

# 計算モデル特論

## ML:関数型プログラミング言語



佐藤一郎

E-mail: ichiro@nii.ac.jp

Ichiro Satoh

## ▶ ML:関数型プログラミング言語

- 関数型言語であるMLは関数定義と利用に関する柔軟で強力な機構を提供する
- MLは定理証明システムEdinburg LCFのメタ言語(Meta Language)として開発
- R.Milnerによるlet式を含む多相型推論機構に基づく実装

Ichiro Satoh

## ▶ 関数定義

関数の定義:

```
fun fname param = expression ;
```

関数の定義例:

```
-fun double x = x*2;  
val double=fn:int->int  
-double 2;  
val it=4:int  
fun f(3,double 4);  
val it=14:int
```

Ichiro Satoh

## ▶ 関数定義

関数型言語であるMLは関数定義と利用に関する柔軟で強力な機構を提供する

関数の定義:

```
fun fname param = expression ;
```

関数の定義例:

```
-fun fib n = if n = 0 then 0  
            else if n = 1 then 1  
            else fib (n-1) + fib(n-2);  
val fib=fn:int->int  
-fib 10;  
val it=55:int
```

Ichiro Satoh

## ▶ 名前なし関数定義

関数型言語であるMLは関数定義と利用に関する柔軟で強力な機構を提供する

関数の定義:

```
fn param => expression ;
```

関数の定義例:

```
-fn x => x+1;  
val it=fn:int->int  
-(fn x=>x+1) 3 * 10  
val it=40:int
```

Ichiro Satoh

## ▶ 第一級のデータとしての関数

MLでは関数は整数などと同様にデータとして取り扱う

$m, n$ の組を受け取り $n^m$ を返す関数

```
-fun power(m,n) = if m=0 then 1 else n*power(m-1,n);  
val power=fn:int*int->int  
-power(3,2);  
val it=8:int
```

$m$ を受け取り、任意の $n$ に対して $n^m$ を計算する関数

```
-fun power m n = if m=0 then 1 else n*power(m-1) n;  
val power=fn:int->(int->int)  
-power 3 2;  
val it=8:int
```

Ichiro Satoh

## ▶ 局所データの宣言

宣言内で使用する変数の宣言

let式の定義:

```
fn valまたはfunの宣言リスト in expression end
```

let式の定義例:

```
-fun fib n = let fun f n a b =  
                if n = 0 then a  
                else if n = 1 then b  
                else f (n-1) b (a+b)  
              in f n 0 1 end;  
val fib=fn:int->int
```

Ichiro Satoh