Compactness and Effectiveness

Outline

Infinite?

"Given" an infinite set Σ of wffs and a single wff τ, can we "tell" whether $\Sigma \models \tau$?

Four Big Results

- (Compactness.) $\Sigma \models \tau$ iff for some finite $\Sigma_0 \subseteq \Sigma$, we have $\Sigma_0 \models \tau$.

Four Big Results

- (Compactness.) $\Sigma \models \tau$ iff for some finite $\Sigma_0 \subseteq \Sigma$, we have $\Sigma_0 \models \tau$.
- Suppose Σ is recursively enumerable (i.e., a computer can print the (possibly infinitely many) elements of Σ). Then the set of tautological consequences of Σ is also r.e.

Four Big Results

- (Compactness.) $\Sigma \vDash \tau$ iff for some finite $\Sigma_0 \subseteq \Sigma$, we have $\Sigma_0 \vDash \tau$.
- Suppose Σ is recursively enumerable (i.e., a computer can print the (possibly infinitely many) elements of Σ). Then the set of tautological consequences of Σ is also r.e.
- There is a set S that is r.e. but not decidable (i.e., no computer that halts on all inputs can, given x, determine whether $x \in S$).

Four Big Results

- (Compactness.) $\Sigma \models \tau$ iff for some finite $\Sigma_0 \subseteq \Sigma$, we have $\Sigma_0 \models \tau$.
- Suppose Σ is recursively enumerable (i.e., a computer can print the (possibly infinitely many) elements of Σ). Then the set of tautological consequences of Σ is also r.e.
- There is a set S that is r.e. but not decidable (i.e., no computer that halts on all inputs can, given x, determine whether $x \in S$).
- There is a decidable set Σ such that $\overline{\Sigma} = \{\tau : \Sigma \models \tau\}$ is not decidable.

Compactness Implies R.E. Consequences

Compactness implies $\{\tau : \Sigma \models \tau\}$ is r.e.:

Compactness Implies R.E. Consequences

Compactness implies $\{\tau : \Sigma \models \tau\}$ is r.e.:

- Enumerate $\Sigma = \langle \sigma_1, \sigma_2, \ldots, \rangle$ and WFF $= \langle \tau_1, \tau_2, \ldots, \rangle$.

Compactness Implies R.E. Consequences

Compactness implies $\{\tau : \Sigma \models \tau\}$ is r.e.:

- Enumerate $\Sigma = \langle \sigma_1, \sigma_2, \ldots, \rangle$ and WFF $= \langle \tau_1, \tau_2, \ldots, \rangle$.
- Try all τ_i and $\{\sigma_1, \sigma_2, \ldots, \sigma_j\}$; if $\{\sigma_1, \sigma_2, \ldots, \sigma_j\} \models \tau_i$, output τ.

Compactness Implies R.E. Consequences

Compactness implies $\{\tau : \Sigma \vDash \tau\}$ is r.e.:

- Enumerate $\Sigma = \langle \sigma_1, \sigma_2, \dots, \rangle$ and WFF $= \langle \tau_1, \tau_2, \dots, \rangle$.
- Try all τ_i and $\{\sigma_1, \sigma_2, \dots, \sigma_j\}$; if $\{\sigma_1, \sigma_2, \dots, \sigma_j\} \vDash \tau_i$, output τ.

If some $\{\sigma_1, \sigma_2, \dots, \sigma_j\} \vDash \tau_i$, then (obviously) $\Sigma \vDash \tau_i$, so output is always a consequence.

Compactness Implies R.E. Consequences

Compactness implies $\{\tau : \Sigma \vDash \tau\}$ is r.e.:

- Enumerate $\Sigma = \langle \sigma_1, \sigma_2, \ldots, \rangle$ and WFF $= \langle \tau_1, \tau_2, \ldots, \rangle$.
- Try all τ_i and $\{\sigma_1, \sigma_2, \ldots, \sigma_j\}$; if $\{\sigma_1, \sigma_2, \ldots, \sigma_j\} \vDash \tau_i$, output τ.

If some $\{\sigma_1, \sigma_2, \ldots, \sigma_j\} \vDash \tau_i$, then (obviously) $\Sigma \vDash \tau_i$, so output is always a consequence.
If $\Sigma \vDash \tau_i$, then, by compactness, some $\{\sigma_1, \sigma_2, \ldots, \sigma_j\} \vDash \tau_i$. So we don't miss any consequences.

An R.E. Undecidable Set Implies Undecidable Consequences

- Let $S \subseteq \mathbb{N}$ be r.e. but not decidable, and let M enumerate S.

An R.E. Undecidable Set Implies Undecidable Consequences

- Let $S \subseteq \mathbb{N}$ be r.e. but not decidable, and let M enumerate S.
- Then $S' = \{(n, t) : M \text{ prints } n \text{ in } \leq t \text{ steps}\}$ is decidable.

An R.E. Undecidable Set Implies Undecidable Consequences

- Let $S \subseteq \mathbb{N}$ be r.e. but not decidable, and let M enumerate S.
- Then $S' = \{(n, t) : M \text{ prints } n \text{ in } \leq t \text{ steps}\}$ is decidable.
- Let $\Sigma = \{\overbrace{\mathbf{A}_n \wedge \mathbf{A}_n \wedge \cdots \wedge \mathbf{A}_n}^{t} : (n, t) \in S'\}$; then Σ is decidable.

An R.E. Undecidable Set Implies Undecidable Consequences

- Let $S \subseteq \mathbb{N}$ be r.e. but not decidable, and let M enumerate S.
- Then $S' = \{(n,t) : M \text{ prints } n \text{ in } \leq t \text{ steps}\}$ is decidable.
- Let $\Sigma = \{\overbrace{\mathbf{A}_n \wedge \mathbf{A}_n \wedge \cdots \wedge \mathbf{A}_n}^{t} : (n,t) \in S'\}$; then Σ is decidable.
- $\Sigma \vDash \mathbf{A}_n$ iff $\exists t\ M$ prints n in t steps—undecidable.

Significance

We can not decide which wffs are the consequences of some "scenario" specified by Σ, but we can enumerate those consequences (analogs of "theorems" in first-order logic).
This narrowly characterizes the "wildness" of $\{\tau : \Sigma \models \tau\}$.
(Additional comments later.)

We Seek the Truth

Why investigate $\Sigma \models \tau$ for infinite, non-trivial Σ?
Consider

We Seek the Truth

Why investigate $\Sigma \vDash \tau$ for infinite, non-trivial Σ?
Consider

- Tautologies, $\Sigma = \varnothing$. Since τ is finite, check $\vDash \tau$ using truth tables. Proving $\vDash \tau$ amounts to finding a tautology—doing logic, not math.

We Seek the Truth

Why investigate $\Sigma \vDash \tau$ for infinite, non-trivial Σ?
Consider

- Tautologies, $\Sigma = \emptyset$. Since τ is finite, check $\vDash \tau$ using truth tables. Proving $\vDash \tau$ amounts to finding a tautology—doing logic, not math.
- Finite $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$. Then $\Sigma \vDash \tau$ iff $\vDash \sigma_1 \wedge \cdots \wedge \sigma_n \rightarrow \tau$.

We Seek the Truth

Why investigate $\Sigma \models \tau$ for infinite, non-trivial Σ?
Consider

- Tautologies, $\Sigma = \emptyset$. Since τ is finite, check $\models \tau$ using truth tables. Proving $\models \tau$ amounts to finding a tautology—doing logic, not math.
- Finite $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$. Then $\Sigma \models \tau$ iff $\models \sigma_1 \wedge \cdots \wedge \sigma_n \to \tau$.
- Suppose, for all i, either $\mathbf{A}_i \in \Sigma$ or $\neg\mathbf{A}_i \in \Sigma$, decidably. Then $\Sigma \models \tau$ iff v satisfies τ for the truth assignment v induced by Σ. Showing $\Sigma \models \tau$ is showing this one v satisfies τ.

We Seek the Truth

Why investigate $\Sigma \models \tau$ for infinite, non-trivial Σ?
Consider

- Tautologies, $\Sigma = \varnothing$. Since τ is finite, check $\models \tau$ using truth tables. Proving $\models \tau$ amounts to finding a tautology—doing logic, not math.
- Finite $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$. Then $\Sigma \models \tau$ iff $\models \sigma_1 \wedge \cdots \wedge \sigma_n \rightarrow \tau$.
- Suppose, for all i, either $\mathbf{A}_i \in \Sigma$ or $\neg\mathbf{A}_i \in \Sigma$, decidably. Then $\Sigma \models \tau$ iff v satisfies τ for the truth assignment v induced by Σ. Showing $\Sigma \models \tau$ is showing this one v satisfies τ.

The interesting case: Infinite Σ. Several v's satisfy all $\sigma \in \Sigma$ and several v's do not. Showing $\Sigma \models \tau$ says something (new?) about a large set of v's.

We Seek the Truth

Why investigate $\Sigma \models \tau$ for infinite, non-trivial Σ?
Consider

- Tautologies, $\Sigma = \varnothing$. Since τ is finite, check $\models \tau$ using truth tables. Proving $\models \tau$ amounts to finding a tautology—doing logic, not math.
- Finite $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$. Then $\Sigma \models \tau$ iff $\models \sigma_1 \wedge \cdots \wedge \sigma_n \to \tau$.
- Suppose, for all i, either $\mathbf{A}_i \in \Sigma$ or $\neg \mathbf{A}_i \in \Sigma$, decidably. Then $\Sigma \models \tau$ iff v satisfies τ for the truth assignment v induced by Σ. Showing $\Sigma \models \tau$ is showing this one v satisfies τ.

The interesting case: Infinite Σ. Several v's satisfy all $\sigma \in \Sigma$ and several v's do not. Showing $\Sigma \models \tau$ says something (new?) about a large set of v's. What is true in the theory of Σ?

Example—Multiplication

$$
\begin{array}{ccc}
\cdots & \mathbf{A}_4 & \mathbf{A}_0 \\
\cdots & \mathbf{A}_5 & \mathbf{A}_1 \\
\hline
\cdots & \mathbf{A}_6 & \mathbf{A}_2 \\
\cdots & \mathbf{A}_7 & \\
\cdots & & \\
\hline
\cdots & \mathbf{A}_8 & \mathbf{A}_3
\end{array}
$$

Σ can "say" that the above multiplication computation is correct, e.g., $\mathbf{A}_2 \leftrightarrow \mathbf{A}_0 \wedge \mathbf{A}_1$, etc, for any factors.

Example—Multiplication

$$
\begin{array}{ccc}
\cdots & \mathbf{A}_4 & \mathbf{A}_0 \\
\cdots & \mathbf{A}_5 & \mathbf{A}_1 \\
\hline
\cdots & \mathbf{A}_6 & \mathbf{A}_2 \\
\cdots & \mathbf{A}_7 & \\
\cdots & & \\
\hline
\cdots & \mathbf{A}_8 & \mathbf{A}_3
\end{array}
$$

Σ can "say" that the above multiplication computation is correct, e.g., $\mathbf{A}_2 \leftrightarrow \mathbf{A}_0 \wedge \mathbf{A}_1$, etc, for any factors.
What about the following τ?

$$\tau = (\mathbf{A}_0 \leftrightarrow \mathbf{A}_1) \wedge (\mathbf{A}_4 \leftrightarrow \mathbf{A}_5) \rightarrow \neg\mathbf{A}_8$$

Example—Multiplication

$$
\begin{array}{ccc}
\cdots & \mathbf{A}_4 & \mathbf{A}_0 \\
\cdots & \mathbf{A}_5 & \mathbf{A}_1 \\
\hline
\cdots & \mathbf{A}_6 & \mathbf{A}_2 \\
\cdots & \mathbf{A}_7 & \\
\cdots & & \\
\hline
\cdots & \mathbf{A}_8 & \mathbf{A}_3
\end{array}
$$

Σ can "say" that the above multiplication computation is correct, e.g., $\mathbf{A}_2 \leftrightarrow \mathbf{A}_0 \wedge \mathbf{A}_1$, etc, for any factors.
What about the following τ?

$$\tau = (\mathbf{A}_0 \leftrightarrow \mathbf{A}_1) \wedge (\mathbf{A}_4 \leftrightarrow \mathbf{A}_5) \to \neg \mathbf{A}_8$$

$\Sigma \models \tau$ implies x^2 mod 4 is congruent to 0 or 1. ($0^2 = 2^2 = 0$ and $1^2 = 3^2 = 1$.)

Example—Multiplication

$$
\begin{array}{ccc}
\cdots & \mathbf{A}_4 & \mathbf{A}_0 \\
\cdots & \mathbf{A}_5 & \mathbf{A}_1 \\
\hline
\cdots & \mathbf{A}_6 & \mathbf{A}_2 \\
\cdots & \mathbf{A}_7 & \\
\cdots & & \\
\hline
\cdots & \mathbf{A}_8 & \mathbf{A}_3
\end{array}
$$

Σ can "say" that the above multiplication computation is correct, e.g., $\mathbf{A}_2 \leftrightarrow \mathbf{A}_0 \wedge \mathbf{A}_1$, etc, for any factors.

What about the following τ?

$$\tau = (\mathbf{A}_0 \leftrightarrow \mathbf{A}_1) \wedge (\mathbf{A}_4 \leftrightarrow \mathbf{A}_5) \rightarrow \neg \mathbf{A}_8$$

$\Sigma \vDash \tau$ implies x^2 mod 4 is congruent to 0 or 1. ($0^2 = 2^2 = 0$ and $1^2 = 3^2 = 1$.)

This is true under any assigment to the sentence symbols making Σ correct, and interesting.

Not Always So Easy

$$\begin{array}{ccc} \cdots & \mathbf{A}_4 & \mathbf{A}_0 \\ \cdots & \mathbf{A}_5 & \mathbf{A}_1 \\ \hline \cdots & \mathbf{A}_6 & \mathbf{A}_2 \\ \cdots & \mathbf{A}_7 & \\ \hline \cdots & \mathbf{A}_8 & \mathbf{A}_3 \end{array}$$

$$\tau = (\mathbf{A}_0 \leftrightarrow \mathbf{A}_1) \wedge (\mathbf{A}_4 \leftrightarrow \mathbf{A}_5) \to \neg\mathbf{A}_8 = x^2 \in \{0, 1\} \bmod 4$$

Here, one can show that $\mathbf{A}_8$ depends only on $\mathbf{A}_{<8}$, *i.e.*, $\mathbf{A}_2, \mathbf{A}_6, \mathbf{A}_7$ are correct functions of $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_4, \mathbf{A}_5$.

Not Always So Easy

$$\begin{array}{ccc} \cdots & \mathbf{A}_4 & \mathbf{A}_0 \\ \cdots & \mathbf{A}_5 & \mathbf{A}_1 \\ \hline \cdots & \mathbf{A}_6 & \mathbf{A}_2 \\ \cdots & \mathbf{A}_7 & \\ \hline \cdots & \mathbf{A}_8 & \mathbf{A}_3 \end{array}$$

$$\tau = (\mathbf{A}_0 \leftrightarrow \mathbf{A}_1) \wedge (\mathbf{A}_4 \leftrightarrow \mathbf{A}_5) \to \neg\mathbf{A}_8 = x^2 \in \{0, 1\} \bmod 4$$

Here, one can show that $\mathbf{A}_8$ depends only on $\mathbf{A}_{<8}$, *i.e.*, $\mathbf{A}_2, \mathbf{A}_6, \mathbf{A}_7$ are correct functions of $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_4, \mathbf{A}_5$.
That is, there is a finite $\Sigma_0 \subseteq \Sigma$, with $\Sigma \vDash \tau$ iff $\Sigma_0 \vDash \tau$.

Not Always So Easy

$$
\begin{array}{ccc}
\cdots & \mathbf{A}_4 & \mathbf{A}_0 \\
\cdots & \mathbf{A}_5 & \mathbf{A}_1 \\
\hline
\cdots & \mathbf{A}_6 & \mathbf{A}_2 \\
\cdots & \mathbf{A}_7 & \\
\hline
\cdots & \mathbf{A}_8 & \mathbf{A}_3
\end{array}
$$

$$\tau = (\mathbf{A}_0 \leftrightarrow \mathbf{A}_1) \wedge (\mathbf{A}_4 \leftrightarrow \mathbf{A}_5) \to \neg \mathbf{A}_8 = x^2 \in \{0, 1\} \text{ mod } 4$$

Here, one can show that $\mathbf{A}_8$ depends only on $\mathbf{A}_{<8}$, *i.e.*, $\mathbf{A}_2, \mathbf{A}_6, \mathbf{A}_7$ are correct functions of $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_4, \mathbf{A}_5$.
That is, there is a finite $\Sigma_0 \subseteq \Sigma$, with $\Sigma \vDash \tau$ iff $\Sigma_0 \vDash \tau$.
By compactness, there's always some Σ_0, depending on τ.

Not Always So Easy

$$
\begin{array}{ccc}
\cdots & \mathbf{A}_4 & \mathbf{A}_0 \\
\cdots & \mathbf{A}_5 & \mathbf{A}_1 \\
\hline
\cdots & \mathbf{A}_6 & \mathbf{A}_2 \\
\cdots & \mathbf{A}_7 & \\
\hline
\cdots & \mathbf{A}_8 & \mathbf{A}_3
\end{array}
$$

$$\tau = (\mathbf{A}_0 \leftrightarrow \mathbf{A}_1) \wedge (\mathbf{A}_4 \leftrightarrow \mathbf{A}_5) \rightarrow \neg\mathbf{A}_8 = x^2 \in \{0, 1\} \bmod 4$$

Here, one can show that $\mathbf{A}_8$ depends only on $\mathbf{A}_{<8}$, *i.e.*, $\mathbf{A}_2, \mathbf{A}_6, \mathbf{A}_7$ are correct functions of $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_4, \mathbf{A}_5$.

That is, there is a finite $\Sigma_0 \subseteq \Sigma$, with $\Sigma \vDash \tau$ iff $\Sigma_0 \vDash \tau$.

By compactness, there's always some Σ_0, depending on τ.

There is no effective procedure, in general, to find Σ_0.

Example

Suppose $\tau = \mathbf{A}_1$ and

$$\Sigma = \{\mathbf{A}_2, \mathbf{A}_2 \to \mathbf{A}_3 \wedge \mathbf{A}_4, \mathbf{A}_3 \to \mathbf{A}_{16} \wedge \mathbf{A}_{13}, \ldots\}.$$

Do we have $\Sigma \vDash \tau$?

Example

Suppose $\tau = \mathbf{A}_1$ and

$$\Sigma = \{\mathbf{A}_2, \mathbf{A}_2 \to \mathbf{A}_3 \wedge \mathbf{A}_4, \mathbf{A}_3 \to \mathbf{A}_{16} \wedge \mathbf{A}_{13}, \ldots\}.$$

Do we have $\Sigma \vDash \tau$?
A naive approach searches for a path from $\mathbf{A}_2$ to $\mathbf{A}_1$.

Example

Suppose $\tau = \mathbf{A}_1$ and

$$\Sigma = \{\mathbf{A}_2, \mathbf{A}_2 \to \mathbf{A}_3 \land \mathbf{A}_4, \mathbf{A}_3 \to \mathbf{A}_{16} \land \mathbf{A}_{13}, \ldots\}.$$

Do we have $\Sigma \models \tau$?
A naive approach searches for a path from $\mathbf{A}_2$ to $\mathbf{A}_1$.
Insufficient to restrict attention to WFFs in Σ that mention $\mathbf{A}_1$.

Outline

Statement of Compactness

Theorem

A set Σ of WFFs is satisfiable iff every finite subset is satisfiable.

Statement of Compactness

Theorem

A set Σ of WFFs is satisfiable iff every finite subset is satisfiable.

(Space of all truth assignments, under product topology, is compact.)

Statement of Compactness

Theorem

A set Σ of WFFs is satisfiable iff every finite subset is satisfiable.

(Space of all truth assignments, under product topology, is compact.)
Interesting direction: If Σ is finitely satisfiable, then it is satisfiable.

Statement of Compactness

Theorem

A set Σ of WFFs is satisfiable iff every finite subset is satisfiable.

(Space of all truth assignments, under product topology, is compact.)
Interesting direction: If Σ is finitely satisfiable, then it is satisfiable.
Proof has two parts.

- Enlarge Σ to Δ, a maximal such set.
- Show that $v(\phi) = T$ iff $\phi \in \Delta$ works.

Hierarchy of Tautological Implication

Consider an increasing set of Σ's that tautologicaly imply some τ.

Σ	$\{v : v \vDash \Sigma\}$	$\{\tau : \Sigma \vDash \tau\}$	Case Name
$\varnothing$	all v	Tautologies	Tautologies
$\{\mathbf{A}, \mathbf{B} \vee \neg\mathbf{C}\}$	some v, but not all	Tautologies, and e.g., $\mathbf{A} \vee \mathbf{D}$	Finite
Infinite	some v, but not all	some τ, but not all	Interesting
$\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots\}$	one v	$\{\tau : v(\tau) = T\}$	One v
$\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots\}$ and consequences	one v	Σ	Maximal
All wffs (or just $\{\mathbf{A}, \neg\mathbf{A}\}$)	no v's	all wffs	Inconsistent

Hierarchy of Tautological Implication

Consider an increasing set of Σ's that tautologicaly imply some τ.

Σ	$\{v : v \vDash \Sigma\}$	$\{\tau : \Sigma \vDash \tau\}$	Case Name
$\varnothing$	all v	Tautologies	Tautologies
$\{\mathbf{A}, \mathbf{B} \vee \neg\mathbf{C}\}$	some v, but not all	Tautologies, and e.g., $\mathbf{A} \vee \mathbf{D}$	Finite
Infinite	some v, but not all	some τ, but not all	Interesting
$\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots\}$	one v	$\{\tau : v(\tau) = T\}$	One v
$\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots\}$ and consequences	one v	Σ	Maximal
All wffs (or just $\{\mathbf{A}, \neg\mathbf{A}\}$)	no v's	all wffs	Inconsistent

The interesting case is the case of Σ infinite, satisfied by some, but not all v, and tautologically implying some, but not all, τ. Other cases are easy.

Hierarchy of Tautological Implication

Consider an increasing set of Σ's that tautologicaly imply some τ.

Σ	$\{v : v \vDash \Sigma\}$	$\{\tau : \Sigma \vDash \tau\}$	Case Name
$\varnothing$	all v	Tautologies	Tautologies
$\{\mathbf{A}, \mathbf{B} \vee \neg \mathbf{C}\}$	some v, but not all	Tautologies, and e.g., $\mathbf{A} \vee \mathbf{D}$	Finite
Infinite	some v, but not all	some τ, but not all	Interesting
$\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots\}$	one v	$\{\tau : v(\tau) = T\}$	One v
$\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots\}$ and consequences	one v	Σ	Maximal
All wffs (or just $\{\mathbf{A}, \neg \mathbf{A}\}$)	no v's	all wffs	Inconsistent

The interesting case is the case of Σ infinite, satisfied by some, but not all v, and tautologically implying some, but not all, τ. Other cases are easy. The compactness theorem will extend an "Interesting" Σ to a "Maximal" Σ, that is easy to analyze.

Building a Maximal Set

Let $\langle \alpha_0, \alpha_1, \ldots, \rangle$ enumerate the WFFs.

Building a Maximal Set

Let $\langle \alpha_0, \alpha_1, \ldots, \rangle$ enumerate the WFFs. Define

$$\begin{aligned} \Delta_0 &= \Sigma \\ \Delta_{i+1} &= \begin{cases} \Delta_i; \alpha_i & \text{if this is finitely satisfiable} \\ \Delta_i; \neg\alpha_i & \text{otherwise} \end{cases} \end{aligned}$$

Let $\Delta = \bigcup_i \Delta_i$. Then:

- $\Sigma \subseteq \Delta$

Building a Maximal Set

Let $\langle \alpha_0, \alpha_1, \ldots, \rangle$ enumerate the WFFs. Define

$$\begin{aligned} \Delta_0 &= \Sigma \\ \Delta_{i+1} &= \begin{cases} \Delta_i; \alpha_i & \text{if this is finitely satisfiable} \\ \Delta_i; \neg\alpha_i & \text{otherwise} \end{cases} \end{aligned}$$

Let $\Delta = \bigcup_i \Delta_i$. Then:

- $\Sigma \subseteq \Delta$
- For any α, either α or $\neg\alpha$ is in Δ, by construction.

Building a Maximal Set

Let $\langle \alpha_0, \alpha_1, \ldots, \rangle$ enumerate the WFFs. Define

$$\begin{aligned} \Delta_0 &= \Sigma \\ \Delta_{i+1} &= \begin{cases} \Delta_i; \alpha_i & \text{if this is finitely satisfiable} \\ \Delta_i; \neg\alpha_i & \text{otherwise} \end{cases} \end{aligned}$$

Let $\Delta = \bigcup_i \Delta_i$. Then:

- $\Sigma \subseteq \Delta$
- For any α, either α or $\neg\alpha$ is in Δ, by construction.
- $\Delta_0 = \Sigma$ is finitely satisfiable. (Base case.)

Building a Maximal Set

Let $\langle \alpha_0, \alpha_1, \ldots, \rangle$ enumerate the WFFs. Define

$$\begin{array}{rcl} \Delta_0 & = & \Sigma \\ \Delta_{i+1} & = & \begin{cases} \Delta_i ; \alpha_i & \text{if this is finitely satisfiable} \\ \Delta_i ; \neg\alpha_i & \text{otherwise} \end{cases} \end{array}$$

Let $\Delta = \bigcup_i \Delta_i$. Then:

- $\Sigma \subseteq \Delta$
- For any α, either α or $\neg\alpha$ is in Δ, by construction.
- $\Delta_0 = \Sigma$ is finitely satisfiable. (Base case.)
- All $\Delta_{>0}$ are finitely satisfiable. (Inductive case, next...)

Building a Maximal Set

Let $\langle \alpha_0, \alpha_1, \ldots, \rangle$ enumerate the WFFs. Define

$$\begin{aligned} \Delta_0 &= \Sigma \\ \Delta_{i+1} &= \begin{cases} \Delta_i; \alpha_i & \text{if this is finitely satisfiable} \\ \Delta_i; \neg\alpha_i & \text{otherwise} \end{cases} \end{aligned}$$

Let $\Delta = \bigcup_i \Delta_i$. Then:

- $\Sigma \subseteq \Delta$
- For any α, either α or $\neg\alpha$ is in Δ, by construction.
- $\Delta_0 = \Sigma$ is finitely satisfiable. (Base case.)
- All $\Delta_{>0}$ are finitely satisfiable. (Inductive case, next...)
- Δ is finitely satisfiable (any finite $\Delta' \subseteq \Delta$ is also a subset of some Δ_i).

All Δ_i are finitely satisfiable

$$\Delta_{i+1} = \begin{cases} \Delta_i; \alpha_i & \text{if this is finitely satisfiable} \\ \Delta_i; \neg\alpha_i & \text{otherwise} \end{cases}$$

To show Δ_{i+1} is finitely satisfiable:

- By induction, Δ_i is finitely satisfiable.

All Δ_i are finitely satisfiable

$$\Delta_{i+1} \quad = \quad \begin{cases} \Delta_i; \alpha_i & \text{if this is finitely satisfiable} \\ \Delta_i; \neg\alpha_i & \text{otherwise} \end{cases}$$

To show Δ_{i+1} is finitely satisfiable:

- By induction, Δ_i is finitely satisfiable.
- If $\Delta_i; \alpha_i$ is finitely satisfiable, this is Δ_{i+1}—done.

All Δ_i are finitely satisfiable

$$\Delta_{i+1} \quad = \quad \begin{cases} \Delta_i;\alpha_i & \text{if this is finitely satisfiable} \\ \Delta_i;\neg\alpha_i & \text{otherwise} \end{cases}$$

To show Δ_{i+1} is finitely satisfiable:

- By induction, Δ_i is finitely satisfiable.
- If $\Delta_i;\alpha_i$ is finitely satisfiable, this is Δ_{i+1}—done.
- Otherwise, there's there's some finite, unsatisfiable $\Delta'' \subseteq \Delta_i;\alpha_i$. Since Δ_i is finitely sastisfiable, $\Delta'' \nsubseteq \Delta_i$, so $\alpha_i \in \Delta''$ by induction.

All Δ_i are finitely satisfiable

$$\Delta_{i+1} = \begin{cases} \Delta_i; \alpha_i & \text{if this is finitely satisfiable} \\ \Delta_i; \neg\alpha_i & \text{otherwise} \end{cases}$$

To show Δ_{i+1} is finitely satisfiable:

- By induction, Δ_i is finitely satisfiable.
- If $\Delta_i; \alpha_i$ is finitely satisfiable, this is Δ_{i+1}—done.
- Otherwise, there's there's some finite, unsatisfiable $\Delta'' \subseteq \Delta_i; \alpha_i$. Since Δ_i is finitely sastisfiable, $\Delta'' \nsubseteq \Delta_i$, so $\alpha_i \in \Delta''$ by induction.
- Toward a contradiction, assume Δ' is a finite, unsatisfiable subset of $\Delta_{i+1} = \Delta_i; \neg\alpha_i$. Similar to the above, $\neg\alpha_i \in \Delta'$.

All Δ_i are finitely satisfiable

$$\Delta_{i+1} = \begin{cases} \Delta_i; \alpha_i & \text{if this is finitely satisfiable} \\ \Delta_i; \neg\alpha_i & \text{otherwise} \end{cases}$$

To show Δ_{i+1} is finitely satisfiable:

- By induction, Δ_i is finitely satisfiable.
- If $\Delta_i; \alpha_i$ is finitely satisfiable, this is Δ_{i+1}—done.
- Otherwise, there's there's some finite, unsatisfiable $\Delta'' \subseteq \Delta_i; \alpha_i$. Since Δ_i is finitely sastisfiable, $\Delta'' \nsubseteq \Delta_i$, so $\alpha_i \in \Delta''$ by induction.
- Toward a contradiction, assume Δ' is a finite, unsatisfiable subset of $\Delta_{i+1} = \Delta_i; \neg\alpha_i$. Similar to the above, $\neg\alpha_i \in \Delta'$.
- $\widehat{\Delta} = (\Delta' - \neg\alpha_i) \cup (\Delta'' - \alpha_i) \subseteq \Delta_i$ is satisfiable by induction, by some $\overline{v}$.

All Δ_i are finitely satisfiable

$$\Delta_{i+1} = \begin{cases} \Delta_i;\alpha_i & \text{if this is finitely satisfiable} \\ \Delta_i;\neg\alpha_i & \text{otherwise} \end{cases}$$

To show Δ_{i+1} is finitely satisfiable:

- By induction, Δ_i is finitely satisfiable.
- If $\Delta_i;\alpha_i$ is finitely satisfiable, this is Δ_{i+1}—done.
- Otherwise, there's there's some finite, unsatisfiable $\Delta'' \subseteq \Delta_i;\alpha_i$. Since Δ_i is finitely sastisfiable, $\Delta'' \nsubseteq \Delta_i$, so $\alpha_i \in \Delta''$ by induction.
- Toward a contradiction, assume Δ' is a finite, unsatisfiable subset of $\Delta_{i+1} = \Delta_i;\neg\alpha_i$. Similar to the above, $\neg\alpha_i \in \Delta'$.
- $\widehat{\Delta} = (\Delta' - \neg\alpha_i) \cup (\Delta'' - \alpha_i) \subseteq \Delta_i$ is satisfiable by induction, by some $\overline{v}$.
- If $\overline{v}(\alpha_i) = T$, then $\overline{v}$ satisfies $\widehat{\Delta};\alpha_i \supseteq \Delta''$. Else $\overline{v}(\neg\alpha_i) = T$, and $\overline{v}$ satisfies $\widehat{\Delta};\neg\alpha_i \supseteq \Delta'$.

Defining a Satisfying Assignment

Define $v(\mathbf{A}) = T$ iff $\mathbf{A} \in \Delta$ and extend to $\overline{v}$.

Defining a Satisfying Assignment

Define $v(\mathbf{A}) = T$ iff $\mathbf{A} \in \Delta$ and extend to $\overline{v}$.
Claim: $\overline{v}$ satisfies ϕ iff $\phi \in \Delta$.

Defining a Satisfying Assignment

Define $v(\mathbf{A}) = T$ iff $\mathbf{A} \in \Delta$ and extend to $\overline{v}$.
Claim: $\overline{v}$ satisfies ϕ iff $\phi \in \Delta$.
By structural induction. E.g., suppose $\phi = \neg\psi$. Then

Defining a Satisfying Assignment

Define $v(\mathbf{A}) = T$ iff $\mathbf{A} \in \Delta$ and extend to $\overline{v}$.

Claim: $\overline{v}$ satisfies ϕ iff $\phi \in \Delta$.

By structural induction. E.g., suppose $\phi = \neg\psi$. Then

$$\begin{array}{lll} \overline{v} \text{ satisfies } \phi & \text{iff} & \overline{v} \text{ does not satisfy } \psi \\ & \text{iff} & \psi \notin \Delta \\ & \text{iff} & \phi = \neg\psi \in \Delta. \end{array}$$

Defining a Satisfying Assignment

Define $v(\mathbf{A}) = T$ iff $\mathbf{A} \in \Delta$ and extend to $\overline{v}$.

Claim: $\overline{v}$ satisfies ϕ iff $\phi \in \Delta$.

By structural induction. E.g., suppose $\phi = \neg\psi$. Then

$$\begin{array}{lll} \overline{v} \text{ satisfies } \phi & \text{iff} & \overline{v} \text{ does not satisfy } \psi \\ & \text{iff} & \psi \notin \Delta \\ & \text{iff} & \phi = \neg\psi \in \Delta. \end{array}$$

In particular, $\overline{v}$ satisfies Σ.

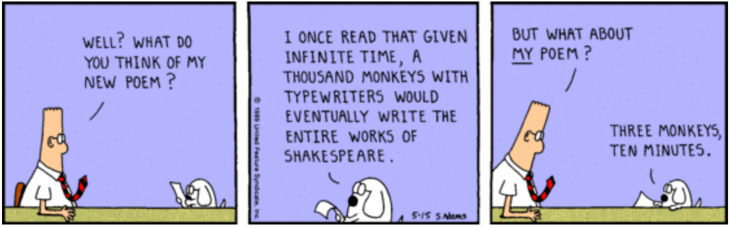
Corollary

An equivalent formulation:
$\Sigma \models \tau$ iff there is a finite $\Sigma_0 \subseteq \Sigma$ with $\Sigma_0 \models \tau$.

Outline

Enumeration

http://dilbert.com



Recursive Enumerability

A set S (of numbers, expressions, or WFFs) is recursively enumerable (r.e.) if there is some computer program (machine) M that prints a (possibly infinite) list that contains exactly the elements of $S = L(M)$.

Recursive Enumerability

A set S (of numbers, expressions, or WFFs) is recursively enumerable (r.e.) if there is some computer program (machine) M that prints a (possibly infinite) list that contains exactly the elements of $S = L(M)$.

A set is semi-decidable if there is a computer program that, on input x, halts (and answers "yes") or does not halt depending on whether $x \in S$ or $x \notin S$, respectively.

Equivalence

Theorem

S is r.e. iff S is semi-decidable.

Equivalence

Theorem

S is r.e. iff S is semi-decidable.

Proof.

(Harder direction): Suppose S is semi-decidable, by M. To enumerate S, try $M(x_0), M(x_1), \ldots$, in interleaving processes, and output x_i if $M(x_i)$ accepts.

Equivalence

Theorem

S is r.e. iff S is semi-decidable.

Proof.

(Harder direction): Suppose S is semi-decidable, by M. To enumerate S, try $M(x_0), M(x_1), \ldots$, in interleaving processes, and output x_i if $M(x_i)$ accepts.
Interleaving processes: Proceed one step of $M(x_0)$ at time 1, 3, 5, 7,. . . . Proceed one step of $M(x_1)$ at time 2, 6, 10, 14,. . . . Proceed one step of $M(x_i)$ at time odd$\cdot 2^i$. □

Decidability

- By contrast, a set S is decidable if a computer program that halts on all inputs x outputs "yes" or "no" depending on whether $x \in S$.

Decidability

- By contrast, a set S is decidable if a computer program that halts on all inputs x outputs “yes” or “no” depending on whether $x \in S$.
- There are sets that are not decidable. (Most of the uncountably-many sets are not decidable by any of the countably-many computer programs.)

Decidability

- By contrast, a set S is decidable if a computer program that halts on all inputs x outputs "yes" or "no" depending on whether $x \in S$.
- There are sets that are not decidable. (Most of the uncountably-many sets are not decidable by any of the countably-many computer programs.)
- There are sets that are r.e. but not decidable. (Later...)

Practical Analogy

TCP, a communications protocol underlying the web, requires:

- If a server is normal, it must send an acknowledgement of clients' packets.
- If a server is overloaded, it must signal this by not sending acknowledgements. (That may be all it can manage.)

(Im-)morally speaking, the set of times at which a server is normal is r.e. but not decidable.

Tautological Consequences is R.E.

Theorem

If Σ is r.e., then so is the set $\overline{\Sigma}$ of its tautological consequences.

Proof.

- Let $\Sigma = \langle \sigma_1, \sigma_2, \ldots, \rangle$ and WFF $= \langle \tau_1, \tau_2, \ldots, \rangle$.

Tautological Consequences is R.E.

Theorem

If Σ is r.e., then so is the set $\overline{\Sigma}$ of its tautological consequences.

Proof.

- Let $\Sigma = \langle \sigma_1, \sigma_2, \ldots, \rangle$ and WFF $= \langle \tau_1, \tau_2, \ldots, \rangle$.
- Try all τ_i and $\{\sigma_1, \sigma_2, \ldots, \sigma_j\}$

Tautological Consequences is R.E.

Theorem

If Σ is r.e., then so is the set $\overline{\Sigma}$ of its tautological consequences.

Proof.

- Let $\Sigma = \langle \sigma_1, \sigma_2, \ldots, \rangle$ and WFF $= \langle \tau_1, \tau_2, \ldots, \rangle$.
- Try all τ_i and $\{\sigma_1, \sigma_2, \ldots, \sigma_j\}$
- Output τ_i if $\{\sigma_1, \sigma_2, \ldots, \sigma_j\} \models \tau_i$

Tautological Consequences is R.E.

Theorem

If Σ is r.e., then so is the set $\overline{\Sigma}$ of its tautological consequences.

Proof.

- Let $\Sigma = \langle \sigma_1, \sigma_2, \ldots, \rangle$ and WFF $= \langle \tau_1, \tau_2, \ldots, \rangle$.
- Try all τ_i and $\{\sigma_1, \sigma_2, \ldots, \sigma_j\}$
- Output τ_i if $\{\sigma_1, \sigma_2, \ldots, \sigma_j\} \models \tau_i$

How to try all? Next...

Tautological Consequences is R.E.

Theorem

If Σ is r.e., then so is the set $\overline{\Sigma}$ of its tautological consequences.

Proof.

- Let $\Sigma = \langle \sigma_1, \sigma_2, \ldots, \rangle$ and WFF $= \langle \tau_1, \tau_2, \ldots, \rangle$.
- Try all τ_i and $\{\sigma_1, \sigma_2, \ldots, \sigma_j\}$
- Output τ_i if $\{\sigma_1, \sigma_2, \ldots, \sigma_j\} \vDash \tau_i$

How to try all? Next...
By compactness, some $\{\sigma_1, \sigma_2, \ldots, \sigma_j\} \vDash \tau_i$, iff $\Sigma \vDash \tau_i$. $\square$

Busy Beaver

How to try all $M(\tau_i, \{\sigma_1, \ldots, \sigma_j\})$?

Busy Beaver

How to try all $M(\tau_i, \{\sigma_1, \ldots, \sigma_j\})$?
Enumerate all pairs of natural numbers.

$f(i,j)$	1	2	3	4
0	0	1	3	6
1	2	4	7	
2	5	8		

Busy Beaver

How to try all $M(\tau_i, \{\sigma_1, \ldots, \sigma_j\})$?
Enumerate all pairs of natural numbers.

$f(i,j)$	1	2	3	4
0	0	1	3	6
1	2	4	7	
2	5	8		

Interleave all processes: advance $f(i,j)$ one step at time odd$\cdot 2^{f(i,j)}$.

$f(i,j)$	Times when $f(i,j)$ is active
0	$1, 3, 5, 7, 9, 11, 13, 15, \ldots$
1	$2, 6, 10, 14, \ldots$
2	$4, 12, \ldots$
3	$8, \ldots$

Outline

The Reals are Uncountable

Suppose not. List the reals:

Position	Real
1	.12345...
2	.67514...
3	.14159...

The Reals are Uncountable

Suppose not. List the reals:

Position	Real
1	.12345...
2	.67514...
3	.14159...

Let x differ from the diagonal by at least 2 in each position; e.g., $x = .909\ldots$.

The Reals are Uncountable

Suppose not. List the reals:

Position	Real
1	.12345...
2	.67514...
3	.14159...

Let x differ from the diagonal by at least 2 in each position; e.g.,

$x = .909\ldots$.

x cannot be on the list.

Diagonalization, Generally

- Get an explicit transcendental number (not the root of any integer polynomial) by diagonalization

Diagonalization, Generally

- Get an explicit transcendental number (not the root of any integer polynomial) by diagonalization
- List the integer polynomials (low degrees first; use pairing)

k	poly	roots	
1	$3x - 1$	$1/3$	
2	$4x^2 - 1$	$\pm 1/2$	
3	$5x - 4$	$.8$	
4	$2x^2 - 1$	$\pm\sqrt{1/2}$	
$\vdots$			

Diagonalization, Generally

- Get an explicit transcendental number (not the root of any integer polynomial) by diagonalization
- List the integer polynomials (low degrees first; use pairing)

k	poly	roots	$x\downarrow$
1	$3x-1$	$1/3$	9
2	$4x^2-1$	$\pm 1/2$	0
3	$5x-4$	$.8$	0
4	$2x^2-1$	$\pm\sqrt{1/2}$	0
$\vdots$			

- Define k'th digit of x to be 0 or 9 so that x is at least $2\cdot 10^{-k}$ away from any root of the k'th polynomial, when k'th digit is chosen. Here, $x = 0.9000\ldots$

Diagonalization, Generally

- Get an explicit transcendental number (not the root of any integer polynomial) by diagonalization
- List the integer polynomials (low degrees first; use pairing)

k	poly	roots	$x\downarrow$
1	$3x-1$	$1/3$	9
2	$4x^2-1$	$\pm 1/2$	0
3	$5x-4$	$.8$	0
4	$2x^2-1$	$\pm\sqrt{1/2}$	0
$\vdots$			

- Define k'th digit of x to be 0 or 9 so that x is at least $2\cdot 10^{-k}$ away from any root of the k'th polynomial, when k'th digit is chosen. Here, $x = 0.9000\ldots$
- However the lower-order bits are set, x is not the root of the k'th polynomial; it is at least 10^{-k} away.

Undecidability of the Halting and Related Problems

The Acceptance problem is given by

$$A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$$

i.e., the set of (M, x) such that computer program ("machine") M, when run on input x, halts and answers yes.

Undecidability of the Halting and Related Problems

The Acceptance problem is given by

$$A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$$

i.e., the set of (M, x) such that computer program ("machine") M, when run on input x, halts and answers yes.
E.g., a malware detector might want to decide A, by examining code for M without running M.

Undecidability of the Halting and Related Problems

The Acceptance problem is given by

$$A = \{(M,x) : M(x) \text{ halts and } M(x) = \text{yes}\}$$

i.e., the set of (M,x) such that computer program ("machine") M, when run on input x, halts and answers yes.
E.g., a malware detector might want to decide A, by examining code for M without running M.
Suppose there were a decider H for A, i.e.,

$$H(M,x) = \begin{cases} \text{yes} & M(x) \text{ halts and } M(x) = \text{yes} \\ \text{no} & M(x) \text{ runs forever or } M(x) = \text{no} \end{cases}$$

Diagonalization

Situation:

$$A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$$

$$H(M, x) = \begin{cases} \text{yes} & M(x) \text{ halts and } M(x) = \text{yes} \\ \text{no} & M(x) \text{ runs forever or } M(x) = \text{no} \end{cases}$$

Diagonalization

Situation:

$$A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$$

$$H(M, x) = \begin{cases} \text{yes} & M(x) \text{ halts and } M(x) = \text{yes} \\ \text{no} & M(x) \text{ runs forever or } M(x) = \text{no} \end{cases}$$

Consider the program $D(M)$ that simulates $H(M, M)$ and returns the opposite, i.e.,

Diagonalization

Situation:

$$A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$$

$$H(M, x) = \begin{cases} \text{yes} & M(x) \text{ halts and } M(x) = \text{yes} \\ \text{no} & M(x) \text{ runs forever or } M(x) = \text{no} \end{cases}$$

Consider the program $D(M)$ that simulates $H(M, M)$ and returns the opposite, i.e.,

$$D(M) = \begin{cases} \text{yes} & H(M, M) = \text{no} \\ \text{no} & H(M, M) = \text{yes} \end{cases}$$

Diagonalization

Situation:

$$A = \{(M,x) : M(x) \text{ halts and } M(x) = \text{yes}\}$$

$$H(M,x) = \begin{cases} \text{yes} & M(x) \text{ halts and } M(x) = \text{yes} \\ \text{no} & M(x) \text{ runs forever or } M(x) = \text{no} \end{cases}$$

Consider the program $D(M)$ that simulates $H(M,M)$ and returns the opposite, i.e.,

$$D(M) = \begin{cases} \text{yes} & H(M,M) = \text{no} \\ \text{no} & H(M,M) = \text{yes} \end{cases}$$

D halts on all inputs, since H does.

Punchline

We have $A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$,

$$H(M,x) = \begin{cases} \text{yes} & M(x) \text{ halts and } M(x) = \text{yes} \\ \text{no} & M(x) \text{ runs forever or } M(x) = \text{no} \end{cases}$$

and $D(M) = \neg H(M, M)$.

Punchline

We have $A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$,

$$H(M, x) = \begin{cases} \text{yes} & M(x) \text{ halts and } M(x) = \text{yes} \\ \text{no} & M(x) \text{ runs forever or } M(x) = \text{no} \end{cases}$$

and $D(M) = \neg H(M, M)$. What about $D(D)$?

Punchline

We have $A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$,

$$H(M, x) = \begin{cases} \text{yes} & M(x) \text{ halts and } M(x) = \text{yes} \\ \text{no} & M(x) \text{ runs forever or } M(x) = \text{no} \end{cases}$$

and $D(M) = \neg H(M, M)$. What about $D(D)$?

$$\begin{aligned} D(D) = \text{yes} \quad & \Rightarrow \quad H(D, D) = \text{no} \\ & \Rightarrow \quad H(D, D) = \text{no} \\ & \Rightarrow \quad D(D) \text{ runs forever } \text{ or } \ D(D) = \text{no} \end{aligned}$$

and

Punchline

We have $A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$,

$$H(M,x) = \begin{cases} \text{yes} & M(x) \text{ halts and } M(x) = \text{yes} \\ \text{no} & M(x) \text{ runs forever or } M(x) = \text{no} \end{cases}$$

and $D(M) = \neg H(M, M)$. What about $D(D)$?

$$\begin{aligned} D(D) = \text{yes} \quad &\Rightarrow \quad H(D, D) = \text{no} \\ &\Rightarrow \quad H(D, D) = \text{no} \\ &\Rightarrow \quad D(D) \text{ runs forever} \quad \text{or} \quad D(D) = \text{no} \end{aligned}$$

and

$$\begin{aligned} D(D) = \text{no} \quad &\Rightarrow \quad H(D, D) = \text{yes} \\ &\Rightarrow \quad D(D) \text{ halts and } D(D) = \text{yes} \end{aligned}$$

Contradiction! Thus decider H does not exist. $\boxed{A \text{ is not decidable.}}$

...but r.e.

But A is r.e.: Semi-decider S_A works as follows:

…but r.e.

But A is r.e.: Semi-decider S_A works as follows:
On input (M, x), run $M(x)$.

...but r.e.

But A is r.e.: Semi-decider S_A works as follows:
On input (M, x), run $M(x)$.

- If M halts and accepts x, then S_A halts and accepts (M, x).

...but r.e.

But A is r.e.: Semi-decider S_A works as follows:
On input (M, x), run $M(x)$.

- If M halts and accepts x, then S_A halts and accepts (M, x).
- If M halts and rejects x, then S_A runs forever.

...but r.e.

But A is r.e.: Semi-decider S_A works as follows:
On input (M, x), run $M(x)$.

- If M halts and accepts x, then S_A halts and accepts (M, x).
- If M halts and rejects x, then S_A runs forever.
- If M runs forever, then S_A runs forever.

Compare with Transcendentals

To produce transendental x, list integer polynomials and make sure the k'th polynomial p does not "accept" x, *i.e.*, $p(x) \neq 0$. To insure the condition for p_k, look at the k'th bit of x.

Compare with Transcendentals

To produce transendental x, list integer polynomials and make sure the k'th polynomial p does not "accept" x, *i.e.*, $p(x) \neq 0$. To insure the condition for p_k, look at the k'th bit of x.

To produce undecidable set A, list all machines and make sure machine M does not decide A, *i.e.*, there's some input y on which M does not halt or $M(y)$'s output doesn't match A. To insure the condition for machine M, look at the input $y = M$ for machine M.

Consequences are Undecidable

- Suppose S_A semi-decides $A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$.

Consequences are Undecidable

- Suppose S_A semi-decides $A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$.
- Pair $n = (M, x)$.

Consequences are Undecidable

- Suppose S_A semi-decides $A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$.
- Pair $n = (M, x)$.
- Put $A_\leq = \{(n, t) : S_A \text{ accepts } (M, x) \text{ in } \leq t \text{ steps}\}$. Then $A_\leq$ is decidable.

Consequences are Undecidable

- Suppose S_A semi-decides $A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$.
- Pair $n = (M, x)$.
- Put $A_{\leq} = \{(n, t) : S_A \text{ accepts } (M, x) \text{ in} \leq t \text{ steps}\}$. Then $A_{\leq}$ is decidable.
- Let $\Sigma = \{\overbrace{\mathbf{A}_n \wedge \mathbf{A}_n \wedge \cdots \wedge \mathbf{A}_n}^{t} : (n, t) \in A_{\leq}\}$; then Σ is decidable.

Consequences are Undecidable

- Suppose S_A semi-decides $A = \{(M, x) : M(x) \text{ halts and } M(x) = \text{yes}\}$.
- Pair $n = (M, x)$.
- Put $A_{\leq} = \{(n, t) : S_A \text{ accepts } (M, x) \text{ in } \leq t \text{ steps}\}$. Then $A_{\leq}$ is decidable.
- Let $\Sigma = \{\overbrace{\mathbf{A}_n \wedge \mathbf{A}_n \wedge \cdots \wedge \mathbf{A}_n}^{t} : (n, t) \in A_{\leq}\}$; then Σ is decidable.
- $\Sigma \models \mathbf{A}_n$ iff $\exists t\ \overbrace{\mathbf{A}_n \wedge \mathbf{A}_n \wedge \cdots \wedge \mathbf{A}_n}^{t} \in \Sigma$ iff $\exists t\ \ S_A$ accepts (M, x) in t steps iff $(M, x) \in A$—undecidable!

Outline

More is True...

We now show that there is a true but unprovable sentence in Boolean logic.

More is True...

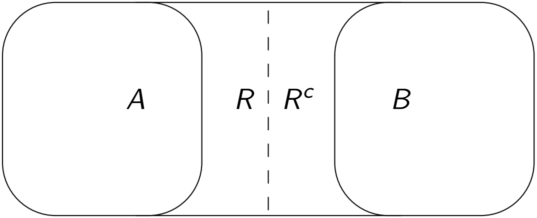
We now show that there is a true but unprovable sentence in Boolean logic.
(More importantly, we formulate this carefully!)

More is True...

We now show that there is a true but unprovable sentence in Boolean logic.
(More importantly, we formulate this carefully!)
First, a bit more recursion theory.

Recursive Inseparability

Two sets A and B are Recursively Inseparable if there is no decidable set R with $A \subseteq R$ and $B \subseteq R^c$.



Example Inseparable Sets

Theorem

The sets

$$\begin{aligned} L_Y &= \{M : M(M) \text{ halts and } M(M) = \text{yes}\} \\ L_N &= \{M : M(M) \text{ halts and } M(M) = \text{no}\} \end{aligned}$$

are recursively inseparable.

Proof.

Example Inseparable Sets

Theorem

The sets

$$\begin{aligned} L_Y &= \{M : M(M) \textit{ halts and } M(M) = \textit{yes}\} \\ L_N &= \{M : M(M) \textit{ halts and } M(M) = \textit{no}\} \end{aligned}$$

are recursively inseparable.

Proof.

Suppose wlog $L_\mathrm{N} \subseteq R$ and $R \cap L_\mathrm{Y} = \varnothing$ with M deciding R.

Example Inseparable Sets

Theorem

The sets

$$\begin{aligned} L_Y &= \{M : M(M) \text{ halts and } M(M) = \mathit{yes}\} \\ L_N &= \{M : M(M) \text{ halts and } M(M) = \mathit{no}\} \end{aligned}$$

are recursively inseparable.

Proof.

Suppose wlog $L_N \subseteq R$ and $R \cap L_Y = \varnothing$ with M deciding R. What is $M(M)$?

Example Inseparable Sets

Theorem

The sets

$$\begin{aligned} L_Y &= \{M : M(M) \text{ halts and } M(M) = yes\} \\ L_N &= \{M : M(M) \text{ halts and } M(M) = no\} \end{aligned}$$

are recursively inseparable.

Proof.

Suppose wlog $L_\mathrm{N} \subseteq R$ and $R \cap L_\mathrm{Y} = \varnothing$ with M deciding R. What is $M(M)$?

$$M(M) = \text{yes} \quad \Rightarrow \quad M \in L_\mathrm{Y}, \text{ by def of } L_\mathrm{Y}$$

Example Inseparable Sets

Theorem

The sets

$$\begin{aligned} L_Y &= \{M : M(M) \text{ halts and } M(M) = \text{yes}\} \\ L_N &= \{M : M(M) \text{ halts and } M(M) = \text{no}\} \end{aligned}$$

are recursively inseparable.

Proof.

Suppose wlog $L_\mathrm{N} \subseteq R$ and $R \cap L_\mathrm{Y} = \varnothing$ with M deciding R. What is $M(M)$?

$$\begin{aligned} M(M) = \text{yes} &\Rightarrow M \in L_\mathrm{Y}, \text{ by def of } L_\mathrm{Y} \\ &\Rightarrow M \notin R, \text{ since } R \cap L_\mathrm{Y} = \varnothing \end{aligned}$$

Example Inseparable Sets

Theorem

The sets

$$
\begin{aligned}
L_Y &= \{M : M(M) \textit{ halts and } M(M) = \textit{yes}\} \\
L_N &= \{M : M(M) \textit{ halts and } M(M) = \textit{no}\}
\end{aligned}
$$

are recursively inseparable.

Proof.

Suppose wlog $L_\mathrm{N} \subseteq R$ and $R \cap L_\mathrm{Y} = \varnothing$ with M deciding R. What is $M(M)$?

$$
\begin{aligned}
M(M) = \text{ yes} \quad &\Rightarrow \quad M \in L_\mathrm{Y}, \text{ by def of } L_\mathrm{Y} \\
&\Rightarrow \quad M \notin R, \text{ since } R \cap L_\mathrm{Y} = \varnothing \\
&\Rightarrow \quad M(M) \text{ halts and says no, since } M \text{ decides } R.
\end{aligned}
$$

Example Inseparable Sets

Theorem

The sets

$$\begin{aligned} L_Y &= \{M : M(M) \text{ halts and } M(M) = yes\} \\ L_N &= \{M : M(M) \text{ halts and } M(M) = no\} \end{aligned}$$

are recursively inseparable.

Proof.

Suppose wlog $L_{\mathrm{N}} \subseteq R$ and $R \cap L_{\mathrm{Y}} = \varnothing$ with M deciding R. What is $M(M)$?

$$\begin{aligned} M(M) = \text{ yes} \quad &\Rightarrow \quad M \in L_{\mathrm{Y}}, \text{ by def of } L_{\mathrm{Y}} \\ &\Rightarrow \quad M \notin R, \text{ since } R \cap L_{\mathrm{Y}} = \varnothing \\ &\Rightarrow \quad M(M) \text{ halts and says no, since } M \text{ decides } R. \end{aligned}$$

Similar contradiction if $M(M) = $ no.

Example Inseparable Sets

Theorem

The sets

$$\begin{aligned} L_Y &= \{M : M(M) \text{ halts and } M(M) = \text{yes}\} \\ L_N &= \{M : M(M) \text{ halts and } M(M) = \text{no}\} \end{aligned}$$

are recursively inseparable.

Proof.

Suppose wlog $L_N \subseteq R$ and $R \cap L_Y = \varnothing$ with M deciding R. What is $M(M)$?

$$\begin{aligned} M(M) = \text{ yes} \quad &\Rightarrow \quad M \in L_Y, \text{ by def of } L_Y \\ &\Rightarrow \quad M \notin R, \text{ since } R \cap L_Y = \varnothing \\ &\Rightarrow \quad M(M) \text{ halts and says no, since } M \text{ decides } R. \end{aligned}$$

Similar contradiction if $M(M) =$ no.

Contradiction! $\boxed{\text{No recursive separating } R.}$ □

Back to WFFs

Put

$$
\begin{aligned}
L_{\mathsf{Y}} &= \{M : M(M) = \text{ yes}\} \\
L_{\mathsf{Y},\leq} &= \{(M,t) : M(M) = \text{ yes in} \leq t \text{ steps}\} \\
\Sigma_{\mathsf{Y},\leq} &= \{\overbrace{\mathbf{A}_M \wedge \mathbf{A}_M \wedge \cdots \wedge \mathbf{A}_M}^{t} : (M,t) \in L_{\mathsf{Y},\leq}\}
\end{aligned}
$$

Back to WFFs

Put

$$\begin{aligned} L_{\mathrm{Y}} &= \{M : M(M) = \text{ yes}\} \\ L_{\mathrm{Y},\le} &= \{(M,t) : M(M) = \text{ yes in } \le t \text{ steps}\} \\ \Sigma_{\mathrm{Y},\le} &= \{\overbrace{\mathbf{A}_M \wedge \mathbf{A}_M \wedge \cdots \wedge \mathbf{A}_M}^{t} : (M,t) \in L_{\mathrm{Y},\le}\} \end{aligned}$$

and

$$\begin{aligned} L_{\mathrm{N}} &= \{M : M(M) = \text{ no}\} \\ L_{\mathrm{N},\le} &= \{(M,t) : M(M) = \text{ no in } \le t \text{ steps}\} \\ \Sigma_{\mathrm{N},\le} &= \{\overbrace{\mathbf{A}_M \wedge \mathbf{A}_M \wedge \cdots \wedge \mathbf{A}_M}^{t} : (M,t) \in L_{\mathrm{N},\le}\} \end{aligned}$$

Back to WFFs

Put

$$
\begin{aligned}
L_{\mathrm{Y}} &= \{M : M(M) = \text{ yes}\} \\
L_{\mathrm{Y},\le} &= \{(M,t) : M(M) = \text{ yes in} \le t \text{ steps}\} \\
\Sigma_{\mathrm{Y},\le} &= \{\overbrace{\mathbf{A}_M \wedge \mathbf{A}_M \wedge \cdots \wedge \mathbf{A}_M}^{t} : (M,t) \in L_{\mathrm{Y},\le}\}
\end{aligned}
$$

and

$$
\begin{aligned}
L_{\mathrm{N}} &= \{M : M(M) = \text{ no}\} \\
L_{\mathrm{N},\le} &= \{(M,t) : M(M) = \text{ no in} \le t \text{ steps}\} \\
\Sigma_{\mathrm{N},\le} &= \{\overbrace{\mathbf{A}_M \wedge \mathbf{A}_M \wedge \cdots \wedge \mathbf{A}_M}^{t} : (M,t) \in L_{\mathrm{N},\le}\}
\end{aligned}
$$

Then $\Sigma_{\mathrm{Y},\le}$ and $\Sigma_{\mathrm{N},\le}$ are decidable.

Back to WFFs

Put

$$
\begin{aligned}
L_{\mathrm{Y}} &= \{M : M(M) = \text{ yes}\} \\
L_{\mathrm{Y},\le} &= \{(M,t) : M(M) = \text{ yes in } \le t \text{ steps}\} \\
\Sigma_{\mathrm{Y},\le} &= \{\overbrace{\mathbf{A}_M \wedge \mathbf{A}_M \wedge \cdots \wedge \mathbf{A}_M}^{t} : (M,t) \in L_{\mathrm{Y},\le}\}
\end{aligned}
$$

and

$$
\begin{aligned}
L_{\mathrm{N}} &= \{M : M(M) = \text{ no}\} \\
L_{\mathrm{N},\le} &= \{(M,t) : M(M) = \text{ no in } \le t \text{ steps}\} \\
\Sigma_{\mathrm{N},\le} &= \{\overbrace{\mathbf{A}_M \wedge \mathbf{A}_M \wedge \cdots \wedge \mathbf{A}_M}^{t} : (M,t) \in L_{\mathrm{N},\le}\}
\end{aligned}
$$

Then $\Sigma_{\mathrm{Y},\le}$ and $\Sigma_{\mathrm{N},\le}$ are decidable.
Also, note $M \in L_i$ iff $\Sigma_{i,\le} \models \mathbf{A}_M$, $i = $Y,N.

Inseparable

Note $M \in L_i$ iff $\Sigma_{i,\leq} \vDash \mathbf{A}_M$, i =Y,N.

Theorem

$\overline{\Sigma_{Y,\leq}}$ and $\overline{\Sigma_{N,\leq}}$ are recursively inseparable.

Proof.

If some recursive R' separated $\overline{\Sigma_{\mathrm{Y},\leq}}$ and $\overline{\Sigma_{\mathrm{N},\leq}}$, then $R = \{M : \mathbf{A}_M \in R'\}$ would separate L_{Y} and L_{N}. $\square$

Significance

Theorem

Suppose

- *v is any truth assignment such that $\overline{v}$ satisfies every $\sigma_Y \in \Sigma_{Y,\leq}$ and satisfies no $\sigma_N \in \Sigma_{N,\leq}$; equivalently, $\overline{v}$ satisfies $\Sigma_{\leq} = \Sigma_{Y,\leq} \cup \{\neg\sigma : \sigma \in \Sigma_{N,\leq}\}$ (v is any reasonable notion of truth compatible with $\Sigma_{\leq}$), and*

Significance

Theorem

Suppose

- *v is any truth assignment such that $\overline{v}$ satisfies every $\sigma_Y \in \Sigma_{Y,\leq}$ and satisfies no $\sigma_N \in \Sigma_{N,\leq}$; equivalently, $\overline{v}$ satisfies $\Sigma_{\leq} = \Sigma_{Y,\leq} \cup \{\neg\sigma : \sigma \in \Sigma_{N,\leq}\}$ (v is any reasonable notion of truth compatible with $\Sigma_{\leq}$), and*
- *C is any r.e. set that is closed under tautological implication and $C \neq$ WFF (C is any reasonable notion of proof).*

Significance

Theorem

Suppose

- *v is any truth assignment such that $\overline{v}$ satisfies every $\sigma_Y \in \Sigma_{Y,\leq}$ and satisfies no $\sigma_N \in \Sigma_{N,\leq}$; equivalently, $\overline{v}$ satisfies $\Sigma_{\leq} = \Sigma_{Y,\leq} \cup \{\neg\sigma : \sigma \in \Sigma_{N,\leq}\}$ (v is any reasonable notion of truth compatible with $\Sigma_{\leq}$), and*
- *C is any r.e. set that is closed under tautological implication and $C \neq$ WFF (C is any reasonable notion of proof).*

Then there is a true but unprovable sentence.

Significance

Theorem

Suppose

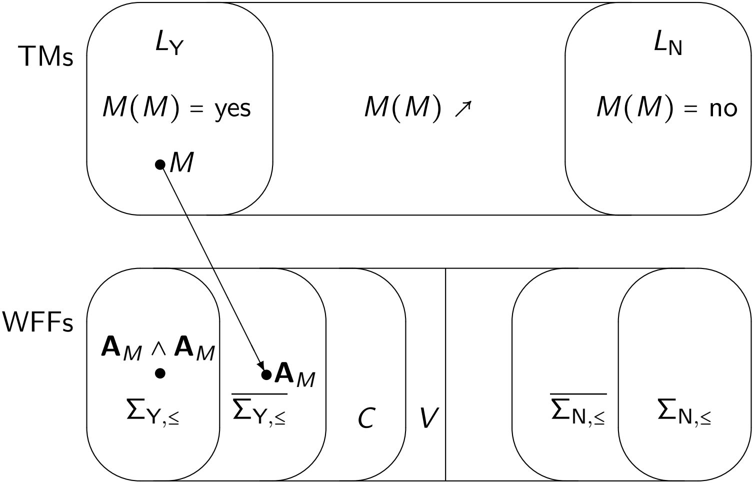
- *v is any truth assignment such that $\overline{v}$ satisfies every $\sigma_Y \in \Sigma_{Y,\leq}$ and satisfies no $\sigma_N \in \Sigma_{N,\leq}$; equivalently, $\overline{v}$ satisfies $\Sigma_{\leq} = \Sigma_{Y,\leq} \cup \{\neg\sigma : \sigma \in \Sigma_{N,\leq}\}$ (v is any reasonable notion of truth compatible with $\Sigma_{\leq}$), and*
- *C is any r.e. set that is closed under tautological implication and $C \neq$ WFF (C is any reasonable notion of proof).*

Then there is a true but unprovable sentence.

Proof.

- $V = \{\sigma : \overline{v}(\sigma) = T\}$ is not decidable, so $C \neq V$.
- If $V \subseteq C$, then $\exists\sigma \in V$ with $\sigma \in C$ and $\neg\sigma \in C$.
- Since C is closed, $C =$ WFF. (Contradiction.)

Picture


TMs
L_{Y}
$M(M)$ = yes
M
$M(M) \nearrow$
L_{N}
$M(M)$ = no
WFFs
$\mathbf{A}_M \wedge \mathbf{A}_M$
$\Sigma_{\mathrm{Y},\leq}$
$\mathbf{A}_M$
$\overline{\Sigma_{\mathrm{Y},\leq}}$
C
V
$\overline{\Sigma_{\mathrm{N},\leq}}$
$\Sigma_{\mathrm{N},\leq}$

Some [sic] notion of truth?

Some (different) examples:

1. $v(\mathbf{A}_M)$ is true iff $M(M)$ halts and says yes.
2. $v(\mathbf{A}_M)$ is true unless $M(M)$ halts and says no.
3. $v(\mathbf{A}_M)$ is true iff $M(M)$ halts and says yes, or if $M(M)$ loops and $|M|$ is a prime number.

Fix?

Suppose ϕ is true but unprovable. (Note: ϕ can be a sentence symbol.)
What about $\Sigma_{\leq}; \phi$?

Fix?

Suppose ϕ is true but unprovable. (Note: ϕ can be a sentence symbol.)
What about $\Sigma_{\leq}; \phi$?

- $\Sigma_{\leq}; \phi$ is decidable, since each of $\Sigma_{\leq}$ and $\{\phi\}$ is.

Fix?

Suppose ϕ is true but unprovable. (Note: ϕ can be a sentence symbol.) What about $\Sigma_{\leq}; \phi$?

- $\Sigma_{\leq}; \phi$ is decidable, since each of $\Sigma_{\leq}$ and $\{\phi\}$ is.
- For each possible proof system C, there's some *other* ψ, with $\Sigma_{\leq}; \phi \vDash \psi$ unprovable.

Fix?

Suppose ϕ is true but unprovable. (Note: ϕ can be a sentence symbol.) What about $\Sigma_{\leq}; \phi$?

- $\Sigma_{\leq}; \phi$ is decidable, since each of $\Sigma_{\leq}$ and $\{\phi\}$ is.
- For each possible proof system C, there's some *other* ψ, with $\Sigma_{\leq}; \phi \vDash \psi$ unprovable.
- Throw in ψ, too?

Fix?

Suppose ϕ is true but unprovable. (Note: ϕ can be a sentence symbol.)
What about $\Sigma_{\leq}; \phi$?

- $\Sigma_{\leq}; \phi$ is decidable, since each of $\Sigma_{\leq}$ and $\{\phi\}$ is.
- For each possible proof system C, there's some *other* ψ, with $\Sigma_{\leq}; \phi \vDash \psi$ unprovable.
- Throw in ψ, too?
  - Throwing in an r.e. set of ψ's makes no qualitative progress.

Fix?

Suppose ϕ is true but unprovable. (Note: ϕ can be a sentence symbol.) What about $\Sigma_{\leq};\phi$?

- $\Sigma_{\leq};\phi$ is decidable, since each of $\Sigma_{\leq}$ and $\{\phi\}$ is.
- For each possible proof system C, there's some *other* ψ, with $\Sigma_{\leq};\phi \vDash \psi$ unprovable.
- Throw in ψ, too?
    - Throwing in an r.e. set of ψ's makes no qualitative progress.
    - Throwing in more than an r.e. set makes the new Σ not r.e. (unreasonable).

Fix?

Suppose ϕ is true but unprovable. (Note: ϕ can be a sentence symbol.) What about $\Sigma_{\leq}; \phi$?

- $\Sigma_{\leq}; \phi$ is decidable, since each of $\Sigma_{\leq}$ and $\{\phi\}$ is.
- For each possible proof system C, there's some *other* ψ, with $\Sigma_{\leq}; \phi \vDash \psi$ unprovable.
- Throw in ψ, too?
    - Throwing in an r.e. set of ψ's makes no qualitative progress.
    - Throwing in more than an r.e. set makes the new Σ not r.e. (unreasonable).

Later, in first-order logic, the definition of $\mathbf{A}_M$ becomes part of the logic.

Where We Stand

Done with Boolean Logic (we'll build on it, next)

Where We Stand

Done with Boolean Logic (we'll build on it, next)
Gave algorithms for

Where We Stand

Done with Boolean Logic (we'll build on it, next)
Gave algorithms for

- testing whether σ is a WFF
- writing a Boolean function over $\{\wedge, \neg\}$, but not over $\{\wedge, \vee\}$, etc.
- testing whether σ is true under $\overline{v}$
- $\Sigma \models \sigma$ for finite Σ.

Where We Stand

Done with Boolean Logic (we'll build on it, next)
Gave algorithms for

- testing whether σ is a WFF
- writing a Boolean function over $\{\wedge, \neg\}$, but not over $\{\wedge, \vee\}$, etc.
- testing whether σ is true under $\overline{v}$
- $\Sigma \models \sigma$ for finite Σ.

For decidable Σ,

Where We Stand

Done with Boolean Logic (we'll build on it, next)
Gave algorithms for

- testing whether σ is a WFF
- writing a Boolean function over $\{\wedge, \neg\}$, but not over $\{\wedge, \vee\}$, etc.
- testing whether σ is true under $\overline{v}$
- $\Sigma \models \sigma$ for finite Σ.

For decidable Σ,

- $\Sigma \models \tau$ is recursively enumerable.
- $\Sigma \models \tau$ may not be decidable.

Where We Stand

Done with Boolean Logic (we'll build on it, next)
Gave algorithms for

- testing whether σ is a WFF
- writing a Boolean function over $\{\wedge, \neg\}$, but not over $\{\wedge, \vee\}$, etc.
- testing whether σ is true under $\overline{v}$
- $\Sigma \models \sigma$ for finite Σ.

For decidable Σ,

- $\Sigma \models \tau$ is recursively enumerable.
- $\Sigma \models \tau$ may not be decidable.

Reformulation:
For certain Σ, given any v compatible with Σ and any r.e., closed, not-all-WFF proof system C, there is a true but unprovable sentence.