

3. 変数

変数(variable)は、値(value)を保持することのできる「箱」のようなものであり、それぞれ識別子(identifier)という名前を持っている。また、変数には型(type)が関連づけられており、変数に格納できる値はその型に属するものに制限される。

識別子

MLには、2種類の識別子が用意されている。

- アルファベット・数字識別子(alphanumeric identifier): 1文字目が英文字か ' (一重引用符)、2文字目以降が英文字、数字、'、_ からなる識別子。通常の変数名や関数名に用いる。1文字目が ' で始まる変数は型変数(type variable)といい、用途が制限される(後述)。
- シンボル識別子(symbolic identifier) 以下の20文字のみからなる識別子。演算子に用いる。+ - / * < > = ! @ # \$ % ^ & ` ~ \ | ? :
 - 実はMLでは、演算子もすべて関数として定義されている。例えばdivは、2つのint型の引数からint型の値を返す関数である。

環境

ある時点でのプログラムの計算は、そのとき有効な変数集合の値に基づいて行われる。この変数集合のことを環境(environment)という。また、格納されている値の集合のことをストア(store)という。

Cなどの言語の計算は、ストアを書き換えていく(副作用, side-effect)ことで行われる。一方、MLでは、ストアを書き換えることはできない。その代わりに、現在の環境に新しい値の束縛(value binding)を追加することで計算が行われる。

MLを起動すると、トップレベル環境(top-level environment)という環境が用意される。この中にはfloor, ceil, + など、これまで述べてきた演算子や組み込み関数などの定義が含まれている。後述するように、MLでは、関数と変数は同じように扱われる。例えば関数floorは、floorという名前を持ち、関数型の値を格納できる変数とみなすことができる。

環境に新たに識別子を追加するには、次のような構文を用いる。

```
val 識別子 = 式;

val pi = 3.14159;
val pi = 3.14159 : real
val radius = 4.0;
val radius = 4.0 : real
val area = pi * radius * radius;
val area = 50.26544 : real
```

MLでは、いったん識別子が環境に追加されると、その値は書き換えることができない。変数の値を(見かけ上)変更するには、同じ識別子を環境に追加する。このとき、元の環境は全く変更されない。このため、MLの計算は副作用がないと言われる。

環境に同じ識別子が複数含まれていた場合は、一番上のものだけが有効になる。

```
val radius = 5.0;
val radius = 5.0 : real
pi * radius * radius;
val it = 78.53975 : real
val radius = 4;
```

```
val radius = 4 : int  
radius * radius;  
val it = 16 : int
```