



RCHAIN
COOPERATIVE

Rholang V 1.1

Lucius Gregory Meredith



RCHAIN
COOPERATIVE

Rholang V1.1

New and improved for-comprehension

```
for(  
  ptrn11 ← src11 & ... & ptrn1n ← src1n;  
  ...  
  ptrnm1 ← srcm1 & ... & ptrnmn ← srcmn;  
) { P }
```

where $\text{src} ::= x \mid x?! \mid x!?(a_1, \dots, a_k)$

and ‘&’ replaces the old meaning of ‘;’



Rholang V1.1

New and improved for-comprehension desugared

```
[[for(  
  ptrn11 ← x11!?( a1, ..., ak ) & ... & ptrn1n ← src1n;  
  ...  
  ptrnm1 ← srcm1 & ... & ptrnmn ← srcmn;  
){P}]]  
=  
new r11 in  
  x11!( a1, ..., ak, *r11 )  
  | [[for( ptrn11 ← r11 & ... & ptrn1n ← src1n;  
  ...  
  ptrnm1 ← srcm1 & ... & ptrnmn ← srcmn;  
){P}]]
```

removing send/recv's: x₁₁!?(a₁, ..., a_k)



Rholang V1.1

New and improved for-comprehension desugared

```
[[for(  
  ptrn11 ← x11?! & ... & ptrn1n ← src1n;  
  ...  
  ptrnm1 ← srcm1 & ... & ptrnmn ← srcmn;  
){P}]]  
  
=  
  
[[for(  
  (ptrn11, r) ← x11 & ... & ptrn1n ← src1n;  
  ...  
  ptrnm1 ← srcm1 & ... & ptrnmn ← srcmn;  
){ r!() | P}]]
```

removing recv/send's: $x_{11}?!$

where r is fresh for the whole context



Rholang V1.1

New and improved for-comprehension desugared

```
[[for(  
  ptrn11 ← x11 & ... & ptrn1n ← x1n;  
  ptrnm1 ← srcm1 & ... & ptrnmn ← srcmn;  
  ...  
  ptrnm1 ← srcm1 & ... & ptrnmn ← srcmn  
){P}]]  
=  
for(  
  ptrn11 ← x11 & ... & ptrn1n ← x1n  
)  
[[for(  
  ptrnm1 ← srcm1 & ... & ptrnmn ← srcmn;  
  ...  
  ptrnm1 ← srcm1 & ... & ptrnmn ← srcmn;  
){P}]]  
}
```

removing ;'s



Rholang V1.1

sequential output

$x!(v);P$

Allows for sequences of sends

Rholang V1.1

sequential send expressions desugared

$$\llbracket x!(v);P \rrbracket = \text{new } r \text{ in } x!((v,*r)) \mid \text{for}(_ \leftarrow r)\{ \llbracket P \rrbracket \}$$

removing ;'s

Rholang V1.1

let expressions

```
let ptrn1 ← v1 ; ... ; ptrnm ← vm in P
```

```
let ptrn1 ← v1 & ... & ptrnm ← vm in P
```

These provide immutable variables much like Scala's

```
val x = v; P
```




RCHAIN
COOPERATIVE

Rholang V1.1

let expressions desugared

```
[[let ptrn1 ← v1 ; ... ; ptrnn ← vn in P ]]  
=  
new x1 in  
  x1!(v1)  
  | for( ptrn1 ← x1 ){  
    [[let ptrn2 ← v2 ; ... ; ptrnn ← vn in P ]]  
  }
```

removing ;'s

Rholang V1.1

let expressions desugared

$$\begin{aligned} & \llbracket \text{let ptrn}_1 \leftarrow v_1 \ \& \ \dots \ \& \ \text{ptrn}_n \leftarrow v_n \ \text{in } P \rrbracket \\ & = \\ & \text{new } x_1 \ \dots \ x_n \ \text{in} \\ & \quad x_1!(v_1) \ | \ \dots \ | \ x_n!(v_n) \\ & \quad | \ \text{for}(\text{ptrn}_1 \leftarrow x_1 \ \& \ \dots \ \& \ \text{ptrn}_n \leftarrow x_n) \{ \llbracket P \rrbracket \} \end{aligned}$$

removing &'s