계산이론

2022년 1학기

이은주

4장 정규 언어의 성질
정규 언어의 폐포 성질
비정규 언어의 식별

- 정규 언어에 **연산**을 수행할 경우 그 결과가 어떻게 될까?

  접합과 같은 간단한 집합 연산들,
  언어의 각 문자열을 바꾸는 연산들

  - 예) 연산을 수행한 결과의 언어가 여전히 정규 언어일까?
    - 폐포(closure) 성질을 묻는 질문

- 어떤 성질에 대한 결정을 할 수 있는가?
  - 예) 어떤 언어가 유한(finite)인지 무한(infinite)인지를 결정할 수 있는가?

- 주어진 언어의 정규 언어 여부 판정
  - 정규 언어 : 정규 표현, dfa, 혹은 정규 문법 제시

- 어떤 언어가 정규 언어가 아님을 보이는 한 가지 방법
  - 모든 정규 언어들의 일반적인 성질(모든 정규 언어가 공유하는 특성)들

정규 언어의 폐포 성질

- 주어진 두 정규 언어 L_1과 L_2에 대해, 이들의 합집합 정규 언어인가?
  - 모든 정규 언어 L_1과 L_2에 대하여도 성립하는가?
    - **정규 언어군**(family of regular languages)은 **합집합**에 대해 **폐포**되어(closed) 있음

간단한 집합 연산에 대한 폐포 성질

- [정리 4.1] (115 page) L_1과 L_2가 정규 언어인 경우
  $L_1 \cup L_2, L_1 \cap L_2, L_1L_2, \bar{L}_1,$ 그리고 $L^*{}_1$도 정규 언어
- 정규 언어 : 합집합, 교집합, 접합, 여집합, 스타-폐포 연산에 대해 모두 폐포

증명) L_1과 L_2가 정규 언어이면, $L_1 = L(r_1)$과 $L_2 = L(r_2)$를 만족하는 정규 표현 r_1과 r_2가 존재

- $r_1 + r_2, r_1r_2$와 $r_1{}^*$: 각각 $L_1 \cup L_2, L_1L_2$와 $L^*{}_1$를 표현하는 정규 표현

$\therefore$ 합집합, 접합, 스타-폐포 연산에 대한 폐포 성질 성립

간단한 집합 연산에 대한 폐포 성질

- [정리 4.1] (115 page) – 계속 (여집합에 대하여)

dfa $M = (Q, \Sigma, \delta, q_0, F)$: L_1을 인식하는 dfa

$\bar{L}_1$을 인식하는 dfa $\widehat{M}$: $\widehat{M} = (Q, \Sigma, \delta, q_0, Q - F)$

dfa의 정의에서 δ^*는 전체 함수(total function)

- 모든 $w \in \Sigma^*$에 대해 $\delta^*(q_0, w)$가 정의됨
  - $w \in L$인 경우 $\delta^*(q_0, w)$는 항상 승인 상태
  - $w \in \bar{L}$인 경우 $\delta^*(q_0, w) \in Q - F$

∴ 여집합에 대해 폐포 성질 성립

간단한 집합 연산에 대한 폐포 성질

- [정리 4.1] (115 page) – 계속 (교집합에 대하여)

$L_1 = L(M_1), L_2 = L(M_2)$일 때,

- dfa는 $M_1 = (Q, \Sigma, \delta_1, q_0, F_1)$, $M_2 = (P, \Sigma, \delta_2, p_0, F_2)$

M_1과 M_2를 결합한 오토마타 $\widehat{M} = \left(\hat{Q}, \Sigma, \hat{\delta}, (q_0, p_0), \hat{F}\right)$을 구성

- 상태들의 집합 $\hat{Q} = Q \times P : \left(q_i, p_j\right)$
- 전이 함수 $\hat{\delta} : M_1$이 상태 q_i 에 있고 M_2가 상태 p_j에 있을 경우
  - 항상 상태 $\left(q_i, p_j\right)$에 있도록
  - 전이 함수 $\hat{\delta}$의 정의

$\delta_1(q_i, a) = q_k$ 이고 $\delta_2\left(q_j, a\right) = q_l$ 일 경우

$\hat{\delta}\left(\left(q_i, p_j\right), a\right) = (q_k, p_l)$

- $\hat{F} : q_i \in F_1$이고 $p_j \in F_2$인 모든 쌍 $\left(q_i, p_j\right)$들의 집합
  - $w \in L_1 \cap L_2$일 때 w 가 $\widehat{M}$에 의하여 승인됨

    $\therefore$ $L_1 \cap L_2$은 정규 언어

구성적인 증명(constructive proof)

간단한 집합 연산에 대한 폐포 성질

- [정리 4.1] (115 page) – 계속 (교집합에 대하여)
- 비구성적인 증명

  교집합에 대한 폐포 성질에 대하여,

  $L_1 \cap L_2 = \overline{\overline{L_1 \cap L_2}} = \overline{(\overline{L}_1 \cup \overline{L}_2)}$ (DeMorgan의 법칙)

  - L_1과 L_2가 정규 언어라면 여집합에 대한 폐포 성질에 의하여
    - $\overline{L}_1$과 $\overline{L}_2$도 정규 언어

      $\overline{(\overline{L}_1 \cup \overline{L}_2)} = L_1 \cap L_2$ (정규 언어)

간단한 집합 연산에 대한 폐포 성질

- 예제 4.1 (116 page) 정규 언어군이 차집합에 대해 폐포 성질이 성립함
  - L_1과 L_2정규 언어일 경우 : $L_1 - L_2$도 정규 언어

    $L_1 - L_2 = L_1 \cap \bar{L}_2$(차집합의 정의에 따라)

  - L_2가 정규 언어 : $\bar{L}_2$역시 정규 언어

정규 언어는 교집합에 대해 폐포 성질 성립 : $L_1 \cap \bar{L}_2$는 정규 언어

간단한 집합 연산에 대한 폐포 성질

- [정리 4.2] (116 page) 정규 언어군은 역(reversal)에 대해 폐포 성질이 성립한다.
  증명) (2.3절 연습문제 7)
  L이 정규 언어일 때 승인 상태를 하나만 갖는 L 에 대한 nfa를 구성
  - 이 nfa에 대한 전이 그래프를 다음 과정으로 수정
    - 초기 상태를 승인 상태로
    - 승인 상태는 초기 상태로
    - 모든 간선들의 방향을 역으로 바꿈
  - 수정된 nfa가 w^R을 승인 : 원래의 nfa가 w를 승인할 수 있음
  - 수정된 nfa는 L^R을 인식 : 역에 대한 폐포 성질 성립

기타 연산들에 대한 폐포 성질

- [정의 4.1] (117 page) 준동형(homomorphism)
  - Σ와 Γ가 알파벳일 때 다음과 같이 정의된 함수 h

    $$h : \Sigma \to \Gamma^*$$

    단일 심벌을 문자열로 대체하는 치환(substitution)

    만약 $w = a_1 a_2 \cdots a_n$인 경우, $h(w) = h(a_1)h(a_2)\cdots h(a_n)$

    - L이 Σ에 대한 언어이면, L 의 준동형 상(homomorphic image)은

      $$h(L) = \{h(L) : w \in L\}$$

간단한 집합 연산에 대한 폐포 성질

- 예제 4.2 (117 page) Σ = {a, b}, Γ = {a, b, c}이고, h는 다음과 같을 때

$$h(a) = ab$$

$$h(b) = bbc$$

- $h(aba) = abbbcab$ 이며, $L = \{aa, aba\}$의 준동형 상

$$h(L) = \{abab, abbbcab\}$$

- 언어 L에 대한 정규 표현이 r이면
  - h(L)에 대한 정규 표현
    - r을 구성하는 각 Σ 심벌에 준동형 함수를 적용

간단한 집합 연산에 대한 폐포 성질

- 예제 4.3 (118 page) Σ = {a, b}, Γ = {a, b, c}, h는 다음과 같을 때

$$h(a) = abcc$$

$$h(b) = bdc$$

  - 정규 언어 L이 다음의 정규 표현에 의하여 표현된다면,

$$r = (a + b^*)(aa)^*$$

  - 다음의 정규 표현은 정규 언어 h(L)을 표현

$$r_l = (abcc + (bdc)^*)(dbccdbcc)^*$$

기타 연산들에 대한 폐포 성질

- [정리 4.3] (118 page) 함수 h가 준동형일 때
  L이 정규 언어인 경우 : L 의 준동형상인 $L(h)$도 정규 언어
  - 정규 언어군은 임의의 준동형에 대해 폐포

  증명) L을 어떤 정규 표현 r에 의하여 표현되는 정규 언어라 하자
  - $h(r)$: r에 나타나는 Σ의 모든 심벌 a 를 준동형 함수 $h(a)$로 치환하여 얻어진 결과
    - 정규 표현의 정의에 의해서 $h(r)$은 정규 표현
    - 결과의 정규 표현 $h(r)$은 $h(L)$을 나타냄

  (증명 과정에서 보일 내용)
    - 모든 w ∈ L(r)에 대하여, 대응되는 h(w)가 L(h(r))에 속한다는 것
    - 역으로 모든 v ∈ L(h(r))에 대하여, v = h(w)을 만족하는 w가 L에 존재한다는 것
  - $h(L)$은 정규 언어

기타 연산들에 대한 폐포 성질

- [정의 4.2] (119 page)

  L_1과 L_2가 동일한 알파벳에 대한 정규 언어일 때 L_2에 대한 L_1의 우측 몫(right quotient)의 정의

  $$L_1/L_2 = \{x : xy \in L_1 \text{ for some } y \in L_2\}$$

- L_2에 대한 L_1의 우측 몫을 구성하기 위해,
  - L_2에 속하는 문자열을 후위부(suffix)로 갖는 L_1에 속한 모든 문자열을 모아서 후위부를 제거하고 남은 문자열
    - L_1/L_2에 속하게 됨

기타 연산들에 대한 폐포 성질

- [정리 4.4] (121 page) L_1과 L_2가 정규 언어인 경우, L_1/L_2는 역시 정규 언어이다. 즉 정규 언어군은 우측 몫 연산에 대해 폐포 성질이 성립한다.

증명) $L_1 = L(M)$일 때 dfa는 $M = (Q, \Sigma, \delta, q_0, F)$

또다른 dfa $\widehat{M} = \left(Q, \Sigma, \delta, q_0, \widehat{F}\right)$는 $q_i \in Q$에 대하여 다음을 만족하는 문자열 $y \in L_2$가 존재하는지 결정

$$\delta^*(q_i, y) = q_f \in F$$

- 오토마타 M_i : M의 시작 상태 q_0를 $q_i \in Q$로 대체

- L_2에도 속하면서, $L(M_i)$에 속하는 문자열 y가 존재하는지 결정
  - $L_2 \cap L(M_i)$에 대한 전이 그래프 구성 (정리 4.1)
    - 모든 $q_i \in Q$에 대해서 반복 : $\widehat{F}$가 결정되고, $\widehat{M}$이 구성됨

기타 연산들에 대한 폐포 성질

- [정리 4.4] (121 page) – 계속 $L(\widehat{M}) = L_1/L_2$을 보이기 위해, L_1/L_2에 속한 임의의 문자열 x에 대해
  - $xy \in L_1$를 만족하는 문자열 $y \in L_2$가 반드시 존재

$$\delta^*(q_0, xy) \in F$$

  다음을 만족하는 $q \in Q$ 존재

$$\delta^*(q_0, x) \in q,\ \delta^*(q, y) \in F$$

- $\widehat{M}$는 x를 승인 : $q \in \hat{F}$ 이고, $\delta^*(q_0, x) \in \hat{F}$이므로

  역으로 $\widehat{M}$에 의하여 승인되는 모든 문자열 x에 대해

$$\delta^*(q_0, x) = q \in \hat{F}$$

- 위의 과정을 적용 : $\delta^*(q, y) \in F$ 조건을 만족하는 $y \in L_2$ 존재
  - $xy \in L_1$, $x \in L_1/L_2$ 만족 : $L(\widehat{M}) = L_1/L_2$ 성립

  ∴ L_1/L_2정규 언어

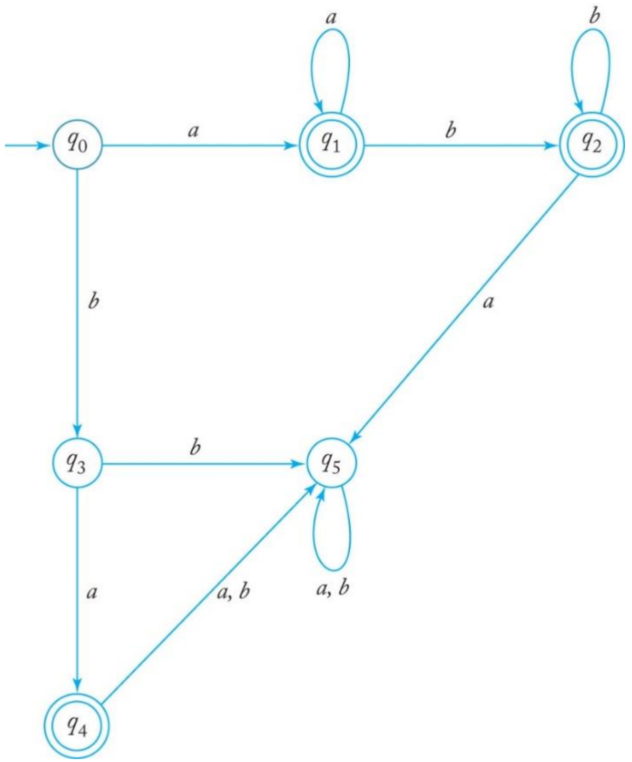
간단한 집합 연산에 대한 폐포 성질

- 예제 4.4 (119 page)

  $L_1 = \{a^n b^m : n \geq 1, m \geq 0\}$ 이고 $L_2 = \{b^m : m \geq 1\}$인 경우

$$L_1 \,/\, L_2 = \{a^n b^m : n \geq 1, m \geq 0\}$$

  - $L_1 = \{a^n b^m : n \geq 1, m \geq 0\}$에 대한 dfa $M_1 = (Q, \Sigma, \delta, q_0, F)$



간단한 집합 연산에 대한 폐포 성질

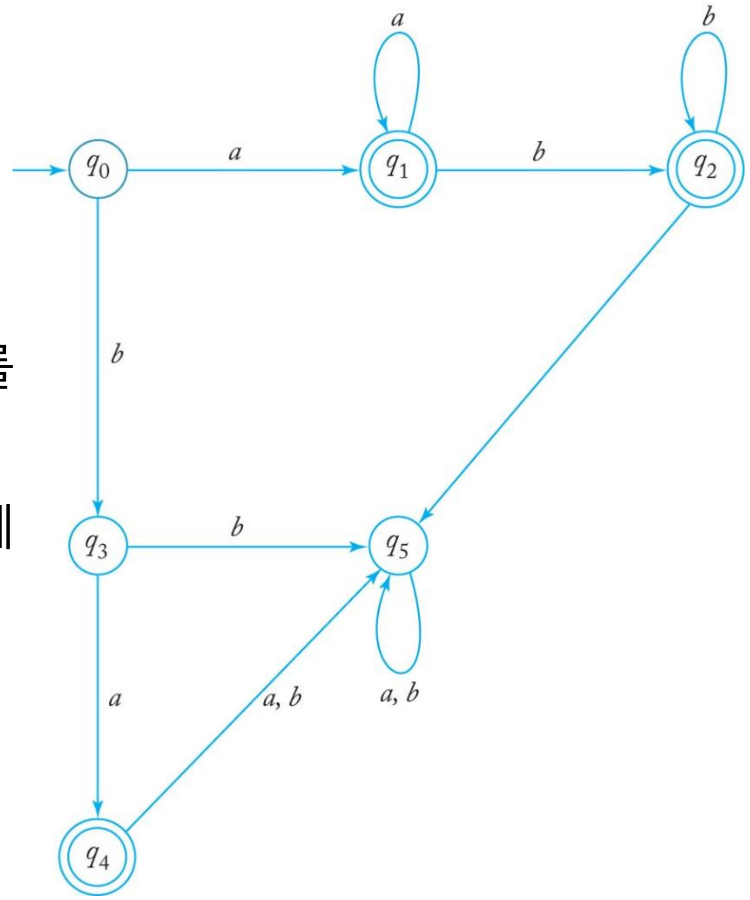
- 예제 4.4 (119 page) - 계속

  $L_1 \,/\, L_2 = \{a^n b^m : n \geq 1, m \geq 0\}$에 대한 dfa

  - L_1에 속한 문자열들의 어떠한 전위부라도 승인해야 함

– $xy \in L_1$, $y \in L_2$를 만족하는 어떤 문자열 y가 존재하는 가를 알아내야 함 (어려운 문제)

  - 각각의 $q \in Q$에 대해 이 상태로부터 한 승인 상태로의 L_2에 속한 v를 라벨로 갖는 보행이 존재하는지를 결정
  - $\delta^*(q_0, x) = q$를 만족하는 모든 문자열 x는 $L_1 \,/\, L_2$에 속하게 됨
  - q 가 승인 상태가 되도록 M_1을 수정



정규 언어와 정규 문법간의 동치성

- [정리 4.6] (127 page) 표준 표현으로 주어진 정규 언어가 공집합(empty set), 유한 집합(finite set), 혹은 무한 집합 (infinite set)인지를 판별할 수 있는 알고리즘이 존재한다.
  - 언어를 인식하는 dfa의 전이 그래프로 표현
  - 초기 정점으로부터 임의의 승인 정점까지의 단순 경로 존재
    - 공집합이 아님
  - 무한 집합 : 사이클의 기지(base)가 되는 모든 정점들 가운데 어느 정점이 초기 정점으로부터 임의의 승인 정점까지의 단순 경로 상에 있는 언어
  - 유한집합 : 무한집합이 아닌 경우

정규 언어와 정규 문법간의 동치성

- [정리 4.7] (128 page)

  두 정규 언어 L_1과 L_2가 표준 표현으로 주어졌을 경우, $L_1 = L_2$인지 아닌지를 판별할 수 있는 알고리즘이 존재한다.

  증명) L_1과 L_2를 이용하여 L_3를 다음과 같이 정의

$$L_3 = (L_1 \cap \ \bar{L}_2) \cup (\bar{L}_1 \cap \ L_2)$$

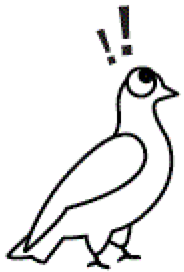
- 정규 언어의 합집합, 교집합, 여집합에 대해 폐포 : L_3는 정규 언어

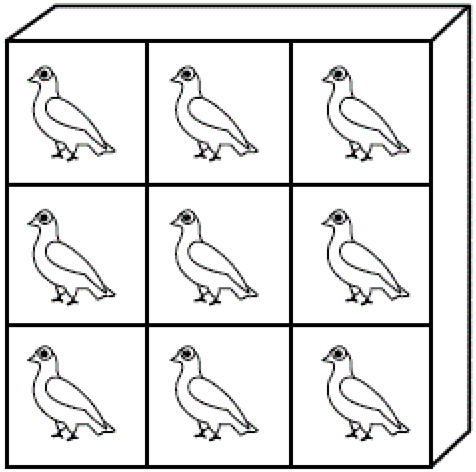
- 비정규 언어 식별
  - 피전홀 원리
  - 펌핑 보조정리

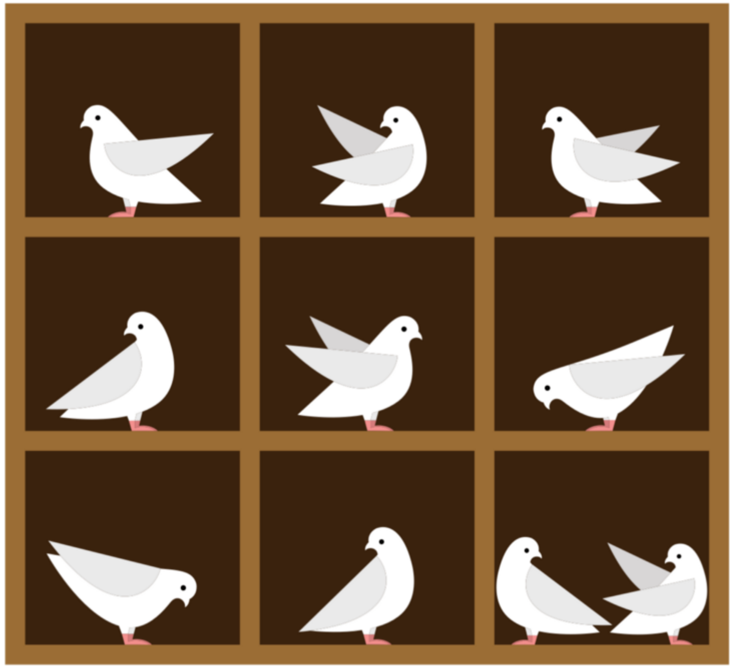
Type	문법(Grammar)	문법(Grammar)	오토마타(Automata)
0	재귀적 열거 가능 언어	무제한 문법	튜링 기계
1	문맥 의존 언어	문맥 의존 문법(CSG)	선형한계 오토마타
2	문맥-자유 언어	문맥자유(무관) 문법(CFG)	푸시다운 오토마타
3	**정규 언어**	**정규 문법** (우선형문법(RLG), 좌선형문법(LLG))	**유한 오토마타**

비정규 언어의 식별 : 피전홀 원리의 이용

- 피전홀(pigeonhole) 원리
  - m개의 상자 (여기서는 상자가 피전홀임)에 n개의 물건을 넣는 경우
    - n>m : 적어도 한 개의 상자에는 두 개 이상의 물건을 넣어야 함







피전홀(pigeonhole) 원리 : 응용

바구니와 사과를 통한 비둘기 집 원리

비둘기 집 원리는 다음과 같은 응용 문제에 적용될 수 있다.

(1) 18개의 터미널을 가지고 있는 컴퓨터 시스템에서 24개의 작업을 수행하고자 할 때 적어도 6개의 작업은 다른 작업이 끝날 때까지 기다려야 한다.

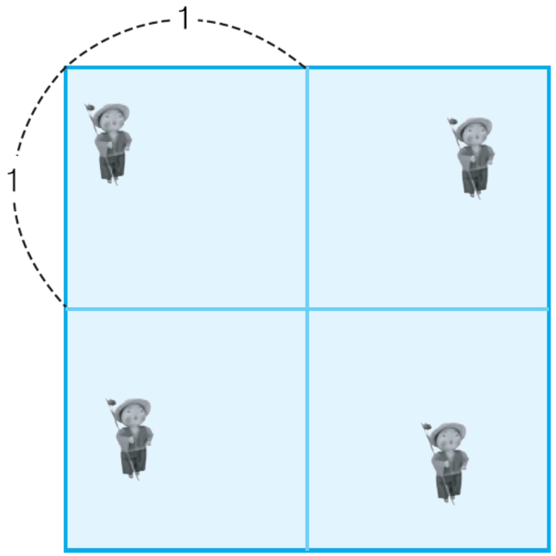
(2) 만약 n개의 공이 m개 상자에 있을 때 n이 m보다 크면, 이 상자들 중의 하나에는 적어도 두 개의 공을 담아야 된다.

피전홀(pigeonhole) 원리 : 예제 1

학예회에서 창작연극을 하기 위해 6명의 학생들이 역할을 나누기로 하였다. 등장하는 인물의 직업은 가수, 재즈댄서, 국악인, 개그맨, 앵커 등이다. 이때 최소한 두 명은 동일한 직업을 가지게 된다는 것을 보여라.

피전홀(pigeonhole) 원리 : 예제 2

가로, 세로 길이가 2m인 방안에 5개의 인형을 놓을 때 임의의 두 인형은 반드시 $\sqrt{2}$m 이내에 있게 됨을 보여라.



피전홀(pigeonhole) 원리

- 일반화된 비둘기집 원리
  - N개의 물체를 k개의 상자에 넣는다면, 적어도 하나의 상자에는 $\lceil {}^{N}/_{k} \rceil$개의 문제가 들어가게 된다.

물체가 $\lceil N/k \rceil$ -1 개 보다 많이 들어가는 상자는 없다고 하자.

$\lceil N/k \rceil < \lceil N/k \rceil + 1$ 에 의하여 물체의 수는 다음의 수식과 같이 N보다 작게 된다.

$$k\left(\left\lceil \frac{N}{k} \right\rceil - 1\right) < k\left(\left(\frac{N}{k} + 1\right) - 1\right) = N,$$

N개의 물체가 있다는 사실에 모순된다.

$\lceil N/k \rceil \geq r$ 일 경우 r개 이상의 물체가 들어가 있는 상자가 존재한다.

피전홀(pigeonhole) 원리 : 예제3

서울에서 머리카락 수가 똑같은 사람은 몇 명일까?

학생 수가 30명인 반이 있는 데 이 반 학생 중 같은 요일에 태어난 사람은 적어도 몇 명일까?

비정규 언어의 식별 : 피전홀 원리의 이용

- 예제 4.6 (130 page) 언어 $L = \{a^n b^n : n \geq 0\}$은 정규 언어가 아님을 증명 (귀류법 사용)
  - L 이 정규 언어일 때 L 을 인식하는 결정적 유한 오토마타
    - dfa $M = (Q, \{a, b\}, \delta, q_0, F)$이 존재
    - 모든 $i = 1, 2, 3, \ldots$에 대해 $\delta^*(q_0, q_i)$일 때
      - i는 무한, M 의 상태들의 개수는 유한
    - 피전홀 원리에 의하여 다음의 조건을 만족하는 어떤 한 상태 q가 존재함

$$\delta^*(q_0, a^n) = q, \delta^*(q_0, a^m) = q, \text{ 단}, n \neq m.$$

비정규 언어의 식별 : 피전홀 원리의 이용

- 예제 4.6 (128 page) - 계속

  가정($L = \{a^n b^n : n \geq 0\}$)에 의해 M은 $a^n b^n$을 승인하므로 다음이 성립

$$\delta^*(q, b^n) = q_f \in \mathrm{F}$$

  - M이 $a^n b^m$을 승인할 수 있음

$$\delta^*(q_0, a^n) = q, \delta^*(q_0, a^m) = q, \ \text{단}, n \neq m.$$

$$\delta^*(q_0, a^m b^n) = \delta^*(\delta^*(q_0, a^m), b^n) = \delta^*(q, b^n) = q_f$$

    - M 이 단지 $n = m$인 경우에만 $a^m b^n$을 승인할 수 있다는 가정에 모순 : L은 정규 언어가 될 수 없음

비정규 언어의 식별 : 피전홀 원리의 이용

- 정규 언어에 대한 전이 그래프
  - 전이 그래프가 사이클을 가지지 않으면 해당 언어는 유한 : 정규 언어
  - 전이 그래프가 비어 있지 않은 라벨(nonempty label)이 포함된 사이클을 가지면 해당 언어는 무한
    - 역으로, 모든 무한 정규 언어는 dfa에 그러한 사이클을 가짐
  - 사이클이 있을 경우 이 사이클은 생략될 수도 있고 임의의 횟수만큼 반복될 수도 있음
    - 사이클이 라벨 v를 가지면서 문자열 w_1vw_2가 해당 언어에 속할 경우 w_1w_2, w_1vvw_2, w_1vvvw_2 등도 해당 언어에 속함
  - 주어진 dfa의 어디에 이러한 사이클이 존재하는지는 알 수 없음
    - 다만, 그 dfa가 m개의 상태를 가진다면 m개의 심벌이 읽혀지는 시점에 사이클이 시작됨

비정규 언어의 식별 : 펌핑 보조정리

- [정리 4.8] (131 page) L이 무한 정규 언어(infinite regular language)일 때 다음 성질을 만족하는 양의 정수 m 존재
  - $|w| \geq m$을 만족하는 모든 문자열 $w \in L$는 아래의 조건을 만족하도록 분할 될 수 있다:

$$w = xyz$$

$|xy| \leq m, |y| \geq 1,$ 모든 정수 $i = 1, 2, 3, \ldots$에 대해서

$$w_i = xy^iz \in \mathrm{L}$$

- L에 속한 충분히 긴 모든 문자열은 다음의 성질을 만족하도록 세 부분으로 분리할 수 있다:
  - 중간 부분을 임의의 횟수 반복하더라도 결과의 문자열은 L에 속한 또 다른 문자열
  - 중간 부분이 "펌프"된다고 말하며, 이 결과가 **펌핑 보조정리** 이다.

비정규 언어의 식별 : 펌핑 보조정리

- 무한 언어(infinite language)에 대한 펌핑 보조정리
  - 유한 언어는, 항상 정규 언어이고 펌프될 수 없음
    - 펌프를 하게 되면 자동적으로 무한 집합이 생성되기 때문
  - 이 정리는 유한 언어에 대해서도 자동적으로 성립
    - 펌핑 보조정리의 m을, 어느 문자열도 펌프되지 않도록, 가장 긴 문자열의 길이보다 크게 잡으면 됨
  - 펌핑 보조정리는 어떤 언어가 정규 언어가 아님을 증명하기 위해서 사용
    - 증명 방법은 항상 귀류법 사용

비정규 언어의 식별 : 펌핑 보조정리

- 예제 4.7 (133 page) 펌핑 보조정리를 이용하여

  $L = \{a^n b^n : n \geq 0\}$이 정규 언어가 아님을 보이자.

  - L 정규 언어라 가정 : 펌핑 보조정리 성립
  - m의 값은 $n = m$으로 선택할 수 있음
    - 부문자열 y : 전부 a들로 이루어져야 함
  - $|y| = k$라 가정
  - 식 $w_i = xy^i z$에서 $i = 0$으로 하여 얻어진 문자열

$$w_0 = a^{m-k} b^m (\notin L)$$

    - 펌핑 보조정리에 모순 : L 이 정규 언어 라는 가정 거짓

비정규 언어의 식별 : 펌핑 보조정리

- 펌핑 보조정리의 적용

  1. 상대방이 양의 정수 값 m을 선택한다.

  2. 주어진 m에 대하여, 우리는 L에 속한 길이가 m 이상인 문자열 w를 선택한다.

  w 는 $w \in L$이고 $|w| \geq m$이면 아무 것이든 상관없다.

  3. 상대방은 $|xy| \leq m, |y| \geq 1$을 만족하도록 w를 xyz의 세 부분으로 분할한다. 우리는 상대방이 게임을 이기기 위하여 우리를 어렵게 만드는 선택을 한다고 가정해야 한다.

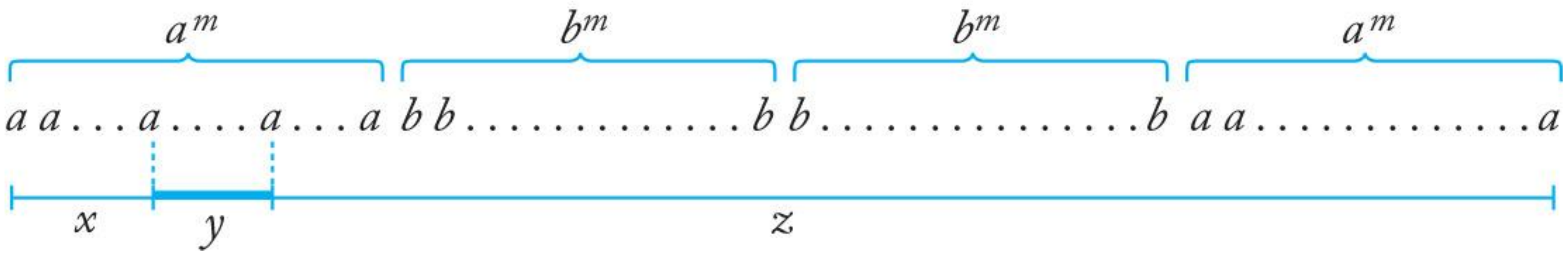
  4. 우리는, 펌프된 문자열 $w_i = xy^iz$가 L에 속하지 않도록 i를 선택한다.

비정규 언어의 식별 : 펌핑 보조정리

- 예제 4.8 (134 page) $\Sigma = \{a, b\}$일 때 다음의 언어는 정규 언어가 아님을 증명

$$L = \{ww^R : w \in \Sigma^*\}$$

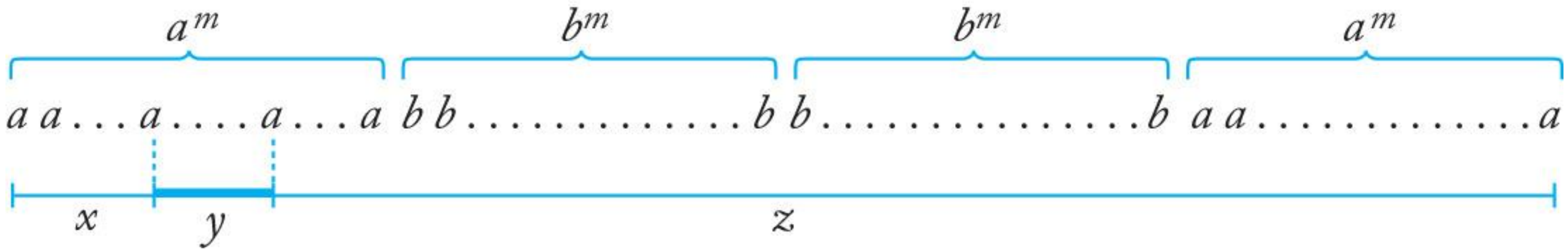
  - 단계 1에서 상대방이 어떤 m을 선택하든 그림에서 보여진 것과 같은 w 를 선택할 수 있음



  - 이 선택과 요구 조건 $|xy| \leq m$때문에, 상대방은 단계 3에서 y가 전부 a들로만 이루어지도록 선택할 수밖에 없음

비정규 언어의 식별 : 펌핑 보조정리

- 예제 4.8 (134 page) -계속



  - 단계 4에서 $i = 0$을 사용
  - 이런 식으로 얻어진 문자열은 왼쪽에 있는 a들의 수는 오른쪽에 있는 수보다 작게 되어 ww^R 형태 의 문자열이 될 수 없음
  - 따라서 L은 정규 언어가 아님

비정규 언어의 식별 : 펌핑 보조정리

- 예제 4.8 (134 page) -계속
  - 너무 짧은 w를 선택하면, 상대방이 y가 짝수개의 b들을 갖도록 선택할 수 있음
    - 마지막 단계에서 펌핑 보조정리가 위반되는 것을 보일 수가 없게 됨
  - 전부 a로만 이루어진 문자열을 w로 선택한다면,

$$w = a^{2m} \in L$$

    - 실패하게 되며 우리를 이기기 위해서, 상대방은 $y = aa$를 선택
    - 모든 i에 대하여, w_i는 L에 속하고, 우리는 지게 된다.

비정규 언어의 식별 : 펌핑 보조정리

- 예제 4.9 (135 page) $\Sigma = \{a, b\}$일 때 다음의 언어가 정규 언어가 아님을 보여라.

$$L = \{w \in \Sigma^*: n_a(w) < n_b(w)\}$$

  - m이 주어졌을 때 $w = a^m b^{m+1}$를 선택
  - $|xy| \leq m$ 이므로, 상대방은 y가 전부 a로만 이루어지는 것을 선택

$$y = a^k, 1 \leq k \leq m$$

$i = 2$를 사용하면, 얻어진 다음의 문자열은 L에 속하지 않음

$$w_2 = a^{m+k} b^{m+1}$$

  - 따라서 펌핑 보조정리는 위반이 되고, L 은 정규 언어가 아님