

$F \vdash D$ ^{declarations}

$d ::= \text{Let } x = e;;$

$D ::= e \mid d \ D$

$\Delta ::= [x \mapsto v, \dots]$

$$\boxed{\Delta \vdash e \Rightarrow v}$$

$$\boxed{\Delta \vdash D \Rightarrow \Delta'}$$

Let $f = \text{Function } a \rightarrow a + 1;;$

Let $n = f \ 2;;$

$$\frac{}{\Delta \vdash e \Rightarrow \Delta}$$

$$\frac{\Delta(x) = v}{\Delta \vdash x \Rightarrow v}$$

$$\frac{d = (\text{Let } x = e;;) \quad \Delta \vdash e \Rightarrow v \quad \Delta \parallel [x \mapsto v] = \Delta' \quad \Delta' \vdash D \Rightarrow \Delta''}{\Delta \vdash d \ D \Rightarrow \Delta''}$$

$TF \vdash \text{DRSA} \approx C$

auto $a = 5;$

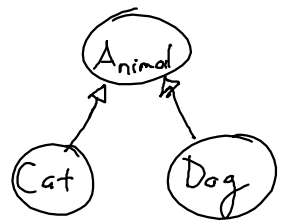
$ESTF \vdash \text{DRSA} \vee O \approx C++$

```
List<Animal> l1 = new ArrayList<Animal>();
```

```
List<Dog> l2 = l1;
```

```
l1.add(new Cat());
```

```
Dog d = l2.get(0);
```



```
List<Dog> l1 = new ArrayList<Dog>();
```

```
List<Animal> l2 = l1;
```

```
l2.add(new Cat());
```

```
Dog d = l1.get(0);
```

Dog <: Animal

List<Animal> ? List<Dog>

List<Dog> ? List<Animal>

List<Dog> ≈ { add: Dog → unit;
get: int → Dog }

List<Animal> ≈ { add: Animal → unit;
get: int → Animal }

only a problem
for mutable lists



int <: int	Dog <: Animal
int → Dog <: int → Animal	

Animal <: Dog	unit <: unit
Dog → unit <: Animal → unit	

{ add: Dog → unit;
get: int → Dog; } <: { add: Animal → unit;
get: int → Animal; }

```
List<Dog> l1 = new ArrayList<>();
```

```
List<? extends Animal> l2 = l1;
```

(∃ x. x <: Animal)

```
List<? super Dog> l3;
```

(∃ x. Dog <: x)

class Dog extends Animal

class Dog : public Animal