

State 1

```
equal = v12:False
+- val v7 = v7:Nil
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Nil
  +- [] case_List = v11:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
      +- [a: v8, b: v9] val v2 = v2:False
```

State 1

```
equal
x: Nil
y: Cons(Zero, Nil)
= case x of
  Nil ->
    case y of
      Nil ->
        Cons(a: Zero, b: Nil) -> False
      Cons(x':, y':) ->
```

False

State 1

```
equal = v12:False
+- val v7 = v7:Nil
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Nil
  +- [] case_List = v11:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
      +- [a: v8, b: v9] val v2 = v2:False
```

State 1 \rightarrow State 2

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [] case_List = v11:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] val v2 = v2:False
```

State 2

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Zero
        |   +- [] case_Nat = v15:True
        |     +- y = v8:Zero
        |     +- [] val v6 = v6:True
      +- [] equal = v18:False
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:False
          +- x = v14:Nil
          +- [] case_List = v17:False
            +- y = v9:Nil
            +- [] val v3 = v3:False
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
```

State 2

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        | +- x = v13:Zero
        | +- [] case_Nat = v15:True
        | +- y = v8:Zero
        | +- [] val v6 = v6:True
      +- [] equal = v18:False
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:False
          +- x = v14:Nil
          +- [] case_List = v17:False
            +- y = v9:Nil
            +- [] val v3 = v3:False
      [] case_List = v11:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
  +- [a: v8, b: v9] val v2 = v2:False
```

State 2

```
equal
x: Cons(Zero, Nil)
y: Cons(Zero, Nil)
= case x of
  Nil ->
  Cons(x': Zero, y': Nil) ->
    case y of
      Nil ->
      Cons(a: Zero, b: Nil) ->
        case equal_nat(Zero, Zero) ... True of
          True ->
            equal
            x: Nil
            y: Nil
            = case x of
              Nil ->
              case y of
                Nil -> False
                Cons(a:, b:) ->
                  Cons(x'::, y'::) ->
                  False ->
```

False

State 2

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Zero
        |   +- [] case_Nat = v15:True
        |     +- y = v8:Zero
        |     +- [] val v6 = v6:True
      +- [] equal = v18:False
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:False
          +- x = v14:Nil
          +- [] case_List = v17:False
            +- y = v9:Nil
            +- [] val v3 = v3:False
      [] case_List = v11:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
  +- [a: v8, b: v9] val v2 = v2:False
```

State 2 → State 3

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Zero
        |   +- [] case_Nat = v15:True
        |     +- y = v8:Zero
        |     +- [] val v6 = v6:True
      +- [] equal = v18:False
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:False
          +- x = v14:Nil
          +- [] case_List = v17:False
            +- y = v9:Nil
            +- [] val v3 = v3:True
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
```

State 2 \rightarrow State 3

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Zero
        |   +- [] case_Nat = v15:True
        |     +- y = v8:Zero
        |     +- [] val v6 = v6:True
      +- [] equal = v18:False
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:False
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
      [] case_List = v11:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
  +- [a: v8, b: v9] val v2 = v2:False
```

State 2 \rightarrow State 3

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Zero
        |   +- [] case_Nat = v15:True
        |     +- y = v8:Zero
        |     +- [] val v6 = v6:True
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
```

State 2 \rightarrow State 3

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:True
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        | +- x = v13:Zero
        | +- [] case_Nat = v15:True
        | +- y = v8:Zero
        | +- [] val v6 = v6:True
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
      [] case_List = v11:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
  +- [a: v8, b: v9] val v2 = v2:False
```

State 2 \rightarrow State 3

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:True
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:True
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        | +- x = v13:Zero
        | +- [] case_Nat = v15:True
        | +- y = v8:Zero
        | +- [] val v6 = v6:True
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
      [] case_List = v11:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
  +- [a: v8, b: v9] val v2 = v2:False
```

State 3

```
equal = v12:True
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:True
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:True
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:True
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        | +- x = v13:Zero
        | +- [] case_Nat = v15:True
        | +- y = v8:Zero
        | +- [] val v6 = v6:True
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
      [] case_List = v11:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
  +- [a: v8, b: v9] val v2 = v2:False
```

State 3

```
equal = v12:True
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:True
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:True
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:True
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Zero
        |   +- [] case_Nat = v15:True
        |     +- y = v8:Zero
        |     +- [] val v6 = v6:True
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
```

State 3

```
equal
x: Cons(Zero, Nil)
y: Cons(Zero, Nil)
= case x of
  Nil ->
  Cons(x': Zero, y': Nil) ->
    case y of
      Nil ->
      Cons(a: Zero, b: Nil) ->
        case equal_nat(Zero, Zero) ... True of
          True ->
            equal
            x: Nil
            y: Nil
            = case x of
              Nil ->
              case y of
                Nil -> True
                Cons(a:, b:) ->
                  Cons(x'::, y'::) ->
                    False ->
```

True

State 3

```
equal = v12:True
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:True
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:True
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:True
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Zero
        |   +- [] case_Nat = v15:True
        |     +- y = v8:Zero
        |     +- [] val v6 = v6:True
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
```

State 3 → State 4

```
equal = v12:True
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:True
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:True
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:True
      +- equal_nat = v16:True
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Succ(v21:Zero)
        |   +- [] case_Nat = v15:True
        |     +- y = v8:Zero
        |     +- [] val v6 = v6:True
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
```

State 4

```
equal = v12:True
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:True
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:True
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:True
      +- equal_nat = v16:True
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Succ(v21:Zero)
        |   +- [x': v21] case_Nat = v22:True
        |     +- y = v8:Zero
        |     +- [] val v4 = v4:True
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
[] case_Nat = v15:True
+- y = v8:Zero
+- [] val v6 = v6:True
```

State 4

```
equal = v12:True
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:True
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:True
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:True
      +- equal_nat = v16:True
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Succ(v21:Zero)
        |   +- [x': v21] case_Nat = v22:True
        |     +- y = v8:Zero
        |     +- [] val v4 = v4:True
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
[] case_Nat = v15:True
+- y = v8:Zero
+- [] val v6 = v6:True
```

State 4

```
equal
x: Cons(Succ(Zero), Nil)
y: Cons(Zero, Nil)
= case x of
  Nil ->
  Cons(x': Succ(Zero), y': Nil) ->
    case y of
      Nil ->
      Cons(a: Zero, b: Nil) ->
        case equal_nat
          x: Succ(Zero)
          y: Zero
          = case x of
            Zero ->
            Succ(x': Zero) ->
              case y of
                Zero -> True
                Succ(a:) ->
  True ->
    equal
    x: Nil
    y: Nil
    = case x of
      Nil ->
      case y of
        Nil -> True
        Cons(a:, b:) ->
        Cons(x', y') ->
  False ->
```

True of

State 4

```
equal = v12:True
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:True
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:True
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:True
      +- equal_nat = v16:True
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Succ(v21:Zero)
        |   +- [x': v21] case_Nat = v22:True
        |     +- y = v8:Zero
        |     +- [] val v4 = v4:True
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
[] case_Nat = v15:True
+- y = v8:Zero
+- [] val v6 = v6:True
```

State 4 \rightarrow State 5

```
equal = v12:True
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:True
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:True
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:True
      +- equal_nat = v16:True
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Succ(v21:Zero)
        |   +- [x': v21] case_Nat = v22:True
        |     +- y = v8:Zero
        |     +- [] val v4 = v4:False
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
[] case_Nat = v15:True
+- y = v8:Zero
+- [] val v6 = v6:True
```

State 4 → State 5

```
equal = v12:True
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:True
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:True
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:True
      +- equal_nat = v16:True
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        |   +- x = v13:Succ(v21:Zero)
        |   +- [x': v21] case_Nat = v22:False
        |     +- y = v8:Zero
        |     +- [] val v4 = v4:False
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
[] case_Nat = v15:True
+- y = v8:Zero
+- [] val v6 = v6:True
```

State 4 → State 5

```
equal = v12:True
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:True
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:True
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:True
      +- equal_nat = v16:False
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:False
        | +- x = v13:Succ(v21:Zero)
        | +- [x': v21] case_Nat = v22:False
        | +- y = v8:Zero
        | +- [] val v4 = v4:False
      +- [] equal = v18:True
        +- y' = v14:Nil
        +- b = v9:Nil
        +- [x: v14, y: v9] case_List = v18:True
          +- x = v14:Nil
          +- [] case_List = v17:True
            +- y = v9:Nil
            +- [] val v3 = v3:True
    [] case_List = v11:False
  +- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
[] case_Nat = v15:True
+- y = v8:Zero
+- [] val v6 = v6:True
```

State 4 → State 5

```
equal = v12:True
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:True
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:True
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:False
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:False
        |   +- x = v13:Succ(v21:Zero)
        |   +- [x': v21] case_Nat = v22:False
        |   +- y = v8:Zero
        |   +- [] val v4 = v4:False
      +- [] val v0 = v0:False
[] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
[] case_Nat = v15:True
+- y = v8:Zero
+- [] val v6 = v6:True
[] equal = v18:True
+- y' = v14:Nil
+- b = v9:Nil
+- [x: v14, y: v9] case_List = v18:True
  +- x = v14:Nil
  +- [] case_List = v17:True
    +- y = v9:Nil
    +- [] val v3 = v3:True
```

State 4 \rightarrow State 5

```
equal = v12:True
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:True
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:False
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:False
        |   +- x = v13:Succ(v21:Zero)
        |   +- [x': v21] case_Nat = v22:False
        |   +- y = v8:Zero
        |   +- [] val v4 = v4:False
      +- [] val v0 = v0:False
[] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
[] case_Nat = v15:True
+- y = v8:Zero
+- [] val v6 = v6:True
[] equal = v18:True
+- y' = v14:Nil
+- b = v9:Nil
+- [x: v14, y: v9] case_List = v18:True
  +- x = v14:Nil
  +- [] case_List = v17:True
    +- y = v9:Nil
    +- [] val v3 = v3:True
```

State 5

```
equal = v12:False
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:False
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:False
        |   +- x = v13:Succ(v21:Zero)
        |   +- [x': v21] case_Nat = v22:False
        |   +- y = v8:Zero
        |   +- [] val v4 = v4:False
      +- [] val v0 = v0:False
[] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
[] case_Nat = v15:True
+- y = v8:Zero
+- [] val v6 = v6:True
[] equal = v18:True
+- y' = v14:Nil
+- b = v9:Nil
+- [x: v14, y: v9] case_List = v18:True
  +- x = v14:Nil
  +- [] case_List = v17:True
    +- y = v9:Nil
    +- [] val v3 = v3:True
```

State 5

```
equal = v12:False
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:False
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:False
        |   +- x = v13:Succ(v21:Zero)
        |   +- [x': v21] case_Nat = v22:False
        |     +- y = v8:Zero
        |     +- [] val v4 = v4:False
      +- [] val v0 = v0:False
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Nil)
+- [a: v8, b: v9] val v2 = v2:False
[] case_Nat = v15:True
+- y = v8:Zero
+- [] val v6 = v6:True
[] equal = v18:True
+- y' = v14:Nil
+- b = v9:Nil
+- [x: v14, y: v9] case_List = v18:True
  +- x = v14:Nil
  +- [] case_List = v17:True
    +- y = v9:Nil
    +- [] val v3 = v3:True
```

State 5

```
equal
x: Cons(Succ(Zero), Nil)
y: Cons(Zero, Nil)
= case x of
  Nil ->
  Cons(x': Succ(Zero), y': Nil) ->
    case y of
      Nil ->
      Cons(a: Zero, b: Nil) ->
        case equal_nat
          x: Succ(Zero)
          y: Zero
          = case x of
            Zero ->
            Succ(x': Zero) ->
              case y of
                Zero -> False
                Succ(a:) ->
                  True ->
                  False
```

False

State 5

```
equal = v12:False
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Nil)
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Nil)
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:False
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:False
        |   +- x = v13:Succ(v21:Zero)
        |   +- [x': v21] case_Nat = v22:False
        |     +- y = v8:Zero
        |     +- [] val v4 = v4:False
      +- [] val v0 = v0:False
    [] case_List = v11:False
  +- y = v10:Cons(v8:Zero,v9:Nil)
  +- [a: v8, b: v9] val v2 = v2:False
  [] case_Nat = v15:True
  +- y = v8:Zero
  +- [] val v6 = v6:True
  [] equal = v18:True
  +- y' = v14:Nil
  +- b = v9:Nil
  +- [x: v14, y: v9] case_List = v18:True
    +- x = v14:Nil
    +- [] case_List = v17:True
      +- y = v9:Nil
      +- [] val v3 = v3:True
```

State 6

```
equal = v12:False
+- val v7 = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Succ(v21:Zero),v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:False
        | +- x' = v13:Succ(v21:Zero)
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:False
        |   +- x = v13:Succ(v21:Zero)
        |   +- [x': v21] case_Nat = v22:False
        |     +- y = v8:Zero
        |     +- [] val v4 = v4:False
      +- [] val v0 = v0:False
    [] case_List = v11:False
  +- y = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
  +- [a: v8, b: v9] val v2 = v2:False
  [] case_Nat = v15:True
  +- y = v8:Zero
  +- [] val v6 = v6:True
  [] equal = v18:True
  +- y' = v14:Nil
  +- b = v9:Cons(v23:Zero,v24:Nil)
  +- [x: v14, y: v9] case_List = v18:True
    +- x = v14:Nil
    +- [] case_List = v17:True
      +- y = v9:Cons(v23:Zero,v24:Nil)
      +- [] val v3 = v3:True
```

State 6 → State 7

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:False
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:False
        | +- x = v13:Zero
        | +- [x': v21] case_Nat = v22:False
        | +- y = v8:Zero
        | +- [] val v4 = v4:False
      +- [] val v0 = v0:False
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
+- [a: v8, b: v9] val v2 = v2:False
[] case_Nat = v15:True
+- y = v8:Zero
+- [] val v6 = v6:True
[] equal = v18:True
+- y' = v14:Nil
+- b = v9:Cons(v23:Zero,v24:Nil)
+- [x: v14, y: v9] case_List = v18:True
  +- x = v14:Nil
  +- [] case_List = v17:True
    +- y = v9:Cons(v23:Zero,v24:Nil)
    +- [] val v3 = v3:True
```

State 6 → State 7

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        | +- x = v13:Zero
        | +- [] case_Nat = v15:True
        | +- y = v8:Zero
        | +- [] val v6 = v6:True
      +- [] val v0 = v0:False
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
+- [a: v8, b: v9] val v2 = v2:False
[x': v21] case_Nat = v22:False
+- y = v8:Zero
+- [] val v4 = v4:False
[] equal = v18:True
+- y' = v14:Nil
+- b = v9:Cons(v23:Zero,v24:Nil)
+- [x: v14, y: v9] case_List = v18:True
  +- x = v14:Nil
  +- [] case_List = v17:True
    +- y = v9:Cons(v23:Zero,v24:Nil)
    +- [] val v3 = v3:True
```

State 6 → State 7

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        | +- x = v13:Zero
        | +- [] case_Nat = v15:True
        | +- y = v8:Zero
        | +- [] val v6 = v6:True
      +- [] val v0 = v0:False
  [] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
+- [a: v8, b: v9] val v2 = v2:False
[x': v21] case_Nat = v22:False
+- y = v8:Zero
+- [] val v4 = v4:False
[] equal = v18:True
+- y' = v14:Nil
+- b = v9:Cons(v23:Zero,v24:Nil)
+- [x: v14, y: v9] case_List = v18:True
  +- x = v14:Nil
  +- [] case_List = v17:False
    +- y = v9:Cons(v23:Zero,v24:Nil)
    +- [a: v23, b: v24] val v2 = v2:False
[] val v3 = v3:True
```

State 6 → State 7

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        | +- x = v13:Zero
        | +- [] case_Nat = v15:True
        | +- y = v8:Zero
        | +- [] val v6 = v6:True
      +- [] val v0 = v0:False
[] case_List = v11:False
+- y = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
+- [a: v8, b: v9] val v2 = v2:False
[x': v21] case_Nat = v22:False
+- y = v8:Zero
+- [] val v4 = v4:False
[] equal = v18:False
+- y' = v14:Nil
+- b = v9:Cons(v23:Zero,v24:Nil)
+- [x: v14, y: v9] case_List = v18:False
  +- x = v14:Nil
  +- [] case_List = v17:False
    +- y = v9:Cons(v23:Zero,v24:Nil)
    +- [a: v23, b: v24] val v2 = v2:False
[] val v3 = v3:True
```

State 7

```
equal = v12:False
+- val v7 = v7:Cons(v13:Zero,v14:Nil)
+- val v10 = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
+- [x: v7, y: v10] case_List = v12:False
  +- x = v7:Cons(v13:Zero,v14:Nil)
  +- [x': v13, y': v14] case_List = v20:False
    +- y = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
    +- [a: v8, b: v9] case_Bool = v19:False
      +- equal_nat = v16:True
        | +- x' = v13:Zero
        | +- a = v8:Zero
        | +- [x: v13, y: v8] case_Nat = v16:True
        | +- x = v13:Zero
        | +- [] case_Nat = v15:True
        | +- y = v8:Zero
        | +- [] val v6 = v6:True
      +- [] equal = v18:False
        +- y' = v14:Nil
        +- b = v9:Cons(v23:Zero,v24:Nil)
        +- [x: v14, y: v9] case_List = v18:False
          +- x = v14:Nil
          +- [] case_List = v17:False
            +- y = v9:Cons(v23:Zero,v24:Nil)
            +- [a: v23, b: v24] val v2 = v2:False
      [] case_List = v11:False
    +- y = v10:Cons(v8:Zero,v9:Cons(v23:Zero,v24:Nil))
  +- [a: v8, b: v9] val v2 = v2:False
  [x': v21] case_Nat = v22:False
  +- y = v8:Zero
  +- [] val v4 = v4:False
  [] val v0 = v0:False
  [] val v3 = v3:True
```