Homework 1
SNU 4910.210, Fall 2013
Kwangkeun Yi
due: 9/14(Sat), 24:00

이번 숙제의 목적은:

- 프로그래밍의 기본 부품을 조합해서 프로그램을 구성하는 것을 익힌다.
- 재귀적인 방식으로 프로그램하는 것을 익힌다.

Exercise 1 "유클리드gcd"

입력: 두 개의 음이 아닌 정수.

두 양의 정수의 최대 공약수를 구하는 함수 (`gcd` n m)를 정의하세요. 유클리드(Euclid)가 이용한 성질은, 간단한 성질을 이용한 재귀법이었다: n과 0의 gcd는 n이고; n과 m의($n \geq m$) gcd는 $n - m$과 m의 gcd와 같다. □

Exercise 2 "튜링2"

입력: 하나의 정수.

컴퓨터의 디자인을 선보인 튜링(Alan Turing)의 1936년도 논문[1]에 두 번째 예로 든 프로그램이 $001011011101111\cdots$ 을 출력하는 프로그램이었다. 그 함수의 변종인 `t2`를 작성하세요: (`t2` n)은 $0010110111\cdots0\underbrace{1\cdots1}_{|n|}$를 출력한다. □

Exercise 3 "양휘3"

입력: 하나의 정수.

[1] "On Computable Numbers, with an Application to Eintscheidungsproblem", *Proceedings of the London Mathematical Society, ser.2, vol.42, pp.230-265., 1936.*

중국의 양휘는 13세기에 이미 “파스칼의 삼각형”으로 알려진 것을 고안해서 계산에 이용한 기록이 있다:

$$\begin{array}{ccccccc} & & & 1 & & & \\ & & 1 & & 1 & & \\ & 1 & & 2 & & 1 & \\ 1 & & 3 & & 3 & & 1 \\ & & & \vdots & & & \end{array}$$

정수를 입력으로 받아서 파스칼 삼각형의 그 횡까지를 연속해서 출력하는 함수 `yanghui`을 작성하세요: (`yanghui` n)은 $1111211331\cdots \underbrace{1\cdots 1}_{|n|\text{번째 횡}}$ 를 출력한다. 0번째 횡은 아무것도 존재하지 않는다. □

Exercise 4 “k친수”

일반적으로 k진수($k > 1$)는 다음과 같이 표현한다.

$$d_0 \cdots d_n$$

여기서

$$\forall d_i \in \{0, \cdots, k-1\}.$$

그리고 “$d_0 \cdots d_n$”은 크기가

$$d_0 \times k^0 + \cdots + d_n \times k^n$$

인 정수를 표현한다.

이것을 살짝 확장해서 “k친수”를 다음과 같이 정의해보자. 표현은

$$d_0 \cdots d_n$$

여기서

$$\forall d_i \in \{1-k, \cdots, 0\} \cup \{0, \cdots, k-1\}.$$

그리고 “$d_0 \cdots d_n$”은 크기가

$$d_0 \times k^0 + \cdots + d_n \times k^n$$

인 정수를 표현한다.

예를 들어, 2친수의 경우를 생각하자. 베이스가 $\{-1, 0, 1\}$이 되겠다. `0`이 0을, `+`가 1을 `-`가 -1을 표현한다고 하면, `+` 는 1을, `+0+`는 5를, `+-`는 -1을, `+-0-`는 -9인 정수를 표현한다.

이러한 2친수 N의 집합을 귀납적으로 정의하면 다음과 같다:

$$
\begin{array}{rcl}
N & ::= & \texttt{0} \\
 & | & \texttt{+} \\
 & | & \texttt{-} \\
 & | & \texttt{0}N \\
 & | & \texttt{+}N \\
 & | & \texttt{-}N
\end{array}
$$

그리고, Scheme에서는 2친수 N을 다음과 같은 방법 $\underline{N}$에 의해 Scheme의 리스트로 표현할 수 있다:

$$
\begin{array}{rcl}
\underline{\texttt{0}} & = & \texttt{'z} \\
\underline{\texttt{+}} & = & \texttt{'p} \\
\underline{\texttt{-}} & = & \texttt{'n} \\
\underline{\texttt{0}N} & = & \texttt{(cons 'z } \underline{N}\texttt{)} \\
\underline{\texttt{+}N} & = & \texttt{(cons 'p } \underline{N}\texttt{)} \\
\underline{\texttt{-}N} & = & \texttt{(cons 'n } \underline{N}\texttt{)}
\end{array}
$$

즉, `0+-`는 Scheme에서

```
(cons 'z (cons 'p 'n))
```

로 표현된다, 왜냐하면

$$
\begin{array}{rcl}
\underline{\texttt{0+-}} & = & \texttt{(cons 'z } \underline{\texttt{+-}}\texttt{)} \\
 & = & \texttt{(cons 'z (cons 'p } \underline{\texttt{-}}\texttt{))} \\
 & = & \texttt{(cons 'z (cons 'p 'n)).}
\end{array}
$$

자 이제, 위와 같이 표현되는 2친수를 받아서 그것의 값을 계산하는 함수 `crazy2val`을 정의하라.

$$\texttt{crazy2val} : \text{2친수} \rightarrow \text{정수}.$$

□

Exercise 5 "2친수 더하기"

두 2친수를 받아서 2친수의 합에 해당하는 2친수를 내어놓는 함수 `crazy2add`를 정의하라

$$\texttt{crazy2add} : \text{2친수} \times \text{2친수} \rightarrow \text{2친수}.$$

위의 `crazy2add`는 다음의 성질이 만족되야 한다:

- 당연히, 임의의 2친수 z 과 z'에 대해서

  $$(\texttt{crazy2val (crazy2add } z\ z')) = (\texttt{crazy2val } z) + (\texttt{crazy2val } z').$$

- `crazy2add`은 재귀적으로 정의되야 한다. 2친수를 10진수로 변형후 더해서 다시 2친수로 표현하는 과정은 불허.

  2친수 두 개($d_0 \cdots d_n$, $d'_0 \cdots d''_m$)를 더하는 것은 각 자릿수 d_i과 d'_i를 더해서 만들어 진다. 이전 자릿수를 더해서 이번 자릿수로 넘어온 수(carry) c_i를 생각해야 하므로, $d_i + d'_i + c_i$를 $i = 0$부터 더하면서 구할 수 있다.

□