Homework 10
CS420/KAIST Spring 1999
이 광근
Due: June 3 (목요일), 24:00

1999년 5월 27일

이번 숙제는 종이에 작성해서 조교에게 제출한다.

Exercise 1 "Manual analysis" **30pt**
강의 노트 23 페이지 Example 3의 네가지 프로그램들을 대상으로

1. 강의 노트 24 페이지의 규칙을 **그대로 적용**해서 집합조건식들을 도출하고
2. 도출된 집합식으로 부터 연립방정식을 세운후
3. 그 연립방정식의 최소의 해를 구하는 과정을 보이라.

□

Exercise 2 "Alpha conversion: first step before deriving equations" **20pt**
주어진 프로그램의 변수들을 서로 다른 이름으로 바꾸고 나서 집합관계식을 도출하면 프로그램 분석의 정확도가 향상될 수 있다. 예를 들어, 다음의 프로그램을 보자.

```
let
  f = fn x => (fn y => x y)
  g = fn x => x 0
in
  (f (fn z => z+1)) (g (fn w => w+2))
end
```

위에서 변수 x는 f의 인자이기도 하고 g의 인자이기도 하다. 이 상태에서 집합제약식을 만들면 f의 x에 전달되는 함수 `fn z => z+1`과 g의 x에 전달되는 함수 `fn w => w+2`가 집합변수 X_x에 **모두** 모아지게 된다:

$$X_x \supseteq \{\lambda z.z+1\}$$
$$X_x \supseteq \{\lambda w.w+2\}$$

따라서 x가 가질 수 있는 함수들이 두개가 되다 보니, 위의 프로그램에서 함수 호출 식 "x y"이나 "x 0"를 좀더 경제적인 코드로 변환 할 수 없게된다. (See our lecture note, page 21)

위의 프로그램에서 다른 두개의 x를 다르게 이름을 붙여보자:

```
let
  f = fn x1 => (fn y => x1 y)
  g = fn x2 => x2 0
in
  (f (fn z => z+1)) (g (fn w => w+2))
end
```

위와같이 변환한 후에 집합제약식을 도출하면 다른 변수 X_{x_1} 와 X_{x_2}에 각각 하나의 함수가 정확하게 모인다:

$$X_{x_1} \supseteq \{\lambda z.z + 1\}$$
$$X_{x_2} \supseteq \{\lambda w.w + 2\}$$

이렇게 되면, 함수 호출식 "x1 y"와 "x2 0"가 좀더 경제적인 코드로 변환될 수 있다.

이와같이 다른 변수를 다르게 이름붙인 후에 집합관계식을 취합하는 것이 분석의 정확도를 높이는데 도움이 된다. 이러한 변환(pre-processing)을 "alpha conversion"이라고 한다.

주어진 tcoML 프로그램을 alpha-변환시키는 규칙을 정의하라. □