

강의 개요

Historical Review

혹은 강의 개요

1900년대 초반에 논리식 또는 수식으로 표현할 수 없는 명제(또는 집합; 또는 문제)가 존재하는가에 관한 논의가 활발하였다. 이 문제는 Russel이 논리식으로 표현할 수 없는 집합을 기술하면서(Russel's paradox) 문제가 확실하여 졌다.

수학으로 표현 가능한 문제와 가능하지 않은 문제의 구분은 computer가 할 수 있는 일과 할 수 없는 일의 구분에 관한 논의와 같은 것으로 후에 받아들여져서 computer 또는 program이 수학(논리학)이 같다는 주장의 근거가¹⁾ 되기도 하고 거꾸로 computer의 정의(Turing Machine, Chap. 8)로 받아들여지기도 한다. 교과서에서는 컴퓨터 혹은 수학이 할 수 있는 일과 없는 일을, recursively enumerable(RE; computable, programmable)과 non-recursively enumerable(non-RE; Noncomputable, nonprogrammable)이라는 용어로 구분한다. 이에 관한 자세한 논의는 강의 후반부에서 다루어진다.

한편 1940년대와 50년대에 finite automata라는 간단한 기계가 다루어진다. 이는 regular language, regular expression과 같은(equivalent) 것으로 강의 전반부에서 다루어진다²⁾. 특히 이는 한글모아쓰기 오토마타와 같아서 속제를 통하여 실제로 프로그램해 볼 것이다.

1950년대 말 Noam Chomsky에 의하여 grammar(context-free grammar)에 관한 논의가 시작되고 이는 finite automata에 stack이라는 저장소(memory)가 추가된 pushdown automata과 같고 이것은 context-free language를 정의한다.

이 부분은 강의 중반부에서 다루어지며, 특히 context-free grammar의 **deterministic parsing**은 프로그래밍 언어 컴파일러의 파싱에서 이용되므로 실용적으로 매우 중요³⁾하다. 속제를 통하여 이를 프로그램 해 볼 것이다.

Recursively enumerable 문제 해결이 컴퓨터에서 끝나는 경우(recursive 혹은 terminate; decidable)를 포함하지만 끝나지 않는(non-terminate 혹은 undecidable) 경우도 포함한다. 이 중 끝나는(recursive) 문제가 우리의 관심(algorithm)이 많으며, 컴퓨터가 빠르게 풀 수 있는(tractable) 문제와 빠르게 풀 수 없는(intractable) 문제로 1969년 Cook에 의하여 구분되었으며, 빠르게 풀 수 없는 문제 중 **가장 어려운 문제를 NP-complete**라는 용어로 정의하며 강의의 제일 마지막 부분에서 다루어진다.

이 모든 문제들, non-recursively enumerable, recursively enumerable, recursive, context-free, regular language의 다섯 개의 언어 class는 상의 언어 class가 하위 언어

-
- 1) 논리학 혹은 수학의 한계가 어디까지인가 하는 논쟁은 끝이 없으나, 이 강의에서는 수학 혹은 논리, 프로그램이 할 수 있는 것을 수학의 한계로 보는 Russel이후의 주장에 동의한다.
 - 2) Finite Automata는 deterministic한 경우에서 시작하여 nondeterminism을 허용하는 방법으로 발전할 것이다. 특히 type 3는 regular expression이라는 특별한 언어표현 방식이 있다.
 - 3) Top-down parsing 방법인 LL파싱과 Bottom-up 방법인 LR로 분류되며, LR의 변형인 LALR이 많이 사용된다.

강의 개요

클래스를 포함하며 상위부터 차례로 RE은 type 0로 recursive는 type 1으로, context-free는 type 2로 regular는 type 3로 Chomsky에 의하여 불리어 졌으며 이를 Chomsky's languages hierarchy라고 부른다.

이러한 문제들을 다루기 위하여 수학기초가 강의의 제일 앞부분에서 다루어 질 것이다. 여기서는 집합, 관계(relation), 그래프(graph), 함수(function), 집합의 크기(cardinality of set)와 집합의 상등성(set isomorphism), 유한집합과 무한집합, 셀 수 있는 무한(countably infinite)과 셀 수 없는 무한(uncountably infinite)과 Cantor's diagonal argument 또는 Russel's paradox가 다루어 질 것이다.

Chomsky's Grammars, Languages, and Machine(Automata) Hierarchy

Grammars	Languages	Automata(Machine)
non- type 0	No grammar	Non Recursively Enumerable
type 0 (unrestricted) grammar	Uncomputable, Non-programmable	No Automata
type 1 context-sensitive	Recursively Enumerable	Turing Machine
type 1.H NP-complete	Computable, Programmable	(fa + memory(tape))
type 1.L Polynomial	Recursive, decidable(algorithm)	?
type 2 context-free	intractable	
type 3 regular	tractable	
	context-free	Pushdown Automata (fa + stack)
	regular, regular expression	Finite Automata

강의 순서

- A. Mathematical Previews(Chap. 1 and hand out)
- B. Finite Automata(Chap. 2), Regular Expressions(Chap. 3), and Regular Languages(Chap. 4)
- C. Context-free grammars(Chap. 5), Pushdown Automata(Chap. 6), and Context-free Languages(Chap. 7)
- D. Turing Machine(Chap. 8) and Undecidability(Chap. 9)
- E. Intractable Problems(Chap. 10)