Determine if x encodes a proof of ϕ in PA.
      If it does then output x and STOP.

END OF ALGORITHM

Note If you are given a ϕ that has no proof in PA then this algorithm will run forever.

Peano and Decidability

Lemma The following is **decidable:**

Given a formula ϕ which you are promised is provable in PA, find the proof in PA of ϕ.

PROMISE ALGORITHM

1) Input ϕ
2) For $n = 1$ to ∞
   For all strings x of length n
   (There are a finite number of x.)
      Determine if x encodes a proof of ϕ in PA.
      If it does then output x and STOP.

END OF ALGORITHM

Note If you are given a ϕ that has no proof in PA then this algorithm will run forever. Thats fine.

Peano and Decidability

Lemma The following is **decidable:**

Given a formula ϕ which you are promised is provable in PA, find the proof in PA of ϕ.

PROMISE ALGORITHM

1) Input ϕ
2) For $n = 1$ to ∞

    For all strings x of length n

    (There are a finite number of x.)

        Determine if x encodes a proof of ϕ in PA.

        If it does then output x and STOP.

END OF ALGORITHM

Note If you are given a ϕ that has no proof in PA then this algorithm will run forever. Thats fine.

Note All we use about PA is that it has a finite number of axioms and rules of inference.

What is True about Peano Arithmetic

What is True about Peano Arithmetic

1) Virtually all truths about $\mathbb{N}$ can be derived from PA.

What is True about Peano Arithmetic

1) Virtually all truths about $\mathbb{N}$ can be derived from PA.

2) It may have been thought that **all** truths about $\mathbb{N}$ can be derived from PA.

What is True about Peano Arithmetic

1) Virtually all truths about $\mathbb{N}$ can be derived from PA.

2) It may have been thought that **all** truths about $\mathbb{N}$ can be derived from PA.

3) It was definitely thought that **some** system of axioms would suffice to prove all theorems about $\mathbb{N}$ and with some care all Theorems in Math.

What is True about Peano Arithmetic

1) Virtually all truths about $\mathbb{N}$ can be derived from PA.

2) It may have been thought that **all** truths about $\mathbb{N}$ can be derived from PA.

3) It was definitely thought that **some** system of axioms would suffice to prove all theorems about $\mathbb{N}$ and with some care all Theorems in Math.

4) They were wrong.

Godel's Incompleteness Theorem

Godel's Incompleteness Theorem There exists a sentence ϕ such that

Godel's Incompleteness Theorem

Godel's Incompleteness Theorem There exists a sentence ϕ such that
1) ϕ is true of the natural numbers.

Godel's Incompleteness Theorem

Godel's Incompleteness Theorem There exists a sentence ϕ such that
1) ϕ is true of the natural numbers.
2) ϕ is not provable in PA.

Godel's Incompleteness Theorem

Godel's Incompleteness Theorem There exists a sentence ϕ such that
1) ϕ is true of the natural numbers.
2) ϕ is not provable in PA.

This theorem would not be so impressive if it was tied to PA. However, we will see after the proof that it applies to **any** proof system with a finite number of axioms and rules of inference.

Godel's Inc Theorem: Then and Now

Godel's Inc Theorem: Then and Now

1) Godel's Inc Theorem stunned the math world who thought that there was a proof system that could derive all of math.

Godel's Inc Theorem: Then and Now

1) Godel's Inc Theorem stunned the math world who thought that there was a proof system that could derive all of math.

2) The Proof of Godel's Inc Theorem was very hard and very clever

Godel's Inc Theorem: Then and Now

1) Godel's Inc Theorem stunned the math world who thought that there was a proof system that could derive all of math.

2) The Proof of Godel's Inc Theorem was very hard and very clever

3) We will prove it in **TWO** slides.

Godel's Inc Theorem: Then and Now

1) Godel's Inc Theorem stunned the math world who thought that there was a proof system that could derive all of math.

2) The Proof of Godel's Inc Theorem was very hard and very clever

3) We will prove it in **TWO** slides.
How is that possible?

Godel's Inc Theorem: Then and Now

1) Godel's Inc Theorem stunned the math world who thought that there was a proof system that could derive all of math.

2) The Proof of Godel's Inc Theorem was very hard and very clever

3) We will prove it in **TWO** slides.
How is that possible?

a) Alot of the proof involves coding math statements into numbers. Today we just assume that all works out.

Godel's Inc Theorem: Then and Now

1) Godel's Inc Theorem stunned the math world who thought that there was a proof system that could derive all of math.

2) The Proof of Godel's Inc Theorem was very hard and very clever

3) We will prove it in **TWO** slides.
How is that possible?

a) Alot of the proof involves coding math statements into numbers. Today we just assume that all works out.

b) We will use that Hilbert's Tenth Problem is undecidable.

Recall Hilberts' Tenth Problem

H10 Given $p \in \mathbb{Z}[\vec{x}]$ determine if

$$(\exists \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0].$$

Recall Hilberts' Tenth Problem

H10 Given $p \in \mathbb{Z}[\vec{x}]$ determine if

$$(\exists \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0].$$

Thm $H10$ is undecidable.

Proof of Godel's Inc. Thm: SLIDE ONE

We restate Godel's Theorem in Concrete Terms.

Proof of Godel's Inc. Thm: SLIDE ONE

We restate Godel's Theorem in Concrete Terms.

Thm There exists a polynomial $p \in \mathbb{Z}[\vec{x}]$ such that

Proof of Godel's Inc. Thm: SLIDE ONE

We restate Godel's Theorem in Concrete Terms.

Thm There exists a polynomial $p \in \mathbb{Z}[\vec{x}]$ such that

1) $(\forall \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0]$.

Proof of Godel's Inc. Thm: SLIDE ONE

We restate Godel's Theorem in Concrete Terms.

Thm There exists a polynomial $p \in \mathbb{Z}[\vec{x}]$ such that

1) $(\forall \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0]$.

2) Statement 1 cannot be proven in PA.

Proof of Godel's Inc. Thm: SLIDE TWO

Assume, BWOC, that for every polynomial $p \in \mathbb{Z}[\vec{x}]$ such that $(\forall \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0]$, there is a proof in PA of this.

Proof of Godel's Inc. Thm: SLIDE TWO

Assume, BWOC, that for every polynomial $p \in \mathbb{Z}[\vec{x}]$ such that $(\forall \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0]$, there is a proof in PA of this.

We use this to obtain an algorithm for H10.

Proof of Godel's Inc. Thm: SLIDE TWO

Assume, BWOC, that for every polynomial $p \in \mathbb{Z}[\vec{x}]$ such that $(\forall \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0]$, there is a proof in PA of this.

We use this to obtain an algorithm for H10.

ALGORITHM FOR H10

Proof of Godel's Inc. Thm: SLIDE TWO

Assume, BWOC, that for every polynomial $p \in \mathbb{Z}[\vec{x}]$ such that $(\forall \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0]$, there is a proof in PA of this.

We use this to obtain an algorithm for H10.

ALGORITHM FOR H10

1) Input $p \in \mathbb{Z}[\vec{x}]$.

Proof of Godel's Inc. Thm: SLIDE TWO

Assume, BWOC, that for every polynomial $p \in \mathbb{Z}[\vec{x}]$ such that $(\forall \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0]$, there is a proof in PA of this.

We use this to obtain an algorithm for H10.

ALGORITHM FOR H10

1) Input $p \in \mathbb{Z}[\vec{x}]$.
2) Do the following simultaneously

Proof of Godel's Inc. Thm: SLIDE TWO

Assume, BWOC, that for every polynomial $p \in \mathbb{Z}[\vec{x}]$ such that $(\forall \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0]$, there is a proof in PA of this.

We use this to obtain an algorithm for H10.

ALGORITHM FOR H10

1) Input $p \in \mathbb{Z}[\vec{x}]$.
2) Do the following simultaneously
   a) For all $\vec{a} \in \mathbb{Z}$ compute $p(\vec{a})$.
   If you ever get 0, output YES and STOP.

Proof of Godel's Inc. Thm: SLIDE TWO

Assume, BWOC, that for every polynomial $p \in \mathbb{Z}[\vec{x}]$ such that $(\forall \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0]$, there is a proof in PA of this.

We use this to obtain an algorithm for H10.

ALGORITHM FOR H10

1) Input $p \in \mathbb{Z}[\vec{x}]$.
2) Do the following simultaneously
   a) For all $\vec{a} \in \mathbb{Z}$ compute $p(\vec{a})$.
   If you ever get 0, output YES and STOP.

   b)Run PROMISE on the statement $(\forall \vec{a})[p(\vec{a}) \neq 0]$.
   If produces proof that $(\forall \vec{a})[p(\vec{a}) \neq 0]$, output NO and STOP.

Proof of Godel's Inc. Thm: SLIDE TWO

Assume, BWOC, that for every polynomial $p \in \mathbb{Z}[\vec{x}]$ such that $(\forall \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0]$, there is a proof in PA of this.

We use this to obtain an algorithm for H10.

ALGORITHM FOR H10

1) Input $p \in \mathbb{Z}[\vec{x}]$.
2) Do the following simultaneously
   a) For all $\vec{a} \in \mathbb{Z}$ compute $p(\vec{a})$.
   If you ever get 0, output YES and STOP.

   b)Run PROMISE on the statement $(\forall \vec{a})[p(\vec{a}) \neq 0]$.
   If produces proof that $(\forall \vec{a})[p(\vec{a}) \neq 0]$, output NO and STOP.

END OF ALGORITHM

Proof of Godel's Inc. Thm: SLIDE TWO

Assume, BWOC, that for every polynomial $p \in \mathbb{Z}[\vec{x}]$ such that $(\forall \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0]$, there is a proof in PA of this.

We use this to obtain an algorithm for H10.

ALGORITHM FOR H10

1) Input $p \in \mathbb{Z}[\vec{x}]$.
2) Do the following simultaneously
   a) For all $\vec{a} \in \mathbb{Z}$ compute $p(\vec{a})$.
   If you ever get 0, output YES and STOP.

   b)Run PROMISE on the statement $(\forall \vec{a})[p(\vec{a}) \neq 0]$.
   If produces proof that $(\forall \vec{a})[p(\vec{a}) \neq 0]$, output NO and STOP.

END OF ALGORITHM

Since we are assuming that for every $p \in \mathbb{Z}[\vec{x}]$ such that $(\forall \vec{a})[p(\vec{a}) \neq 0]$. there is a proof of that in PA, the above algorithm always halts with the correct answer and hence solves H10. This is a contradiction.

Proof of Godel's Inc. Thm: SLIDE TWO

Assume, BWOC, that for every polynomial $p \in \mathbb{Z}[\vec{x}]$ such that $(\forall \vec{a} \in \mathbb{Z})[p(\vec{a}) = 0]$, there is a proof in PA of this.

We use this to obtain an algorithm for H10.

ALGORITHM FOR H10

1) Input $p \in \mathbb{Z}[\vec{x}]$.
2) Do the following simultaneously

   a) For all $\vec{a} \in \mathbb{Z}$ compute $p(\vec{a})$.
   If you ever get 0, output YES and STOP.

   b)Run PROMISE on the statement $(\forall \vec{a})[p(\vec{a}) \neq 0]$.
   If produces proof that $(\forall \vec{a})[p(\vec{a}) \neq 0]$, output NO and STOP.

END OF ALGORITHM

Since we are assuming that for every $p \in \mathbb{Z}[\vec{x}]$ such that $(\forall \vec{a})[p(\vec{a}) \neq 0]$. there is a proof of that in PA, the above algorithm always halts with the correct answer and hence solves H10. This is a contradiction.

DONE in two slides!