22-1

Pairs

pair x y = λc. c x y

$M ::= \dots \mid \text{pair } M\ N \mid \text{fst } M \mid \text{snd } M$

$V ::= \dots \mid \text{pair } V\ V$

$E ::= \dots \mid \text{pair } E\ M \mid \text{pair } V\ E \mid \text{fst } E \mid \text{snd } E$

$E[\text{fst } (\text{pair } V\ U)] \Rightarrow E[V]$

$E[\text{snd } (\text{pair } V\ U)] \Rightarrow E[U]$

$T ::= \dots \mid T \times T$

$$\frac{\Gamma \vdash M : T_1 \quad \Gamma \vdash N : T_2}{\Gamma \vdash \text{pair } M\ N : T_1 \times T_2} \qquad \frac{\Gamma \vdash M : T_1 \times T_2}{\Gamma \vdash \text{fst } M : T_1} \qquad \frac{\Gamma \vdash M : T_1 \times T_2}{\Gamma \vdash \text{snd } M : T_2}$$

struct {int x; int y;} posn

posn (x:int)(y:int) = pair x y   posn.x p = fst p   posn.y = snd p

---

Variants   one thing or another   (Disjoint Union)

Animal is either Cat (int, string) or Dog (string)

Animal ::= (int × str) + str

$M ::= \dots \mid \text{inl } M \mid \text{inr } M \mid \text{case } M\ (X \Rightarrow N)\ (Y \Rightarrow L)$

$V ::= \dots \mid \text{inl } V \mid \text{inr } V$

$E ::= \dots \mid \text{inl } E \mid \text{inr } E \mid \text{case } E\ (X \Rightarrow N)\ (Y \Rightarrow L)$

$E[\text{case } (\text{inl } V)\ (X \Rightarrow N)\ (Y \Rightarrow L)] \Rightarrow E[N[X \leftarrow V]]$

$E[\text{case } (\text{inr } U)\ (X \Rightarrow N)\ (Y \Rightarrow L)] \Rightarrow E[L[Y \leftarrow U]]$

fluffy ::= inl (pair 99 "red")

$(\lambda x{:}^{\text{Animal}}.\ \text{case } x\ (c \Rightarrow \overset{}{\underset{\text{snd } c}{\text{"cat"}}})\ (d \Rightarrow \underset{d}{\text{"oddie"}})\ )$

List is either NULL or Node (Data × List)

22-2 / Variants continued...

$T ::= \ldots \mid T + T$

inl M T
inr T N

$$\frac{\Gamma \vdash M : T_1}{\Gamma \vdash \text{inl } M : T_1 + T_2} \qquad \frac{\Gamma \vdash N : T_2}{\Gamma \vdash \text{inr } N : T_1 + T_2}$$

$$\frac{\Gamma \vdash M : T_1 + T_2 \quad \Gamma[X \to T_1] \vdash N : T_R \quad \Gamma[Y \to T_2] \vdash L : T_R}{\Gamma \vdash \text{case } M \ (X \Rightarrow N)\ (Y \Rightarrow L) : T_R}$$

(λx : Animal . cat-fur x)

cat-fur : Cat → Str

x : Animal

case x

cat-fur : Ani ⇒ Str

(C ⇒ cat-fur c)

(D ⇒ error)

" 5+5 " (JS)

```
movq %rax, $5
addq %rax, $5
```

⇒ case ~~$Num 5~~)

(inl (inl (inr 5)))

JS/Py/Rkt

"uni-typed"

x ⇒ case (Num 5)

y ⇒ x + y

_ ⇒ erro

Γ ⊢ RP : TRT

_ ⇒ error

TRT ::= Num int

| Cons TRT TRT

| Vec TRT ...

ISWIM one identity fun

$\lambda x. x$ $\quad\quad\quad$ $\Lambda A. \lambda (x:A). x \;:\; \forall A. A \to A$

But in typed ISWIM there are many

$\lambda x:int. x$ $\quad\quad$ List<X> = MT | Node (X, List<X>)

$\lambda x:bool. x$ $\quad\quad$ Map<K,V>

$\lambda x:cat. x$ $\quad\quad$ PriQueue<X, X → int>

Sort< list<X>, X → X → bool >

Templates (C++) Generics (Java, C#) Polymorphism

C++/Java

$M ::= \ldots \mid (\Lambda A. M) \mid M[T]$ $\quad\quad$ $(\Lambda A. M)[T]$

$V ::= \ldots \mid \Lambda A. M$ $\quad\quad$ $X[T]$

$T ::= \ldots \mid A \mid \forall A. T$

$E ::= \ldots \mid E[T]$ $\quad\quad$ List: $\forall A. (Bool) + (A \times List)$

$E[(\Lambda A. M)[T]] \Rightarrow E[M[A \leftarrow T]]$

$$\frac{\Gamma, A \vdash M : T}{\Gamma \vdash \Lambda A. M : \forall A. T} \qquad \frac{\Gamma \vdash M : \forall A. T_1}{\Gamma \vdash M[T] : T_1[A \leftarrow T]}$$

List<X> = struct { X* data; List<X>* next; }

List<int> = struct { int data; List<int>* next; }

int data2;

M[lookup db "whatgoes"]

M< lookup db "what" >

$(x:int) \to (y:int \mid x<y) \to bool$

refinement