(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년07월04일
(11) 등록번호 10-2682299
(24) 등록일자 2024년07월02일

(51) 국제특허분류(Int. Cl.)
G06F 16/901 (2019.01) *G06F 16/2455* (2019.01)
G06F 16/903 (2019.01) *G06F 16/9035* (2019.01)
(52) CPC특허분류
G06F 16/9024 (2019.01)
G06F 16/24568 (2019.01)
(21) 출원번호 10-2021-0125933
(22) 출원일자 2021년09월23일
심사청구일자 2021년09월23일
(65) 공개번호 10-2023-0042986
(43) 공개일자 2023년03월30일
(56) 선행기술조사문헌
KR101222486 B1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
한요섭
서울특별시 은평구 진관1로 77-8(진관동, 은평뉴타운 폭포동)
한중혁
서울특별시 마포구 토정로18길 11, 102동 1604호 (현석동, 래미안웰스트림)
성시철
강원도 춘천시 우석로101번길 86, 106동 1205호(석사동, 석사대우아파트)
(74) 대리인
특허법인우인

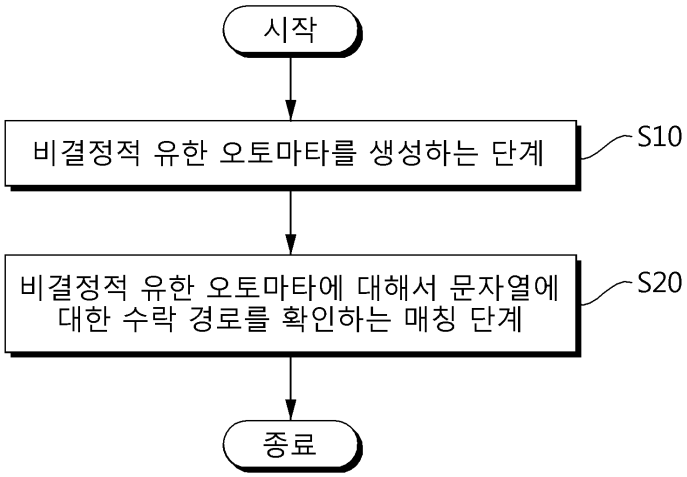
전체 청구항 수 : 총 8 항 심사관 : 안지현

(54) 발명의 명칭 **글루시코프 오토마타 생성과 하이브리드 매칭을 활용한 정규 표현식 엔진에 관한 오토마타 처리 장치 및 방법**

(57) 요 약

본 실시예들은 정규식 패턴을 특정 유형의 비결정적 유한 오토마타(Nondeterministic Finite Automata, NFA)로 변환하고, 비결정적 유한 오토마타에 대해서 확장 문법 포함 여부에 따라 매칭 알고리즘을 선택적으로 적용하여, 시간적 공간적 자원 사용을 최소화하고 ReDoS(Regular expression Denial of Service)를 방지할 수 있는 오토마타 처리 장치 및 방법을 제공한다.

대 표 도 - 도11



(52) CPC특허분류
G06F 16/90344 (2019.01)
G06F 16/9035 (2019.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711126002
과제번호 2018-0-00276-004
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송연구개발사업
연구과제명 딥러닝 기반 악성코드 패턴 룰셋 생성 자동화 원천 기술 개발 (4/5)
기 여 율 1/2
과제수행기관명 연세대학교 산학협력단
연구기간 2021.01.01 ~ 2021.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711126082
과제번호 2020-0-01361-002
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송연구개발사업
연구과제명 인공지능대학원지원사업[1단계] (2/5)
기 여 율 1/2
과제수행기관명 연세대학교 산학협력단
연구기간 2021.01.01 ~ 2021.12.31
공지예외적용 : 있음

명 세 서

청구범위

청구항 1

오토마타 처리 장치에 의한 오토마타 처리 방법에 있어서,

정규식 패턴을 각 노드가 하나의 문자에 대응하도록 변환하여 글루시코프 구조(Glushkov construction)에 따라 글루시코프 오토마타를 생성하는 단계;

상기 글루시코프 오토마타에 대해서 문자열에 대한 수락 경로를 확인하여 상기 정규식 패턴과 상기 문자열의 일치 여부를 확인하는 매칭 단계를 포함하고,

상기 정규식 패턴은 정규 표현식 또는 확장된 정규 표현식으로 표현되며,

상기 매칭 단계는,

상기 정규식 패턴이 상기 확장된 정규 표현식에 해당하는지 여부에 따라 제1 매칭 알고리즘 또는 제2 매칭 알고리즘을 선택적으로 적용하는 것을 특징으로 하는 오토마타 처리 방법.

청구항 2

삭제

청구항 3

삭제

청구항 4

제1항에 있어서,

상기 확장된 정규 표현식은 캡처 그룹, 역참조, 전방 탐색, 또는 이들의 조합을 포함하는 확장 문법이 적용된 것을 특징으로 하는 오토마타 처리 방법.

청구항 5

삭제

청구항 6

제4항에 있어서,

상기 매칭 단계는,

상기 정규식 패턴이 상기 확장 문법을 포함하면,

시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 여러 다음 상태 중 하나를 선택해 경로를 탐색하고, 선택하지 않은 상태는 문자열 상의 위치와 함께 따로 저장하고, 먼저 선택한 상태에서 진행한 경로 중 수락 경로가 있으면 매칭을 종료하고, 수락 경로를 찾지 못할 경우에 가장 최근에 저장된 상태와 위치를 바탕으로 새로운 경로를 탐색하는 상기 제1 매칭 알고리즘을 적용하는 것을 특징으로 하는 오토마타 처리 방법.

청구항 7

제4항에 있어서,

상기 매칭 단계는,

상기 정규식 패턴이 상기 확장 문법을 포함하지 않으면,

시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 모든 다음 상태를 동시에 고려하고, 모든 문자를 소비

한 시점에 현재 상태가 수락 상태를 포함하는 경우에 수락 경로가 존재한다고 판단하는 상기 제2 매칭 알고리즘을 적용하는 것을 특징으로 하는 오토마타 처리 방법.

청구항 8

프로세서 및 상기 프로세서에 의해 실행되는 프로그램을 저장하는 메모리를 포함하는 오토마타 처리 장치에 있어서,

상기 프로세서는,

정규식 패턴을 각 노드가 하나의 문자에 대응하도록 변환하여 글루시코프 구조(Glushkov construction)에 따라 글루시코프 오토마타를 생성하고,

상기 글루시코프 오토마타에 대해서 문자열에 대한 수락 경로를 확인하여 상기 정규식 패턴과 상기 문자열의 일치 여부를 확인하는 매칭을 수행하고,

상기 정규식 패턴은 정규 표현식 또는 확장된 정규 표현식으로 표현되며,

상기 프로세서는,

상기 정규식 패턴이 상기 확장된 정규 표현식에 해당하는지 여부에 따라 제1 매칭 알고리즘 또는 제2 매칭 알고리즘을 선택적으로 적용하여 상기 매칭을 수행하는 것을 특징으로 하는 오토마타 처리 장치.

청구항 9

삭제

청구항 10

삭제

청구항 11

제8항에 있어서,

상기 확장된 정규 표현식은 캡처 그룹, 역참조, 전방 탐색, 또는 이들의 조합을 포함하는 확장 문법이 적용된 것을 특징으로 하는 오토마타 처리 장치.

청구항 12

삭제

청구항 13

제11항에 있어서,

상기 프로세서는,

상기 정규식 패턴이 상기 확장 문법을 포함하면,

시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 여러 다음 상태 중 하나를 선택해 경로를 탐색하고, 선택하지 않은 상태는 문자열 상의 위치와 함께 따로 저장하고, 먼저 선택한 상태에서 진행한 경로 중 수락 경로가 있으면 매칭을 종료하고, 수락 경로를 찾지 못할 경우에 가장 최근에 저장된 상태와 위치를 바탕으로 새로운 경로를 탐색하는 상기 제1 매칭 알고리즘을 적용하는 것을 특징으로 하는 오토마타 처리 장치.

청구항 14

제11항에 있어서,

상기 프로세서는,

상기 정규식 패턴이 상기 확장 문법을 포함하지 않으면,

시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 모든 다음 상태를 동시에 고려하고, 모든 문자를 소비

한 시점에 현재 상태가 수락 상태를 포함하는 경우에 수락 경로가 존재한다고 판단하는 상기 제2 매칭 알고리즘을 적용하는 것을 특징으로 하는 오토마타 처리 장치.

발명의 설명

기 술 분 야

[0001] 본 발명이 속하는 기술 분야는 비결정적 유한 오토마타 처리 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 이 부분에 기술된 내용은 단순히 본 실시예에 대한 배경 정보를 제공할 뿐 종래기술을 구성하는 것은 아니다.

[0003] 정규 표현식(regular expression)은 특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용하는 형식 언어이다. 컴퓨터를 비롯한 연산 장치에서 문자열을 비교하거나 검색할 때 찾고자 하는 문자열을 표현하는 용도로 많이 사용된다.

[0004] 정규 표현식은 아무 내용도 없는 문자열을 의미하는 ε(엡실론)과, 한 문자로만 이루어진 정규 표현식(예를 들면, a, b, c 등)을 기본으로 하며, 이어 붙이기(abc, bbbb, baba 등), 선택(ab|c, ab|ba 등), 반복(c* 등)과 같은 연산자를 이용하여 기본적인 정규 표현식을 조합하여 다양한 패턴의 문자열을 나타낼 수 있다.

[0005] 정규 표현식이 너무 길어지거나 복잡해지는 경우가 발생할 수 있기 때문에, 사용상의 편의를 위해 다양한 확장 문법을 덧붙인 형태의 정규 표현식도 존재한다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) WO 2012-133976 (2012.10.04)

(특허문헌 0002) US 9563399 (2017.02.07)

(특허문헌 0003) KR 10-1222486 (2013.01.16)

(특허문헌 0004) KR 10-1645890 (2016.08.05)

발명의 내용

해결하려는 과제

[0007] 본 발명의 실시예들은 정규식 패턴을 특정 유형의 비결정적 유한 오토마타(Nondeterministic Finite Automata, NFA)로 변환하고, 비결정적 유한 오토마타에 대해서 확장 문법 포함 여부에 따라 매칭 알고리즘을 선택적으로 적용하여, 시간적 공간적 자원 사용을 최소화하고 ReDoS(Regular expression Denial of Service) 공격에 강인한 정규 표현식 엔진을 제공하는데 주된 목적이 있다.

[0008] 본 발명의 명시되지 않은 또 다른 목적들은 하기의 상세한 설명 및 그 효과로부터 용이하게 추론할 수 있는 범위 내에서 추가적으로 고려될 수 있다.

과제의 해결 수단

[0009] 본 실시예의 일 측면에 의하면 오토마타 처리 장치에 의한 오토마타 처리 방법에 있어서, 정규식 패턴을 기반으로 특정 유형의 비결정적 유한 오토마타를 생성하는 단계; 상기 비결정적 유한 오토마타에 대해서 문자열에 대한 수락 경로를 확인하는 매칭 단계를 포함하는 오토마타 처리 방법을 제공한다.

[0010] 상기 비결정적 유한 오토마타를 생성하는 단계는, 각 노드가 하나의 문자에 대응하도록 변환할 수 있다.

[0011] 상기 비결정적 유한 오토마타를 생성하는 단계는, 상기 정규식 패턴을 글루시코프 구조(Glushkov construction)에 따라 글루시코프 오토마타로 변환할 수 있다.

[0012] 상기 정규식 패턴은 정규 표현식 또는 확장된 정규 표현식으로 표현되며, 상기 확장된 정규 표현식은 캡처

그룹, 역참조, 전방 탐색, 또는 이들의 조합을 포함하는 확장 문법이 적용될 수 있다.

[0013] 상기 매칭 단계는, 상기 정규식 패턴이 상기 확장된 정규 표현식에 해당 여부에 따라 제1 매칭 알고리즘 또는 제2 매칭 알고리즘을 선택적으로 적용할 수 있다.

[0014] 상기 매칭 단계는, 상기 정규식 패턴이 상기 확장 문법을 포함하면, 시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 여러 다음 상태 중 하나를 선택해 경로를 탐색하고, 선택하지 않은 상태는 문자열 상의 위치와 함께 따로 저장하고, 먼저 선택한 상태에서 진행한 경로 중 수락 경로가 있으면 매칭을 종료하고, 수락 경로를 찾지 못할 경우에 가장 최근에 저장된 상태와 위치를 바탕으로 새로운 경로를 탐색하는 상기 제1 매칭 알고리즘을 적용할 수 있다.

[0015] 상기 매칭 단계는, 상기 정규식 패턴이 상기 확장 문법을 포함하지 않으면, 시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 모든 다음 상태를 동시에 고려하고, 모든 문자를 소비한 시점에 현재 상태가 수락 상태를 포함하는 경우에 수락 경로가 존재한다고 판단하는 상기 제2 매칭 알고리즘을 적용할 수 있다.

[0016] 본 실시예의 다른 측면에 의하면 프로세서 및 상기 프로세서에 의해 실행되는 프로그램을 저장하는 메모리를 포함하는 오토마타 처리 장치에 있어서, 상기 프로세서는, 정규식 패턴을 기반으로 특정 유형의 비결정적 유한 오토마타를 생성하고, 상기 비결정적 유한 오토마타에 대해서 문자열에 대한 수락 경로를 확인하는 매칭을 수행하는 것을 특징으로 하는 오토마타 처리 장치를 제공한다.

[0017] 상기 프로세서는, 각 노드가 하나의 문자에 대응하도록 변환하여 상기 비결정적 유한 오토마타를 생성할 수 있다.

[0018] 상기 프로세서는, 상기 정규식 패턴을 글루시코프 구조(Glushkov construction)에 따라 글루시코프 오토마타로 변환하여 상기 비결정적 유한 오토마타를 생성할 수 있다.

[0019] 상기 정규식 패턴은 정규 표현식 또는 확장된 정규 표현식으로 표현되며, 상기 확장된 정규 표현식은 캡처 그룹, 역참조, 전방 탐색, 또는 이들의 조합을 포함하는 확장 문법이 적용될 수 있다.

[0020] 상기 프로세서는, 상기 정규식 패턴이 상기 확장된 정규 표현식에 해당 여부에 따라 제1 매칭 알고리즘 또는 제2 매칭 알고리즘을 선택적으로 적용하여 매칭을 수행할 수 있다.

[0021] 상기 프로세서는, 상기 정규식 패턴이 상기 확장 문법을 포함하면, 시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 여러 다음 상태 중 하나를 선택해 경로를 탐색하고, 선택하지 않은 상태는 문자열 상의 위치와 함께 따로 저장하고, 먼저 선택한 상태에서 진행한 경로 중 수락 경로가 있으면 매칭을 종료하고, 수락 경로를 찾지 못할 경우에 가장 최근에 저장된 상태와 위치를 바탕으로 새로운 경로를 탐색하는 상기 제1 매칭 알고리즘을 적용할 수 있다.

[0022] 상기 프로세서는, 상기 정규식 패턴이 상기 확장 문법을 포함하지 않으면, 시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 모든 다음 상태를 동시에 고려하고, 모든 문자를 소비한 시점에 현재 상태가 수락 상태를 포함하는 경우에 수락 경로가 존재한다고 판단하는 상기 제2 매칭 알고리즘을 적용할 수 있다.

발명의 효과

[0023] 이상에서 설명한 바와 같이 본 발명의 실시예들에 의하면, 정규식 패턴을 특정 유형의 비결정적 유한 오토마타(Nondeterministic Finite Automata, NFA)로 변환하고, 비결정적 유한 오토마타에 대해서 확장 문법 포함 여부에 따라 매칭 알고리즘을 선택적으로 적용하여, 시간적 공간적 자원 사용을 최소화하고 ReDoS(Regular expression Denial of Service)를 방지할 수 있는 효과가 있다.

[0024] 여기에서 명시적으로 언급되지 않은 효과라 하더라도, 본 발명의 기술적 특징에 의해 기대되는 이하의 명세서에서 기재된 효과 및 그 잠정적인 효과는 본 발명의 명세서에 기재된 것과 같이 취급된다.

도면의 간단한 설명

[0025] 도 1은 본 발명의 일 실시예에 따른 오토마타 처리 장치를 예시한 블록도이다.

도 2는 확장 문법이 포함된 톰슨(Tohmpson) 오토마톤을 예시한 도면이다.

도 3은 본 발명의 일 실시예에 따른 오토마타 처리 장치가 처리하는 확장 문법이 포함된 글루시코프(Glushkov) 오토마톤을 예시한 도면이다.

도 4는 본 발명의 일 실시예에 따른 오토마타 처리 장치가 처리하는 확장 문법이 포함되지 않은 글루시코프(Glushkov) 오토마톤을 예시한 도면이다.

도 5a 내지 도 5c는 도 2의 톰슨 오토마톤에 스펜서(Spencer) 알고리즘을 적용한 결과를 트리 형태로 나타낸 도면이다.

도 6은 도 3의 글루시코프 오토마톤에 스펜서(Spencer) 알고리즘을 적용하여 문자열 일치를 확인하는 과정을 나타낸 도면이다.

도 7a 내지 도 7c는 도 3의 글루시코프 오토마톤에 스펜서(Spencer) 알고리즘을 적용한 결과를 트리 형태로 나타낸 도면이다.

도 8a 내지 도 8c는 도 4의 글루시코프 오토마톤에 스펜서(Spencer) 알고리즘을 적용한 결과를 트리 형태로 나타낸 도면이다.

도 9는 도 4의 글루시코프 오토마톤에 클래시컬 매칭(Classical matching) 알고리즘을 적용하여 문자열 일치를 확인하는 과정을 나타낸 도면이다.

도 10a 내지 도 10c는 도 4의 글루시코프 오토마톤에 클래시컬 매칭(Classical matching) 알고리즘을 적용한 결과를 트리 형태로 나타낸 도면이다.

도 11은 본 발명의 다른 실시예에 따른 오토마타 처리 방법을 예시한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0026] 이하, 본 발명을 설명함에 있어서 관련된 공지기능에 대하여 이 분야의 기술자에게 자명한 사항으로서 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하고, 본 발명의 일부 실시예들을 예시적인 도면을 통해 상세하게 설명한다.

[0027] 정규 표현식 엔진을 사용하는 서비스 제공자가 유해한 정규식 패턴을 사용할 경우, 엔진은 DoS(Denial of Service) 공격의 매개체로 사용될 수 있다. 이를 ReDoS(Regular expression Denial of Service)라 한다. ReDoS는 엔진이 유해한 패턴과 문자열이 일치하는지 확인하는데 필요한 시간적, 공간적 자원이 문자열의 길이에 비해 과도하게 (지수적으로) 크기 때문에 발생한다. 현존하는 많은 프로그램은 정규 표현식 엔진을 사용하며, 이로 인하여 ReDoS 공격의 위험에 노출되어 있다.

[0028] 본 명세서에서는 기존의 방식보다 더 적은 시간적, 공간적 자원을 요구하는 새로운 정규 표현식 엔진을 제안한다. 더 빠른 정규식 패턴 일치 확인을 가능하게 하고, 더욱 안정화된 프로그램을 작성할 수 있다.

[0029] 본 실시예에 따른 오토마타 처리 장치는 클래시컬 매칭(Classical matching) 알고리즘을 적용하여 ReDoS를 원천적으로 차단하고, 확장 문법이 적용된 정규 표현식에 대해서 스펜서(Spencer) 알고리즘을 사용해야 하는 경우에도 글루시코프(Glushkov) 오토마타를 통해 ReDoS를 방지할 수 있다.

[0030] 본 실시예에 따른 오토마타 처리 장치는 글루시코프(Glushkov) 오토마타에 해당하는 비결정적 유한 오토마타(Nondeterministic Finite Automata, NFA)를 생성하고, 확장 문법 포함 여부에 따라 스펜서(Spencer) 알고리즘 또는 클래시컬 매칭(Classical matching) 알고리즘을 선택적으로 적용한다.

[0031] 본 실시예에 따른 오토마타 처리 장치가 처리하는 정규식 패턴은 정규 표현식, 혹은 확장된 정규 표현식으로 표현된 문자열의 패턴을 의미한다. 정규 표현식 엔진은 정규식 패턴과 문자열이 일치하는지 확인하기 위하여 사용되며, 이는 정규식 패턴에 해당하는 비결정적 유한 오토마톤(Nondeterministic finite state automaton, NFA)을 만드는 NFA 생성 과정과 해당 NFA에 문자열에 대한 수락 경로가 있는지 확인하는 매칭 과정을 포함한다.

[0032] 오토마타 처리 장치는 NFA 생성 과정에서 정규식 패턴을 매칭에 효율적인 형태의 NFA인 Glushkov 오토마타로 변환한다. 정규식 패턴에 따라 Spencer 알고리즘과 Classical matching 알고리즘을 선택적으로 적용하는 하이브리드 매칭 과정을 수행한다. Thompson 오토마타, Spencer 알고리즘을 사용하는 종래 기술과 비교하여 빠른 시간 내로 정규식 패턴 일치를 확인할 수 있다.

[0033] 임의의 문자 σ는 정규 표현식이며, 정규 표현식 r_1, r_2에 대하여 $(r_1\ r_2)$, $(r_1|r_2)$, (r_1*) 또한 정규 표현식이다. 정규 표현식 r이 나타내는 언어 L(r)은 다음과 같이 정의된다.

[0034] (1) $L(\sigma)=\{\sigma\}$

[0035] (2) $L(r_1r_2)=L(r_1)L(r_2)$

[0036] (3) $L(r_1 | r_2)=L(r_1)\cup L(r_2)$

[0037] (4) $L(r_1*)=L(r_1)^*$

[0038] 이와 같이 정의된 정규 표현식은 실생활의 적용을 위해 캡처 그룹, 역참조와 전방 탐색이라는 개념을 활용하여 문법을 확장할 수 있다.

[0039] 정규 표현식의 사용에 따라 정규 표현식을 정규식 패턴 또는 패턴이라고 칭할 수 있다.

[0040] 캡처 그룹 $(_n)_n$과 역참조 \n는 정규 표현식의 일부에 일치된 부분 문자열을 재사용하고자 할 때 사용된다. 캡처 그룹은 그룹 내부의 정규 표현식에 일치된 부분 문자열을 저장하며, 역참조는 캡처 그룹에서 저장된 부분 문자열에 일치한다. 예를 들어 $(_1ab|ba)1\backslash 1$를 abab와 일치시키는 경우, 캡처 그룹 $(_1)_1$은 abab의 먼저 나온 ab에 ab|ba가 일치함을 확인하고 ab를 저장한다. 이후 역참조 \1은 캡처 그룹 $(_1)_1$가 저장한 ab를 참조하여 abab의 뒤쪽의 ab를 일치시킨다. 마찬가지로 해당 패턴은 abab와 baba에는 일치하지만, abba나 baab에는 역참조에서 참조하는 문자열과 실제 일치를 시도하는 문자열이 다르다. 즉, 패턴 $(_1ab|ba)_1\backslash 1$는 abba와 baab의 문자열과 일치하지 않는다.

[0041] 전방 탐색 (?=)은 이후 나올 문자열의 앞부분이 전방 탐색 내부의 패턴에 일치하는지 판단하기 위해서만 사용되고 실제 일치시키지는 않는다. 예를 들어 패턴 a(?=b)(a | b)*에서 (?=b)는 전방 탐색이며 해당 전방 탐색 내부의 패턴은 b이다. 패턴 a(?=b)(a | b)*을 aba에 일치시키는 경우, 패턴의 a와 문자열의 a를 일치시킨 이후 전방 탐색 (?=b)는 남은 문자열인 ba의 앞부분이 정규 표현식 b와 일치하는지 판단한다. 전방 탐색은 이를 확인한 후, 실제 일치시키지는 않기에 뒷부분의 정규 표현식 (a | b)*는 문자열 a가 아닌 ba와 일치를 시도한다. 이 둘이 일치하므로 패턴 a(?=b)(a | b)*전체는 전체 문자열 aba과 일치한다. 마찬가지로 해당 패턴은 aba와 abb에는 일치한다. 이와 달리 aab나 aaa와 같은 문자열은 전방 탐색 (?=b)에서 b와 일치하지 못하므로 전체 패턴 a(?=b)(a | b)*에 일치하지 못한다.

[0042] 캡처 그룹, 역참조, 전방 탐색 등을 확장 문법이라 하며, 이들을 포함한 정규 표현식을 확장된 정규 표현식이라 한다. 본 발명은 확장된 정규 표현식을 지원하며 효율적으로 문자열과 정규식 패턴의 일치를 판단하는 정규 표현식 엔진이다.

[0043] 도 1은 본 발명의 일 실시예에 따른 오토마타 처리 장치를 예시한 블록도이다. 도 2는 확장 문법이 포함된 톰슨(Tohmpson) 오토마톤을 예시한 도면이고, 도 3은 본 발명의 일 실시예에 따른 오토마타 처리 장치가 처리하는 확장 문법이 포함된 글루시코프(Glushkov) 오토마톤을 예시한 도면이고, 도 4는 본 발명의 일 실시예에 따른 오토마타 처리 장치가 처리하는 확장 문법이 포함되지 않은 글루시코프(Glushkov) 오토마톤을 예시한 도면이다.

[0044] 오토마타 처리 장치(110)는 적어도 하나의 프로세서(120), 컴퓨터 판독 가능한 저장매체(130) 및 통신 버스(170)를 포함한다.

[0045] 프로세서(120)는 오토마타 처리 장치(110)로 동작하도록 제어할 수 있다. 예컨대, 프로세서(120)는 컴퓨터 판독 가능한 저장 매체(130)에 저장된 하나 이상의 프로그램들을 실행할 수 있다. 하나 이상의 프로그램들은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 컴퓨터 실행 가능 명령어는 프로세서(120)에 의해 실행되는 경우 오토마타 처리 장치(110)로 하여금 예시적인 실시예에 따른 동작들을 수행하도록 구성될 수 있다.

[0046] 컴퓨터 판독 가능한 저장 매체(130)는 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보를 저장하도록 구성된다. 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보는 입출력 인터페이스(150)나 통신 인터페이스(160)를 통해서도 주어질 수 있다. 컴퓨터 판독 가능한 저장 매체(130)에 저장된 프로그램(140)은 프로세서(120)에 의해 실행 가능한 명령어의 집합을 포함한다. 일 실시예에서, 컴퓨터 판독 가능한 저장 매체(130)는 메모리(랜덤 액세스 메모리와 같은 휘발성 메모리, 비휘발성 메모리, 또는 이들의 적절한 조합), 하나 이상의 자기 디스크 저장 디바이스들, 광학 디스크 저장 디바이스들, 플래시 메모리 디바이스들, 그 밖에 오토마타 처리 장치(110)에 의해 액세스되고 원하는 정보를 저장할 수 있는 다른 형태의 저장 매체, 또는 이들의 적합한 조합일 수 있다.

[0047] 통신 버스(170)는 프로세서(120), 컴퓨터 판독 가능한 저장 매체(130)를 포함하여 오토마타 처리 장치(110)의

다른 다양한 컴포넌트들을 상호 연결한다.

[0048] 오토마타 처리 장치(110)는 또한 하나 이상의 입출력 장치를 위한 인터페이스를 제공하는 하나 이상의 입출력 인터페이스(150) 및 하나 이상의 통신 인터페이스(160)를 포함할 수 있다. 입출력 인터페이스(150) 및 통신 인터페이스(160)는 통신 버스(170)에 연결된다. 입출력 장치(미도시)는 입출력 인터페이스(150)를 통해 오토마타 처리 장치(110)의 다른 컴포넌트들에 연결될 수 있다.

[0049] 오토마타 처리 장치는 확장된 정규 표현식에 대한 효율적인 일치 확인을 위해 패턴에 대한 Glushkov 오토마타라는 NFA를 생성하며, 일치 확인은 Classical matching 알고리즘과 Spencer 알고리즘 중에서 주어진 정규식 패턴에 따라 효율적인 알고리즘을 사용하는 하이브리드 매칭 알고리즘이다. 핵심이 되는 정규식 패턴에 대한 NFA의 생성 과정과, 패턴과 문자열의 일치를 확인하는 과정을 수행한다.

[0050] 프로세서는 정규식 패턴을 기반으로 특정 유형의 비결정적 유한 오토마타를 생성하고, 비결정적 유한 오토마타에 대해서 문자열에 대한 수락 경로를 확인하는 매칭을 수행한다.

[0051] 프로세서는 각 노드가 하나의 문자에 대응하도록 변환하여 비결정적 유한 오토마타를 생성할 수 있다. 프로세서는 정규식 패턴을 글루시코프 구조(Glushkov construction)에 따라 글루시코프 오토마타로 변환하여 비결정적 유한 오토마타를 생성할 수 있다.

[0052] NFA 생성 과정은 정규식 패턴을 NFA로 변환한다. 정규식 패턴은 정규 표현식 또는 확장된 정규 표현식으로 표현되며, 확장된 정규 표현식은 캡처 그룹, 역참조, 전방 탐색, 또는 이들의 조합을 포함하는 확장 문법이 적용될 수 있다.

[0053] 주어진 정규식 패턴에 대해 Glushkov construction을 사용하여 NFA를 생성한다. Glushkov construction을 통해 생성된 NFA를 Glushkov 오토마타라 한다.

[0054] 도 3을 참조하면 확장된 정규 표현식 $(_1a|ab)_1(\backslash w*)*\backslash 1$에 대한 Glushkov 오토마톤을 나타내고, 도 4를 참조하면 확장 문법이 포함되지 않은 정규식 패턴 (a|\w)*b에 대한 Glushkov 오토마톤을 나타낸다. 이 때, \w는 모든 알파벳에 일치하는 특수 문자이다.

[0055] 매칭 과정은 문자열이 주어진 경우 일치 여부를 확인한다.

[0056] 정규식 패턴에 대한 문자열의 일치 확인 과정을 매칭 과정이라 하며, 이를 위해 생성된 NFA를 활용해 NFA의 시작 상태에서 해당 문자열의 문자를 차례로 모두 소비하여 수락 상태에 도달하는 경로가 존재하는지 확인한다.

[0057] 도 4에서 문자열 aab를 받았을 때, NFA는 시작 상태인 0에서 시작해 a를 읽고 상태 1로 진행한다. 그 다음 문자인 a를 읽고 다시 상태 1로, 마지막 문자인 b를 읽고 수락 상태인 상태 3으로 진행한다. 이러한 경로를 문자열에 대한 경로라 하고 상황에 따라 두 개 이상의 경로가 존재할 수 있다.

[0058] 문자열의 경로 중 수락 상태에 도달하는 경로를 수락 경로라 한다. 수락 경로가 존재한다면 정규식 패턴과 문자열이 일치하며, 그렇지 않은 경우 패턴과 문자열은 일치하지 않는다.

[0059] 본 실시예는 정규식 패턴이 확장 문법을 포함하는지 여부에 따라 다음의 두 가지 알고리즘 중 하나를 택하여 적용한다. 클래시컬 매칭(Classical matching) 알고리즘은 스펜서(Spencer) 알고리즘과 비교하여 수행 시간의 분산이 작으나, 정규 표현식 확장 문법에 대해 (e.g., 역참조, 전방탐색) 적용 불가능한 경우가 있다. 따라서 확장된 정규 표현식에 대해 스펜서(Spencer) 알고리즘을 적용한다.

[0060] 프로세서는 정규식 패턴이 확장된 정규 표현식에 해당 여부에 따라 제1 매칭 알고리즘 또는 제2 매칭 알고리즘을 선택적으로 적용하여 매칭을 수행할 수 있다. 제1 매칭 알고리즘은 스펜서 알고리즘에 대응할 수 있고, 제2 매칭 알고리즘은 클래시컬 매칭 알고리즘에 대응할 수 있다.

[0061] 프로세서는 정규식 패턴이 확장 문법을 포함하면, 시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 여러 다음 상태 중 하나를 선택해 경로를 탐색하고, 선택하지 않은 상태는 문자열 상의 위치와 함께 따로 저장하고, 먼저 선택한 상태에서 진행한 경로 중 수락 경로가 있으면 매칭을 종료하고, 수락 경로를 찾지 못할 경우에 가장 최근에 저장된 상태와 위치를 바탕으로 새로운 경로를 탐색하는 제1 매칭 알고리즘을 적용할 수 있다.

[0062] 프로세서는 정규식 패턴이 확장 문법을 포함하지 않으면, 시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 모든 다음 상태를 동시에 고려하고, 모든 문자를 소비한 시점에 현재 상태가 수락 상태를 포함하는 경우에

수락 경로가 존재한다고 판단하는 제2 매칭 알고리즘을 적용할 수 있다.

[0063] 기존의 엔진들(예컨대, JAVA, Python 등)은 톰슨(Thompson) 오토마타를 기반으로 하며, 표현식 내의 문자들과 연산자에 대하여 재귀적으로 NFA를 생성하는 방식을 적용한다. 이는 NFA의 형태가 직관적이고 구현이 간단하다는 장점을 가지나, 문자를 소비하지 않는 간선을 가지며, 이는 일치 판정을 수행하는데 비효율적인 형태이다.

[0064] 도 2를 참조하면, 도 3과 동일한 정규 표현식에 해당하는 톰슨(Thompson) 오토마톤을 나타낸다. 즉, 확장 문법이 포함된 경우의 톰슨 오토마톤을 나타낸다. 일부 노드는 가독성을 위해 생략한다.

[0065] 본 실시예는 Glushkov 오토마타를 기반으로 하며, 이는 각 노드가 하나의 문자에 대응한다. 결과적으로 Thompson 오토마타에서 나타나는 하나 이상의 노드가 Glushkov 오토마톤에서 하나의 노드로 축약된다.

[0066] 이러한 축약의 구체적인 예시는 도 2의 Thompson 오토마톤에서 직사각형으로 표시된 영역 1, 2, 3의 노드들이 도 3의 Glushkov 오토마톤에서 각각 노드 1, 6, 7로 축약되는 것을 통해 확인할 수 있다.

[0067] NFA는 특정 입력 심볼에 대응되는 다음 상태가 여러 개일 수 있다. ε는 스트링의 길이가 0임을 뜻하는 심볼에 해당하며 엡실론(epsilon)이라고 한다. ε변환은 ε을 보고 갈 수 있는 상태가 존재한다는 것을 의미한다. 입력 심볼이 들어오지 않아도 상태전이가 가능하다.

[0068] 글루시코프 구조는 e변환이 없다. 시작 상태는 내변환이 없다. 각 상태의 모든 내변환은 같은 레이블을 갖는다. 상태의 수는 정규 표현식의 기호 수보다 하나가 더 많다.

[0069] 글루시코프 구조는 정규 표현식의 형태 유형에 따라 재귀적으로 정의된 null, first, last, follow의 4가지 함수를 반복적으로 적용하여 얻을 수 있다.

[0070] A. Bruggemann-Klein, "Regular expressions into finite automata", Theoretical Computer Science, 1993.을 참조하면 글루시코프(Glushkov) 오토마타 생성에 관한 내용을 확인할 수 있다.

[0071] 도 5a 내지 도 5c는 도 2의 톰슨 오토마톤에 스펜서(Spencer) 알고리즘을 적용한 결과를 트리 형태로 나타낸 도면이다.

[0072] 패턴으로부터 NFA를 생성한 후, 기존의 엔진들은 매칭 과정에 있어 Spencer 알고리즘을 기반으로 일치를 수행할 수 있다. Spencer 알고리즘의 모든 경로를 탐색한다는 특징은 확장 문법을 지원하기 위하여 필수적이지만, 그렇지 않은 경우 여러 경로에서 공통되는 부분을 중복하여 확인하는 결과를 야기한다.

[0073] 도 5를 참조하면, 확장 문법을 포함하는 Thompson 오토마톤에 문자열 (a) ab, (b) aab, (c) aaab에 대한 Spencer 알고리즘을 수행한 결과를 각각 트리 형태로 표현한 것으로, 관찰을 통해 Thompson 오토마톤에서는 ReDoS의 원인이 되는 일치 확인 시간의 지수적 증가가 나타나는 것을 확인할 수 있다.

[0074] Spencer 알고리즘이 같은 경로를 중복하여 탐색하는 구체적 예시는 도 5a 내지 도 5c에서 T로 표기된 과정이 반복되는 것을 통해 확인할 수 있다. 즉, 확장 문법이 포함된 유해 패턴으로 인한 종래 톰슨 오토마톤과 스펜서 알고리즘에서의 ReDoS 발생을 확인할 수 있다.

[0075] 본 실시예는 확장 문법을 사용하지 않는 경우 Classical matching 알고리즘을 사용하여 이를 방지한다.

[0076] 도 6은 도 3의 글루시코프 오토마톤에 스펜서(Spencer) 알고리즘을 적용하여 문자열 일치를 확인하는 과정을 나타낸 도면이다.

[0077] 정규 표현식이 확장 문법을 포함하고 있는 경우 Spencer 알고리즘을 사용하여 매칭을 수행한다. 해당 알고리즘은 시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 여러 다음 상태 중 하나를 선택해 경로를 탐색한다. 이 때 선택하지 않은 상태는 문자열 상의 위치와 함께 따로 저장한다. 먼저 선택한 상태에서 진행한 경로 중 수락 경로가 있으면 매칭을 종료한다. 수락 경로를 찾지 못할 경우, 가장 최근에 저장된 상태와 위치를 바탕으로 새로운 경로를 탐색한다.

[0078] 도 6을 참조하면, 확장 문법을 포함하는 NFA를 바탕으로 문자열 abab와 일치를 확인하는 과정을 확인할 수 있다.

[0079] 도 7a 내지 도 7c는 도 3의 글루시코프 오토마톤에 스펜서(Spencer) 알고리즘을 적용한 결과를 트리 형태로 나타낸 도면이다.

[0080] 도 7은 확장 문법을 포함하는 Glushkov 오토마톤에서 수행한 것을 표현한 것으로, Glushkov 오토마톤에서는 일

치 확인 시간의 지수적 증가가 나타나지 않음을 확인할 수 있다. 도 2 및 도 3에 도시된 두 오토마톤 모두 확장 문법이 포함된 정규식 패턴 $(_1a|ab)_1(\backslash w*)*\backslash 1$에 해당하는 NFA이며, 종래 기술로는 유해성을 띠던 패턴이 본 실시예에선 유해성을 띠지 않는다. 즉, Glushkov 오토마톤을 통한 패턴의 유해성 해소를 확인할 수 있다.

[0081] 도 8a 내지 도 8c는 도 4의 글루시코프 오토마톤에 스펜서(Spencer) 알고리즘을 적용한 결과를 트리 형태로 나타낸 도면이다.

[0082] 도 8을 참조하면, 확장 문법을 포함하지 않는 Glushkov 오토마톤에 문자열 (a) ab, (b) aab, (c) aaab에 대한 Spencer 알고리즘을 수행한 결과를 표현한 것이며, 관찰을 통해 ReDoS의 원인이 되는 일치 확인 시간의 지수적 증가가 나타나는 것을 확인할 수 있다. 즉, 확장 문법이 포함되지 않은 유해 패턴으로 인한 ReDoS 발생을 확인할 수 있다.

[0083] 도 9는 도 4의 글루시코프 오토마톤에 클래시컬 매칭(Classical matching) 알고리즘을 적용하여 문자열 일치를 확인하는 과정을 나타낸 도면이다.

[0084] 정규 표현식이 확장 문법을 포함하고 있지 않은 경우 Classical matching 알고리즘을 사용한다. 해당 알고리즘은 시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 모든 다음 상태를 동시에 고려한다. 모든 문자를 소비한 시점에 현재 상태가 수락 상태를 포함하는 경우 수락 경로가 존재한다고 판단한다.

[0085] 도 9를 참조하면, 확장 문법을 포함하지 않는 NFA를 바탕으로 abab와 일치를 확인하는 과정을 확인할 수 있다.

[0086] 도 10a 내지 도 10c는 도 4의 글루시코프 오토마톤에 클래시컬 매칭(Classical matching) 알고리즘을 적용한 결과를 트리 형태로 나타낸 도면이다.

[0087] 확장 문법을 포함하지 않는 오토마톤과 문자열에 대하여 Classical matching 알고리즘을 수행한 결과를 나타낸 것으로, 이를 통해 Classical matching 알고리즘이 일치 확인 시간의 지수적 증가를 차단하는 것을 확인할 수 있다. 즉, Classical matching을 통한 패턴의 유해성 해소를 확인할 수 있다.

[0088] 도 11은 본 발명의 다른 실시예에 따른 오토마타 처리 방법을 예시한 흐름도이다.

[0089] 오토마타 처리 방법은 오토마타 처리 장치에 의해 수행될 수 있다.

[0090] 단계 S10에서는 정규식 패턴을 기반으로 특정 유형의 비결정적 유한 오토마타를 생성한다.

[0091] 단계 S20에서는 비결정적 유한 오토마타에 대해서 문자열에 대한 수락 경로를 확인하는 매칭을 수행한다.

[0092] 비결정적 유한 오토마타를 생성하는 단계(S10)는, 각 노드가 하나의 문자에 대응하도록 변환할 수 있다. 비결정적 유한 오토마타를 생성하는 단계(S10)는, 정규식 패턴을 글루시코프 구조(Glushkov construction)에 따라 글루시코프 오토마타로 변환할 수 있다.

[0093] 정규식 패턴은 정규 표현식 또는 확장된 정규 표현식으로 표현되며, 확장된 정규 표현식은 캡처 그룹, 역참조, 전방 탐색, 또는 이들의 조합을 포함하는 확장 문법이 적용될 수 있다.

[0094] 매칭 단계(S20)는, 정규식 패턴이 확장된 정규 표현식에 해당 여부에 따라 제1 매칭 알고리즘 또는 제2 매칭 알고리즘을 선택적으로 적용할 수 있다.

[0095] 매칭 단계(S20)는, 정규식 패턴이 확장 문법을 포함하면, 시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 여러 다음 상태 중 하나를 선택해 경로를 탐색하고, 선택하지 않은 상태는 문자열 상의 위치와 함께 따로 저장하고, 먼저 선택한 상태에서 진행한 경로 중 수락 경로가 있으면 매칭을 종료하고, 수락 경로를 찾지 못할 경우에 가장 최근에 저장된 상태와 위치를 바탕으로 새로운 경로를 탐색하는 제1 매칭 알고리즘을 적용할 수 있다.

[0096] 매칭 단계(S20)는, 정규식 패턴이 확장 문법을 포함하지 않으면, 시작 상태에서 출발하여 각 문자를 통해 이동할 수 있는 모든 다음 상태를 동시에 고려하고, 모든 문자를 소비한 시점에 현재 상태가 수락 상태를 포함하는 경우에 수락 경로가 존재한다고 판단하는 제2 매칭 알고리즘을 적용할 수 있다.

[0097] 오토마타 처리 장치는 하드웨어, 펌웨어, 소프트웨어 또는 이들의 조합에 의해 로직회로 내에서 구현될 수 있고, 범용 또는 특정 목적 컴퓨터를 이용하여 구현될 수도 있다. 장치는 고정배선형(Hardwired) 기기, 필드 프로그램 가능한 게이트 어레이(Field Programmable Gate Array, FPGA), 주문형 반도체(Application Specific Integrated Circuit, ASIC) 등을 이용하여 구현될 수 있다. 또한, 장치는 하나 이상의 프로세서 및 컨트롤러를

포함한 시스템온칩(System on Chip, SoC)으로 구현될 수 있다.

[0098] 오토마타 처리 장치는 하드웨어적 요소가 마련된 컴퓨팅 디바이스 또는 서버에 소프트웨어, 하드웨어, 또는 이들의 조합하는 형태로 탑재될 수 있다. 컴퓨팅 디바이스 또는 서버는 각종 기기 또는 유무선 통신망과 통신을 수행하기 위한 통신 모뎀 등의 통신장치, 프로그램을 실행하기 위한 데이터를 저장하는 메모리, 프로그램을 실행하여 연산 및 명령하기 위한 마이크로프로세서 등을 전부 또는 일부 포함한 다양한 장치를 의미할 수 있다.

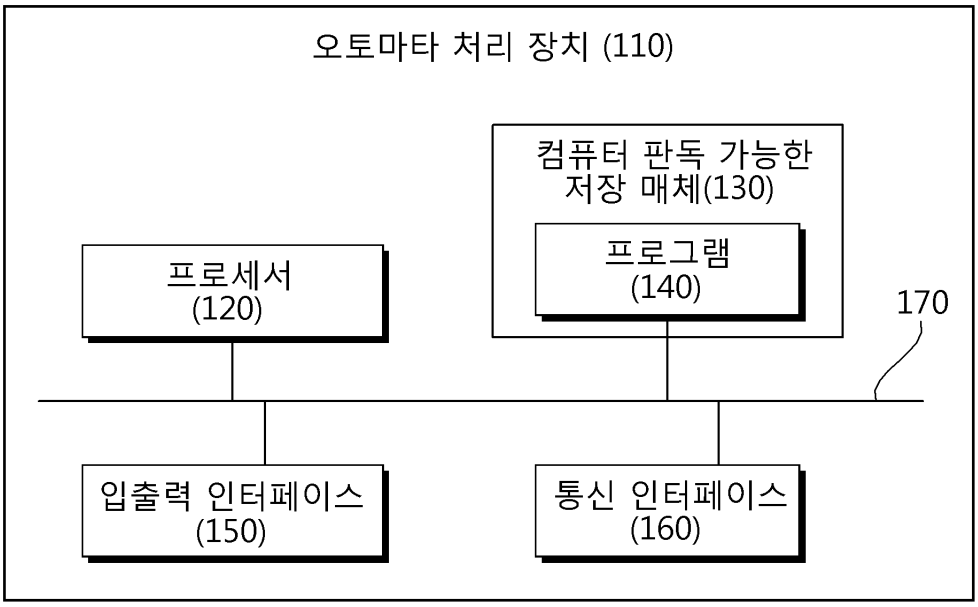
[0099] 도 11에서는 각각의 과정을 순차적으로 실행하는 것으로 기재하고 있으나 이는 예시적으로 설명한 것에 불과하고, 이 분야의 기술자라면 본 발명의 실시예의 본질적인 특성에서 벗어나지 않는 범위에서 도 11에 기재된 순서를 변경하여 실행하거나 또는 하나 이상의 과정을 병렬적으로 실행하거나 다른 과정을 추가하는 것으로 다양하게 수정 및 변형하여 적용 가능할 것이다.

[0100] 본 실시예들에 따른 동작은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능한 매체에 기록될 수 있다. 컴퓨터 판독 가능한 매체는 실행을 위해 프로세서에 명령어를 제공하는데 참여한 임의의 매체를 나타낸다. 컴퓨터 판독 가능한 매체는 프로그램 명령, 데이터 파일, 데이터 구조 또는 이들의 조합을 포함할 수 있다. 예를 들면, 자기 매체, 광기록 매체, 메모리 등이 있을 수 있다. 컴퓨터 프로그램은 네트워크로 연결된 컴퓨터 시스템 상에 분산되어 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수도 있다. 본 실시예를 구현하기 위한 기능적인(Functional) 프로그램, 코드, 및 코드 세그먼트들은 본 실시예가 속하는 기술분야의 프로그래머들에 의해 용이하게 추론될 수 있을 것이다.

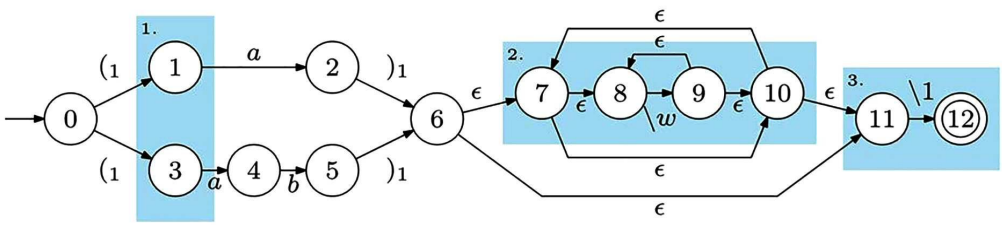
[0101] 본 실시예들은 본 실시예의 기술 사상을 설명하기 위한 것이고, 이러한 실시예에 의하여 본 실시예의 기술 사상의 범위가 한정되는 것은 아니다. 본 실시예의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 실시예의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

도면

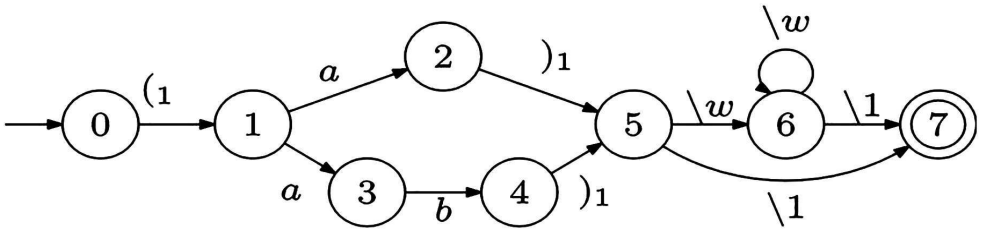
도면1



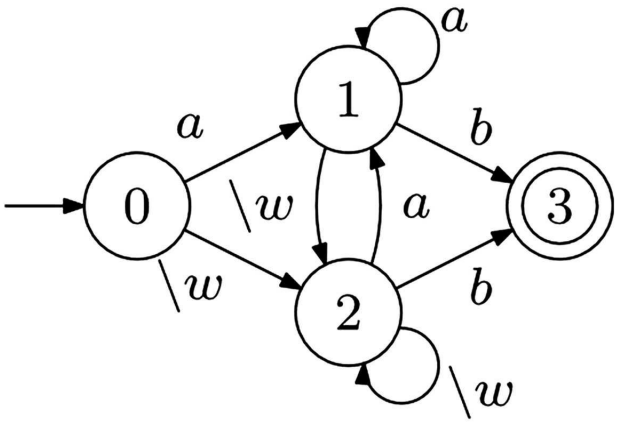
도면2



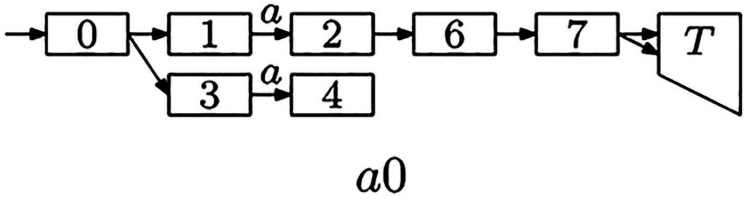
도면3



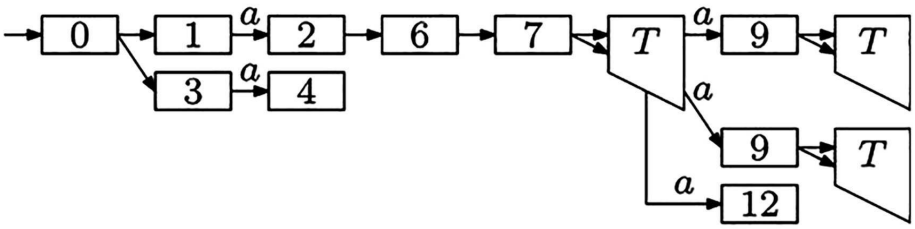
도면4



도면5a

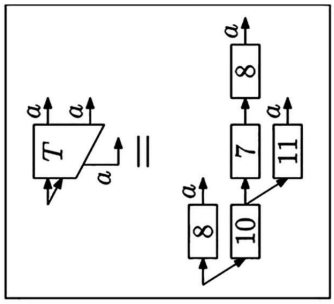


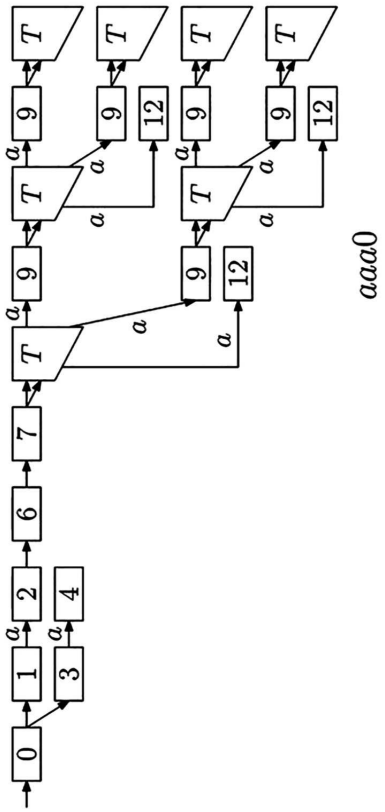
도면5b



$aa0$

도면5c



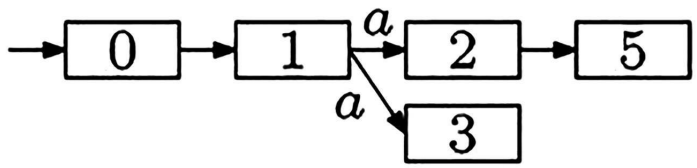


$aaa0$

도면6

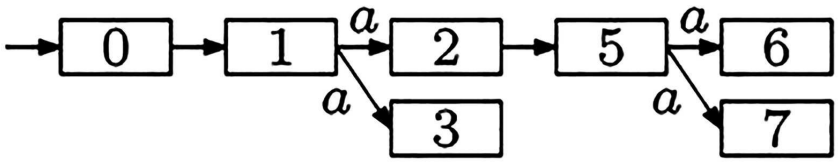
현재 상태	처리 중인 문자	사용 가능한 화살표	그룹 1	저장된 분기점
0	abab	0 → 1	-	
1	abab	1 → 2, 1 → 3	-	
2	abab	2 → 5	-	1) (abab, 1 → 3, -)
5	abab	5 → 6	A	
6	abab	6 → 7	A	
7	abab	없음	A	1)번 불러옴
3	abab	3 → 4	-	
4	abab	4 → 5	-	
5	abab	5 → 7, 그룹 1	ab	
7	abab	일치 확인	ab	

도면7a



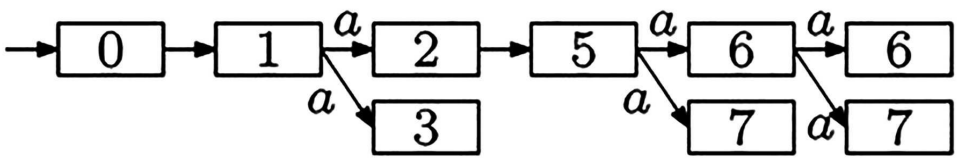
$a0$

도면7b



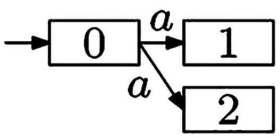
$aa0$

도면7c



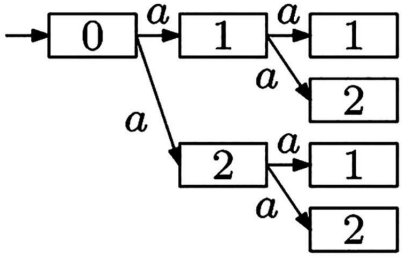
$aaa0$

도면8a



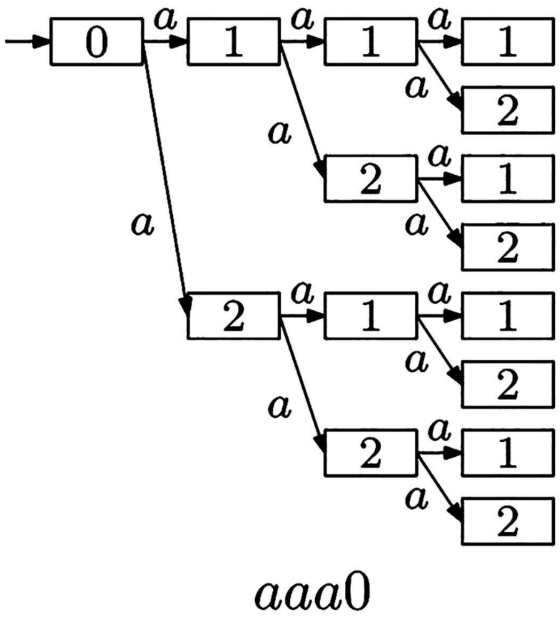
$a0$

도면8b



$aa0$

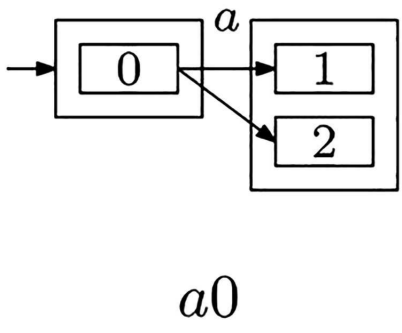
도면8c



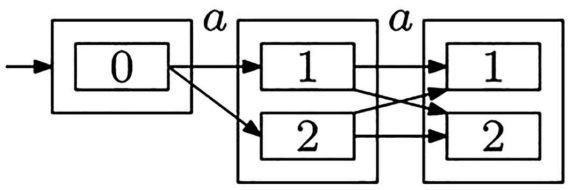
도면9

현재 상태	처리 중인 문자	사용 가능한 화살표
0	abab	0 → 1, 0 → 2
1, 2	abab	1 → 2, 1 → 3, 2 → 3
2, 3	abab	2 → 1
1	abab	1 → 2, 1 → 3
2, 3	abab	일치 확인

도면10a

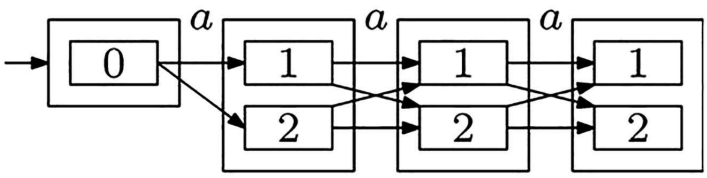


도면10b



$aa0$

도면10c



$aaa0$

도면11