Homework 4
CS420 Spring 1999
이 광근
Due: April 8 (목요일), 24:00

1999년 4월 1일

Exercise 1 "ML-Lex can be more powerful than generating DFA."

Actually, `lex`-generated programs can be strictly more powerful than finite state automata because we can add a general program in the input specification. For example, `ml-lex` can generate a program that recognize nested comments like

```
(*(* nested comments *)*)
```

Remember that language $\{a^n c a^n | n \geq 1\}$ cannot be recognized by a finite-state automaton.

Using `ml-lex`, construct a lexical analyzer that recognizes nested SML comments. Input text is in `[a-z\t\n\ ()]+`. Your lexical analyzer must reject if input string has unbalanced `(*` and `*)`.

Hint: use the option "`%s A;`" for specifying an extra start state `<A>`, and in the rule parts do something like

```
<INITIAL>"(*"  => (YYBEGIN A; comLevel := 1; lex());
<A>"(*"        => (inc comLevel; lex());
<A>"*)"        => (dec comLevel; ...);
```

□

Exercise 2 "Lambda 세상"

Lambda Calculus 는 1930년대에 Alonzo Church 에 의해 고안된 논리체계로서 계산가능한 함수를 정의하는 한 방법으로 알려져있다. 실제로, Lambda Calculus에서 표현가능한 함수는 Turing machine으로 실행시킬 수 있고, 거꾸로 Turing machine에서 실행되는 프로그램은 Lambda Calculus의 함수로 표현될 수 있다는 것이 증명되었다.

특히, 프로그래밍 언어의 핵심이 놀랄만큼 간단한 Lambda Calculus로 표현되기때문에, 프로그래밍 언어의 세계를 기술하고 고민하는 데 가장 적절한 서술양식으로서 Lambda Calculus가 사용되고 있다.

여러분들이 이번학기 사용하는 ML이라는 언어도 Lambda Calculus를 기반으로 연구된 수많은 연구성과를 실용적으로 담아낸 결과인 것이다.

다음은 Lambda Calculus의 식들에 대한 문법이다.

$$
\begin{array}{rcll}
e & ::= & id & \text{변수} \\
  & | & \backslash\, id\ .\ e & \text{함수} \\
  & | & e\, e & \text{함수적용} \\
  & | & (e) &
\end{array}
$$

- id는 `[a-zA-Z][a-zA-Z0-9]*` 로 정의된다.
- 함수적용식은 왼편으로 묶인다 (left-associative). 즉, $e_1\, e_2\, e_3$는 $(e_1\, e_2)\, e_3$으로 간주된다.
- 함수식 $\backslash\, x.\ e$의 몸체 부분인 e는 최대한 가능한 것으로 한다. 즉, $\backslash\, x.\ x\, x$는 $(\backslash\, x.\ x)\, x$로 간주되지않고 $\backslash\, x.\ (x\, x)$로 간주된다.

Ml-lex와 yacc을 사용하여 Lambda 식들이 있는 화일을 읽어들여서 자유변수(free variable)들의 리스트를 만드는 함수 `fv`를 작성하라:

```
fv :  string -> string list
```

Lambda 식에서 자유변수(free variable)들은 다음과 같이 정의된다:

$$
\begin{array}{lcl}
\mathrm{FV}(\mathrm{id}) & = & \{\mathrm{id}\} \\
\mathrm{FV}(\backslash\, \mathrm{id}.\ e) & = & \mathrm{FV}(e)\ \setminus\ \{\mathrm{id}\} \\
\mathrm{FV}(e_1\, e_2) & = & \mathrm{FV}(e_1) \cup \mathrm{FV}(e_2)
\end{array}
$$

□