

SNU 4541.664A Program Analysis

Note 10-1c

Prof. Kwangkeun Yi

실행과정(trace) 요약해석 알고리즘 (1/4)

프로그램 C 의 요약 의미 $[\hat{C}]$ 인 다음을 계산하면 된다:

$$\text{fix}(\hat{F} \stackrel{\text{let}}{=} \lambda \hat{T}. \alpha(T_0) \sqcup \text{Next } \hat{T})$$

여기서

$$\begin{aligned}\hat{F} &\in \text{Trace} \rightarrow \text{Trace} \\ \text{Trace} &= \Delta \rightarrow \text{State} \\ \text{Next} &= (\wp \sqcup) \circ \hat{\pi} \circ (\wp \sqcup \text{next}) \\ \hat{\text{next}} &\in \text{State} \rightarrow 2^{\text{State}}.\end{aligned}$$

$\text{fix } \hat{F}$ 를 펼쳐보면 다음의 Y_i 를 계산하게 된다, 변화가 없을 때까지:

$$Y_0 = \alpha(T_0), \quad Y_{n+1} = \alpha(T_0) \sqcup \hat{\text{Next}}(Y_n)$$

따라서

```
T, T': Δ → State;
begin
  T := T' := α(T₀);
  repeat
    T' := T;
    T := α(T₀) ∪ ((⊆) ∘ π̂)(∪_{i ∈ Δ} next T[i]);
  until T ⊆ T'; (* no more increase *)
return T';
end
```

Figure: 초보 알고리즘

실행과정(trace) 요약해석 알고리즘 (2/4)

축지법(∇)과 좁히기(Δ)가 필요한 경우 아래 두개를 차례대로 구현:

$$\text{Widen}(\hat{F}) = \lim_{i \in \mathbb{N}} \begin{cases} \hat{Y}_0 & = \alpha(T_0) \\ \hat{Y}_{i+1} & = \begin{cases} \hat{Y}_i & \text{if } \hat{F}(\hat{Y}_i) \sqsubseteq \hat{Y}_i \\ \hat{Y}_i \nabla \hat{F}(\hat{Y}_i) & \text{o.w.} \end{cases} \end{cases}$$

$$\text{Narrow}(\hat{m}) = \lim_{i \in \mathbb{N}} \begin{cases} \hat{Z}_0 & = \hat{m} \\ \hat{Z}_{i+1} & = \hat{Z}_i \Delta \hat{F}(\hat{Z}_i) \end{cases}$$

따라서

```
T, T', Y: Δ → State;
begin
  T := T' := α(T0);
  repeat
    T' := T;
    Y := α(T0) ⊔ ((ϕ ⊔) ∘ π)(⋃i ∈ Δ next T[i]);
    T := if Y ⊆ T' then T' else T' ∇ Y;
  until T ⊆ T'; (* no more increase *)

  repeat
    T := T'
    T' Δ := α(T0) ⊔ ((ϕ ⊔) ∘ π)(⋃i ∈ Δ next T[i]);
  until T ⊆ T'; (* no more decrease *)
  return T;
end
```

Figure: 축지법(∇)과 좁히기(Δ)를 사용하는 초보 알고리즘

실행과정(trace) 요약해석 알고리즘 (3/4)

할일만하기 방식(worklist method):

- 매번 모든 인덱스 $i \in \Delta$ 를 훑으며 \hat{next} 를 계산

$$\bigcup_{i \in \Delta} \hat{next} T[i]$$

할 필요는 없다.

- 이전 반복에서 변화된 것들만 다시 계산하면 된다.

```
T, T' : Δ → State;  
W : 2Δ; (* worklist *)  
begin  
  T := T' := α(T0);  W := Δ;  
  repeat  
    T' := T;  
    T := α(T0) ⊔ ((∅ ⊔) ∘  $\hat{\pi}$ )( $\bigcup_{i \in W} \hat{next} T[i]$ );  
    W := {i ∈ Δ | T[i]  $\not\subseteq$  T'[i]};  
  until W = {}; (* no more increase *)  
return T';  
end
```

Figure: 할일만하기 알고리즘

실행과정(trace) 요약해석 알고리즘 (4/4)

```
 $T, T', Y: \Delta \rightarrow \hat{State};$   
 $W: 2^\Delta; (* \text{worklist} *)$   
begin  
   $T := T' := \alpha(T_0); \quad W := \Delta;$   
  repeat  
     $T' := T;$   
     $Y := \alpha(T_0) \sqcup ((\wp \sqcup) \circ \hat{\pi})(\bigcup_{i \in W} \hat{next} T[i]);$   
     $T := \text{if } Y \sqsubseteq T' \text{ then } T' \text{ else } T' \nabla Y;$   
     $W := \{i \in \Delta \mid T[i] \not\sqsubseteq T'[i]\};$   
  until  $W = \{ \}; (* \text{no more increase} *)$   
  
   $W := \Delta;$   
  repeat  
     $T := T';$   
     $T' \Delta := \alpha(T_0) \sqcup ((\wp \sqcup) \circ \hat{\pi})(\bigcup_{i \in W} \hat{next} T[i]);$   
     $W := \{i \in \Delta \mid T[i] \not\sqsubseteq T'[i]\};$   
  until  $W = \{ \}; (* \text{no more decrease} *)$   
  return  $T;$   
end
```

Figure: 축지법(∇)과 좁히기(Δ) + 할일만하기 알고리즘