2110711
THEORY
of
COMPUTATION

ATHASIT SURARERKS
ELITE

Athasit Surarerks
ELITE
Engineering Laboratory in Theoretical Enumerable System
Computer Engineering, Faculty of Engineering
Chulalongkorn University
254 Phayathai road, Patumwan, Bangkok 10330
Tel : 0 2218 6989, Fax : 0 2218 6955
Email : athasit@cp.eng.chula.ac.th
Webpage : http://www.cp.eng.chula.ac.th/~athasit

DESCRIPTION

Computable functions
decidable predicates and
solvable problems;
computational complexity;
NP-complete problems;
automata theory;
formal language;
lambda calculus.

EVALUATION

Mid-Term examination	50 %
Final examination	50 %

TEXTBOOK

Essentials of
Theoretical Computer Science

F. D. Lewis

REFERENCES

Mc Graw Hill

- Introduction to Languages and Theory of Computation(3rd ed.) John C. Martin

Addison Wesley

- Introduction to Automata Theory, Languages, and Computation, J.E. Hopcroft, R. Motwani, J.D. Ullman

W IE

- Introduction to Computer Theory (2nd ed.) Daniel I. A. Cohen
- Languages and Machines: An Introduction to the Theory of Computer Science (2nd ed.) Thomas A. Sudkamp

PRINTICE HALL

- Topology (2nd ed.) James R. Munkres

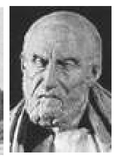
Mc Graw Hill

- Discrete Mathematics and Its Applications (4th ed.)

BACKGROUND

Aristotle (384-322 B.C.)	SYLLOGISTIC REASONING
Euclid of Alexandria (325-265 B.C.)	DEDUCTIVE REASONING
Chrysippus of Soli (279-206 B.C.)	MODAL LOGIC
George Boole (1815-1864 A.D.)	PROPOSITIONAL LOGIC
Augustus De Morgan (1806-1871 A.D.)	DE MORGAN's LAWs





Charles Babbage

1791-1871

- Created the first difference engine (producing the members of the sequence $n^2 + n + 41$ at the rate of about 60 every 5 minutes)
- The 1st drawings of the analytical engine (describes five logical components, the store, the mill, the control, the input and the output)

The construction of modern computers, logically similar to Babbage's design

Kurt Gödel

1906-1978

Proved that there was no algorithm to provide proofs for all the true statements in mathematics.

Universal model for all algorithms.

VARIOUS VERSIONS

OF A UNIVERSAL ALGORITHM MACHINE

- Andrei Andreevich Markov 1856-1922
- Emil Post 1897-1954
- Alonzo Church 1903-1995
- Stephen Kleene 1909-1994
- John von Neumann 1903-1957
- Alain Turing 1912-1954

Alain Turing

1912-1954

•studied problems which today lie at the heart of artificial intelligence.
•proposed the Turing Test which is still today the test people apply in attempting to answer whether a computer can be intelligent.

Computing machinery and intelligence

Warren McCulloch & Walter Pitts

neurophysiologists

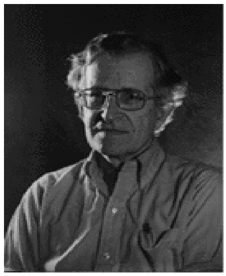
Constructed for a “neural net” was a theoretical machine of the same nature as the one Turing invented.

Modern linguists

Investigated a very similar subject

•What is language in general ?
•How could primitive humans have developed language ?
•How do people understand it ?
•How do they learn it as children ?
•What ideas can be expressed, and in what way ?
•How do people construct sentences from the idea s in their minds ?

Noam
Chomsky



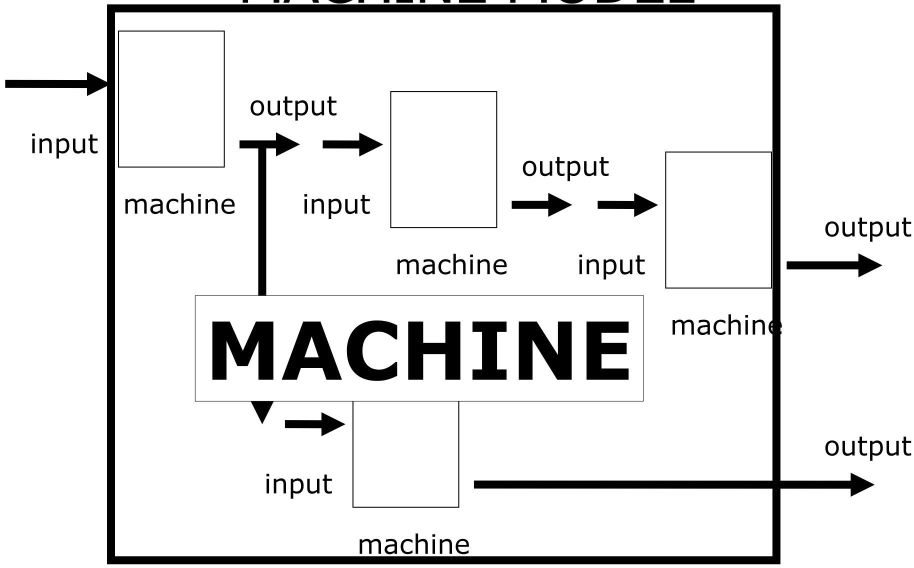
Massachusetts
Institute
of
Technology

Created the subject of mathematical models for the description of languages to answer these questions.

MAIN TOPIC

We shall study different types of theoretical machines that are mathematical models for actual physical processes.

MACHINE MODEL



MAIN CONCLUSIONS

?

" this can be done or it can never be done."

1

LANGUAGES

LANGUAGES

Different entities (in English)

- letters
- words
- sentences
- paragraphs
- coherent stories

COLLECTION
&
SEQUENTIAL

Not all collections of letters form a valid sentence.

Humans agree on which sequences are valid or which are not.

How do they do that ?

COMPUTER LANGUAGES

Different entities
- letters
- words
- commands
- programs
- systems

It is very hard to state all the rules for the language “spoken English”.

Commands can be recognized by certain sequences of words.

Language structure is based on explicitly rules.

LANGUAGE

Language means simply a set of strings involving symbols from alphabet.

THEORY OF FORMAL LANGUAGES

Formal refers

- explicitly rules
  - What sequences of symbols can occur ?
  - No liberties are tolerated.
  - No reference to any "deeper understanding" is required.
- the form of the sequences of symbols
- not the meaning

THEORY OF FORMAL LANGUAGES

STRUCTURE

- One finite set of fundamental units , called "alphabet", denoted Σ.
- An element of alphabet is called "character".
- A certain specified set of strings of characters will be called "language" denoted L.
- Those strings that are permissible in the language we call "words".
- The string without letter is called "empty string" or "null string", denoted by Λ.
- The language that has no word is denoted by $\varnothing$.

SYMBOLS

Union operation	+
Different operation	–
Alphabet	Σ
Empty string	Λ
	ε
Language	L
	Γ
Empty language	$\varnothing$

LANGUAGES

LANGUAGE DEFINING

IMPLICITLY DEFINING

Given an alphabet Σ = { a b c … z ʻ - }.
We can now specify a language L as

{ all words in a standard dictionary },

named "ENGLISH-WORDS".
We define a language Γ as

{ all words in a standard dictionary, blank space,
the usually punctuation marks },

named "ENGLISH-SENTENCES".

LANGUAGES

INFINITE LANGUAGE DEFINING

The trick of defining the language Γ,

By listing all rules of grammar.

This allows us to give a finite description of an infinite language.

Consider this sentence "I eat three Sundays".

RIDICULOUS LANGUAGE

This is grammatically correct.

LANGUAGES

LANGUAGE DEFINING

METHOD OF EXHAUSTION

Let $\Sigma = \{x\}$ be an alphabet.

Language L can be defined by

$L = \{ x \;\; xx \;\; xxx \;\; xxxx \;\; \ldots \}$

$L = \{ x^n \text{ for } n = 1 \;\; 2 \;\; 3 \;\; \ldots \}$.

Language $L_2 = \{ x \;\; xxx \;\; xxxxx \;\; xxxxxxx \;\; \ldots \}$

$L_2 = \{ x^{odd} \}$

$L_2 = \{ x^{2n-1} \text{ for } n = 1 \;\; 2 \;\; 3 \;\; \ldots \}$.

LANGUAGES
SOME DEFINITIONS

We define the function length of a string to be the number of letters in the string.

For example, if a word *a* = xxxx in L, then length(*a*)=4.

In any language that includes Λ, we have length(Λ)=0.

The function reverse is defined by if *a* is a word in L, then reverse(*a*) is the same string of letters spelled backward, called the reverse of *a*.

For example, reverse(123)=321.

Remark: The reverse(*a*) is not necessary in the language of *a*.

LANGUAGES
SOME DEFINITIONS

We define the function $n_a(w)$ of a w to be the number of letter *a* in the string *w*.

For example, if a word w = aabbac in L, then $n_a(w)=3$.

Concatenation of two strings means that two strings are written down side by side.

For example, x^n concatenated with x^m is x^{n+m}

LANGUAGES
SOME DEFINITIONS

Language is called PALINDROME over the alphabet if

Language = { Λ and all strings x such that reserve(x)=x }.

For example, let ∑={ a, b }, and

PALINDROME={ Λ a b aa bb aaa aba bab bbb …}.

Remark: Sometimes, we obtain another word in PALINDROME when we concatenate two words in PALINDROME. We shall see the interesting properties of this language later.

LANGUAGES
SOME DEFINITIONS

Consider the language

PALINDROME={ Λ a b aa bb aaa aba bab bbb …}.

We usually put words in size order and then listed all the words of the same length alphabetically. This order is called lexicographic order.

LANGUAGES

KLEENE CLOSURE

Given an alphabet Σ, the language that any string of characters in Σ are in this language is called the closure of the alphabet. It is denoted by

$$\Sigma^*.$$

This notation is sometimes known as the Kleene star.

Kleene star can be considered as an operation that makes an infinite language. When we say "infinite language", we mean infinitely many words, each of finite length.

LANGUAGES

KLEENE CLOSURE

More general,

if S is a set of words, then by S* we mean the set of all finite strings formed by concatenating words from S and from S*.

Example:

If S = { a ab }then

S* = { Λ and any word composed of factors of a and ab }.

{ Λ and all strings of a and b except strings with double b }.

{ Λ a aa ab aaa aab aba aaaa aaab aaba … }.

LANGUAGES
KLEENE CLOSURE

Example:

If S = { a ab }then

S* = { Λ and any word composed of factors of a and ab }.

{ Λ and all strings of a and b except strings with double b }.

{ Λ a aa ab aaa aab aba aaaa aaab aaba … }.

To prove that a certain word is in the closure language S* , we must show how it can be written as a concatenation of words from the base set S.

Example: abaaba can be factored as (ab)(a)(ab)(a) and it is unique.

LANGUAGES
KLEENE CLOSURE

Example:

If S = { xx xxxxx }then

S* ={ Λ xx xxxx xxxxx xxxxxx xxxxxxx xxxxxxxx … }.

{ Λ and xx and x^n for n = 4 5 6 7 … }.

How can we prove this statement ?

Hence: proof by constructive algorithm (showing how to create it).

LANGUAGES
KLEENE CLOSURE

Example:

If $S = \{ a\ b\ ab \}$ and $T = \{ a\ b\ ba \}$, then $S^* = T^* = \{ a\ b \}^*$.

Proof: It is clear that $\{ a\ b \}^* \subset S^*$ and $\{ a\ b \}^* \subset T^*$.

We have to show that S^* and $T^* \subset \{ a\ b \}^*$.

For $x \in S^*$, in the case that x is composed of ab.

Replace ab in x by a, b which are in $\{ a\ b \}^*$.

Then $S^* \subset \{ a\ b \}^*$.

The proof of $T^* \subset \{ a\ b \}^*$ is similarity. QED

LANGUAGES
POSITIVE CLOSURE

Given an alphabet Σ, the language that any string (not zero) of characters in Σ are in this language is called the positive closure of the alphabet. It is denoted by

$$\Sigma^+.$$

Example: Let $\Gamma = \{ ab \}$.

Then $\Gamma^+ = \{ ab\ \ abab\ \ ababab\ \ \ldots \}$.

LANGUAGES

TRIVIAL REMARK

Given an alphabet Σ={ aa bbb }. Then Σ^* is the set of all strings where a's occur in even clumps and b's in groups of 3, 6, 9…. Some words in Σ^* are

bbb   aabbbaaaa bbbaa

If we concatenate these three elements of Σ^*, we get one big word in Σ^{**}, which is again in Σ^*.

bbbaabbbaaaabbbaa = (bbb)(aa)(bbb)(aa)(aa)(bbb)(aa)

Note : Σ^{**} means $(\Sigma^*)^*$.

LANGUAGES

THEOREM

Theorem

For any set S of strings, we have $S^* = S^{**}$.

Proof: Every words in S^{**} is made up of factors from S^*.

Every words in S^* is made up of factors from S.

Therefore every words in S^{**} is made up of factors from S.

We can write this $S^{**} \subset S^*$.

In general, it is true that $S \subset S^*$. So $S^* \subset S^{**}$.

Then $S^* = S^{**}$. QED

2

RECURSIVELY
DEFINING
LANGUAGES

RECURSIVE DEFINITIONS
LANGUAGE DEFINING

EVEN language

EVEN is the set of all positive whole numbers divisible by 2.

EVEN is the set of all 2n where n = 1 2 3 4 …

Another way we might try this:

The set is defined by these three rules:

Rule1: 2 is in EVEN.

Rule2: if x is in EVEN, then so is x+2.

Rule3: The only elements in the set EVEN are those that can be produced from the two rules above.

The last rule above is completely redundant.

RECURSIVE DEFINITIONS
LANGUAGE DEFINING

EVEN language

The set is defined by these three rules:
Rule1: 2 is in EVEN.
Rule2: if x is in EVEN, then so is x+2.
Rule3: The only elements in the set EVEN are those that can be produced from the two rules above.

PROBLEM: Show that 10 is in this language.

By Rule1, 2 is in EVEN.
By Rule2, 2+2=4 is in EVEN.
By Rule2, 4+2=6 is in EVEN.
By Rule2, 6+2=8 is in EVEN.
By Rule2, 8+2=10 is in EVEN.

PRETTY HORRIBLE !

RECURSIVE DEFINITIONS
LANGUAGE DEFINING

EVEN language

The set is defined by these three rules:
Rule1: 2 is in EVEN.
Rule2: if x,y are in EVEN, then so is x+y.
Rule3: The only elements in the set EVEN are those that can be produced from the two rules above.

PROBLEM: Show that 10 is in this language.

By Rule1, 2 is in EVEN.
By Rule2, 2+2=4 is in EVEN.
By Rule2, 4+4=8 is in EVEN.
By Rule2, 8+2=10 is in EVEN.

DECIDEDLY HARD

RECURSIVE DEFINITIONS
LANGUAGE DEFINING

POSITIVE language

The set is defined by these three rules:
Rule1: 1 is in POSITIVE.
Rule2: if x,y are in POSITIVE, then so is x+y, x-y, x×y and x/y where y is not zero.
Rule3: The only elements in the set POSITIVE are those that can be produced from the two rules above.

PROBLEM: What is POSITIVE language ?

RECURSIVE DEFINITIONS
LANGUAGE DEFINING

POLYNOMIAL language

The set is defined by these four rules:
Rule1: Any number is in POLYNOMIAL
Rule2: Any variable x is in POLYNOMIAL.
Rule3: if x,y are in POLYNOMIAL, then so is x+y, x-y, x×y and (x).
Rule4: The only elements in the set POLYNOMIAL are those that can be produced from the three rules above.

PROBLEM: Show that $3x^2+2x-5$ is in POLYNOMIAL.
Proof:
Rule1: 2, 3, 5 are in POLYNOMIAL, Rule2: x is in POLYNOMIAL,
Rule3: 3x, 2x are in POLYNOMIAL, Rule3: 3xx is in POLYNOMIAL,
Rule3: 3xxx+2x, $3x^2+2x-5$ are in POLYNOMIAL. QED.

RECURSIVE DEFINITIONS
ARITHMETIC EXPRESSIONS

Language:

Let Σ be an alphabet for AE language.

$\Sigma = \{ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 + - * / ( ) \}$.

Define rules for this language.

Problems:

- Show that the language does not contain substring //.
- Show that ((3+4)-(2*6))/5 is in this language.

REMARK
DEFINING LANGUAGES

Languages can be defined by

- $L_1=\{ x^n \text{ for } n = 1\ 2\ 3\ \dots \}$
- $L_2=\{ x^n \text{ for } n = 1\ 3\ 5\ 7\ \dots \}$
- $L_3=\{ x^n \text{ for } n = 1\ 4\ 9\ 16\ \dots \}$
- $L_4=\{ x^n \text{ for } n = 3\ 4\ 8\ 22\ \dots \}$.

More precision and less guesswork are required.