(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0094977
(43) 공개일자 2020년08월10일

(51) 국제특허분류(Int. Cl.)
G06F 16/33 (2019.01) *G06F 17/16* (2006.01)
(52) CPC특허분류
G06F 16/3346 (2019.01)
G06F 17/16 (2013.01)
(21) 출원번호 10-2019-0012472
(22) 출원일자 2019년01월31일
심사청구일자 2019년01월31일

(71) 출원인
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
한요섭
서울특별시 은평구 진관1로 77-8, 403동 204호 (진관동,은평뉴타운폭포동아파트)
코그네타 마르코
서울특별시 마포구 월드컵로3길 14, 202동 1411호(합정동)
(뒷면에 계속)
(74) 대리인
민영준

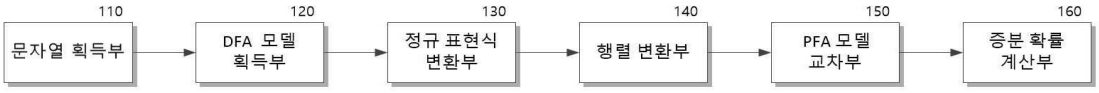
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **오토마타 기반 증분적 중위 확률 계산 장치 및 방법**

(57) 요 약

본 발명은 결정적 유한 오토마타를 기반으로 획득된 DFA 모델을 이용하여 문자열을 정규 표현식으로 획득한 후, 명료한 정규 표현식으로 변환하고, 확률적 유한 오토마타를 이용하여 명료한 정규 표현식으로 표현된 문자열의 출현 확률을 정확하게 계산할 수 있도록 하고, 정규 표현식으로 표현된 문자열의 증분에 따른 출현 확률을 정규 표현식의 증분 방식으로 획득함으로써 용이하게 획득할 수 있도록 하는 오토마타 기반 증분적 중위 확률 계산 장치 및 방법을 제공할 수 있다.

대 표 도



(72) 발명자

권순찬

충청남도 천안시 서북구 두정역길 48, 105동 604호(두정동, 두정역푸르지오아파트)

박준우

경기도 부천시 소향로 11, 20층 B동 2002호(상동, 코오롱이데아폴리스)

이 발명을 지원한 국가연구개발사업

과제고유번호 2018-0-00247

부처명 과학기술정보통신부

연구관리전문기관 정보통신기술진흥센터(NIPA산하)

연구사업명 정보통신방송연구개발사업

연구과제명 [이지바로][주관/창원대학교] GANs를 이용한 딥러닝용 학습데이터 자가 증식 기술 및 유효성 검증 기술 개발 (1/2)

기 여 율 1/1

주관기관 창원대학교 산학협력단

연구기간 2018.04.01 ~ 2018.12.31

명 세 서

청구범위

청구항 1

출현 확률을 계산할 문자열(w)의 각 문자에 대하여 결정적 유한 오토마타(이하 DFA)를 기반으로 정규 표현식 형태의 정규 언어를 획득하기 위해 다수의 상태 및 전이 함수로 구성된 DFA 모델을 획득하는 DFA 모델 획득부;

상기 DFA 모델에 초기 상태 및 단일 최종 상태와 초기 상태 및 단일 최종 상태에 대응하는 전이 함수를 추가한 후, DFA 모델의 각 상태들 간의 경로 중 중첩될 수 있는 경로를 동적으로 반복 제거하여, 명료한 정규 표현식을 표현할 수 있는 DFA 모델로 변환하는 정규 표현식 변환부;

변환된 DFA 모델의 각 상태 및 전이 경로의 확률을 확률적 유한 오토마타(이하 PFA)로 획득되는 PFA 모델을 기반으로 계산하여, 변환된 DFA 모델의 가중치로 적용하는 PFA 모델 교차부; 및

가중치가 적용된 DFA 모델에 문자열(w)을 포함하는 문자열 집합(F(w)) 중 증분된 문자열(wa)이 출현하는 문자열 집합(F(wa))에도 포함되는 문자열(F(w)\F(wa))에 대한 상태와 전이 함수를 추가함으로써, 증분 가중치가 적용된 DFA 모델을 획득하고, 획득된 증분 가중치가 적용된 DFA 모델을 이용하여, 문자열 및 증분된 문자열의 출현 확률을 계산하는 증분 확률 계산부; 를 포함하는 오토마타 기반 증분적 중위 확률 계산 장치.

청구항 2

제1 항에 있어서, 상기 정규 표현식 변환부는

n개의 상태($q = q_1, q_2, \ldots, q_n$)를 포함하는 상태 집합($Q \ni q$)을 갖는 상기 DFA 모델에 초기 상태($q_0$) 및 단일 최종 상태($q_{n+1}$)와 이에 대응하는 전이 함수를 추가하고, 수학식

$$\alpha_{i,j}^{k} = \alpha_{i,j}^{k-1} + \alpha_{i,k}^{k-1}(\alpha_{k,k}^{k-1})^{*}\alpha_{k,j}^{k-1}$$

(여기서, $\alpha_{i,j}^{k}$ 는 상태(q_i)로부터 시작하여, $k(1 \le k \le n)$번째 반복 제거에서 중간 상태(q_l)(여기서 $l < k$)인 경로로 상태(q_j)로 전이되는 문자열의 집합을 나타내고, $\alpha_{i,k}^{k-1}(\alpha_{k,k}^{k-1})^{*}\alpha_{k,j}^{k-1}$ 는 상태(q_i)로부터 시작하여, 중간 상태(q_k)를 통해 상태(q_j)로 전이되는 모든 문자열을 나타낸다)에 따라 경로를 반복 제거하는 오토마타 기반 증분적 중위 확률 계산 장치.

청구항 3

제2 항에 있어서, 상기 PFA 모델 교차부는

PFA 를 기반으로 모델링되는 PFA 모델(P)을 기반으로 문자열(w)에 대응하여 명료한 정규 표현식을 표현하도록 변환된 DFA 모델(D)의 가중치를 수학식

$$[\mathcal{D} \cap \mathcal{P}](w) = \begin{cases} \mathcal{P}(w), & w \in L(\mathcal{D}); \\ 0, & \text{otherwise.} \end{cases}$$

(여기서 L(D)는 변환된 DFA 모델(D)로 표현되는 문자열(w)에 대한 명료한 정규 표현식)

에 따라 획득하여 적용하는 오토마타 기반 증분적 중위 확률 계산 장치.

청구항 4

제3 항에 있어서, 상기 증분 확률 계산부는

가중치가 적용된 DFA 모델에 문자열(w)을 포함하는 문자열 집합(F(w)) 중 증분된 문자열(wa)이 출현하는 문자열 집합(F(wa))에도 포함되는 중복 문자열(F(w)\F(wa))에 대한 상태와 전이 함수를 수학식

$$(\alpha_{k,k}^{k-1})^{*}\alpha_{k,k+1}^{k-1} = \mathcal{F}(w_1 \cdots w_{k-1}) \backslash \mathcal{F}(w_1 \cdots w_k)$$

에 따라 획득하여, 가중치가 적용된 DFA 모델에 추가함으로써, 증분 가중치가 적용된 DFA 모델을 획득하는 오토마타 기반 증분적 중위 확률 계산 장치.

청구항 5

제1 항에 있어서, 상기 오토마타 기반 증분적 중위 확률 계산 장치는

상기 정규 표현식 변환부에서 명료한 정규 표현식을 표현하도록 변환된 DFA 모델에서 정규 표현식의 각 문자를 행렬 형식으로 변환하는 행렬 변환부; 를 더 포함하는 오토마타 기반 증분적 중위 확률 계산 장치.

청구항 6

출현 확률을 계산할 문자열(w)의 각 문자에 대하여 결정적 유한 오토마타(이하 DFA)를 기반으로 정규 표현식 형태의 정규 언어를 획득하기 위해 다수의 상태 및 전이 함수로 구성된 DFA 모델을 획득하는 단계;

상기 DFA 모델에 초기 상태 및 단일 최종 상태와 초기 상태 및 단일 최종 상태에 대응하는 전이 함수를 추가한 후, DFA 모델의 각 상태들 간의 경로 중 중첩될 수 있는 경로를 동적으로 반복 제거하여, 명료한 정규 표현식을 표현할 수 있는 DFA 모델로 변환하는 단계;

변환된 DFA 모델의 각 상태 및 전이 경로의 확률을 확률적 유한 오토마타(이하 PFA)로 획득되는 PFA 모델을 기반으로 계산하여, 변환된 DFA 모델의 가중치로 적용하는 단계; 및

가중치가 적용된 DFA 모델에 문자열(w)을 포함하는 문자열 집합(F(w)) 중 증분된 문자열(wa)이 출현하는 문자열 집합(F(wa))에도 포함되는 문자열(F(w)\F(wa))에 대한 상태와 전이 함수를 추가함으로써, 증분 가중치가 적용된 DFA 모델을 획득하고, 획득된 증분 가중치가 적용된 DFA 모델을 이용하여, 문자열 및 증분된 문자열의 출현 확률을 계산하는 단계; 를 포함하는 오토마타 기반 증분적 중위 확률 계산 방법.

청구항 7

제6 항에 있어서, 상기 DFA 모델로 변환하는 단계는

n개의 상태($q = q_1, q_2, \ldots, q_n$)를 포함하는 상태 집합($Q \ni q$)을 갖는 상기 DFA 모델에 초기 상태($q_0$) 및 단일 최종 상태($q_{n+1}$)와 이에 대응하는 전이 함수를 추가하는 단계; 및

수학식

$$\alpha_{i,j}^{k} = \alpha_{i,j}^{k-1} + \alpha_{i,k}^{k-1}(\alpha_{k,k}^{k-1})^{*}\alpha_{k,j}^{k-1}$$

(여기서, $\alpha_{i,j}^{k}$ 는 상태(q_i)로부터 시작하여, k($1 \le k \le n$)번째 반복 제거에서 중간 상태(q_l)(여기서 $l < k$)인 경로로 상태(q_j)로 전이되는 문자열의 집합을 나타내고, $\alpha_{i,k}^{k-1}(\alpha_{k,k}^{k-1})^{*}\alpha_{k,j}^{k-1}$ 는 상태(q_i)로부터 시작하여, 중간 상태(q_k)를 통해 상태(q_j)로 전이되는 모든 문자열을 나타낸다)에 따라 경로를 반복 제거하는 단계; 를 포함하는 오토마타 기반 증분적 중위 확률 계산 방법.

청구항 8

제7 항에 있어서, 상기 DFA 모델의 가중치로 적용하는 단계는

PFA 를 기반으로 모델링되는 PFA 모델(P)을 기반으로 문자열(w)에 대응하여 명료한 정규 표현식을 표현하도록 변환된 DFA 모델(D)의 가중치를 수학식

$$[\mathcal{D} \cap \mathcal{P}](w) = \begin{cases} \mathcal{P}(w), & w \in L(\mathcal{D}); \\ 0, & \text{otherwise.} \end{cases}$$

(여기서 L(D)는 변환된 DFA 모델(D)로 표현되는 문자열(w)에 대한 명료한 정규 표현식)

에 따라 획득하여 적용하는 오토마타 기반 증분적 중위 확률 계산 방법.

청구항 9

제8 항에 있어서, 상기 출현 확률을 계산하는 단계는

가중치가 적용된 DFA 모델에 문자열(w)을 포함하는 문자열 집합(F(w)) 중 증분된 문자열(wa)이 출현하는 문자열 집합(F(wa))에도 포함되는 중복 문자열(F(w)\F(wa))에 대한 상태와 전이 함수를 수학식

$$(\alpha_{k,k}^{k-1})^* \alpha_{k,k+1}^{k-1} = \mathcal{F}(w_1 \cdots w_{k-1}) \backslash \mathcal{F}(w_1 \cdots w_k)$$

에 따라 획득하여, 가중치가 적용된 DFA 모델에 추가하는 단계;

상태와 전이 함수가 추가된 가중치가 적용된 DFA 모델에 따라 증분 가중치가 적용된 DFA 모델을 획득하는 단계; 및

증분 가중치가 적용된 DFA 모델을 이용하여 증분 문자열의 출현 확률을 계산하는 단계; 를 포함하는 오토마타 기반 증분적 중위 확률 계산 방법.

청구항 10

제6 항에 있어서, 상기 오토마타 기반 증분적 중위 확률 계산 방법은

명료한 정규 표현식을 표현하도록 변환된 DFA 모델에서 정규 표현식의 각 문자를 행렬 형식으로 변환하는 단계; 를 더 포함하는 오토마타 기반 증분적 중위 확률 계산 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 중위 확률 계산 장치 및 방법에 관한 것으로, 오토마타 기반 증분적 중위 확률 계산 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 자연어 처리에서 주요 작업 중 하나는 주어진 구문(phrase)의 출현 확률 또는 주어진 패턴과 매칭되는 모든 구문의 출현 확률을 계산하는 것이다.

[0003] 일반적으로 주어진 패턴과 매칭되는 문자열(string)의 분포를 언어 모델로 모델링하기 위해 확률적 문맥 자유 문법(probabilistic context-free grammar) 또는 확률적 유한 오토마타(Probabilistic Finite Automata: 이하 PFA)가 주로 사용하고 있다. 이중에서도 확률적 유한 오토마타는 많은 확률적 언어 현상을 간단하지만 강력하고 잘 이해할 수 있는 표현으로 제공할 수 있다는 장점으로 인해, 현재 음성 처리 작업의 대부분이 PFA를 이용하고 있다.

[0004] PFA에서 중요한 문제는 주어진 패턴 분포에 대한 접사(affix)의 확률을 계산하는 것이다. 즉 PFA와 문자열(w)이 주어지면, PFA가 모델링한 문자열의 분포에서 문자열(w)이 다양한 위치에 나타날 확률을 계산하는 것이다. 예를 들어, PFA에 의해 모델링된 문자열 분포에서 문자열(w)은 다른 임의의 문자 또는 문자열 x의 전단에 배치되어 wx와 같은 접두사의 형태로 나타나거나, 후단에 배치되어 xw와 같은 접미사의 형태로 나타날 수 있다. 또한 임의의 문자 또는 문자열들 x, y의 사이에 배치되어 xwy와 같이 가운데 나타날 수 있으며, PFA는 이러한 모든 경우에 대한 확률의 합을 계산해야 한다.

[0005] 주어진 문자열(w)이 접두사로 나타나는 경우에 대한 확률은 계산이 상대적으로 용이한 반면, 접미사 또는 중위어로 나타나는 경우에 대한 확률은 문자열(w)이 반복적으로 나타날 수 있다는 문제로 인해 확률의 계산이 용이

하지 않다. 그럼에도 문자열(w)이 접두사뿐만 아니라, 접미사나 중위어로 나타날 확률에 대한 계산 방법 또한 이미 많은 연구가 수행되어 공지되었다.

[0006] 다만 기존의 연구에서는 주어진 문자열(w)이 나타날 확률을 계산할 수 있는 반면, 문자열(w)의 증분(또는 확장이라고도 함)에 대한 확률은 계산할 수 없다는 한계가 있다. 예를 들면, PFA가 모델링한 문자열의 분포에서 문자열(w)이 나타날 확률은 계산할 수 있는 반면, 문자(character)(a)가 추가된 문자열(wa)이 나타날 확률은 별도로 다시 계산해야 한다. 마찬가지로 문자열(wa)에 대한 확률을 계산하더라도, 문자열(w)에 대한 확률을 별도로 계산해야 한다.

[0007] 따라서 문자열의 증분에 따른 확률을 각각 별도로 계산해야 하므로, 매우 긴 계산 시간을 요구할 뿐만 아니라, 많은 자원을 요구한다는 문제가 있다.

선행기술문헌

특허문헌

[0008] (특허문헌 0001) 한국 등록 특허 제10-1645890호 (2016.07.29 등록)

발명의 내용

해결하려는 과제

[0009] 본 발명의 목적은 문자열의 증분에 따른 확률을 정확하고 신속하게 계산할 수 있어, 증분적 중위 확률을 용이하게 획득할 수 있는 오토마타 기반 증분적 중위 확률 계산 장치 및 방법을 제공하는데 있다.

[0010] 본 발명의 다른 목적은 문자열의 증분적 중위 확률을 동시에 계산하여, 고속으로 문자열의 증분에 따른 확률을 계산할 수 있는 오토마타 기반 증분적 중위 확률 계산 장치 및 방법을 제공하는데 있다.

과제의 해결 수단

[0011] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른 오토마타 기반 증분적 중위 확률 계산 장치는 출현 확률을 계산할 문자열(w)의 각 문자에 대하여 결정적 유한 오토마타(이하 DFA)를 기반으로 정규 표현식 형태의 정규 언어를 획득하기 위해 다수의 상태 및 전이 함수로 구성된 DFA 모델을 획득하는 DFA 모델 획득부; 상기 DFA 모델에 초기 상태 및 단일 최종 상태와 초기 상태 및 단일 최종 상태에 대응하는 전이 함수를 추가한 후, DFA 모델의 각 상태들 간의 경로 중 중첩될 수 있는 경로를 동적으로 반복 제거하여, 명료한 정규 표현식을 표현할 수 있는 DFA 모델로 변환하는 정규 표현식 변환부; 변환된 DFA 모델의 각 상태 및 전이 경로의 확률을 확률적 유한 오토마타(이하 PFA)로 획득되는 PFA 모델을 기반으로 계산하여, 변환된 DFA 모델의 가중치로 적용하는 PFA 모델 교차부; 및 가중치가 적용된 DFA 모델에 문자열(w)을 포함하는 문자열 집합(F(w)) 중 증분된 문자열(wa)이 출현하는 문자열 집합(F(wa))에도 포함되는 문자열(F(w)\F(wa))에 대한 상태와 전이 함수를 추가함으로써, 증분 가중치가 적용된 DFA 모델을 획득하고, 획득된 증분 가중치가 적용된 DFA 모델을 이용하여, 문자열 및 증분된 문자열의 출현 확률을 계산하는 증분 확률 계산부; 를 포함한다.

[0012] 상기 목적을 달성하기 위한 본 발명의 다른 실시예에 따른 오토마타 기반 증분적 중위 확률 계산 방법은 출현 확률을 계산할 문자열(w)의 각 문자에 대하여 결정적 유한 오토마타(이하 DFA)를 기반으로 정규 표현식 형태의 정규 언어를 획득하기 위해 다수의 상태 및 전이 함수로 구성된 DFA 모델을 획득하는 단계; 상기 DFA 모델에 초기 상태 및 단일 최종 상태와 초기 상태 및 단일 최종 상태에 대응하는 전이 함수를 추가한 후, DFA 모델의 각 상태들 간의 경로 중 중첩될 수 있는 경로를 동적으로 반복 제거하여, 명료한 정규 표현식을 표현할 수 있는 DFA 모델로 변환하는 단계; 변환된 DFA 모델의 각 상태 및 전이 경로의 확률을 확률적 유한 오토마타(이하 PFA)로 획득되는 PFA 모델을 기반으로 계산하여, 변환된 DFA 모델의 가중치로 적용하는 단계; 및 가중치가 적용된 DFA 모델에 문자열(w)을 포함하는 문자열 집합(F(w)) 중 증분된 문자열(wa)이 출현하는 문자열 집합(F(wa))에도 포함되는 문자열(F(w)\F(wa))에 대한 상태와 전이 함수를 추가함으로써, 증분 가중치가 적용된 DFA 모델을 획득하고, 획득된 증분 가중치가 적용된 DFA 모델을 이용하여, 문자열 및 증분된 문자열의 출현 확률을 계산하는 단계; 를 포함한다.

발명의 효과

[0013] 따라서, 본 발명의 실시예에 따른 오토마타 기반 증분적 중위 확률 계산 장치 및 방법은 결정적 유한 오토마타를 기반으로 획득된 DFA 모델을 이용하여 문자열을 정규 표현식으로 획득한 후, 명료한 정규 표현식으로 변환하고, 확률적 유한 오토마타를 이용하여 명료한 정규 표현식으로 표현된 문자열의 출현 확률을 정확하게 계산할 수 있도록 한다. 또한 정규 표현식으로 표현된 문자열의 증분에 따른 출현 확률을 정규 표현식의 증분 방식으로 획득함으로써 용이하게 획득할 수 있도록 한다.

도면의 간단한 설명

[0014] 도1 은 본 발명의 일 실시예에 따른 오토마타 기반 증분적 중위 확률 계산 장치의 개략적 구조를 나타낸다.

도2 는 도1 의 정규 표현식 변환부가 DFA 모델의 상태 및 전이를 추가하는 일예를 나타낸다.

도3 은 DFA 모델과 PFA 모델의 교차 알고리즘을 나타낸다.

도4 는 문자열과 그 증분에 대한 정규 표현식의 관계를 나타낸다.

도5 는 도1 의 증분 확률 계산부가 DFA 모델로부터 문자열와 그 증분 문자열이 출연할 확률을 누적하여 계산하는 알고리즘을 나타낸다.

도6 은 문자열의 길이와 상태 집합의 크기에 따른 증분 및 교차 방식의 적용에 따른 연산 소요 시간 측정 결과를 나타낸다.

도7 은 본 발명의 일 실시예에 따른 오토마타 기반 증분적 중위 확률 계산 방법을 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0015] 본 발명과 본 발명의 동작상의 이점 및 본 발명의 실시에 의하여 달성되는 목적을 충분히 이해하기 위해서는 본 발명의 바람직한 실시예를 예시하는 첨부 도면 및 첨부 도면에 기재된 내용을 참조하여야만 한다.

[0016] 이하, 첨부한 도면을 참조하여 본 발명의 바람직한 실시예를 설명함으로써, 본 발명을 상세히 설명한다. 그러나, 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며, 설명하는 실시예에 한정되는 것이 아니다. 그리고, 본 발명을 명확하게 설명하기 위하여 설명과 관계없는 부분은 생략되며, 도면의 동일한 참조부호는 동일한 부재임을 나타낸다.

[0017] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라, 다른 구성요소를 더 포함할 수 있는 것을 의미한다. 또한, 명세서에 기재된 "...부", "...기", "모듈", "블록" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어나 소프트웨어 또는 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.

[0018] 도1 은 본 발명의 일 실시예에 따른 오토마타 기반 증분적 중위 확률 계산 장치의 개략적 구조를 나타낸다.

[0019] 도1 을 참조하면, 본 실시예에 따른 오토마타 기반 증분적 중위 확률 계산 장치는 문자열 획득부(110), DFA 모델 획득부(120), 정규 표현식 변환부(130), 행렬 변환부(140), PFA 모델 교차부(150) 및 증분 확률 계산부(160)를 포함한다.

[0020] 우선 문자열 획득부(110)는 출현 확률을 계산할 문자열(w)을 획득한다. 이때 문자열 획득부(110)는 문자열(w)이 포함되었는지 탐색해야 하는 언어(문장)를 함께 획득할 수도 있다. 이하에서는 문자의 집합(set of characters)을 Σ라고 하고, 모든 문자열의 집합(set of all strings)을 Σ^*라 하며, 탐색할 문자열(w)은 문자열 집합(Σ^*)의 원소($w = w_1, w_2, \ldots, w_n \in \Sigma^*$)이다. 그리고 문자열(w)의 길이(n)는 $|w| = n$이며, 공백 문자열(empty string)은 λ라고 한다.

[0021] DFA 모델 획득부(120)는 문자열(w)의 각 문자에 대하여 유일한 상태변화를 갖는 결정적 유한 오토마타(Deterministic Finite Automata: 이하 DFA)를 기반으로 모델링된 DFA 모델(D)을 획득한다.

[0022] DFA 모델(D)은 탐색해야 하는 문자열(w)을 정규 언어(Regular Language)(L(D))로 변환하기 위한 모델로서, 일반적으로 $(Q, \Sigma, \delta, q_1, F)$의 5-튜플(5-tuple)을 갖도록 모델링된다. 여기서 Q는 다수의 상태들($q = q_1, q_2,$

..., $q_n \in Q$)의 유한 집합(finite set of states)을 나타내고, δ는 전이 함수(transition function)로서 상태 집합(Q)에 대해 문자열 집합(Σ)의 전이 경로는 상태 집합(Q)의 상태 중 적어도 하나에 포함되는 $\delta: Q \times \Sigma \rightarrow Q$의 조건을 만족한다. 그리고, q_1은 상태 집합(Q) 중 초기 상태(initial state)($q_1 \in Q$)를 나타내고, F는 적어도 하나의 상태를 갖는 최종 상태 집합(set of final state)으로 상태 집합(Q)에 포함되는 집합($F \subset Q$)이다.

[0023] 언어는 문자열의 집합이므로, DFA에 의해 미리 의해 모델링된 DFA 모델(D)은 획득된 언어를 정규 언어(L(D))로 인식할 수 있다. 이때 DFA 모델(D)은 정규 언어(L(D))를 정규 표현식(Unambiguous Regular Expression)의 형태로 획득하여 인식할 수 있다.

[0024] 정규 표현식은 정규 언어의 일반적인 표현 방식으로, 기지정된 다양한 메타문자를 이용하여 정규 언어(L(D))를 간략하게 표현할 수 있도록 한다. 일예로, 정규 표현식은 a, aa, aaa, ... 와 같은 정규 언어를 0회 이상의 출현을 의미하는 메타문자 "*"를 이용하여 a^* 로 간략하게 표현할 수 있도록 한다. 또한 정규 표현식에서 "+"는 1번 이상의 발생을 의미하는 메타문자로서 "ab+c" 는 "abc", "abbc", "abbbc" 등을 의미할 수 있다.

[0025] 다만 DFA 모델(D)이 정규 언어(L(D))를 인식하여 정규 표현식으로 변환하는 경우, 불명료한 정규 표현식(ambiguous Regular Expression)이 생성될 가능성이 높다.

[0026] 일 예로, 문자열의 집합(Σ)의 문자가 a, b 인 경우($\Sigma = \{a, b\}$)에 대한 정규 표현식($((a \cup b)^*aa(a \cup b)^*)$)에서 문자열 baa를 탐색하는 경우, 문자열(baa)은 둘 이상의 위치에서 탐색될 수 있다. 이를 명확히 하기 위해 상기의 정규 표현식($((a \cup b)^*aa(a \cup b)^*)$)에 첨자를 부여하면, 정규 표현식($((a_1 \cup b_1)^*a_2a_3(a_4 \cup b_2)^*)$)으로 표현될 수 있다. 그리고 정규 표현식($((a_1 \cup b_1)^*a_2a_3(a_4 \cup b_2)^*)$)에서 문자열(baa) 는 $b_1a_1a_2a_3$로도 탐색될 수 있고, $b_1a_2a_3a_4$로도 탐색될 수 있다. 즉 하나의 정규 표현식($((a \cup b)^*aa(a \cup b)^*)$)에서 하나의 문자열(baa)이 중복으로 탐색될 수 있다. 이는 문자열(baa)의 출연 확률을 중복으로 계산할 수 있도록 하는 오류를 발생시키는 불명료한 정규 표현식이다.

[0027] 그에 반해, 정규 표현식(b*a(bb*a)*a(a∪b)*)은 문자열(baa)을 탐색할 때, 중복 탐색이 발생하지 않는 명료한 표현식이다.

[0028] 이러한 오류가 발생하는 것을 방지하기 위해, 정규 표현식 변환부(130)는 DFA 모델 획득부(120)에서 획득된 DFA 모델(D)에 의해 인식된 정규 언어(L(D))으로부터 명료한 정규 표현식의 형태로 변환한다.

[0029] DFA 모델(D)이 $D = (Q, \Sigma, \delta, q_1, F)$이고, 상태 집합(Q)의 길이($|Q|$) = n이고, 1에서 n까지 순차 정렬된 경우, 정규 표현식 변환부(130)는 DFA 모델(D)의 지정된 상태 집합(Q)에 2개의 새로운 상태(q_0, q_{n+1})를 추가한다. 그리고 추가된 상태(q_0)는 새로이 지정된 시작 상태가 되도록 전이($\delta(q_0, \lambda) = q_1$)를 DFA 모델(D)에 추가하고, 추가된 상태(q_{n+1})는 새로이 지정된 유일한 최종 상태가 되도록 같은 전이($\forall q \in F, \delta(q, \lambda) = q_{n+1}$)를 DFA 모델(D)에 추가한다.

[0030] 도2 는 도1 의 정규 표현식 변환부가 DFA 모델의 상태 및 전이를 추가하는 일예를 나타낸다.

[0031] 도2 에서 (a) 및 (b)는 각각 DFA 모델 획득부(120)에서 획득된 DFA 모델(D)의 일예를 나타내고, (c) 및 (d)는 각각 정규 표현식 변환부(130)가 (a) 및 (b)의 DFA 모델(D)에 새로운 상태 및 전이를 추가한 모델을 나타낸다.

[0032] 즉 새로운 시작 상태(q_0)에서는 공백 문자열(λ), 즉 별도의 조건없이 이전의 시작 상태(q_1)로 전이되고, 이전의 적어도 하나의 최종 상태(F) 모두는 새로운 유일한 최종 상태(q_{n+1})로 별도의 조건없이 전이되도록 한다.

[0033] 그리고 수학식 1을 이용하여, 상태 및 전이가 추가된 DFA 모델(D)에서 상태들($q = q_1, q_2, ..., q_n \in Q$)을 동적으로 반복하여 제거한다.

수학식 1

[0034] $$\alpha_{i,j}^{k} = \alpha_{i,j}^{k-1} + \alpha_{i,k}^{k-1}(\alpha_{k,k}^{k-1})^{*}\alpha_{k,j}^{k-1}$$

[0035] 수학식 1에서 $\alpha_{i,j}^{k}$는 상태(q_i)로부터 시작하여, k(1 ≤ k ≤ n)번째 반복 제거에서 중간 상태(q_l)(여기서 l < k)인 경로로 상태(q_j)로 전이되는 문자열의 집합에 대응하며, 유사하게 $\alpha_{i,k}^{k-1}(\alpha_{k,k}^{k-1})^{*}\alpha_{k,j}^{k-1}$는 상태($q_i$)로부터 시작하여, 중간 상태($q_k$)를 통해 상태($q_j$)로 전이되는 모든 문자열에 대응한다.

[0036] 수학식 1은 정규 표현식에 대한 일반적 연결, 결합 및 Kleene star 규칙을 따르며, 수학식 1 및 2로부터 불명료하게 나타날 수 있는 중첩될 수 있는 정규 표현식의 경로를 제거함으로써, 정규 언어(L(D))에 대한 명료한 정규 표현식(R)은 $\alpha_{0,n+1}^{n}$로서 획득될 수 있다.

[0037] 여기서 $\alpha_{i,j}^{0}$는 수학식 2의 조건에 따라 결정된다.

수학식 2

[0038] $$\alpha_{i,j}^{0} = \begin{cases} \lambda, & i = 0, j = 1; \\ \lambda, & q_i \in F \wedge j = n+1; \\ \{c \mid \delta(q_i, c) = q_j\}, & \text{otherwise.} \end{cases}$$

[0039] 또한 수학식 1에서 문자(c)와 공집합($\emptyset$)에 대한 연산은 수학식 3의 4가지 연산 방식으로 수행된다.

수학식 3

[0040]
- $\emptyset + c = c + \emptyset = c$, for $c \in \Sigma$
- $\emptyset c = c\emptyset = \emptyset$, for $c \in \Sigma$
- $\lambda c = c\lambda = c$, for $c \in \Sigma$
- $\emptyset^{*} = \lambda$

[0041] 정규 표현식 변환부(130)에 의해 DFA 모델(D)이 수정되고, 반복적 상태 제걸르 통해 명료한 정규 표현식(R)이 획득되면, 행렬 변환부(140)가 획득된 명료한 정규 표현식(R)을 행렬 형태로 변환한다.

[0042] 행렬 변환부(140)는 명료한 정규 표현식(R)을 수학식 4에 따라 맵핑하여 행렬로 변환한다.

수학식 4

[0043]
- $\emptyset \rightarrow \mathbf{0}$
- $\lambda \rightarrow \mathbf{1}$
- $c \in \Sigma \rightarrow \mathbb{M}(c)$

[0044] 여기서 임의의 2개의 정규 표현식(R, S)에 대응하는 행렬을 각각 $\mathbb{M}(R)$ 및 $\mathbb{M}(S)$라 하면, 2개의 정규 표현식(R, S) 사이의 여러 연산은 수학식 5와 같은 행렬의 연산 형태로 표현될 수 있다.

수학식 5

[0045]
- $R \cup S \rightarrow \mathbb{M}(R) + \mathbb{M}(S)$
- $RS \rightarrow \mathbb{M}(R)\mathbb{M}(S)$
- $R^* \rightarrow (\mathbf{1} - \mathbb{M}(R))^{-1}$

[0046] PFA 모델 교차부(150)는 확률적 유한 오토마타(probabilistic finite automata: 이하 PFA)에 의해 모델링된 PFA 모델(P)을 기반으로 명료한 정규 표현식(R)에 대한 정규 표현 행렬($\mathbb{M}(R)$)로부터 정규 표현식(R)의 가중치를 획득하여, 정규 표현식 변환부(130)에 의해 수정된 DFA 모델(D)의 각 상태 및 전이 경로에 대해 확률로 표현되는 가중치를 적용함으로써, PFA 모델(P)과 교차된 DFA 모델(D)을 획득한다.

[0047] PFA 모델(P)은 모든 문자열 집합(Σ^*)에 대한 [0, 1]에 대응하는 값의 가중치를 갖는 확률 함수를 획득하기 위한 모델로서, 문자열(w)의 가중치(P(w)) 및 경로(path)(π)의 가중치(P(π))를 계산한다.

[0048] 탐색할 문자열($w = w_1, w_2, \ldots, w_n \in \Sigma^*$)을 고려할 때, 대응하는 경로(π)는 PFA 모델(P)에서 $\pi = (q_0, w_1, q_1), (q_1, w_2, q_2), \ldots, (q_{n-1}, w_n, q_n)$ 이다.

[0049] PFA 모델(P)은 DFA 모델(D)과 유사하게, (Q, Σ, δ, I, F)의 5-튜플(5-tuple)을 갖도록 모델링되고, 여기서 PFA 모델(P)에서 전이 함수(δ)는 $\delta : Q \times \Sigma \times Q \rightarrow [0, 1]$의 조건을 만족하며, I와 F는 각각 적어도 상태를 갖는 초기 상태 집합(set of initial state)의 각 상태의 확률($I: Q \rightarrow [0, 1]$)을 나타내고, F는 적어도 하나의 상태를 갖는 최종 상태 집합(set of final state)의 각 상태의 확률($F: Q \rightarrow [0, 1]$)을 나타낸다. 여기서 전이 함수(δ)의 디폴트 값은 0으로 가정된다. 즉 상태들 사이에 전이가 존재하지 않으면, 가중치가 0인 것으로 고려된다.

[0050] 그리고 PFA 모델(P)에서는 모든 초기 상태 집합의 확률의 합은 1 ($\sum_{q \in Q} I(q) = 1$)이고, 모든 상태에 대해 최종 상태의 확률과 경로별 확률의 합은 1($\forall q \in Q,\ F(q) + \sum_{q' \in Q, c \in \Sigma} \delta(q, c, q') = 1$)이고, 모든 상태는 접근 가능하거나 상태간 상호 접근이 가능하다는 조건을 만족한다.

[0051] 상기한 조건을 만족한다면, 문자열(w)에 대한 가중치(P(w))는 $0 \leq P(w) \leq 1$이고, $\Sigma_{w \in \Sigma^*} P(w) = 1$이다.

[0052] 이에 PFA 모델(P)에서 경로(π)에 대한 가중치, 즉 확률은 수학식 6으로 계산될 수 있다.

수학식 6

[0053] $$\mathcal{P}(\pi) = I(q_0)\left(\prod_{i=1}^{n}\delta(q_{i-1}, w_i, q_i)\right)F(q_n)$$

[0054] 문자열(w)에 대응하는 모든 경로의 집합을 Φ_w라 하면, PFA 모델(P)에서 문자열(w)에 대한 가중치, 즉 확률은 $\sum_{\pi\in\Phi_w}\mathcal{P}(\pi)$ 로 계산된다.

[0055] 한편, 수학식 4에서와 같이 문자(c)를 행렬($\mathbb{M}(c)$)로 표현한 경우, PFA 모델(P)는 $\mathcal{P} = (Q, \Sigma, \{\mathbb{M}(c)\}_{c\in\Sigma}, \mathbb{I}, \mathbb{F})$와 같이 행렬 형태의 튜플을 갖는다. 여기서 $\{\mathbb{M}(c)\}_{c\in\Sigma}$는 $\mathbb{M}(c)_{i,j} = \delta(q_i, c, q_j)$인 |Q| × |Q| 전이 행렬의 집합이다. 그리고 $\mathbb{I}$는 $\mathbb{I}_i = I(q_i)$ 인 1 × |Q|이고, $\mathbb{F}$는 $\mathbb{F}_j = F(q_j)$인 |Q| × 1 벡터이다.

[0056] 이에 문자열(w)을 행렬 형태로 표현하는 경우, 문자열(w) 행렬에 대한 확률은 수학식 7과 같이 표현된다.

수학식 7

[0057] $$\mathbb{I}\prod_{i=1}^{|w|}\mathbb{M}(w_i)\mathbb{F}$$

[0058] 간결함을 위해 $\mathbb{M}(\Sigma) = \sum_{c\in\Sigma}\mathbb{M}(c)$로 표현하고, 0과 1을 각각 0 행렬 및 항등 행렬(identity matrix)로 표현할 수 있다.

[0059] PFA 모델(P)에서 문자의 집합(Σ)에 대한 행렬의 가중치는 $\sum_{i=0}^{\infty}\mathbb{M}(\Sigma)^i = (\mathbf{1} - \mathbb{M}(\Sigma))^{-1}$ 로 계산될 수 있다.

[0060] 이에 PFA 모델(P)에서 문자열(w)이 문자열 집합(Σ^*)에 대한 접두사($w\Sigma^*$) 또는 접미사(Σ^*w)로 나타날 확률은 각각 수학식 8 및 9로 계산될 수 있다.

수학식 8

[0061] $$\mathcal{P}(w\Sigma^*) = \mathbb{I}\left(\prod_{i=1}^{|w|}\mathbb{M}(w_i)\right)\mathbb{M}(\Sigma^*)\mathbb{F}$$

수학식 9

[0062] $$\mathcal{P}(\Sigma^*w) = \mathbb{I}\mathbb{M}(\Sigma^*)\left(\prod_{i=1}^{|w|}\mathbb{M}(w_i)\right)\mathbb{F}$$

[0063] 만일 오토마톤(M)이 $\Sigma_{q\in Q}I(q) \leq 1$ 또는 $\forall q \in Q$ 를 제외한 PFA에서 요구되는 $F(q) + \Sigma_{q'\in Q, c\in\Sigma}\ \delta(q, c, q')$

≤ 1 을 모두 만족하면, 오토마톤(M)은 서브-PFA(sub-PFA)라고 할 수 있다. 그리고 서브-PFA(M)는 문자의 집합(Σ)에서 어떤 문자열(w)에 대한 가중치(M(w))로 0 ≤ M(w) ≤ 1를 가지며, $\Sigma_{w\in\Sigma^*}M(w) \leq 1$이다.

[0064] 문자의 집합(Σ)에 대한 확률적 언어(stochastic language)는 $S \subseteq \Sigma^*$인 집합(S)이고, 여기서 집합(S)의 각 문자열은 연관 확률($Pr_S(w)$)로 $0 \leq Pr_S(w) \leq 1$를 가지며, $\Sigma_{w\in\Sigma^*} Pr_S(w) = 1$이다. 주어진 확률적 언어(S)에 대해 $\forall w \in \Sigma^*$이고, $Pr_S(w) = Pr_S(w)$이면, 확률적 언어(S)는 정류 확률적 언어(regular stochastic language)라 한다.

[0065] 그리고 DFA 모델(D)과 PFA 모델(P)에 교차 적용하는 방식은 정규 언어(L(D))의 가중치를 계산하여 적용하는 형태로 수행될 수 있으며, 수학식 10 및 1에 따라 서브-PFA([D ∩ P])를 생성함으로써, 획득할 수 있다.

수학식 10

[0066]
$$[\mathcal{D}\cap\mathcal{P}](w) = \begin{cases} \mathcal{P}(w), & w \in L(\mathcal{D}); \\ 0, & \text{otherwise.} \end{cases}$$

수학식 11

[0067]
$$\sum_{w\in\Sigma^*}[\mathcal{D}\cap\mathcal{P}](w) = \sum_{w\in L(\mathcal{D})}\mathcal{P}(w)$$

[0068] 수학식 10 및 11에 따른 서브-PFA([D ∩ P])의 교차 알고리즘은 DFA 모델(D)과 PFA 모델(P)이 주어진 경우, 상태 $Q_W = Q_D \times Q_P$를 갖는 새로운 모델(W)을 구성하는 것으로, 모델(W)은 두 상태((x, y), (x', y'))와 문자(c ∈ Σ)에 대해 $\delta_D(x, c) = x'$이면 $\delta_W((x, y), c, (x', y')) = \delta_P(y, c, y')$이고, 그렇지 않으면 0으로 나타난다. 유사하게 DFA 모델(D)의 초기 상태가 p이면, $I_W((p, q)) = I_P(q)$이고, DFA 모델(D)의 최종 상태가 p'이면, $F_W((p', q')) = F_P(q')$이며, 이외엔 0이다.

[0069] DFA 모델(D)과 PFA 모델(P)의 교차 알고리즘은 결과적으로 도3 과 같이 정리될 수 있다.

[0070] 도3 은 DFA 모델과 PFA 모델의 교차 알고리즘을 나타낸다.

[0071] 도3 에 나타난 알고리즘에 따르면, 정규 언어(L(D))에서 모든 문자열의 가중치의 합은 수학식 12로 계산될 수 있다.

수학식 12

[0072]
$$\mathbb{I}_{[\mathcal{D}\cap\mathcal{P}]}(\mathbf{1} - \mathbb{M}_{[\mathcal{D}\cap\mathcal{P}]}(\Sigma))^{-1}\mathbb{F}_{[\mathcal{D}\cap\mathcal{P}]}$$

[0073] PFA 모델 교차부(150)는 수학식 11과 수학식 12로부터 명료한 정규 언어(L(D))의 정규 표현 행렬($\mathbb{M}(R)$)에 대한 가중치를 수학식 13으로 획득할 수 있다.

수학식 13

[0074] $$\mathbb{I}_{\mathcal{P}}\mathbb{M}_{\mathcal{P}}(R)\mathbb{F}_{\mathcal{P}} = \sum_{w\in L(\mathcal{D})}\mathcal{P}(w)$$

[0075] PFA 모델 교차부(150)에서 PFA 모델(P)과 교차된 DFA 모델(D)은 문자열(w)이 나타날 확률을 계산하는 모델이다. 따라서 문자열(w)의 증분(예를 들면, wa)이 나타날 확률을 계산하지 못한다.

[0076] 이에 증분 확률 계산부(160)는 문자열(w)의 증분이 나타날 확률을 계산할 수 있도록 증분에 대응하는 PFA 모델(P)과 교차된 DFA 모델(D)을 획득하고, 획득된 증분에 대한 PFA 모델(P)과 교차된 DFA 모델(D)을 이용하여, 문자열(w)과 그 증분 문자열이 출연할 확률을 계산한다.

[0077] 본 실시예에서는 문자열(w)이 1회만 발생하는 문자열의 집합인 문자열 언어(F(w))를 정의한다. 여기서 문자열 언어(F(w))는 w가 접미사로만 1회 나타나는 문자열의 집합을 의미한다. 따라서 문자열 집합(F(w)○Σ^*)은 문자열(w)이 포함되는 모든 문자열의 집합이며, 문자열 언어(F(w))에 대한 명료한 정규 표현식이 주어지면, 문자열 집합(Σ^*)과 결합하여, 문자열(w)이 포함되는 모든 문자열에 대한 명료한 정규 표현식을 생성할 수 있다.

[0078] 문자열 언어(F(w))에 대한 출현 확률은 PFA 모델 교차부(150)에서 획득된 PFA 모델(P)과 교차된 DFA 모델(D)을 이용하여 계산 될 수 있다.

[0079] 한편, 문자열(w)에 증분을 야기하는 문자(a ∈ Σ)에 대해, F(wa) = F(w)○L를 만족하는 정규 언어(L)를 탐색한다. F(wa) = F(w)○L을 만족하는 정규 언어(L)는 F(w)\F(wa)로 획득될 수 있다. 여기서 \는 두 언어(R, S)가 주어질 때, $R\backslash S = \{y \mid \exists x \in R \text{ such that } xy \in S\}$를 만족하는 언어 연산자이다.

[0080] 따라서 F(w)와 F(w)\F(wa)가 주어지면, F(wa)를 획득할 수 있다. 즉 본 실시예에서는 F(wa)를 직접 획득하지 않고, 문자열 언어(F(w)) 중 문자열 언어(F(wa))에도 속하는 문자열 언어(F(w)\F(wa))를 이용하여, F(wa) = F(w)○F(w)\F(wa)를 획득한다.

[0081] 도4 는 문자열과 그 증분에 대한 정규 표현식의 관계를 나타낸다.

[0082] 도4 에서는 문자열(w)이 문자(a)이고 증분 문자가 a, ab로 확장되는 경우를 나타낸다.

[0083] 도4 을 참조하면, 문자열의 집합(Σ)의 문자가 a, b 인 경우(Σ = {a, b}) 문자열 언어(F(w))는 대한 정규 표현식의 형태 F(a) = b^*a로 표현된다. 그리고, 문자열(w)이 증분 문자(a)로 증분된 문자열 언어(F(wa))는 정규 표현식(F(aa) = b^*a○$(bb^*a)^*a$)로 표현될 수 있다. 여기서 b^*a는 F(a)이며, $(bb^*a)^*a$는 F(a)\F(aa)이다. 유사하게 증분 문자(b)가 추가로 증분된 문자열 언어(F(wab))는 F(aa)와 F(aa)\F(aab)에 의해 정규 표현식(F(aab) = b^*a○$(bb^*a)^*a$○a^*b)로 표현될 수 있다.

[0084] 이는 이전 획득된 문자열 언어(F(w))를 기반으로 추가로 증분된 문자열 언어(F(wa), F(wab))에 대응하는 F(aa)\F(aab)에 대한 명료한 정규 표현식을 표현하는 PFA 모델(P)과 교차된 DFA 모델(D), 즉 가중치가 적용된 DFA 모델(D)을 획득하면, 증분된 문자열 언어(F(wa), F(wab))에 대한 가중치가 적용된 DFA 모델(D)을 용이하게 획득할 수 있음을 의미한다.

[0085] 문자열 언어(F(w = w_1, w_2, ..., w_n))에 대한 최종 상태(q_{n+1})를 포함한 n+1개의 상태를 갖도록 획득한 DFA 모델(D)로부터 명료한 정규 표현식은 $\alpha_{0,n}^{n-1}$ 로 추출될 수 있으며, 유사하게 문자열 언어(F(w = w_1, w_2, ..., w_k))에 대한 명료한 정규 표현식은 $\alpha_{0,k+1}^{k}$로 추출될 수 있다.

[0086] DFA 모델(D)에서 상태 제거 절차의 단계(k)에서 초기 상태(q_0)는 k-1까지의 상태들에만 연결되므로, $\alpha_{0,k+1}^{k-1} = \emptyset$이다. 따라서 수학식 14를 이용하여, DFA 모델(D)의 상태들(q = q_1, q_2, ..., q_n ∈ Q)은 동적으로 반복하여 제거될 수 있다.

수학식 14

[0087] $$\alpha_{0,k+1}^{k} = \alpha_{0,k+1}^{k-1} + \alpha_{0,k}^{k-1}(\alpha_{k,k}^{k-1})^{*}\alpha_{k,k+1}^{k-1}$$

[0088] 한편, 수학식 14 는 수학식 15와 같이 단순하게 표현될 수 있다.

수학식 15

[0089] $$\alpha_{0,k+1}^{k} = \alpha_{0,k}^{k-1}(\alpha_{k,k}^{k-1})^{*}\alpha_{k,k+1}^{k-1}$$

[0090] 수학식 15 에서 $\alpha_{0,k}^{k-1} = \mathcal{F}(w_1 w_2 \cdots w_{k-1})$ 이며, 따라서 수학식 15는 수학식 16과 같이 표현될 수 있다.

수학식 16

[0091] $$\mathcal{F}(w_1 \cdots w_k) = \mathcal{F}(w_1 \cdots w_{k-1})(\alpha_{k,k}^{k-1})^{*}\alpha_{k,k+1}^{k-1}$$

[0092] 그리고 수학식 16으로부터 $F(w = w_1, w_2, \ldots, w_{k-1}) \setminus F(w = w_1, w_2, \ldots, w_k)$는 수학식 17로 계산될 수 있다.

수학식 17

[0093] $$(\alpha_{k,k}^{k-1})^{*}\alpha_{k,k+1}^{k-1} = \mathcal{F}(w_1 \cdots w_{k-1}) \setminus \mathcal{F}(w_1 \cdots w_k)$$

[0094] 수학식 17에 따라 F(aa)\F(aab)에 대한 명료한 정규 표현식을 표현하는 을 획득할 수 있으며, 이에 대응하는 상태 및 전이 함수를 가중치가 적용된 DFA 모델(D)에 추가함으로써, 가중치가 적용된 DFA 모델(D)이 증분된 문자열 집합(F(wa) = F(w)∘F(w)\F(wa))에 대응하도록 변형할 수 있다. 그리고 증분된 문자열(wa)에 대응하여 변형된 가중치가 적용된 DFA 모델(D)은 증분된 문자열(wa)에 대한 출현 확률을 계산하여 출력할 수 있다.

[0095] 도5 는 도1 의 증분 확률 계산부가 DFA 모델로부터 문자열와 그 증분 문자열이 출연할 확률을 누적하여 계산하는 알고리즘을 나타낸다.

[0096] 도5 의 알고리즘에서 DFA 모델(D)에 (n+3) × (n+3) 테이블(T, T')을 생성하는 것은 상기한 바와 같이, DFA 모델(D)이 문자열 언어($F(w = w_1, w_2, \ldots, w_n)$)에 대한 최종 상태($q_{n+1}$)를 포함한 n+1개의 상태를 갖도록 획득되었기 때문이다.

[0097] 그리고 벡터($\mathbb{V}$)는 이전 문자열로부터 기록된 결과이고, 초기값은 $\mathbb{IM}(\mathcal{F}(\lambda)) = \mathbb{I}$로 정해진다. 문자열($w = w_1, w_2, \ldots, w_k$)에 대한 확률은 $\mathbb{V}\mathbb{M}_{\mathcal{P}}(\Sigma^{*})\mathbb{F}_{\mathcal{P}}$로 획득되며, 문자열($w = w_1, w_2, \ldots, w_k$)은 도5 에 도시된 바와 같이, 증분될 수 있다.

[0098] 도6 은 문자열의 길이와 상태 집합의 크기에 따른 증분 및 교차 방식의 적용에 따른 연산 소요 시간 측정 결과를 나타낸다.

[0099] 기존의 방식에서는 증분되는 모든 문자열에 대해 각각 출현 확률을 계산해야하므로 증분되는 모든 문자열의 개수에 따라 계산 시간이 기하급수적으로 증가되는 반면, 본 실시에에서는 증분되는 문자의 개수가 증가될 수록 상대적으로 문장열의 출현 확률을 계산하는 속도가 더욱 저감된다.

[0100] 도6 에 도시된 예에서 본 실시예에 따른 오토마타 기반 증분적 중위 확률 계산 장치를 이용하는 경우, 상태들의

집합(Q)의 크기(|Q|)가 1455이고, 문자열의 길이가 9까지 증분될 때, 최대 560.76%의 계산 속도 향상을 획득할 수 있음이 확인되었다.

[0101] 도7 은 본 발명의 일 실시예에 따른 오토마타 기반 증분적 중위 확률 계산 방법을 나타낸다.

[0102] 도1 을 참조하여, 도7 의 오토마타 기반 증분적 중위 확률 계산 방법을 설명하면, 우선 출현 확률을 계산할 문자열(w)을 획득한다(S10). 그리고, 획득된 문자열(w)의 각 문자에 대하여 DFA를 기반으로 DFA 모델(D)을 모델링하여 획득한다(S20).

[0103] DFA 모델(D)은 획득된 문자열(w)에 대응하는 정규 언어(L(D))를 획득하기 위해 모델링되며, 정규 언어(L(D))를 정규 표현식의 형태로 획득할 수 있다.

[0104] DFA 모델(D)이 획득되면, DFA 모델(D)의 상태 집합(Q)에 새로운 초기 상태(q_0)와 새로운 단일 최종 상태(q_{n+1})를 추가하고 이에 대응하는 전이 함수(δ)를 추가하고, 상태 및 전이가 추가된 DFA 모델(D)에서 상태들($q = q_1, q_2, \ldots, q_n \in Q$)을 수학식 1에 따라 동적으로 반복하여 제거함으로써, DFA 모델(D)이 명료한 정규 표현식을 획득할 수 있도록 변환한다(S30).

[0105] DFA 모델(D)에 상태 및 전이가 추가되어 변형되면, 정규 표현식의 각 문자를 기지정된 행렬로 맵핑하여 행렬 형태의 정규 표현 행렬로 변환한다(S40).

[0106] 그리고 PFA에 의해 모델링된 PFA 모델(P)을 기반으로 정규 표현 행렬로 변환된 DFA 모델(D)의 각 상태(즉 문자) 및 전이 경로의 가중치(P(w), P(π))를 계산하여, 변환된 DFA 모델(D)에 적용하여, PFA 모델(P)과 교차된 DFA 모델(D)을 획득한다(S50).

[0107] PFA 모델(P)과 교차된 DFA 모델(D)은 문자열(w)에 대한 출현 확률을 계산할 수 있으나, 증분된 문자열에 대한 출현 확률을 계산할 수는 없으므로, 문자열(w)이 접미사로 1회만 출현하는 문자열 집합(F(w))에 대한 출현 확률과 문자열 집합(F(w))에서 증분된 문자열(wa)이 1회만 접미사로 출현하는 문자열 집합(F(wa))을 차감한 F(w)\F(wa)가 출현할 확률을 계산할 수 있도록 PFA 모델(P)과 교차된 DFA 모델(D)에 추가 상태 및 전이 함수를 누적하여 적용하고, 추가 상태 및 전이 함수를 누적하여 적용된 DFA 모델(D)을 이용하여 F(wa) = F(w)○F(w)\F(wa)에 따라 증분된 문자열 집합(F(wa))에 대한 출현 확률을 계산한다(S60).

[0108] 본 발명에 따른 방법은 컴퓨터에서 실행 시키기 위한 매체에 저장된 컴퓨터 프로그램으로 구현될 수 있다. 여기서 컴퓨터 판독가능 매체는 컴퓨터에 의해 액세스 될 수 있는 임의의 가용 매체일 수 있고, 또한 컴퓨터 저장 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함하며, ROM(판독 전용 메모리), RAM(랜덤 액세스 메모리), CD(컴팩트 디스크)-ROM, DVD(디지털 비디오 디스크)-ROM, 자기 테이프, 플로피 디스크, 광데이터 저장장치 등을 포함할 수 있다.

[0109] 본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다.

[0110] 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

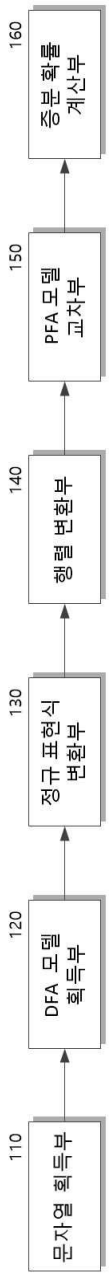
[0111] 110: 문자열 획득부 120: DFA 모델 획득부

130: 정규 표현식 변환부 140: 행렬 변환부

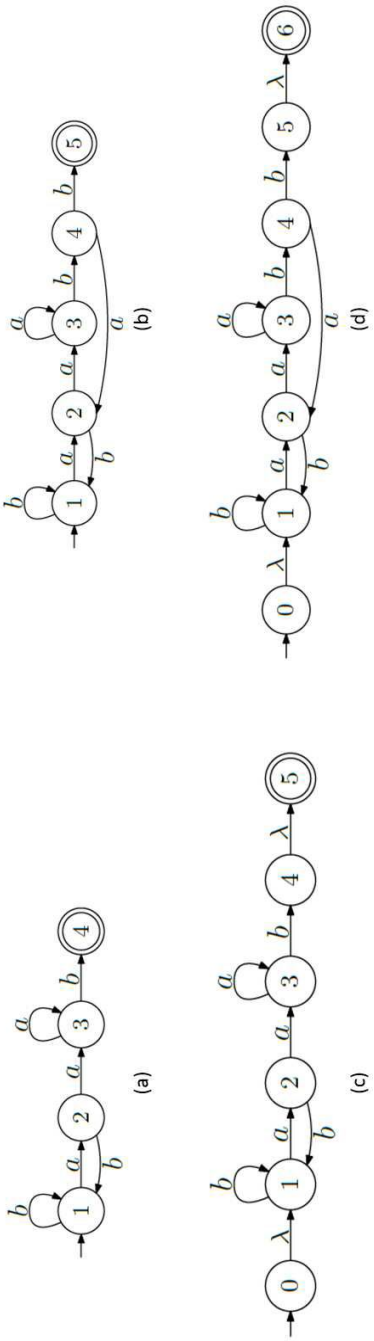
150: PFA 모델 교차부 160: 증분 확률 계산부

도면

도면1


110
문자열 획득부
120
DFA 모델 획득부
130
정규 표현식 변환부
140
행렬 변환부
150
PFA 모델 교차부
160
증분 확률 계산부

도면2



도면3

Algorithm 1 DFA/PFA intersection

procedure INTERSECT(DFA $\mathcal{D}$, PFA $\mathcal{P}$)
  $Q' = Q_{\mathcal{D}} \times Q_{\mathcal{P}}$
  for $(d, p) \in Q'$ **do**
    if d is q_0 **then**
      $I'((d, p)) = I(p)$
    else
      $I'((d, p)) = 0$
    end if
    if $d \in F_{\mathcal{D}}$ **then**
      $F'((d, p)) = F(p)$
    else
      $F'((d, p)) = 0$
    end if
    for $c \in \Sigma$, $(d', p') \in Q'$ **do**
      if $\delta_{\mathcal{D}}(d, c) = d'$ **then**
        $\delta'((d, p), c, (d', p')) = \delta_{\mathcal{P}}(p, c, p')$
      else
        $\delta'((d, p), c, (d', p')) = 0$
      end if
    end for
  end for
  return $[\mathcal{D} \cap \mathcal{P}] = (Q', \Sigma, \delta', I', F')$
end procedure

도면4

$$\mathcal{F}(a) = b^*a$$

$$\mathcal{F}(aa) = \underbrace{b^*a}_{\mathcal{F}(a)} \cdot \underbrace{(bb^*a)^*a}_{\mathcal{F}(a)\setminus\mathcal{F}(aa)}$$

$$\mathcal{F}(aab) = \underbrace{b^*a(bb^*a)^*a}_{\mathcal{F}(aa)} \cdot \underbrace{a^*b}_{\mathcal{F}(aa)\setminus\mathcal{F}(aab)}$$

도면5

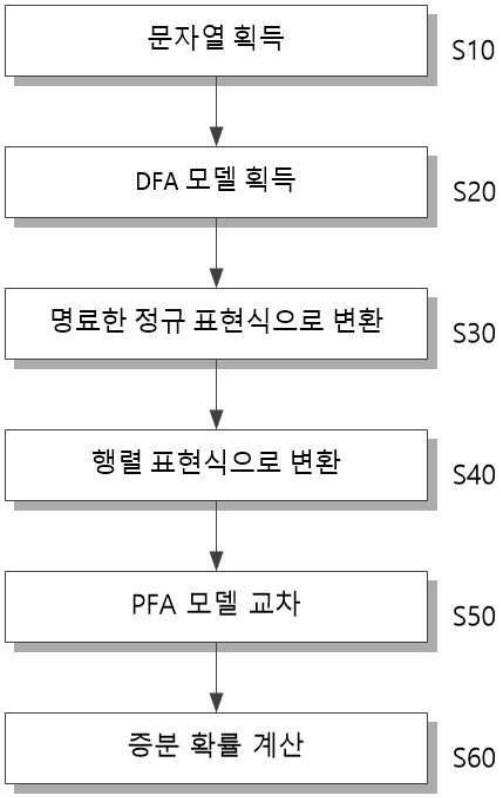
Algorithm 2 Offline Incremental Infix

procedure INFIX($w = w_1 w_2 \cdots w_n \in \Sigma^*$)
  $\mathcal{D} \leftarrow$ DFA accepting $\mathcal{F}(w)$
  $T \leftarrow (n+3) \times (n+3)$ table
  $T_{0,1} \leftarrow \mathbf{1}$
  $T_{n+1,n+2} \leftarrow \mathbf{1}$
  for $i \in [1, n+2]; j \in [1, n+2]; c \in \Sigma$ **do**
    if $\delta(q_i, c) = q_j$ **then**
      $T_{i,j} \leftarrow T_{i,j} + \mathbb{M}(c)$
    end if
  end for
  $\mathbb{V} \leftarrow \mathbb{I}$
  for $k \in [0, n+1]$ **do**
    $\mathbb{V} \leftarrow \mathbb{V}(T_{k,k})^* T_{k,k+1}$
    yield $\mathbb{V}\mathbb{M}(\Sigma^*)\mathbb{F}$
    $T' \leftarrow (n+3) \times (n+3)$ table
    for $i \in [0, n+2]; j \in [0, n+2]$ **do**
      $T'_{i,j} \leftarrow T_{i,j} + T_{i,k}(T_{k,k})^* T_{k,j}$
    end for
    $T \leftarrow T'$
  end for
end procedure

도면6

States	$\|Q\| = 614$		$\|Q\| = 1028$		$\|Q\| = 1455$	
Infix Length	Incremental	Intersection	Incremental	Intersection	Incremental	Intersection
1	0.226	0.147	0.857	0.468	2.383	1.079
2	0.272	0.316	1.072	1.235	3.000	3.112
3	0.334	0.637	1.327	2.634	3.693	6.997
4	0.399	1.133	1.586	4.864	4.442	13.250
5	0.465	1.934	1.855	8.104	5.124	22.357
6	0.527	3.375	2.088	12.562	5.815	35.065
7	0.584	4.129	2.347	18.414	6.593	51.709
8	0.649	5.791	2.591	25.614	7.224	72.512
9	0.711	7.879	2.851	34.959	7.950	99.347
Total	4.169	25.342	16.574	108.853	46.224	305.428

도면7


문자열 획득
S10
DFA 모델 획득
S20
명료한 정규 표현식으로 변환
S30
행렬 표현식으로 변환
S40
PFA 모델 교차
S50
증분 확률 계산
S60