Extra argument for the recorder object

```
1: (%4, %5, %1, %2, %3)
  %6 = saveir!(%5, <original IR>)
  %7 = TapeConstant(%1)
  %8 = trackedargument(%5, %7, nothing, 1, $(QuoteNode(§1:%1)))
  %9 = record!(%5, %8)
  %10 = TapeConstant(%2)
  %11 = trackedargument(%5, %10, nothing, 2, $(QuoteNode(§1:%2)))
  %12 = record!(%5, %11)
  %13 = TapeConstant(%3)
  %14 = trackedargument(%5, %13, nothing, 2, $(QuoteNode(§1:%3)))
  %15 = record!(%5, %14)
  %16 = TapeConstant(Main.rand)
  %17 = Base.tuple()
  %18 = trackedcall(%5, %16, %17, $(QuoteNode(§1:%4)))
  %19 = record!(%5, %18)
  %20 = TapeConstant(Main.:<)
  %21 = trackedvariable(%5, $(QuoteNode(%4)), %19)
  %22 = trackedvariable(%5, $(QuoteNode(%3)), %3)
  %23 = Base.tuple(%21, %22)
  %24 = trackedcall(%5, %20, %23, $(QuoteNode(§1:%5)))
  %25 = record!(%5, %24)
  %26 = Base.tuple()
  %27 = trackedvariable(%5, $(QuoteNode(%5)), %25)
  %28 = trackedjump(%5, 2, %26, %27, $(QuoteNode(§1:&1)))
  %29 = trackedvariable(%5, $(QuoteNode(%2)), %2)
  %30 = trackedreturn(%5, %29, $(QuoteNode(§1:&2)))
  br 2 (%28) unless %25
  br 3 (%2, %30)
2: (%31)
  %32 = record!(%5, %31)
  %33 = TapeConstant(Main.:+)
  %34 = trackedvariable(%5, $(QuoteNode(%2)), %2)
  %35 = Base.tuple(%34, $(QuoteNode((1))))
  %36 = trackedcall(%5, %33, %35, $(QuoteNode(§2:%6)))
  %37 = record!(%5, %36)
  %38 = TapeConstant(Main.geom)
  %39 = trackedvariable(%5, $(QuoteNode(%6)), %37)
  %40 = trackedvariable(%5, $(QuoteNode(%3)), %3)
  %41 = Base.tuple(%39, %40)
  %42 = trackedcall(%5, %38, %41, $(QuoteNode(§2:%7)))
  %43 = record!(%5, %42)
  %44 = trackedvariable(%5, $(QuoteNode(%7)), %43)
  %45 = trackedreturn(%5, %44, $(QuoteNode(§2:&1)))
  br 3 (%43, %45)
3: (%46, %47)
  %48 = record!(%5, %47)
  return %46
```

Original IR, stored in recorder

All arguments are recorded as constants in special nodes

```
1: (%1, %2, %3)
  %4 = Main.rand()
  %5 = %4 < %3
  br 2 unless %5
  return %2
```

Jumps and returns are passed down to the next block

Actual jump is recorded in next block

First argument

Called function

Argument tuple

Function call expression

SSA statements are translated to trackedcall calls

```
2:
  %6 = %2 + 1
  %7 = Main.geom(%6, %3)
  return %7
```