MA/CSSE 474

Theory of Computation

Math Review
More on Languages

Questions?

- Syllabus
- Yesterday's discussion
- Reading Assignment

I often put more in the slides and quizzes than I think we'll get through before the end of class…
… just in case things go faster than I expect.
… as a preview of things to come.

Leftovers from Day 1

The big question:

Given a language description, which strings are in the language?

Example Language Definitions

$L = \{x \in \{\texttt{a}, \texttt{b}\}^* :$ all `a`'s precede all `b`'s$\}$

ε, `a`, `aa`, `aabbb`, and `bb` are in L.

`aba`, `ba`, and `abc` are not in L.

$L = \{x : \exists y \in \{\texttt{a}, \texttt{b}\}^* : x = y\texttt{a}\}$

Simple English description?

$L = \{\texttt{a}^n : n \geq 0\}$

This definition uses replication

$L = \varnothing = \{\ \}$

$L = \{\varepsilon\}$

Note that the last two are different languages

The Perils of English descriptions

$L = \{x\#y$: x, $y \in$ {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}* and, when x and y are viewed as the decimal representations of natural numbers, *square*(x) = $y\}$.

In L: `3#9` `12#144`

Not in L: `3#8` `12` `12#12#12`

In L? `#`

Natural Languages are Tricky

$L = \{w$: w is a sentence in English$\}$.

Examples:

Kerry hit the ball.

Colorless green ideas sleep furiously.

The window needs fixed.

Ball the Stacy hit blue.

A *Halting Problem* Language

L = {w: w is a C program that always halts, no matter what input it is given}.

- Well-specified.
- But can we decide which strings L contains?

Languages and Prefixes

What are the following languages:

$L = \{w \in \{\texttt{a}, \texttt{b}\}^*$: no prefix of w contains $\texttt{b}\}$

$L = \{w \in \{\texttt{a}, \texttt{b}\}^*$: no prefix of w starts with $\texttt{a}\}$

$L = \{w \in \{\texttt{a}, \texttt{b}\}^*$: every prefix of w starts with $\texttt{a}\}$

Q1

Sets and Relations

Defining a (possibly infinite)Set

- Write a program that ***enumerates*** the elements of *S*.
- Write a program that ***decides*** *S* by implementing the ***characteristic function*** of *S*. Such a program returns *True* if run on an element that is in *S* and *False* if run on an element that is not in *S*.

Q2

Cardinality

The cardinality of every set we will consider is:

- a natural number (if *S* is finite),
- "countably infinite" (if *S* has the same number of elements as there are integers), or
- "uncountably infinite" (if *S* has more elements than there are integers).

Sets of Sets

- The ***power set*** of *A* is the set of all subsets of *A*.

  Let $A = \{1, 2, 3\}$. Then:

  $\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$.

- $\Pi \subseteq P(A)$ is a ***partition*** of a set *A* iff:
  - no element of Π is empty,
  - all pairs of elements of Π are disjoint , and
  - the union of all the elements of Π equals *A*.

  Partitions of *A*:

  $\{\{1\}, \{2, 3\}\}$ or $\{\{1, 3\}, \{2\}\}$ or $\{\{1, 2, 3\}\}$.

Closure

- A set S is closed under binary operation *op* iff
  $\forall x,y \in S$ (x *op* y $\in$ S)
- $\mathbb{N}$ is closed under addition and multiplication but not subtraction or division.
- The set of finite sets is closed under union and intersection.

Equivalence Relations

A relation $R \subseteq A \times A$ is an ***equivalence relation*** iff it is:

- reflexive,
- symmetric, and
- transitive.

Examples:

- Equality
- Lives-at-Same-Address-As
- Same-Length-As

Q3, 4

Operations on Languages

Concatenation of Languages

If L_1 and L_2 are languages over Σ:

$$L_1L_2 = \{w \in \Sigma^* : \exists s \in L_1 (\exists t \in L_2 (w = st))\}$$

Examples:

$L_1 = \{$`cat`, `dog`$\}$
$L_2 = \{$`apple`, `pear`$\}$
$L_1\ L_2 = \{$`catapple`, `catpear`, `dogapple`, `dogpear`$\}$

$L_1 = \mathtt{a}^*$ $L_2 = \mathtt{b}^*$
$L_1\ L_2 =$

Concatenation of Languages

$\{\varepsilon\}$ is the identity for concatenation:

$L\{\varepsilon\} = \{\varepsilon\}L = L$

$\varnothing$ is a zero for concatenation:

$L\,\varnothing = \varnothing\, L = \varnothing$

$L_1 = \{\mathtt{a}^n: \ n \geq 0\}$
$L_2 = \{\mathtt{b}^n : n \geq 0\}$

$L_1\, L_2 = \{\mathtt{a}^n\mathtt{b}^m : n, m \geq 0\}$

$L_1 L_2 \neq \{\mathtt{a}^n\mathtt{b}^n : n \geq 0\}$

Q5a

Kleene Star

$L^* = \{\varepsilon\} \cup$
$\{w \in \Sigma^* : \exists k \geq 1$
$(\exists w_1, w_2, \ldots w_k \in L \ (w = w_1\, w_2 \ldots w_k))\}$

Example:
$L = \{$dog, cat, fish$\}$
$L^* = \{\varepsilon$, dog, cat, fish, dogdog,
dogcat, fishcatfish,
fishdogdogfishcat, …$\}$

Q5b

The + Operator

$L^+ = L\, L^*$

$L^+ = L^* - \{\varepsilon\}$ iff $\varepsilon \notin L$

L^+ is the closure of L under concatenation.

Concatenation and Reverse of Languages

Theorem: $(L_1\, L_2)^R = L_2^R\, L_1^R$.

Proof:

$\forall x\, (\forall y\, ((xy)^R = y^R x^R))$ Theorem 2.1

$(L_1\, L_2)^R = \{(xy)^R : x \in L_1 \text{ and } y \in L_2\}$ Definition of concatenation of languages

$= \{y^R x^R : x \in L_1 \text{ and } y \in L_2\}$ Lines 1 and 2

$= L_2^R\, L_1^R$ Definition of concatenation of languages

Determining Language Membership

Computational approach:

- **Generator** (enumerator)
  When it is asked, it gives us the next element of the language.
  Any given element of the language will appear within a finite amount of time.

- **Recognizer**
  Given a string s, recognizer accepts s if it is in the language.
  If not, it either rejects s or keeps running forever.

Enumeration

Enumeration:

- Arbitrary order

- More useful: ***lexicographic order***
  - Shortest first
  - Within a length, dictionary order

The lexicographic enumeration of:

- $\{w \in \{\mathtt{a}, \mathtt{b}\}^* : |w| \text{ is even}\}$:

Review: How Large is a Language?

The smallest language over any Σ is $\varnothing$, with cardinality 0.

The largest is Σ^*. How big is it?

Theorem: If $\Sigma \neq \varnothing$ then Σ^* is countably infinite.

Proof: The elements of Σ^* can be lexicographically enumerated by the following procedure:

- Enumerate all strings of length 0, then length 1, then length 2, and so forth.
- Within the strings of a given length, enumerate them in dictionary order.

This enumeration is infinite since there is no longest string in Σ^*. Since there exists an infinite enumeration of Σ^*, it is countably infinite.

How Many Languages Are There?

Theorem: If $\Sigma \neq \varnothing$ then the set of languages over Σ is uncountably infinite.

Proof: The set of languages defined on Σ is $\mathcal{P}(\Sigma^*)$. Σ^* is countably infinite. If S is a countably infinite set, $\mathcal{P}(S)$ is uncountably infinite. So $\mathcal{P}(\Sigma^*)$ is uncountably infinite.

Logic: Propositional and first-order

From Rich, Appendix A

Most of this material also appears in Grimaldi's Discrete Math book, Chapter 2

Boolean (Propositional) Logic Wffs

A ***wff*** (well-formed formula) is any string that is formed according to the following rules:

1. A propositional symbol (variable or constant) is a wff.
2. If P is a wff, then $\neg P$ is a wff.
3. If P and Q are wffs, then so are:
   $P \vee Q$, $P \wedge Q$, $P \rightarrow Q$, $P \leftrightarrow Q$, and (P).

P	Q	$\neg P$	$P \vee Q$	$P \wedge Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
True	*True*	*False*	*True*	*True*	*True*	*True*
True	*False*	*False*	*True*	*False*	*False*	*False*
False	*True*	*True*	*True*	*False*	*True*	*False*
False	*False*	*True*	*False*	*False*	*True*	*True*

When Wffs are True

- A wff is ***valid*** or is a ***tautology*** iff it is true for all assignments of truth values to the variables it contains.
- A wff is ***satisfiable*** iff it is true for at least one assignment of truth values to the variables it contains.
- A wff is ***unsatisfiable*** iff it is false for all assignments of truth values to the variables it contains.
- Two wffs P and Q are ***equivalent***, written $P \equiv Q$, iff they have the same truth values for every assignment of truth values to the variables they contain.

$P \vee \neg P$ is a tautology:

P	$\neg P$	$P \vee \neg P$
True	*False*	*True*
False	*True*	*True*

Q6

Entailment

A set S of wffs ***logically implies*** or ***entails*** a conclusion Q iff, whenever all of the wffs in S are true, Q is also true.

Example:

$\{A \wedge B \wedge C, D\}$ entails $A \rightarrow D$

Inference Rules

- An inference rule is ***sound*** iff, whenever it is applied to a set *A* of axioms, any conclusion that it produces is entailed by *A*.
- An entire proof is sound iff it consists of a sequence of inference steps each of which was constructed using a sound inference rule.
- A set of inference rules *R* is ***complete*** iff, given any set *A* of axioms, all statements that are entailed by *A* can be proved by applying the rules in *R*.

Q7

Some Sound Inference Rules

- ***Modus ponens***: From $(P \rightarrow Q)$ and P, conclude Q.
- ***Modus tollens***: From $(P \rightarrow Q)$ and $\neg Q$, conclude $\neg P$.
- ***Or introduction***: From P, conclude $(P \vee Q)$.
- ***And introduction***: From P and Q, conclude $(P \wedge Q)$.
- ***And elimination***: From $(P \wedge Q)$, conclude P or conclude Q.
- ***Syllogism***: From $(P \rightarrow Q)$ and $(Q \rightarrow R)$, conclude $(P \rightarrow R)$.

Q3

Additional Sound Inference Rules

- ***Quantifier exchange***:
  - From $\neg\exists x\ (P)$, conclude $\forall x\ (\neg P)$.
  - From $\forall x\ (\neg P)$, conclude $\neg\exists x\ (P)$.
  - From $\neg\forall x\ (P)$, conclude $\exists x\ (\neg P)$.
  - From $\exists x\ (\neg P)$, conclude $\neg\forall x\ (P)$.
- ***Universal instantiation***: For any constant C, from $\forall x\ (P(x))$, conclude $P(C)$.
- ***Existential generalization***: For any constant C, from $P(C)$ conclude $\exists x\ (P(x))$.

First-Order Logic

A **term** is a variable, constant, or function application.
A ***well-formed formula (wff)*** in first-order logic is an expression that can be formed by:

- If P is an n-ary predicate and each of the expressions $x_1, x_2, \ldots, x_n$ is a term, then an expression of the form $P(x_1, x_2, \ldots, x_n)$ is a wff. If any variable occurs in such a wff, then that variable occurs ***free*** *in* $P(x_1, x_2, \ldots, x_n)$.
- If P is a wff, then $\neg P$ is a wff.
- If P and Q are wffs, then so are $P \vee Q$, $P \wedge Q$, $P \rightarrow Q$, and $P \leftrightarrow Q$.
- If P is a wff, then (P) is a wff.
- If P is a wff, then $\forall x\ (P)$ and $\exists x\ (P)$ are wffs. Any free instance of x in P is ***bound*** by the quantifier and is then no longer free.

Q8

Sentences

A wff with no free variables is called a ***sentence*** or a ***statement***.

1. *Bear*(*Smokey*).
2. $\forall x\ (Bear(x) \rightarrow Animal(x))$.
3. $\forall x\ (Animal(x) \rightarrow Bear(x))$.
4. $\forall x\ (Animal(x) \rightarrow \exists y\ (Mother\text{-}of(y, x)))$.
5. $\forall x\ ((Animal(x) \wedge \neg Dead(x)) \rightarrow Alive(x))$.

Which of these sentences are true in the everyday world?

A ***ground instance*** is a sentence that contains no variables, such as #1

Q9

Interpretations and Models

- An ***interpretation*** for a sentence *w* is a pair (*D*, *I*), where *D* is a universe of objects. *I* assigns meaning to the symbols of *w*: it assigns values, drawn from *D*, to the constants in *w* and it assigns functions and predicates (whose domains and ranges are subsets of *D*) to the function and predicate symbols of *w*.
- A ***model*** of a sentence *w* is an interpretation that makes *w* true. For example, let *w* be the sentence:

  $\forall x\ (\exists y\ (y < x))$.
- A sentence *w* is ***valid*** iff it is true in all interpretations.
- A sentence *w* is ***satisfiable*** iff there exists *some* interpretation in which *w* is true.
- A sentence *w* is ***unsatisfiable*** iff $\neg w$ is valid.

Q10

Examples

- $\forall x\ ((P(x) \wedge Q(Smokey)) \rightarrow P(x))$.
- $\neg(\forall x\ (P(x) \vee \neg(P(x)))$.
- $\forall x\ (P(x, x))$.

A Simple Proof

Assume the following three axioms:

[1]	$\forall x\ (P(x) \wedge Q(x) \rightarrow R(x))$.	
[2]	$P(X_1)$.	
[3]	$Q(X_1)$.	

We prove $R(X_1)$ as follows:

[4]	$P(X_1) \wedge Q(X_1) \rightarrow R(X_1)$.	(Universal instantiation, [1].)
[5]	$P(X_1) \wedge Q(X_1)$.	(And introduction, [2], [3].)
[6]	$R(X_1)$.	(Modus ponens, [5], [4].)

Q11-12

Definition of a Theory

- A first-order **theory** is a set of axioms and the set of all theorems that can be proved, using a set of sound and complete inference rules, from those axioms.
- A theory is logically **complete** iff, for every sentence P in the language of the theory, either P or $\neg$P is a theorem.
- A theory is **consistent** iff there is no sentence P such that both P and $\neg$ P are theorems.
  - If there is such a sentence, then the theory contains a **contradiction** and is **inconsistent**.
- Let w be an interpretation of a theory. The theory is **sound** with respect to w if every theorem in the theory corresponds to a statement that is true in w.