

Functions—Take 2

SetA \rightarrow SetB

Functions—Take 2

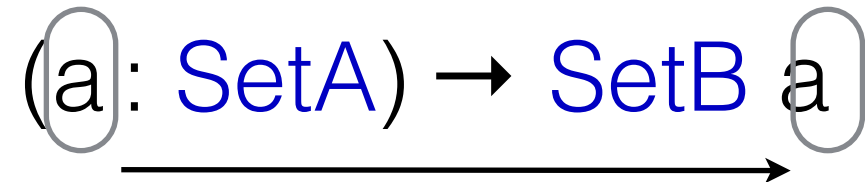
$(a : \text{SetA}) \rightarrow \text{SetB}$

Functions—Take 2

$(a: \text{SetA}) \rightarrow \text{SetB}$

Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a$



Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a$

Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a$

Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : ? \rightarrow ?$

Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : ?$

Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : ? \rightarrow ?$

Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : ? \rightarrow ?$

Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : \text{SetA} \rightarrow ?$

Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : \text{SetA} \rightarrow ?$

Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : \text{SetA} \rightarrow ?$

$\text{data SetName} : \text{Set where}$

Dependent Functions


$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : \text{SetA} \rightarrow ?$

$\text{data SetName} : \text{Set where}$

Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : \text{SetA} \rightarrow \boxed{\text{Set}}$
 $\text{data SetName} : \boxed{\text{Set}} \text{ where}$


Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : \text{SetA} \rightarrow \text{Set}$

$\text{data SetName} : \text{Set where}$

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

`SetB` : `SetA` \rightarrow `Set`


`data` `SetName` : `Set` `where`

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

SetB : SetA \rightarrow Set

data SetB : Set where

A black arrow points from the SetB in the line above to the SetB in the line below.

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : \text{SetA} \rightarrow \text{Set}$

$\text{data SetB} : \text{Set} \text{ where}$

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : \text{SetA} \rightarrow \text{Set}$

$\text{data SetB} : \text{Set} \text{ where}$

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : \text{SetA} \rightarrow \text{Set}$

$\text{data SetB} : \text{SetA} \rightarrow \text{Set} \text{ where}$

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

$\text{SetB} : \text{SetA} \rightarrow \text{Set}$

$\text{data } \text{SetB} : \text{SetA} \rightarrow \text{Set} \text{ where}$

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

data SetB : SetA → Set where

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

`data SetB : SetA → Set where`
`constructor1...`

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

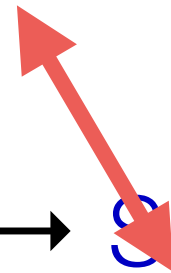
`data SetB : SetA → Set where`
`constructor1 : SetB`

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

`data SetB : SetA → Set where`

`constructor1 : SetB`



Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

`data SetA : Set where cstrA : SetA`

`data SetB : SetA → Set where`

`constructor1 : SetB` 

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

`data SetA : Set where` cstrA `: SetA`

`data SetB : SetA → Set where`

`constructor1 : SetB` cstrA

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

`data SetB : SetA → Set where`

`constructor1 : (a : SetA) → SetB a`

Dependent Types

$(a : \text{SetA}) \rightarrow \text{SetB } a$

data SetB : SetA \rightarrow Set where

constructor1 : $(a : \text{SetA}) \rightarrow \text{SetB } a$

Dependent Types

```
data SetB : SetA → Set where  
  constructor1 : (a : SetA) → SetB a
```

Dependent Types

data SetA : Set where cstrA : SetA

data SetB : SetA → Set where

constructor1 : SetB cstrA

Dependent Types

data SetA : Set where cstrA : SetA

data SetB : SetA → Set where

constructor1 : SetB cstrA

value : SetB cstrA

Dependent Types

```
data SetA : Set where cstrA : SetA
```

```
data SetB : SetA → Set where
```

```
  constructor1 : SetB cstrA
```

```
value : SetB cstrA
```

```
value = constructor1
```

Dependent Types

data SetA : Set where cstrA : SetA

data SetB : SetA → Set where

constructor1 : (a : SetA) → SetB a

Dependent Types

```
data SetA : Set where cstrA : SetA
```

```
data SetB : SetA → Set where
```

```
  constructor1 : (a : SetA) → SetB a
```

```
value : SetB cstrA
```

```
value = constructor1 cstrA
```

Dependent Types

```
data SetA : Set where cstrA : SetA
```

```
data SetB : SetA → Set where  
  constructor1 : (a : SetA) → SetB a
```

```
value : SetB cstrA
```

```
value = constructor1 cstrA
```

Dependent Types

data SetA : Set where cstrA : SetA

data SetB : SetA → Set where

constructor1 : (a : SetA) → SetB a

value : SetB cstrA

value = constructor1 cstrA

Dependent Functions

$(a : \text{SetA}) \rightarrow \text{SetB } a \rightarrow \text{SetC}$

Dependent Functions

$(a : \text{SetA}) \rightarrow (b : \text{SetB } a) \rightarrow \text{SetC}$

Dependent Functions


$(a : \text{SetA}) \rightarrow (b : \text{SetB } a) \rightarrow \text{SetC } b$



The diagram illustrates a dependent function type. The expression is $(a : \text{SetA}) \rightarrow (b : \text{SetB } a) \rightarrow \text{SetC } b$. The variables SetA , SetB , and SetC are written in blue. The variable a is in black. The variable b appears twice: once as a parameter in the second function argument and once as an argument to SetC . Both occurrences of b are enclosed in a light gray oval. A horizontal arrow points from the first b (the parameter) to the second b (the argument to SetC), indicating the dependency of the second b on the first b .

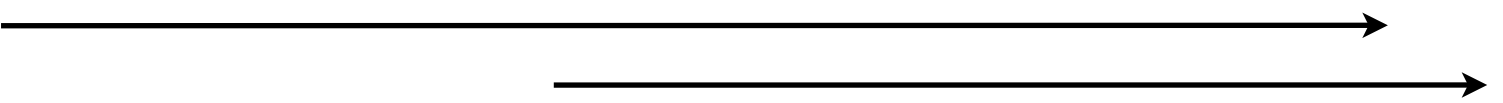
Dependent Functions

$(a : \text{SetA}) \rightarrow (b : \text{SetB } a) \rightarrow \text{SetC } a$



Dependent Functions

$(a : \text{SetA}) \rightarrow (b : \text{SetB } a) \rightarrow \text{SetC } a \ b$



Dependent Functions

$(a : \text{SetA}) \rightarrow (b : \text{SetB } a) \rightarrow \text{SetC } a \ b$

$\text{SetC} : ?$

Dependent Functions

$(a : \text{SetA}) \rightarrow (b : \text{SetB } a) \rightarrow \text{SetC } a \ b$

$\text{SetC} : ? \rightarrow ? \rightarrow ?$

Dependent Functions

$(a : \text{SetA}) \rightarrow (b : \text{SetB } a) \rightarrow \text{SetC } a \ b$

$\text{SetC} : \text{SetA} \rightarrow ? \rightarrow ?$

Dependent Functions

$(a : \text{SetA}) \rightarrow (b : \text{SetB } a) \rightarrow \text{SetC } a \text{ } b$

$\text{SetC} : \text{SetA} \rightarrow \boxed{\text{SetB}} \rightarrow ?$

Dependent Functions

$(a : \text{SetA}) \rightarrow (b : \text{SetB } a) \rightarrow \text{SetC } a \ b$

$\text{SetC} : \text{SetA} \rightarrow \text{SetB} \rightarrow \text{Set}$

Dependent Functions

$(a : \text{SetA}) \rightarrow (b : \text{SetB } a) \rightarrow \text{SetC } a \ b$

$\text{SetC} : \text{SetA} \rightarrow \text{SetB} \rightarrow \text{Set}$

Dependent Types

```
data Name : (a : A) → (b : B a) → Set where  
  constructor1 : (a : A) → (b : B a) → Name a b  
  constructor2 : (a : A) → (b : B a) → Name a b
```

Dependent Types

```
data Name : (a : A) → (b : B a) → Set where  
  constructor1 : (a : A) → (b : B a) → Name a b  
  constructor2 : (a : A) → (b : B a) → Name a b
```

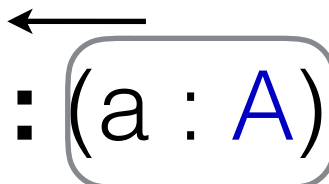
Dependent Types

`data Name : (a : A) → (b : B a) → Set where`

`constructor1 : (a : A) → (b : B a) → Name a b`

`constructor2 : (a : A) → (b : B a) → Name a b`

Dependent Types


data Name : (a : A) → (b : B a) → Set where
constructor1 : (a : A) → (b : B a) → Name a b
constructor2 : (a : A) → (b : B a) → Name a b

Dependent Types

```
data Name (a : A) : (b : B a) → Set where  
  constructor1 : (b : B a) → Name a b  
  constructor2 : (b : B a) → Name a b
```

Dependent Types

```
data Name (a : A) : (b : B a) → Set where  
  constructor1 : (b : B a) → Name a b  
  constructor2 : (b : B a) → Name a b
```

Constructors Limit Indexes

data Name : $(a : A) \rightarrow (b : B\ a) \rightarrow \text{Set}$ where

constructor1 : $(a : A) \rightarrow (b : B\ a) \rightarrow \text{Name}\ a\ b$

constructor2 : $(a : A) \rightarrow (b : B\ a) \rightarrow \text{Name}\ a\ b$