SNU 4541.664A Program Analysis, Spring 2006

Final Exam

6/13/2006, 13:00-15:00

학번: 이름:

Problem 1 [20 × (5pts, -3pts)] O/X로 답하라.

1. 귀납 규칙 집합 Φ는 아래의 함수 ϕ을 정의:

$$\phi(Y) = \{x \mid \frac{X}{x} \in \Phi,\ X \subseteq Y\}$$

하고, Φ 규칙들이 정의하는 집합은 아래의 집합이다:

$$\bigcup\{X|\phi(X) \subseteq X\}$$

2. 분석하고자 하는 프로그램의 소스 언어에 따라서 정적 프로그램 분석이 완벽(sound and complete)할 수 있다.

3. 분석하고자 하는 프로그램의 성질에 따라서 정적 프로그램 분석이 완벽(sound and complete)할 수 있다.

4. 현재의 집합 제약식을 이용한 분석(set-based analysis)들은 임의의 집합 제약식을 이용하는 자유가 없다.

5. 요약 해석(abstract interpretation)에서 요약 공간(abstract domain)이 무한하도록 정의될 수 있다.

6. 정적 프로그램 분석기술을 이용해서 소프트웨어의 모든 오류를 자동으로 찾는 것은 가능하다.

7. 수업시간에 다룬 다형 타입시스템(polymorphic type system)은 단순 타입 시스템(simple type system) 보다 분석의 정확도가 높은 것이다.

8. 타입 시스템에 기초한 프로그램 분석은 주어진 증명규칙으로 프로그램에 대한 증명을 만들어 내는 것이다.

9. 다음과 같이 프로그램 분석을 디자인하면 그 분석은 옳다:

   - 주어진 프로그램의 실제 의미는 연속함수 $F : 2^S \to 2^S$의 최소고정점으로 정의한다. (2^S의 원소들 사이의 순서는 집합포함순: $x \sqsubseteq y = x \subseteq y$)
   - 주어진 프로그램의 요약된 의미는 단조함수 $\hat{F} : \hat{S} \to \hat{S}$의 최소 고정점으로 정의한다.
   - 2^S와 $\hat{S}$는 갈로아 연결

$$2^S \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \hat{S}$$

   되어있다.

10. 축지법/넓히기(widening) $\triangledown$의 조건은

    - $\forall a, b \in \hat{D} : (a \sqsubseteq a \triangledown b) \wedge (b \sqsubseteq a \triangledown b)$
    - $\forall$증가하는 체인$\{x_i\}_i$: 체인 $y_0 = x_0, y_{i+1} = y_i \triangledown x_{i+1}$ 는 유한

    이다.

11. 좁히기(narrowing) $\triangle$의 조건은

    - $\forall a, b \in \hat{D} : x \sqsupseteq y \Rightarrow (x \sqsupseteq (x \triangle y)) \wedge (y \sqsupseteq (x \triangle y))$
    - $\forall$감소하는 체인$\{x_i\}_i$: 체인 $y_0 = x_0, y_{i+1} = y_i \triangle x_{i+1}$ 는 유한

    이다.

12. A, B는 집합이다. 그러면 다음 두 CPO사이에 갈로아 연결이 가능하다:

$$2^{A \overset{\text{fin}}{\to} B} \to 2^B \underset{\alpha?}{\overset{\gamma?}{\leftrightarrows}} (A \overset{\text{fin}}{\to} 2^B) \to 2^B$$

13. A, B는 집합이고 2^B와 $\hat{B}$사이는 갈로아 연결되어 있다. 그러면 다음 두 CPO사이에 갈로아 연결이 가능하다:

$$2^{A \overset{\text{fin}}{\to} 2^B} \to 2^B \underset{\alpha?}{\overset{\gamma?}{\leftrightarrows}} (A \overset{\text{fin}}{\to} \hat{B}) \to \hat{B}$$

14. A는 집합이고 A^ω는 길이가 무한 할 수도 있는 A원소들의 리스트들의 집합이다. 2^A와 $\hat{A}$사이는 갈로아 연결되어 있다. 그러면 다음 두 CPO사이에 갈로아 연결이 가능하다:

$$2^{A^\omega} \underset{\alpha?}{\overset{\gamma?}{\leftrightarrows}} \hat{A}$$

15. A는 집합이고 A^ω는 길이가 무한 할 수도 있는 A원소들의 리스트들의 집합이다. 2^A와 $\hat{A}$사이는 갈로아 연결되어 있다. A의 각 원소를 유한한 인덱스 집합 I의 한 원소로 맺어주는 함수가 존재한다. 이 때 다음 두 CPO사이에 갈로아 연결이 가능하다:

$$2^{A^\omega} \underset{\alpha?}{\overset{\gamma?}{\leftrightarrows}} I \to \hat{A}$$

16. 다음의 정수식 프로그래밍 언어를 타겟으로 하는 분석기를 정의하려고 한다.

$$\begin{array}{rcll} e & ::= & n & (n \in \mathbb{Z}) \\ & | & e\,\texttt{+-} & \\ & | & e\,\texttt{mod}\,e & \end{array}$$

$e\,\texttt{mod}\,0$ 는 e의 값과 상관없이 임의의 양수가되고, $n\texttt{+-}$는 $n+1$과 $n-1$중에서 임의로 선택된다.

요약공간을 만드는 갈로아 연결

$$2^{\mathbb{Z}} \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \{\bot, 0, >0, <0, \top\}$$

을

$$\begin{array}{rcl} \alpha\emptyset & = & \bot \\ \alpha\{0\} & = & 0 \\ \alpha X & = & >0 \quad \text{if } \forall x \in X : x > 0 \\ \alpha X & = & <0 \quad \text{if } \forall x \in X : x < 0 \\ \alpha X & = & \top \quad \text{otherwise} \end{array}$$

로 하고 안전한 $\hat{\texttt{+-}}$을 가장 정확하게 정의해서

$$\begin{array}{rcl} \bot\,\hat{\texttt{+-}} & = & \bot \\ v\,\hat{\texttt{+-}} & = & \top \end{array}$$

로 했다. 옳은가?

17. 또, 위의 요약공간에서 안전한 $\hat{\texttt{mod}}$를 가장 정확하게 정의하면 아래와 같다:

$$\begin{array}{rcl} \bot\,\hat{\texttt{mod}}\,\star & = & \bot \\ \text{else} \quad \star\,\hat{\texttt{mod}}\,\bot & = & \bot \\ \text{else} \quad \star\,\hat{\texttt{mod}}\,0 & = & >0 \\ \text{else} \quad 0\,\hat{\texttt{mod}}\,\star & = & 0 \\ \text{else} \quad \star\,\hat{\texttt{mod}}\,\star & = & \top \end{array}$$

옳은가?

18. 타입 시스템의 안전성을 증명할 때 "Subject Reduction Lemma"를 증명하는데, 그 내용은 타입이 있는 것으로 증명된 프로그램은 한 스텝 실행이 진행되도 같은 타입을 가지는 것으로 증명된다는 것이다.

19. 타입 시스템의 구현은 고정점(fixpoint) 알고리즘 대신에 주로 동일화(unification)알고리즘을 이용한다.

20. 수업시간에 다룬 단순 타입시스템의 안전한 알고리즘은 아래와 같이 정의될 수 있다:

$$M : \mathit{TyEnv} \times \mathit{Exp} \times \mathit{Type} \to \mathit{Subst}$$

$$\begin{array}{rcl} M(\Gamma, n, \tau) & = & \text{unify}(\iota, \tau) \\ M(\Gamma, x, \tau) & = & \text{unify}(\tau, \tau') \quad \text{if } x : \tau' \in \Gamma \\ M(\Gamma, \lambda x.e, \tau) & = & \text{let } S = \text{unify}(\alpha_1 \to \alpha_2, \tau) \quad \text{new } \alpha_1, \alpha_2 \\ & & \quad S' = M(S\Gamma + x : S\alpha_1, e, S\alpha_2) \\ & & \text{in } S' \\ M(\Gamma, e\ e', \tau) & = & \text{let } S = M(\Gamma, e, \alpha \to \tau) \quad \text{new } \alpha \\ & & \quad S' = M(S\Gamma, e', S\alpha) \\ & & \text{in } S' \\ M(\Gamma, e\ \texttt{+}\ e', \tau) & = & \text{let } S = \text{unify}(\iota, \tau) \\ & & \quad S' = M(S\Gamma, e, \iota) \\ & & \quad S'' = M(S'S\Gamma, e', \iota) \\ & & \text{in } S'' \end{array}$$