가장 간단한 컴퓨터로 만드는 수열 *

권도용 (포항공과대학교)

2009년 1월 8-9일

컴퓨터란 계산을 하는 기계이다. 현재 컴퓨터를 만드는 회사 중 하나인 애플컴퓨터의 상표는 누군가가 한입 베어 먹은 사과인데 이것은 1954년 6월 7일 영국의 수학자 튜링(Alan Mathison Turing)이 베어 먹은 것이다.

> 임의의 수학 명제에 대하여 이것이 참인지 거짓인지를 판별하는 기계(알고리듬, algorithm)를 만들 수 있을까?[1]

이 질문은 20세기 초, 수학자 힐버트(David Hilbert)가 그 시대 수학자 모두에게 던진 질문이었다. 사과를 베어 먹기 약 20년 전, 튜링은 '계산하는 기계'를 수학적으로 정의하고 힐버트의 질문에 부정적인 답을 증명했다. 튜링이 정의한 이 기계를 그의 이름을 따서 *튜링기계*라 부른다. 오늘날의 모든 디지탈 컴퓨터는 조금 복잡한 튜링기계에 불과하다.

본 강의에서 우리가 다룰 계산 기계는 오토마타(automata)라는 것으로서 튜링기계의 중앙처리장치(CPU)에 해당된다. 튜링기계보다는 계산 능력이 떨어지는데, 이는 사람이 연습장 없이 계산할 때 그 능력이 더 떨어지는 것과 같은 원리이다.

우리는 오토마타로 생성되는 문자열의 성질을 살펴보겠다. 여기서 수열이란 문자열을 이루는 문자가 숫자인 특별한 경우의 문자열이다. 먼저, 오토마타를 간단히 정의한 후 이것의 계산 능력을 예제를 통하여 알아본다. 또, 오토마타로부터 문자열을 만들어내는 방법을 소개하고 이러한 문자열의 성질을 공부한다. 마지막으로는 오토마타로 만들어진 수열이 현재의 첨단수학과는 어떤 관련이 있는지 소개한다.

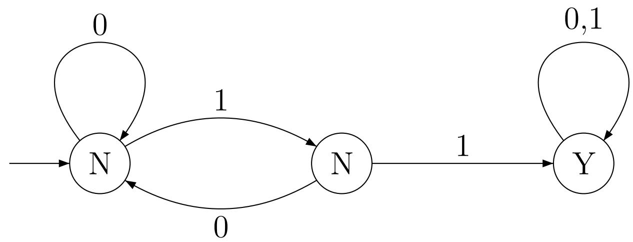
1 오토마타의 정의

앞으로 대문자 I, O는 유한집합을 나타낸다. 또 그 집합의 원소를 *문자*라 하고, 유한개의 문자를 연달아 늘어 놓은 것을 *단어*라 하겠다. 이제 오토

*본 강의록은 http://math.postech.ac.kr/~doyong 에서 구할 수 있습니다.

[1] 그 증명은 모르더라도.

마타를 정의하자. 두 문자집합 I, O 가 있을 때, *오토마타*라는 것은 '*꼭지점*'과 '*방향을 가지는 변*'으로 이루어진 도형이다. 각 변과 꼭지점에는 이름을 붙여 주는데, 변의 이름은 I에 속한 문자로, 꼭지점의 이름은 O에 속한 문자로 짓는다. 여기서 다른 변이 같은 이름을 같는 경우를 허락한다. 꼭지점의 경우도 마찬가지이다. 꼭지점 중 하나를 특별히 표시하고 이를 *시작꼭지점*이라고 부른다. 오토마타에 있어서 변의 길이는 중요하지 않다. 아래 그림은 문자집합이 $I = \{0, 1\}$, $O = \{Y, N\}$인 오토마타의 예이다.



위의 그림에서 제일 왼쪽의 화살표는 그 꼭지점이 시작꼭지점임을 나타내고, 이름이 "$0, 1$"인 변은 이름이 '0'인 변과 '1'인 변을 편의상 하나로 그린 것을 나타낸다. 각 꼭지점에서 나가는 변을 살펴보면 I 속하는 각 문자를 이름으로 갖는 변이 정확히 하나씩만 존재한다. 이와 같이 각 꼭지점에서 나가는 변의 이름에 I의 문자가 모두 나타나고, 또 정확히 하나씩만 나타나는 오토마타를 *결정적 오토마타*라고 부른다. 앞으로 우리가 다룰 오토마타는 모두 결정적 오토마타이다.

I의 문자로 이루어진 단어가 하나 주어졌을 때, 위의 오토마타는 다음의 규칙에 의해 꼭지점 하나를 대응시킨다. 출발은 시작꼭지점으로부터 하고 단어를 한 문자씩 왼쪽에서 오른쪽으로 읽으며 그에 해당되는 변을 따라 다음 꼭지점으로 이동한다. 단어를 모두 읽고나서 도착하는 꼭지점이 그 단어에 대응되는 꼭지점이다.[2] 가령, 단어 '0110'에 대응되는 꼭지점은 이름이 'Y'인 가장 오른쪽 꼭지점이다. 위의 오토마타는 다음의 질문에 항상 올바른 답을 준다.

> **질문**: I의 문자로 이뤄진 단어가 있을 때, 그것은 '11'을 포함하고 있는가?

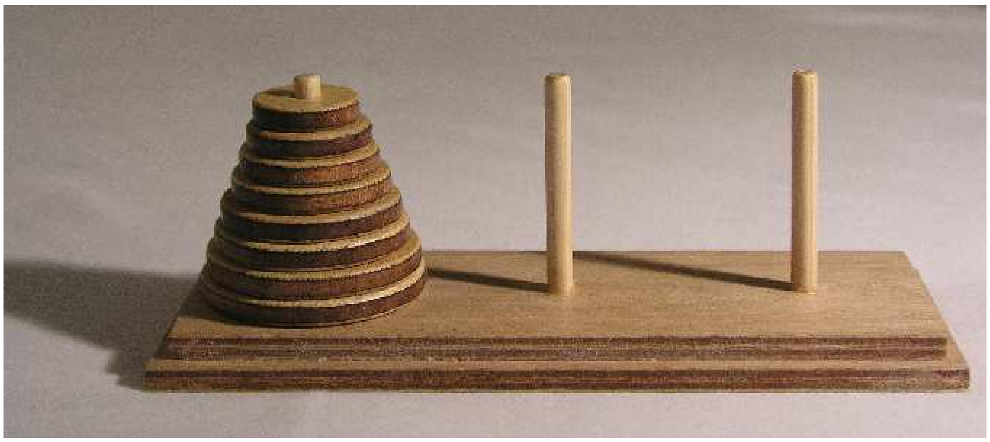
그 단어에 대응되는 꼭지점의 이름이 'Y'이면 질문의 답은 '예'이고, 'N'이면 '아니오'이다.

[2]즉, 이 오토마타는 I의 문자로 이뤄진 단어의 집합에서 꼭지점의 집합으로 가는 함수이다. 여기서, 문자집합 I와 O가 각각 입력(Input)과 출력(Output)의 이니셜이라는 것을 눈치 챈 학생이 있을 수도 있다.

우리는 오토마타를 계산을 하는 기계(의 설계도)라 볼 수 있다. 그 초라한 모습에 이것이 무슨 계산을 할 수 있을까 싶지만, 다음 장에서 소개되는 예를 보면 반드시 그렇지만은 않다라고 느낄 수 있다.

2 오토마타의 예

2.1 하노이 탑 문제



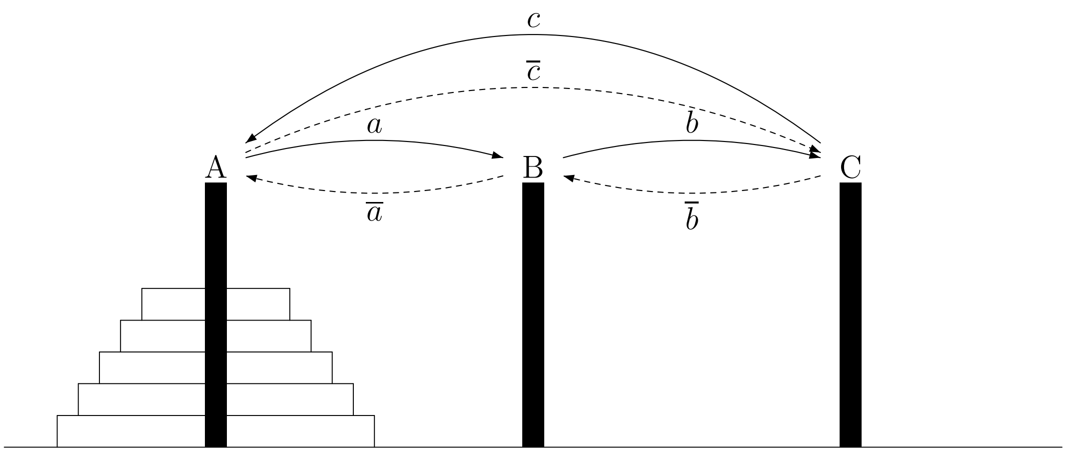
위의 그림에서 세 개의 기둥이 있고, 그 중 하나의 기둥에 다른 크기의 원판 n개가 작은 것이 위에 놓이도록 쌓여있다. 하노이 탑 문제는 다음의 규칙에 의해 움직여 원판을 모두 다른 하나의 기둥으로 옮기는 문제이다.

1. 한 번에 하나의 원판만 옮긴다.
2. 큰 원판이 작은 원판 위에 있을 수 없다.

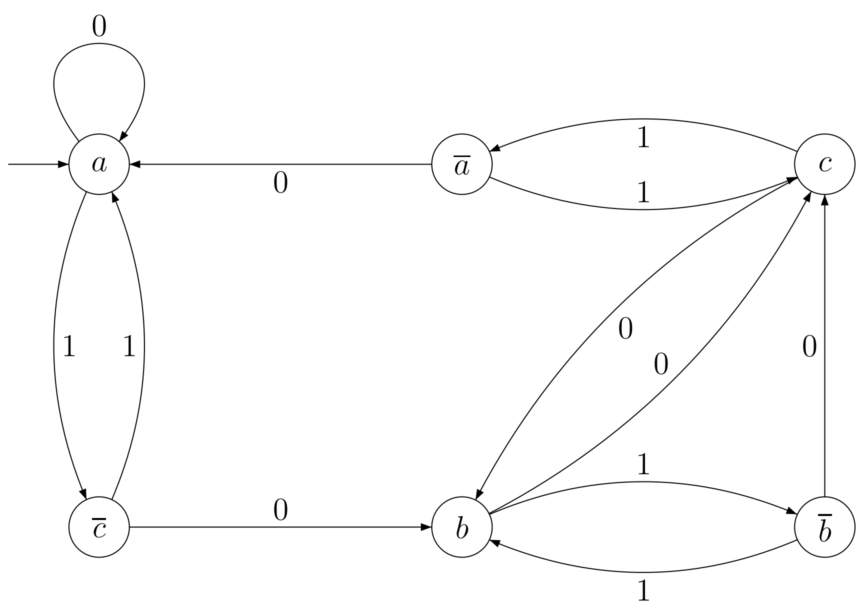
n개의 원판을 모두 다른 기둥으로 옮기는데 필요한 최소한의 움직임 횟수를 T_n이라 하면, $T_1 = 1$임은 당연하다. 한편, $n+1$개의 원판을 모두 다른 기둥으로 옮기는데 필요한 횟수 T_{n+1}은 일단 n개의 원판을 다른 한 기둥으로 옮긴 후 ($= T_n$ 회), 가장 큰 원판을 나머지 한기둥으로 옮기고 ($= 1$ 회), 마지막으로 n개의 원판을 가장 큰 원판 위로 쌓으면 ($= T_n$ 회) 되므로

$$T_{n+1} = 2T_n + 1, \quad n \geq 1$$

로 주어진다. 이 관계식을 이용하면 $T_n = 2^n - 1$임을 쉽게 알 수 있다.



이제 n개의 원판이 기둥 A에 꽂혀 있다고 가정하자. 원판을 A에서 B로 옮기는 것을 a, B에서 C로 옮기는 것을 b, C에서 A로 옮기는 것을 c라 적고, 그 반대 방향으로 옮기는 것을 각각 $\overline{a}, \overline{b}, \overline{c}$라 적는다. 그러면 문자집합 $I = \{0, 1\}$ 와 $O = \{a, b, c, \overline{a}, \overline{b}, \overline{c}\}$에 대한 다음의 오토마타는 하노이 탑 문제의 답을 우리에게 알려준다.



정수 $m \geq 0$에 대하여 m의 2진법 전개를 $(m)_2$라고 쓰면, 우리는

$$(0)_2 = 0,\ (1)_2 = 1,\ (2)_2 = 10,\ (3)_2 = 11,\ (4)_2 = 100,\ (5)_2 = 101, \ldots$$

와 같이 구할 수 있다. 이제 하노이 탑 문제를 풀기위해 $m+1$번째에 취해야할 동작은 단어 $(m)_2$에 대응되는 꼭지점의 이름이다. 처음 몇 개를 정리하면 다음 표와 같다.

$(0)_2 = 0$	$\mapsto$	a	$(5)_2 = 101$	$\mapsto$	$\bar{b}$	$(10)_2 = 1010$	$\mapsto$	c
$(1)_2 = 1$	$\mapsto$	$\bar{c}$	$(6)_2 = 110$	$\mapsto$	a	$(11)_2 = 1011$	$\mapsto$	b
$(2)_2 = 10$	$\mapsto$	b	$(7)_2 = 111$	$\mapsto$	$\bar{c}$	$(12)_2 = 1100$	$\mapsto$	a
$(3)_2 = 11$	$\mapsto$	a	$(8)_2 = 1000$	$\mapsto$	b	$(13)_2 = 1101$	$\mapsto$	$\bar{c}$
$(4)_2 = 100$	$\mapsto$	c	$(9)_2 = 1001$	$\mapsto$	$\bar{a}$	$(14)_2 = 1110$	$\mapsto$	b

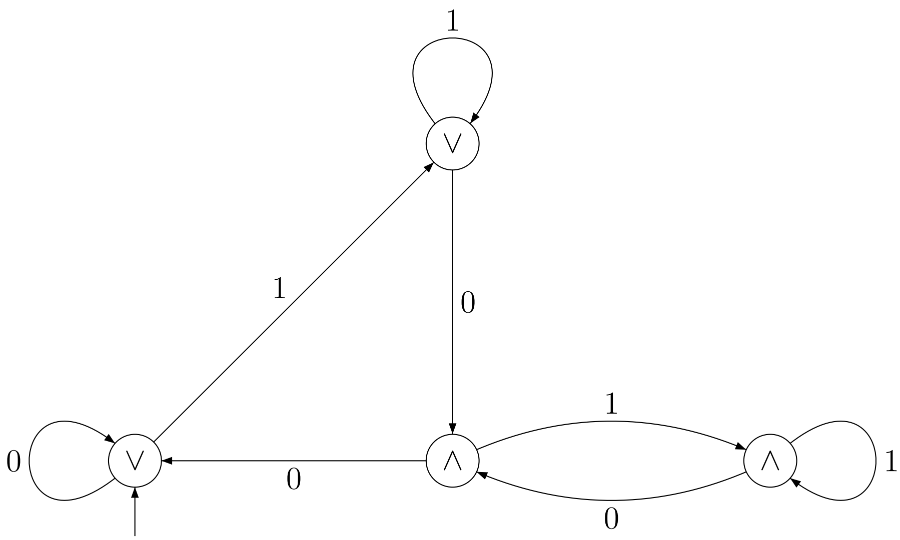
연습문제. 원판이 네 개인 하노이 탑 문제를 위의 표를 이용해 풀어보시오.

2.2 종이접기

반듯한 종이를 책상 위에 올려놓자. 그리고 오른쪽 반이 왼쪽 반 위에 올라가도록 접은 후 다시 펴보자. 그러면 가운데 골짜기가 하나 생길 것이다. 이번에는 처음의 종이를 접은 후 다시 한 번 같은 방법으로 접은 후 펴보자. 그러면 왼쪽부터 차례대로 골짜기, 골짜기, 산이 생길 것이다. 골짜기를 '$\vee$'로 산을 '$\wedge$'로 표시하면 다음의 표와 같이 나온다.

접은 횟수	모양
1	$\vee$
2	$\vee\ \vee\ \wedge$
3	$\vee\ \vee\ \wedge\ \vee\ \vee\ \wedge\ \wedge$
4	$\vee\ \vee\ \wedge\ \vee\ \vee\ \wedge\ \wedge\ \vee\ \vee\ \vee\ \wedge\ \wedge\ \vee\ \wedge\ \wedge$
$\vdots$	$\vdots$

위의 표로부터 n번 접었을 때 나오는 골짜기와 산의 개수의 합은 $2^n - 1$임을 알 수 있다. 또 그 모양은 $n+1$번 접었을 때 모양의 처음 $2^n - 1$개와 같음을 관찰 할 수 있다. 이제 다음 오토마타의 단어 $(m)_2$에 대응되는 꼭지점의 이름은 $m+1$번째가 골짜기인지 산인지를 알려준다.



3 오토마타로 만드는 문자열

문자집합 I, O를 가지는 오토마타가 하나 주어졌다고 가정하자. 자연수 $b \geq 2$에 대하여, 다음의 조건을 만족하는 오토마타를 <u>*b진 오토마타*</u>라고 한다.

1. 결정적 오토마타이다.

2. 문자집합 I는 $I = \{0, 1, \ldots, b-1\}$로 주어진다.

b진 오토마타의 '진'은 2진법, 10진법에서의 '진'으로부터 왔다. 왜 그런지는 아래에서 나온다.

이제 주어진 b진 오토마타로 부터 문자열 $a_0a_1a_2\cdots$를 다음과 같이 정의한다.

> a_n은 n의 b진법 전개 $(n)_b$에 대응되는 b진 오토마타의 꼭지점의 이름.

따라서 모든 a_n는 문자집합 O에 속하는 문자이다.

연습문제. 단어 $a\bar{c}bac\bar{b}a\bar{c}b\bar{a}cba\bar{c}b$는 어떤 2진 오토마타로 부터 만들어진 문자열의 0항부터 14항까지임을 보이시오.

4 오토마타로 만들어진 문자열의 성질

이번 장에서는 문자집합 S의 문자로 이뤄진 문자열을 정의하는 또 다른 방법을 생각해 본다.

문자집합 S에 대하여 *길이 b인 치환*이란, S의 각 문자를 길이 b인 단어로 바꿔주는 규칙이다. 길이 b인 단어란 b개의 문자로 이뤄진 단어란 뜻이고, 여기에 쓰인 문자도 또한 모두 S에 속해있다. 다음의 예를 보면 의미가 분명해진다.

예 1. 문자집합 S가 $S = \{0, 1\}$이라 하고, 이에 대한 길이 2인 치환을

$$0 \mapsto 01, \quad 1 \mapsto 10$$

이라 정의하자. 그렇다면 가령 단어 '1101'은 이 치환에 의해

$$1101 \mapsto 10100110$$

으로 변형된다. 이제 이 치환을 0에 적용하고, 그 결과에 다시 치환을 적용하고, 그 결과에 다시 치환을 적용하는 과정을 반복하면

$$\begin{aligned} 0 &\mapsto 01 \\ 01 &\mapsto 0110 \\ 0110 &\mapsto 01101001 \\ 01101001 &\mapsto 0110100110010110 \\ 0110100110010110 &\mapsto 01101001100101101001011001101001 \\ &\vdots \end{aligned}$$

를 얻는다. 이 과정을 (무한히) 반복하면 우리는 0과 1로 이루어진 문자열을 하나 얻게 된다.

일반적으로 길이 b인 치환이 어떤 문자 'a'를 'a'로 시작하는 단어로 바꿀 때, a에 이 치환을 (무한) 반복 적용하면 문자열을 하나 얻게 된다. 위의 (예 1)에서의 치환은 문자 '0'을 '0'으로 시작하는 단어 01로 변환한다.

우리는 지금까지 문자열을 정의하는 두 가지 방법을 알아보았다. 하나는 b진 오토마타에 의해서였고, 다른 하나는 길이 b인 치환에 의해서였다. 전혀 관련 없어 보이는 이 두 가지 문자열이 실은 거의 같은 개념임이 1972년 Cobham이라는 수학자에 의해 증명되었다.

정리. *(1) 문자집합이 $I = \{0, 1, \ldots, b-1\}$, O인 b진 오토마타로 만들어진 문자열은 다음과 같이 만들 수 있다: 어떤 문자집합 S에 정의된 길이 b인 치환에 의해 문자열을 만든 후, 문자열의 각 문자를 적당한 함수*

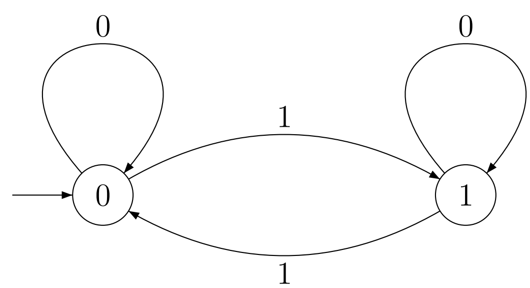
$f : S \to O$의 규칙으로 모두 바꿔 준다.
(2) 또 그 역도 성립한다.

증명의 착안. (1) b진 오토마타의 꼭지점이 n개라면 문자집합 S를 $S = \{0, 1, \ldots, n-1\}$이라 하고 각 꼭지점에 0에서 $n-1$까지의 번호를 붙인다. 그럼 S에 길이 b인 치환을 다음과 같이 정의하자:

> $0 \le i \le n-1$인 i에 대하여 $i \mapsto a_0 a_1 \cdots a_{b-1}$, 여기서 a_k는 번호 i인 꼭지점이 입력 k에 의해 옮겨지는 꼭지점의 번호이다.

함수 $f : S \to O$를 '$f(i)$는 번호 i인 꼭지점의 문자집합 O에 의한 이름'이라 정의하자. 이제 위의 치환을 $0 \in S$에 (무한) 반복 적용하여 문자열을 만들고, 각 문자를 함수 f의 규칙에 의해 바꿔 주면 원하는 문자열을 얻는다.

연습문제. 문자집합 $I = \{0, 1\}$, $O = \{0, 1\}$를 가지는 다음의 2진 오토마타가 만들어 내는 문자열이 (예 1)의 치환이 만들어 내는 문자열과 일치함을 확인해 보시오.



연습문제. 문자집합 $O = \{a, b, c, \overline{a}, \overline{b}, \overline{c}\}$에 대하여 길이 2인 치환을

$$a \mapsto a\overline{c},\ b \mapsto c\overline{b},\ c \mapsto b\overline{a},\ \overline{a} \mapsto ac,\ \overline{b} \mapsto cb,\ \overline{c} \mapsto ba$$

라 정의하자. 문자 'a'에 이 치환을 (무한) 반복 적용하여 얻은 문자열의 처음 15개 항을 적으시오.

연습문제. 문자집합 $S = \{a, b, c, d\}$에 대하여 길이 2인 치환을

$$a \mapsto ad,\ b \mapsto ac,\ c \mapsto bc,\ d \mapsto bd$$

라 하고, 함수 $f : S \to O = \{\vee, \wedge\}$를 $f(a) = f(d) = \vee$, $f(b) = f(c) = \wedge$라 정의하자. 문자 'a'에 이 치환을 (무한) 반복 적용하여 얻은 문자열을 함수 f에 의해 각 문자를 바꿔주면 처음 15개 항은 어떻게 되겠는가?

5 오토마타와 수학

앞에서 언급한 Cobham은 다음과 같은 질문을 수학자들에게 던졌다.

> $\sqrt{2}$의 10진 전개는 10진 오토마타로 만들어낼 수 있을까? 일반적으로, $\sqrt{2}$의 b진 전개는 b진 오토마타로 만들어낼 수 있을까?

이 문제는 오랫동안 미해결 문제로 남아오다 2007년에 와서야 불가능하다는 사실이 증명되었다. 그 결과는 수학에서 가장 좋은 저널 중 하나인 Annals of Mathematics에 출판되었다.

정리. *정수 b를 1보다 크다고 하면, $\sqrt{2}$의 b진 전개는 b진 오토마타로 만들어낼 수 없다.*

참고문헌

- 김홍종, *Turing 기계*, http://www.math.snu.ac.kr/ hongjong/현대수학입문/2001/TuringMachine.pdf
- B. Adamczewski and Y. Bugeaud, On the complexity of algebraic numbers I. expansions in integer bases. *Annals of Mathematics* **165** (2007) 547–565.
- J.-P. Allouche and J. Shallit, *Automatic Sequences : Theory, Applications, Generalizations*, Cambridge University Press, Cambridge, 2003.
- F. Axel and D. Gratias (eds.), *Beyond Quasicrystals*, Springer-Verlag, Berlin, 1995.
- A.M. Turing, On computable numbers, with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society* **42** (1937), 230–265.
- A. Cobham, Uniform tag sequences, *Mathematical Systems Theory* **6** (1972), 164–192.
- Entscheidungsproblem, *Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/wiki/Entscheidungsproblem
- Tower of Hanoi, *Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/wiki/Tower_of_Hanoi