BILL AND NATHAN RECORD LECTURE!!!!

BILL AND NATHAN RECORD LECTURE!!!

FINAL IS FRIDAY May 17 10:30AM-12:30PM

FILL OUT COURSE EVALS for ALL YOUR COURSES!!!

Review for Final

Rules

1. **Begin** Final Tuesday May 17, 10:30PM-12:30PM in CSI 3117. (IF this is a problem for you contact me ASAP!!)

Rules

1. **Begin** Final Tuesday May 17, 10:30PM-12:30PM in CSI 3117. (IF this is a problem for you contact me ASAP!!)
2. **Resources** You can bring two sheets of notes and use both sides.

Rules

1. **Begin** Final Tuesday May 17, 10:30PM-12:30PM in CSI 3117. (IF this is a problem for you contact me ASAP!!)
2. **Resources** You can bring two sheets of notes and use both sides.
3. **Warning** Cramming the entire course on to those pages does not work.

Rules

1. **Begin** Final Tuesday May 17, 10:30PM-12:30PM in CSI 3117. (IF this is a problem for you contact me ASAP!!)
2. **Resources** You can bring two sheets of notes and use both sides.
3. **Warning** Cramming the entire course on to those pages does not work.
4. **Scope of the Exam:** My Slides and the HW.

Turing Machines

Turing Machines

1. For this review we omit definitions and conventions.

Turing Machines

1. For this review we omit definitions and conventions.
2. There is a JAVA program for function f iff there is a TM that computes f.

Turing Machines

1. For this review we omit definitions and conventions.
2. There is a JAVA program for function f iff there is a TM that computes f.
3. Everything computable can be done by a TM.

Decidable Sets

Def A set A is DECIDABLE if there is a Turing Machine M such that

Decidable Sets

Def A set A is DECIDABLE if there is a Turing Machine M such that

$$x \in A \to M(x) = Y$$

Decidable Sets

Def A set A is DECIDABLE if there is a Turing Machine M such that

$$x \in A \to M(x) = Y$$

$$x \notin A \to M(x) = N$$

What is a Theory

What is a Theory

1. All theories have the usual logical symbols, a domain of discourse for the quantifiers, and **Additional Symbols** .

What is a Theory

1. All theories have the usual logical symbols, a domain of discourse for the quantifiers, and **Additional Symbols** .
2. Sentences are combos of Atomic Fmls using $\wedge$, $\vee$, $\neg$, $\exists$ that have all variables quantified over.

What is a Theory

1. All theories have the usual logical symbols, a domain of discourse for the quantifiers, and **Additional Symbols** .
2. Sentences are combos of Atomic Fmls using $\wedge$, $\vee$, $\neg$, $\exists$ that have all variables quantified over.
3. Hence sentences are either TRUE or FALSE.

What is a Theory

1. All theories have the usual logical symbols, a domain of discourse for the quantifiers, and **Additional Symbols** .
2. Sentences are combos of Atomic Fmls using $\wedge$, $\vee$, $\neg$, $\exists$ that have all variables quantified over.
3. Hence sentences are either TRUE or FALSE.
4. Our main question will be **Is this theory decidable?**

WS1S Formulas and Sentences

WS1S Formulas and Sentences

1. Variables x, y, z range over $\mathbb{N}$, X, Y, Z range over finite subsets of $\mathbb{N}$.

WS1S Formulas and Sentences

1. Variables x, y, z range over $\mathbb{N}$, X, Y, Z range over finite subsets of $\mathbb{N}$.
2. Symbols: $<$, $\in$, $\equiv \pmod{}$ (usual meaning), S (meaning $S(x) = x + 1$), $=$ (for numbers and sets).

WS1S Formulas and Sentences

1. Variables x, y, z range over $\mathbb{N}$, X, Y, Z range over finite subsets of $\mathbb{N}$.
2. Symbols: $<$, $\in$, $\equiv \pmod{}$ (usual meaning), S (meaning $S(x) = x + 1$), $=$ (for numbers and sets).
3. Define atomic formulas, formulas, and sentences in the usual way.

TRUE Sets

Def If $\phi(x_1, \ldots, x_n, X_1, \ldots, X_m)$ is a WS1S Formula then $TRUE(\phi)$ is the set

TRUE Sets

Def If $\phi(x_1, \ldots, x_n, X_1, \ldots, X_m)$ is a WS1S Formula then $TRUE(\phi)$ is the set

$$\{(a_1, \ldots, a_n, A_1, \ldots, A_m) \mid \phi(a_1, \ldots, a_n, A_1, \ldots, A_m) = T\}$$

KEY THEOREM

Thm For all WS1S formulas ϕ the set $TRUE_{\phi}$ is regular.

KEY THEOREM

Thm For all WS1S formulas ϕ the set $TRUE_\phi$ is regular.

Need to clarify representation and the define stupid states to make all of this work.

KEY THEOREM

Thm For all WS1S formulas ϕ the set $TRUE_\phi$ is regular.

Need to clarify representation and the define stupid states to make all of this work.

We prove this by induction on the formation of a formula. If you prefer- induction on the LENGTH of a formula.

DECIDABILITY OF WS1S

Thm: WS1S is Decidable.
Proof:

1. Given a SENTENCE in WS1S put it into the form

   $$(Q_1 X_1) \cdots (Q_n X_n)(Q_{n+1} x_1) \cdots (Q_{n+m} x_m)[\phi(x_1, \ldots, x_m, X_1, \ldots, X_n)]$$

2. Assume $Q_1 = \exists$. (If not then negate and negate answer.)
3. View as $(\exists X)[\phi(X)]$, a FORMULA with ONE free var.
4. Construct DFA M for $\{X \mid \phi(X) \text{ is true}\}$.
5. Test if $L(M) = \emptyset$.
6. If $L(M) \neq \emptyset$ then $(\exists X)[\phi(X)]$ is TRUE.
   If $L(M) = \emptyset$ then $(\exists X)[\phi(X)]$ is FALSE.

$(\mathbb{Q}, <)$ Formulas and Sentences

$(\mathbb{Q}, <)$ Formulas and Sentences

1. Variables x, y, z range over $\mathbb{Q}$.

$(\mathbb{Q}, <)$ Formulas and Sentences

1. Variables x, y, z range over $\mathbb{Q}$.
2. Symbols: $<$, $=$ (usual meaning)

$(\mathbb{Q}, <)$ Formulas and Sentences

1. Variables x, y, z range over $\mathbb{Q}$.
2. Symbols: $<$, $=$ (usual meaning)
3. Atomic formulas, formulas, sentences, defined in usual way.

Lemma on Quantifier Elimination

Lemma $\exists$ an algorithm that will, given a sentence of the form

$$(Q_1 x_1) \cdots (Q_{n-1} x_{n-1})(\exists x_n)[\phi(x_1, \ldots, x_n)]$$

(where the Q_i are quantifiers) return a sentence of the form

$$(Q_1 x_1) \cdots (Q_{n-1} x_{n-1})[\phi'(x_1, \ldots, x_{n-1})]$$

$(\mathbb{Q}, <)$ is Decidable: The Algorithm

$(\mathbb{Q}, <)$ is Decidable: The Algorithm

Algorithm

$(\mathbb{Q}, <)$ is Decidable: The Algorithm

Algorithm

1. $(Q_1 x_1) \cdots (Q_n x_n)[\phi(x_1, \ldots, x_n)]$. Replace $\forall$ with $\neg\exists\neg$.

$(\mathbb{Q}, <)$ is Decidable: The Algorithm

Algorithm

1. $(Q_1x_1)\cdots(Q_nx_n)[\phi(x_1,\ldots,x_n)]$. Replace $\forall$ with $\neg\exists\neg$.
2. Apply the Quant Elim Lemma over and over again until either you end up with a TRUE or a FALSE or a sentence with one variable whose truth will be easily discerned.

Undecidability

Notation

Notation

Notation $M_{e,s}(d)$ is the result of running $M_e(d)$ for s steps.

Notation

Notation $M_{e,s}(d)$ is the result of running $M_e(d)$ for s steps.
$M_e(d)\downarrow$ means $M_e(d)$ halts.

Notation

Notation $M_{e,s}(d)$ is the result of running $M_e(d)$ for s steps.
$M_e(d)\downarrow$ means $M_e(d)$ halts.
$M_e(d)\uparrow$ means $M_e(d)$ does not halts.

Notation

Notation $M_{e,s}(d)$ is the result of running $M_e(d)$ for s steps.
$M_e(d)\downarrow$ means $M_e(d)$ halts.
$M_e(d)\uparrow$ means $M_e(d)$ does not halts.
$M_{e,s}(d)\downarrow$ means $M_e(d)$ halts within s steps.

Notation

Notation $M_{e,s}(d)$ is the result of running $M_e(d)$ for s steps.
$M_e(d)\downarrow$ means $M_e(d)$ halts.
$M_e(d)\uparrow$ means $M_e(d)$ does not halts.
$M_{e,s}(d)\downarrow$ means $M_e(d)$ halts within s steps.
$M_{e,s}(d)\downarrow= z$ means $M_e(d)$ halts within s steps and outputs z.

Notation

Notation $M_{e,s}(d)$ is the result of running $M_e(d)$ for s steps.
$M_e(d) \downarrow$ means $M_e(d)$ halts.
$M_e(d) \uparrow$ means $M_e(d)$ does not halts.
$M_{e,s}(d) \downarrow$ means $M_e(d)$ halts within s steps.
$M_{e,s}(d) \downarrow = z$ means $M_e(d)$ halts within s steps and outputs z.
$M_{e,s}(d) \uparrow$ means $M_e(d)$ has not halted within s steps.

Notation

Notation $M_{e,s}(d)$ is the result of running $M_e(d)$ for s steps.
$M_e(d) \downarrow$ means $M_e(d)$ halts.
$M_e(d) \uparrow$ means $M_e(d)$ does not halts.
$M_{e,s}(d) \downarrow$ means $M_e(d)$ halts within s steps.
$M_{e,s}(d) \downarrow = z$ means $M_e(d)$ halts within s steps and outputs z.
$M_{e,s}(d) \uparrow$ means $M_e(d)$ has not halted within s steps.

Noncomputable Sets

Are there any noncomputable sets?

1. Yes—ALL SETS: uncountable. DEC Sets: countable, hence there exists an uncountable number of noncomputable sets.
2. YES—HALT is undecidable, and once you have that you have many other sets undec.
3. YES—the problem of telling if a $p \in \mathbb{Z}[x_1, \ldots, x_n]$ has an int solution is undecidable.
4. YES—there are other natural problems that are undecidable.

The HALTING Problem

Def The HALTING set is the set

$$HALT = \{(e, d) \mid M_e(d) \text{ halts } \}.$$

The HALTING Problem

Def The HALTING set is the set

$$HALT = \{(e, d) \mid M_e(d) \text{ halts } \}.$$

Thm HALT is not computable.

Def $A \in \Sigma_1$ if there exists decidable B such that

$$A = \{x : (\exists y)[(x, y) \in B]\}$$

Def $A \in \Sigma_1$ if there exists decidable B such that

$$A = \{x : (\exists y)[(x, y) \in B]\}$$

Similar to NP.

Beyond Σ_1

Def B is always a decidable set.

Beyond Σ_1

Def B is always a decidable set.

$A \in \Pi_1$ if $A = \{x : (\forall y)[(x, y) \in B]\}$.

Beyond Σ_1

Def B is always a decidable set.

$A \in \Pi_1$ if $A = \{x : (\forall y)[(x, y) \in B]\}$.

$A \in \Sigma_2$ if $A = \{x : (\exists y_1)(\forall y_2)[(x, y_1, y_2) \in B]\}$.

Beyond Σ_1

Def B is always a decidable set.

$A \in \Pi_1$ if $A = \{x : (\forall y)[(x, y) \in B]\}$.

$A \in \Sigma_2$ if $A = \{x : (\exists y_1)(\forall y_2)[(x, y_1, y_2) \in B]\}$.

$A \in \Pi_2$ if $A = \{x : (\forall y_1)(\exists y_2)[(x, y_1, y_2) \in B]\}$.

$\vdots$

Beyond Σ_1

Def B is always a decidable set.

$A \in \Pi_1$ if $A = \{x : (\forall y)[(x, y) \in B]\}$.

$A \in \Sigma_2$ if $A = \{x : (\exists y_1)(\forall y_2)[(x, y_1, y_2) \in B]\}$.

$A \in \Pi_2$ if $A = \{x : (\forall y_1)(\exists y_2)[(x, y_1, y_2) \in B]\}$.

$\vdots$

$TOT = \{x : (\forall y)(\exists s)[M_{x,s}(y) \downarrow]\} \in \Pi_2$.

Beyond Σ_1

Def B is always a decidable set.

$A \in \Pi_1$ if $A = \{x : (\forall y)[(x, y) \in B]\}$.

$A \in \Sigma_2$ if $A = \{x : (\exists y_1)(\forall y_2)[(x, y_1, y_2) \in B]\}$.

$A \in \Pi_2$ if $A = \{x : (\forall y_1)(\exists y_2)[(x, y_1, y_2) \in B]\}$.

$\vdots$

$TOT = \{x : (\forall y)(\exists s)[M_{x,s}(y) \downarrow]\} \in \Pi_2$.

Known: $TOT \notin \Sigma_1 \cup \Pi_1$.

Beyond Σ_1

Def B is always a decidable set.

$A \in \Pi_1$ if $A = \{x : (\forall y)[(x, y) \in B]\}$.

$A \in \Sigma_2$ if $A = \{x : (\exists y_1)(\forall y_2)[(x, y_1, y_2) \in B]\}$.

$A \in \Pi_2$ if $A = \{x : (\forall y_1)(\exists y_2)[(x, y_1, y_2) \in B]\}$.

$\vdots$

$TOT = \{x : (\forall y)(\exists s)[M_{x,s}(y) \downarrow]\} \in \Pi_2$.

Known: $TOT \notin \Sigma_1 \cup \Pi_1$.

Known:

$\Sigma_1 \subset \Sigma_2 \subset \Sigma_3 \cdots$

$\Pi_1 \subset \Pi_2 \subset \Pi_3 \cdots$

Beyond Σ_1

Def B is always a decidable set.

$A \in \Pi_1$ if $A = \{x : (\forall y)[(x, y) \in B]\}$.

$A \in \Sigma_2$ if $A = \{x : (\exists y_1)(\forall y_2)[(x, y_1, y_2) \in B]\}$.

$A \in \Pi_2$ if $A = \{x : (\forall y_1)(\exists y_2)[(x, y_1, y_2) \in B]\}$.

$\vdots$

$TOT = \{x : (\forall y)(\exists s)[M_{x,s}(y) \downarrow]\} \in \Pi_2$.

Known: $TOT \notin \Sigma_1 \cup \Pi_1$.

Known:

$\Sigma_1 \subset \Sigma_2 \subset \Sigma_3 \cdots$

$\Pi_1 \subset \Pi_2 \subset \Pi_3 \cdots$

TOT is **harder** than HALT.

Kolmogorov Complexity

Def of Randomness

Def

Def of Randomness

Def

1. If $x \in \{0,1\}^n$ then $\boldsymbol{C(x)}$ is the length of the shortest TM that, on input e, prints out x. Note that $C(x) \leq n + O(1)$.

Def of Randomness

Def

1. If $x \in \{0,1\}^n$ then **$C(x)$** is the length of the shortest TM that, on input e, prints out x. Note that $C(x) \le n + O(1)$.
2. A string is **Kolmogorov random** if $C(x) \ge n$.

Def of Randomness

Def

1. If $x \in \{0,1\}^n$ then **$C(x)$** is the length of the shortest TM that, on input e, prints out x. Note that $C(x) \le n + O(1)$.
2. A string is **Kolmogorov random** if $C(x) \ge n$.

Note Machine Ind up to additive $O(1)$.

Do Kolm-Random Strings Exist?

Is there a string of length n that has $C(x) \geq n$?

YES- there are more Strings of length n then TMs of length $\leq n-1$.

Applications

Kolm Random Strings were used for:

Applications

Kolm Random Strings were used for:

1. Alternative way to show langs are regular (we did this).

Applications

Kolm Random Strings were used for:

1. Alternative way to show langs are regular (we did this).
2. Gave a string w such that any CFG G with $L(G) = \{w\}$ is large. (this was HW).

Applications

Kolm Random Strings were used for:

1. Alternative way to show langs are regular (we did this).
2. Gave a string w such that any CFG G with $L(G) = \{w\}$ is large. (this was HW).
3. Avg case analysis (we did not do this).

Applications

Kolm Random Strings were used for:

1. Alternative way to show langs are regular (we did this).
2. Gave a string w such that any CFG G with $L(G) = \{w\}$ is large. (this was HW).
3. Avg case analysis (we did not do this).
4. Lower bounds for a variety of models of computation (we did not do this).

BILL AND NATHAN RECORD LECTURE!!!!

BILL AND NATHAN RECORD LECTURE!!!

FINAL IS THURSDAY

May 17

10:30PM-2:30PM

FILL OUT COURSE EVALS for ALL YOUR COURSES!!!