

# 다수의 부분 요약 준비로 문자열 조작 분석하기

김세원

2013. 2. 1 / PLASSE HYU

# 문자열 조작

- 접합 ← 가장 기본, 또는 너무 기본
- 정규식 매칭 / 치환
  - `reg_match("[a-Z]+", id)`  
`reg_replace("<", "&lt;", x)`
  - 자주 쓰임, 밖에서 온 문자열에
  - 대개, 유한 오토마타나 transducer 수준

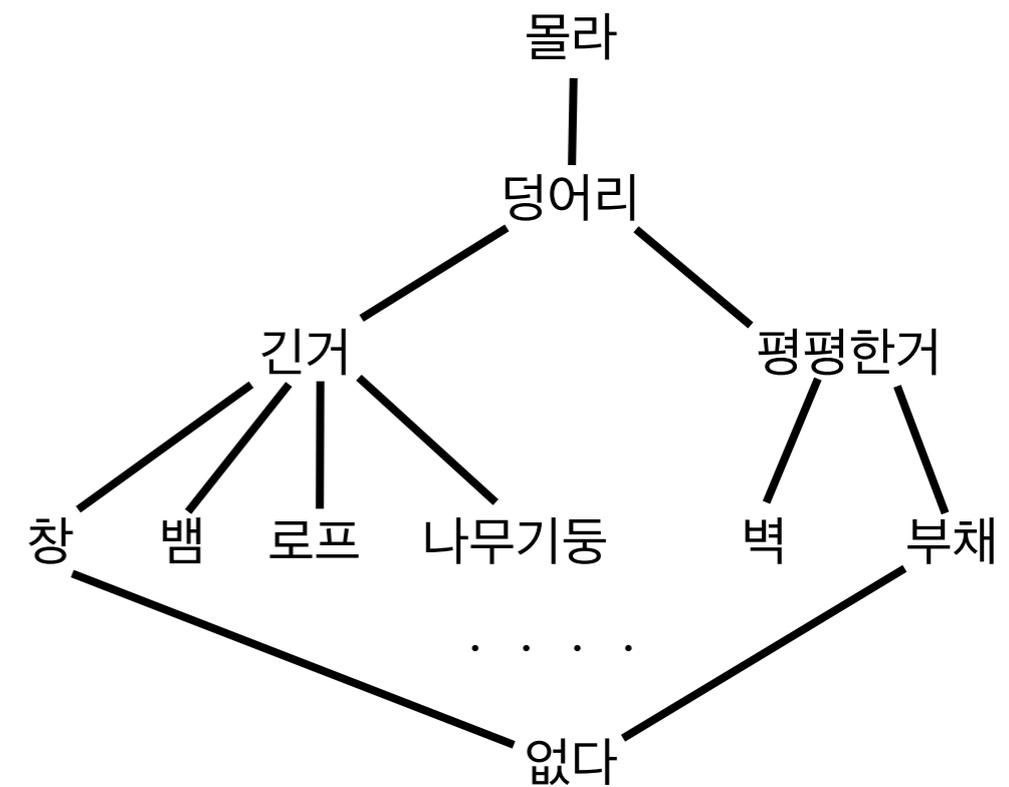
# 분석 필요한 경우

- 최종 검증 목표에 영향

```
x := get_user_input();  
y := "a";  
if (reg_match("a+", x))  
  y := x;  
output y;
```

출력이  $S \rightarrow a \mid aS$  에  
부합하는가?

# 아이디어: 부분 요약 준비



# 아이디어: 부분 요약 준비

- 부분을 관찰한 것이 문제?
- 본 일부를 전체로 착각한 것이 문제!
- 관찰한 부분 정리되면 **정확한 묘사**

코끼리: { 상아  $\mapsto$  창,  
다리  $\mapsto$  나무기둥,  
귀  $\mapsto$  부채, ... }

Q: 상아와 다리는?  
A: 긴것

# 연관 짓기

# 연관 짓기

- 코끼리 ➡ 요약대상 문자열 값 집합
- Q: 상아 모양은? ➡ 정규식 조건 통과
- {상아, 다리, 귀, ...} ➡ {○, X}
- ○: 정규식에 맞는 문자열
- X: 정규식에 맞지 않는 문자열

# 예제

```
x := get_user_input();  
y := "a";  
if (reg_match("a+", x))  
  y := x;  
output y;
```

← 부합?  
 $S \rightarrow a \mid aS$

- 초기화 직후,  $x: \{ \bigcirc \mapsto \{S\}, X \mapsto T \}$
- then branch에서  $\{S\}$  꺼내어  $y$ 에  $\rightarrow$  부합검증!

else 없으니  $X \mapsto \dots$  배도 됨

# 의문

- 조작에서 정규식 고정?
  - 많은 경우
- 뒤에 나오는 조작 어떻게 예상?
  - 전분석

문제: 접합 연산은?

# 조건 전에 접합

- 예제: 부합 분석  $S \rightarrow \langle A \rangle$ ,  $A \rightarrow a \mid aA$

★ →  

```
x := get_user_input();  
x := "<" . x . ">";  
if (reg_match("<a+>", x))  
  output x;
```

★ 에서,

“<” : { ○  $\mapsto$   $\perp$  }

“>” : { ○  $\mapsto$   $\perp$  }

x : { ○  $\mapsto$  { <A>, S } }

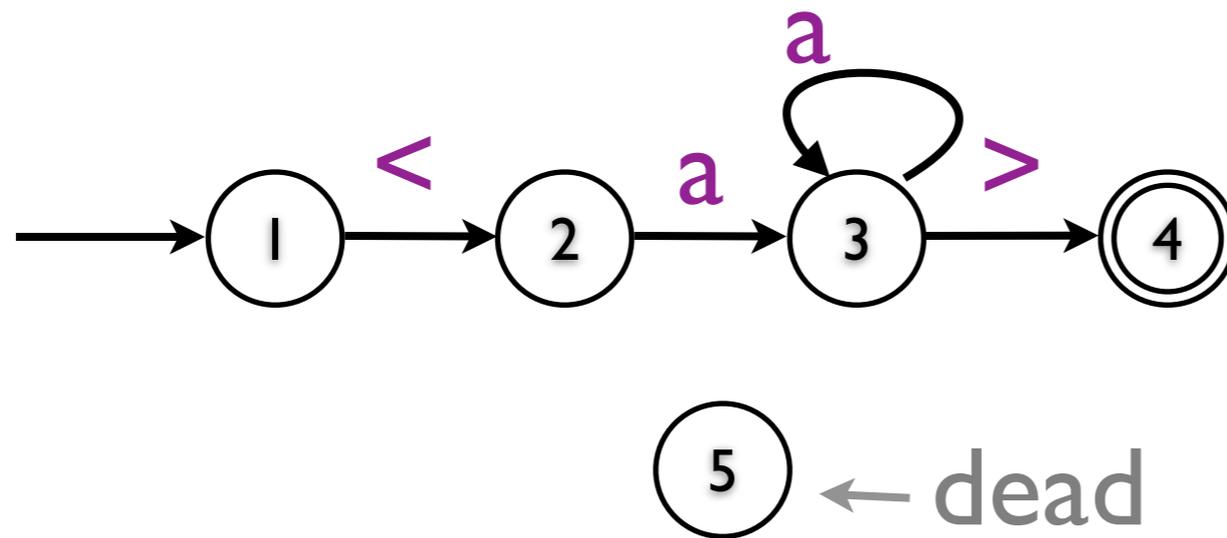
- point-wise 접합, wrong

- “<”.x.“>” 해보면, { ○  $\mapsto$   $\perp$  }

해결: 더 잘게, 여러 부분

# 더 잘게

- 정규식 “<a+>”의 DFA

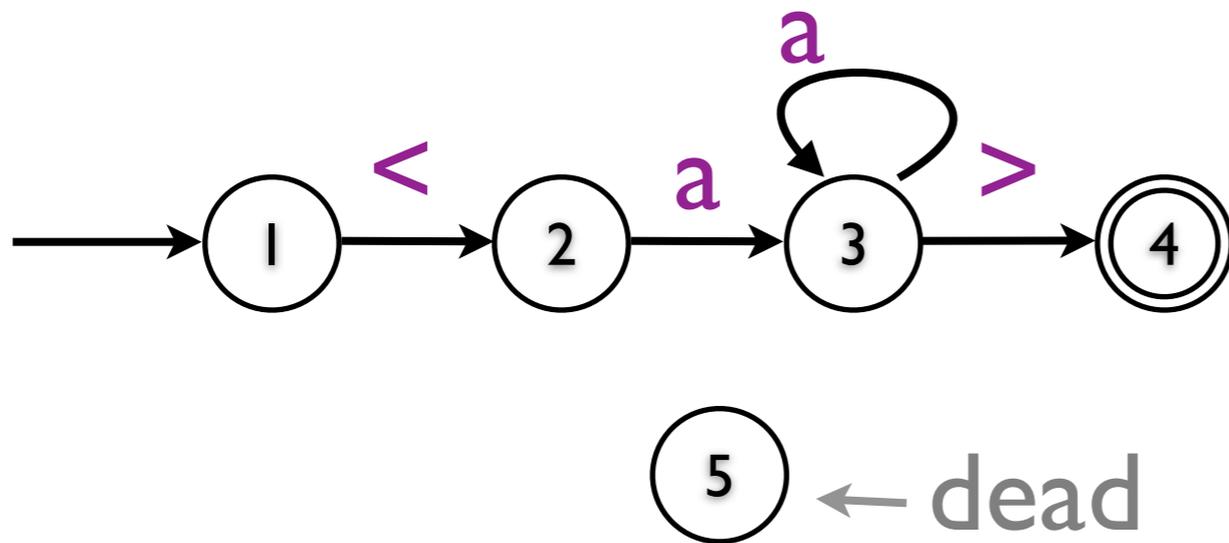


- :  $1 \rightsquigarrow 4$
- ✗ :  $1 \rightsquigarrow 1, 1 \rightsquigarrow 2, 1 \rightsquigarrow 3, 1 \rightsquigarrow 5$

# 더 다양한 부분

- DFA 상태 집합이  $Q = \{1, \dots, n\}$  일때
  - 가능한  $i \rightsquigarrow j$  는  $n \times n$  가지
  - 전부 다 이용
  - 편의상  $n \times n$  행렬로 표시
    - $i \rightsquigarrow j$  에 대응되는 값을  $(i, j)$  위치에

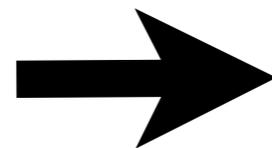
# 예시



$S \rightarrow \langle A \rangle$

$A \rightarrow a \mid aA$

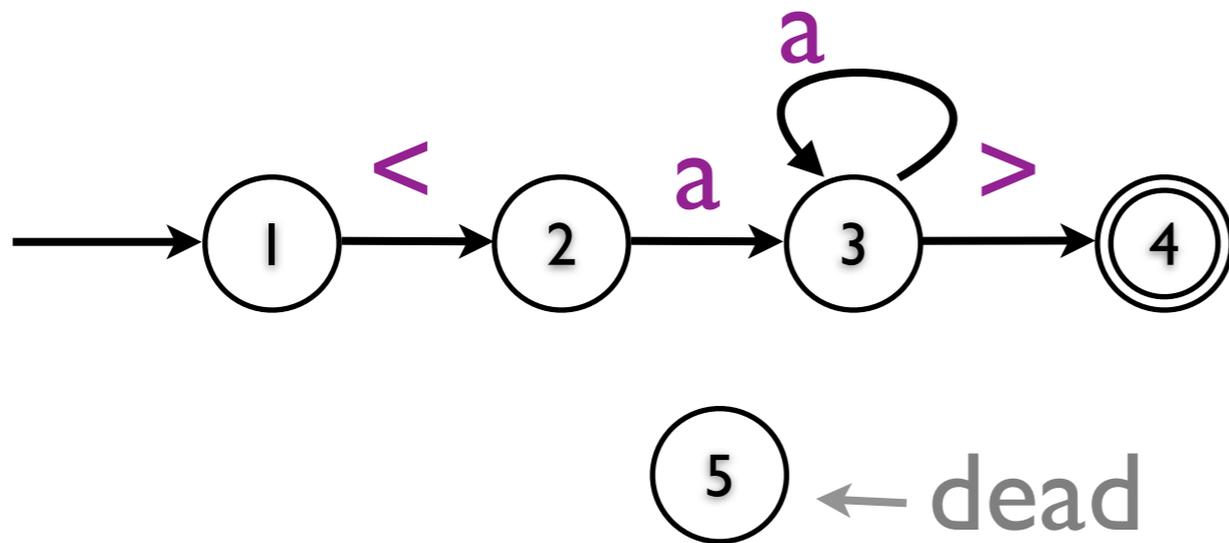
“<” : { ○  $\mapsto$   $\perp$  }



“<” :

$\perp$	{<}	$\perp$	$\perp$	$\perp$
$\perp$	$\perp$	$\perp$	$\perp$	{<}
$\perp$	$\perp$	$\perp$	$\perp$	{<}
$\perp$	$\perp$	$\perp$	$\perp$	{<}
$\perp$	$\perp$	$\perp$	$\perp$	{<}

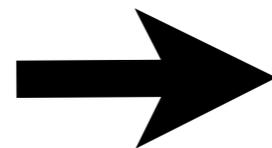
# 예시



$S \rightarrow \langle A \rangle$

$A \rightarrow a \mid aA$

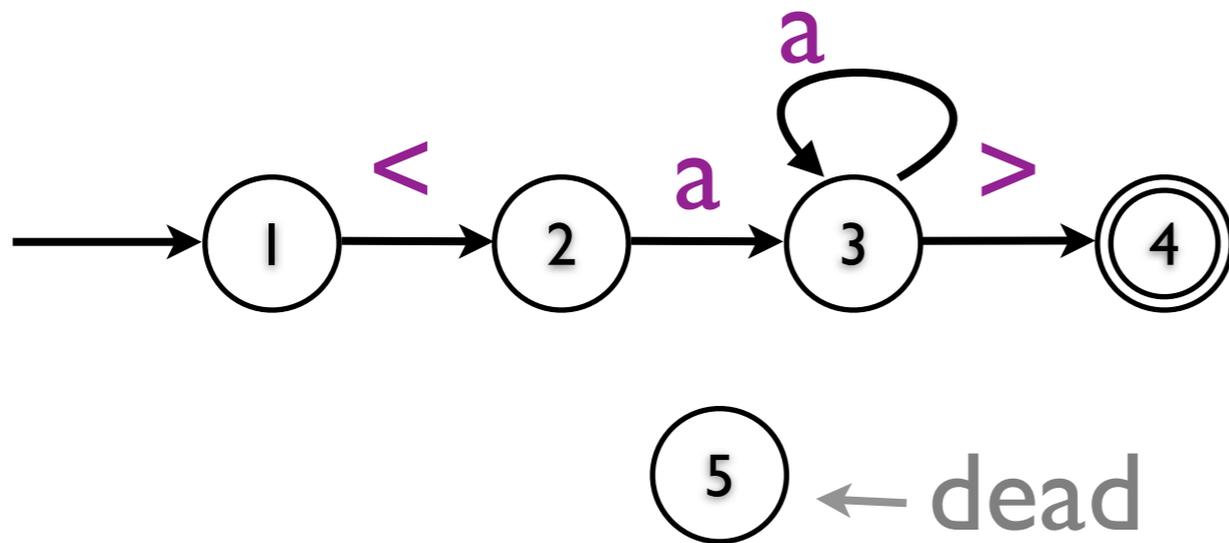
“>” : { ○  $\mapsto \perp$  }



“>” :

$\perp$	$\perp$	$\perp$	$\perp$	{>}
$\perp$	$\perp$	$\perp$	$\perp$	{>}
$\perp$	$\perp$	$\perp$	{>}	$\perp$
$\perp$	$\perp$	$\perp$	$\perp$	{>}
$\perp$	$\perp$	$\perp$	$\perp$	{>}

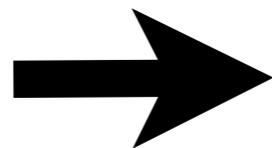
# 예시



$S \rightarrow \langle A \rangle$

$A \rightarrow a \mid aA$

? : {  $\circ \mapsto \perp$  }



? :

{ $\epsilon$ }	{<}	{<A}	{<A>,S}	$\emptyset$
$\perp$	{ $\epsilon$ }	{A}	{A>}	$\emptyset$
$\perp$	$\perp$	$\emptyset$	$\emptyset$	$\emptyset$
$\perp$	$\perp$	$\perp$	{ $\epsilon$ }	$\emptyset$
$\perp$	$\perp$	$\perp$	$\perp$	$\emptyset$

# 접합 연산 $\Rightarrow$ 행렬곱

- 접합은 행렬 곱 형태로 계산 가능
  - 요소간 곱은 원래 분석의 접합
  - 요소간 합은 원래 분석의  $\cup$
- $L(i \rightsquigarrow j)$  :  $i$  에서  $j$  로 가게하는 문자열 집합

$$L(i \rightsquigarrow j) = \bigcup_{k \in Q} [ L(i \rightsquigarrow k) \cdot L(k \rightsquigarrow j) ]$$

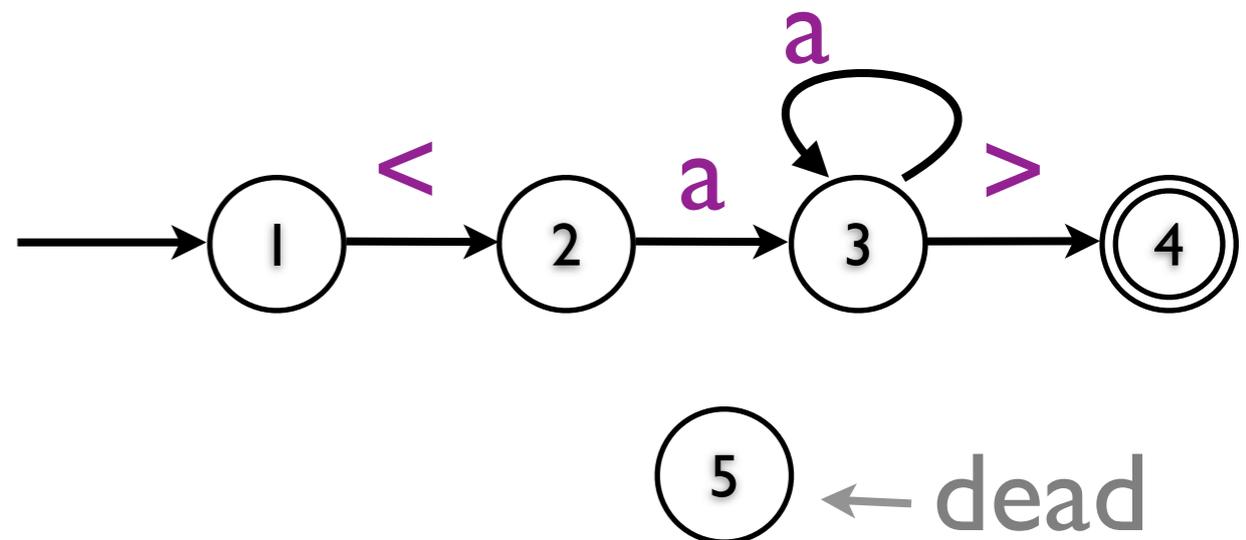
# 예시

- 앞에 나온 “<”, “?”, “>”간의 접합 값을 구하면,

⊥	⊥	⊥	{<A>,S}	∅
⊥	⊥	⊥	⊥	∅
⊥	⊥	⊥	⊥	∅
⊥	⊥	⊥	⊥	∅
⊥	⊥	⊥	⊥	∅

```

x := get_user_input();
x := "<" . x . ">";
if (reg_match("<a+>", x))
  output x;
  
```



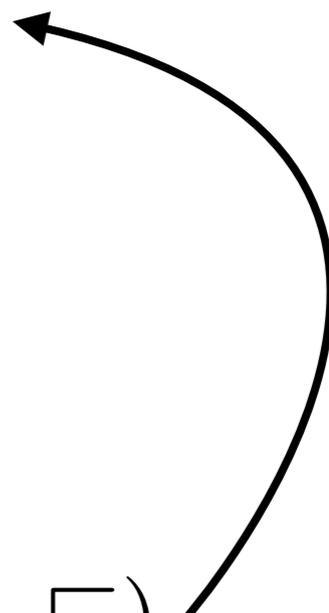
# 중첩 조작

- 주는 것

$$(\mathcal{P}(\Sigma^*), \subseteq) \begin{matrix} \xleftarrow{\gamma_0} \\ \xrightarrow{\alpha_0} \end{matrix} (D_0, \sqsubseteq_0)$$

$$\sqcup_0, \nabla_0, \odot_0, \alpha_0(\{s\})$$

다시 적용 가능  
중첩 조작 해결



- 받는 것

$$(\mathcal{P}(\Sigma^*), \subseteq) \begin{matrix} \xleftarrow{\gamma} \\ \xrightarrow{\alpha} \end{matrix} (Q \times Q \rightarrow D_0, \sqsubseteq)$$

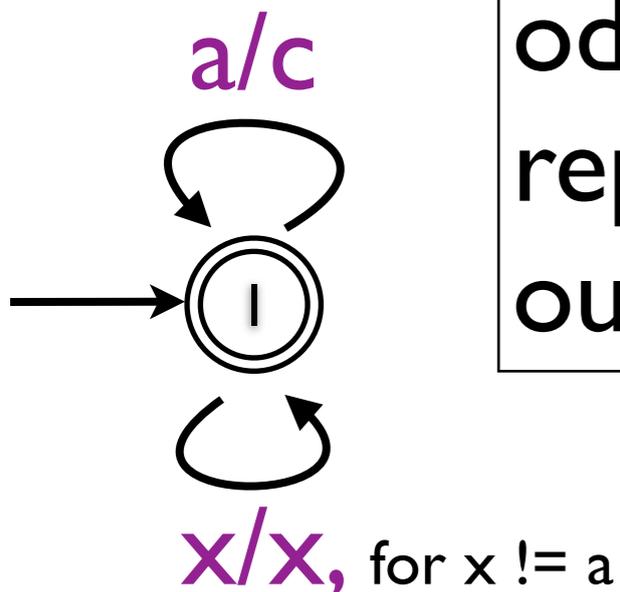
$$\sqcup, \nabla, \odot, \alpha(\{s\}), \text{해당 조작 통과 연산}$$

유한 transducer는?

# transducer의 경우

- 받아들이는 input 말고 그때 나오는 output 요약

```
x := "a";  
while (?) do  
  x := "<" . x . ">";  
od  
replace("a", "c", x);  
output x
```



$S \rightarrow c \mid \langle S \rangle$

“<” : [ {<} ]

“>” : [ {>} ]

“a” : [ {c, S} ]



출력 위치에서  $x : [ \{S\} ]$   
부합 검증!

# 맷기

- 아이디어: 부분 질문에 답변 미리 준비하기
  - 모든  $i \rightsquigarrow j$  준비하면 접합과 중첩 조작 해결
- 어떤 문자열 분석이든지 접합만 주면
  - 유한 상태 기반 조작은 해결

감사합니다

# 질문 / 답변