

StructuralSemantics.pdf



Juanma21_



Teoría de Los Lenguajes de Programación



4º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga



Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos y tiene como una garantía de hasta 100.000 euros por depositante. Consulta más información en inglés.



13/1/25, 13:45

StructuralSemantics.hs

```
1  --
2  -- Structural Operational Semantics for WHILE 2020.
3  -- Examen de Lenguajes de Programación. UMA.
4  --
5  -- Apellidos, Nombre: Cardeñosa Borrego, Juan Manuel
6  --
7
8 module StructuralSemantics where
9
10
11 --
12 -- NO MODIFICAR EL CODIGO DE ABAJO
13 --
14
15 import Data.List.HT (takeUntil)
16
17 import While20
18
19 -- representation of configurations for While
20
21 data Config = Inter Stm State -- <S, s>
22           | Final State      -- s
23           | Stuck Stm State  -- <S, s>
24
25 isFinal :: Config -> Bool
26 isFinal (Final _) = True
27 isFinal _          = False
28
29 isInter :: Config -> Bool
30 isInter (Inter _) = True
31 isInter _          = False
32
33 isStuck :: Config -> Bool
34 isStuck (Stuck _) = True
35 isStuck _          = False
36
37 -- representation of the transition relation <S, s> -> s'
38
39 sosStm :: Config -> Config
40
41 -- x := a
42
43 sosStm (Inter (Ass x a) s) = Final (update s x (aVal a s))
44   where
45     update s x v y = if x == y then v else s y
46
47 -- skip
48
49 sosStm (Inter Skip s) = Final s
50
51 -- s1; s2
```

Consulta condiciones aquí



do your thing

WUOLAH

localhost:50364/4f34bec0-b08c-4ee4-9aa2-262692922db8/

1/3

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

```

52
53 sosStm (Inter (Comp ss1 ss2) s)
54   | isFinal next = Inter ss2 s'
55   where
56     next = sosStm (Inter ss1 s)
57     Final s' = next
58
59 sosStm (Inter (Comp ss1 ss2) s)
60   | isStuck next = Stuck (Comp stm ss2) s'
61   where
62     next = sosStm (Inter ss1 s)
63     Stuck stm s' = next
64
65 sosStm (Inter (Comp ss1 ss2) s)
66   | isInter next = Inter (Comp ss1' ss2) s'
67   where
68     next = sosStm (Inter ss1 s)
69     Inter ss1' s' = next
70
71 -- if b then s1 else s2
72
73 sosStm (Inter (If b ss1 ss2) s)
74   | bVal b s = Inter ss1 s
75
76 sosStm (Inter (If b ss1 ss2) s)
77   | not (bVal b s) = Inter ss2 s
78
79 -----
80 -- NO MODIFICAR EL CODIGO DE ARRIBA
81 -----
82
83 {-}
84
85 [swap sos] -----
86           <Swap x y, s> => <z:= y; y := x; x := z, s'>
87 -}
88
89 -- Sentencia Swap
90
91 sosStm (Inter (Swap x y) s) = Inter stm s''
92   where
93     Inter stm s' = sosStm (Inter ((Comp (Comp (Ass z (V "y")) (Ass y (V "z")))) (Ass x (V "z")))) s)
94     z = "z"
95     -- Para re establecer el valor de z, si fuese necesario, sino, sería la salida (Inter stm s')
96     Final s'' = sosStm (Inter (Ass z (N (s z))) s')
97
98 {-}
99
100 [for sos] -----

```

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](#)

Que te den 10 € para gastar
es una fantasía.
ING lo hace realidad.

Abre la **Cuenta NoCuenta** con el código
[WUOLAH10](#), haz tu primer pago y llévate 10 €.

Quiero el cash

[Consulta condiciones aquí](#)



do your thing

Teoría de Los Lenguajes de P...



Comparte estos flyers en tu clase y consigue más dinero y recompensas



- 1 Imprime esta hoja
- 2 Recorta por la mitad
- 3 Coloca en un lugar visible para que tus compañeros puedan escanear y acceder a apuntes
- 4 Llévate dinero por cada descarga de los documentos descargados a través de tu QR

WUOLAH



```
102     <For s1 b s2 s3, s> => <if b then (s1;s3,s2;(For Skip b s2 s3)) else Skip,  
103     s>  
104  
105 -}  
106  
107 -- For  
108 sosStm (Inter (For s1 b s2 s3) s) = Inter (If b (Comp (Comp (Comp (s1) (s3)) (s2)) (For Skip  
109 b s2 s3)) (Skip)) s  
110  
111
```

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos y tiene como una garantía de hasta 100.000 euros por depositante. Consulta más información en inglés.

13/1/25, 13:45

StructuralSemantics.hs

```
1  --
2  -- Structural Operational Semantics for WHILE 2020.
3  -- Examen de Lenguajes de Programación. UMA.
4  --
5  -- Apellidos, Nombre: Cardeñosa Borrego, Juan Manuel
6  --
7
8 module StructuralSemantics where
9
10
11 --
12 -- NO MODIFICAR EL CODIGO DE ABAJO
13 --
14
15 import Data.List.HT (takeUntil)
16
17 import While20
18
19 -- representation of configurations for While
20
21 data Config = Inter Stm State -- <S, s>
22           | Final State      -- s
23           | Stuck Stm State  -- <S, s>
24
25 isFinal :: Config -> Bool
26 isFinal (Final _) = True
27 isFinal _          = False
28
29 isInter :: Config -> Bool
30 isInter (Inter _) = True
31 isInter _          = False
32
33 isStuck :: Config -> Bool
34 isStuck (Stuck _) = True
35 isStuck _          = False
36
37 -- representation of the transition relation <S, s> -> s'
38
39 sosStm :: Config -> Config
40
41 -- x := a
42
43 sosStm (Inter (Ass x a) s) = Final (update s x (aVal a s))
44   where
45     update s x v y = if x == y then v else s y
46
47 -- skip
48
49 sosStm (Inter Skip s) = Final s
50
51 -- s1; s2
```

Consulta condiciones aquí



do your thing

WUOLAH

localhost:50364/4f34bec0-b08c-4ee4-9aa2-262692922db8/

1/3

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

```

52
53 sosStm (Inter (Comp ss1 ss2) s)
54   | isFinal next = Inter ss2 s'
55   where
56     next = sosStm (Inter ss1 s)
57     Final s' = next
58
59 sosStm (Inter (Comp ss1 ss2) s)
60   | isStuck next = Stuck (Comp stm ss2) s'
61   where
62     next = sosStm (Inter ss1 s)
63     Stuck stm s' = next
64
65 sosStm (Inter (Comp ss1 ss2) s)
66   | isInter next = Inter (Comp ss1' ss2) s'
67   where
68     next = sosStm (Inter ss1 s)
69     Inter ss1' s' = next
70
71 -- if b then s1 else s2
72
73 sosStm (Inter (If b ss1 ss2) s)
74   | bVal b s = Inter ss1 s
75
76 sosStm (Inter (If b ss1 ss2) s)
77   | not (bVal b s) = Inter ss2 s
78
79 -----
80 -- NO MODIFICAR EL CODIGO DE ARRIBA
81 -----
82
83 {-}
84
85 [swap sos] -----
86           <Swap x y, s> => <z:= y; y := x; x := z, s'>
87 -}
88
89 -- Sentencia Swap
90
91 sosStm (Inter (Swap x y) s) = Inter stm s''
92   where
93     Inter stm s' = sosStm (Inter ((Comp (Comp (Ass z (V "y")) (Ass y (V "z")))) (Ass x (V "z")))) s)
94     z = "z"
95     -- Para re establecer el valor de z, si fuese necesario, sino, sería la salida (Inter stm s')
96     Final s'' = sosStm (Inter (Ass z (N (s z))) s')
97
98 {-}
99
100 [for sos] -----

```

```
102          <For s1 b s2 s3, s> => <if b then (s1;s3,s2;(For Skip b s2 s3)) else Skip,  
103          s>  
104      -}  
105  
106      -- For  
107      sosStm (Inter (For s1 b s2 s3) s) = Inter (If b (Comp (Comp (Comp (s1) (s3)) (s2)) (For Skip  
108      b s2 s3)) (Skip)) s  
109  
110  
111
```

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos y tiene como una garantía de hasta 100.000 euros por depositante. Consulta más información en inglés.



13/1/25, 13:45

StructuralSemantics.hs

```
1  --
2  -- Structural Operational Semantics for WHILE 2020.
3  -- Examen de Lenguajes de Programación. UMA.
4  --
5  -- Apellidos, Nombre: Cardeñosa Borrego, Juan Manuel
6  --
7
8 module StructuralSemantics where
9
10
11 --
12 -- NO MODIFICAR EL CODIGO DE ABAJO
13 --
14
15 import Data.List.HT (takeUntil)
16
17 import While20
18
19 -- representation of configurations for While
20
21 data Config = Inter Stm State -- <S, s>
22           | Final State      -- s
23           | Stuck Stm State  -- <S, s>
24
25 isFinal :: Config -> Bool
26 isFinal (Final _) = True
27 isFinal _          = False
28
29 isInter :: Config -> Bool
30 isInter (Inter _) = True
31 isInter _          = False
32
33 isStuck :: Config -> Bool
34 isStuck (Stuck _) = True
35 isStuck _          = False
36
37 -- representation of the transition relation <S, s> -> s'
38
39 sosStm :: Config -> Config
40
41 -- x := a
42
43 sosStm (Inter (Ass x a) s) = Final (update s x (aVal a s))
44   where
45     update s x v y = if x == y then v else s y
46
47 -- skip
48
49 sosStm (Inter Skip s) = Final s
50
51 -- s1; s2
```

Consulta condiciones aquí



do your thing

WUOLAH

localhost:50364/4f34bec0-b08c-4ee4-9aa2-262692922db8/

1/3

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

```

52
53 sosStm (Inter (Comp ss1 ss2) s)
54   | isFinal next = Inter ss2 s'
55   where
56     next = sosStm (Inter ss1 s)
57     Final s' = next
58
59 sosStm (Inter (Comp ss1 ss2) s)
60   | isStuck next = Stuck (Comp stm ss2) s'
61   where
62     next = sosStm (Inter ss1 s)
63     Stuck stm s' = next
64
65 sosStm (Inter (Comp ss1 ss2) s)
66   | isInter next = Inter (Comp ss1' ss2) s'
67   where
68     next = sosStm (Inter ss1 s)
69     Inter ss1' s' = next
70
71 -- if b then s1 else s2
72
73 sosStm (Inter (If b ss1 ss2) s)
74   | bVal b s = Inter ss1 s
75
76 sosStm (Inter (If b ss1 ss2) s)
77   | not (bVal b s) = Inter ss2 s
78
79 -----
80 -- NO MODIFICAR EL CODIGO DE ARRIBA
81 -----
82
83 {-}
84
85 [swap sos] -----
86           <Swap x y, s> => <z:= y; y := x; x := z, s'>
87 -}
88
89 -- Sentencia Swap
90
91 sosStm (Inter (Swap x y) s) = Inter stm s''
92   where
93     Inter stm s' = sosStm (Inter ((Comp (Comp (Ass z (V "y")) (Ass y (V "z")))) (Ass x (V "z")))) s)
94     z = "z"
95     -- Para re establecer el valor de z, si fuese necesario, sino, sería la salida (Inter stm s')
96     Final s'' = sosStm (Inter (Ass z (N (s z))) s')
97
98 {-}
99
100 [for sos] -----

```

```
102     <For s1 b s2 s3, s> => <if b then (s1;s3,s2;(For Skip b s2 s3)) else Skip,  
103     s>  
104  
105 -}  
106  
107 -- For  
108 sosStm (Inter (For s1 b s2 s3) s) = Inter (If b (Comp (Comp (Comp (s1) (s3)) (s2)) (For Skip  
109 b s2 s3)) (Skip)) s  
110  
111
```

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos y tiene como una garantía de hasta 100.000 euros por depositante. Consulta más información en inglés.



13/1/25, 13:45

StructuralSemantics.hs

```
1  --
2  -- Structural Operational Semantics for WHILE 2020.
3  -- Examen de Lenguajes de Programación. UMA.
4  --
5  -- Apellidos, Nombre: Cardeñosa Borrego, Juan Manuel
6  --
7
8 module StructuralSemantics where
9
10
11 --
12 -- NO MODIFICAR EL CODIGO DE ABAJO
13 --
14
15 import Data.List.HT (takeUntil)
16
17 import While20
18
19 -- representation of configurations for While
20
21 data Config = Inter Stm State -- <S, s>
22           | Final State      -- s
23           | Stuck Stm State  -- <S, s>
24
25 isFinal :: Config -> Bool
26 isFinal (Final _) = True
27 isFinal _          = False
28
29 isInter :: Config -> Bool
30 isInter (Inter _) = True
31 isInter _          = False
32
33 isStuck :: Config -> Bool
34 isStuck (Stuck _) = True
35 isStuck _          = False
36
37 -- representation of the transition relation <S, s> -> s'
38
39 sosStm :: Config -> Config
40
41 -- x := a
42
43 sosStm (Inter (Ass x a) s) = Final (update s x (aVal a s))
44   where
45     update s x v y = if x == y then v else s y
46
47 -- skip
48
49 sosStm (Inter Skip s) = Final s
50
51 -- s1; s2
```

Consulta condiciones aquí



do your thing

WUOLAH

localhost:50364/4f34bec0-b08c-4ee4-9aa2-262692922db8/

1/3

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

```

52
53 sosStm (Inter (Comp ss1 ss2) s)
54   | isFinal next = Inter ss2 s'
55   where
56     next = sosStm (Inter ss1 s)
57     Final s' = next
58
59 sosStm (Inter (Comp ss1 ss2) s)
60   | isStuck next = Stuck (Comp stm ss2) s'
61   where
62     next = sosStm (Inter ss1 s)
63     Stuck stm s' = next
64
65 sosStm (Inter (Comp ss1 ss2) s)
66   | isInter next = Inter (Comp ss1' ss2) s'
67   where
68     next = sosStm (Inter ss1 s)
69     Inter ss1' s' = next
70
71 -- if b then s1 else s2
72
73 sosStm (Inter (If b ss1 ss2) s)
74   | bVal b s = Inter ss1 s
75
76 sosStm (Inter (If b ss1 ss2) s)
77   | not (bVal b s) = Inter ss2 s
78
79 -----
80 -- NO MODIFICAR EL CODIGO DE ARRIBA
81 -----
82
83 {-}
84
85 [swap sos] -----
86           <Swap x y, s> => <z:= y; y := x; x := z, s'>
87 -}
88
89 -- Sentencia Swap
90
91 sosStm (Inter (Swap x y) s) = Inter stm s''
92   where
93     Inter stm s' = sosStm (Inter ((Comp (Comp (Ass z (V "y")) (Ass y (V "z")))) (Ass x (V "z")))) s)
94     z = "z"
95     -- Para re establecer el valor de z, si fuese necesario, sino, sería la salida (Inter stm s')
96     Final s'' = sosStm (Inter (Ass z (N (s z))) s')
97
98 {-}
99
100 [for sos] -----

```

```
102          <For s1 b s2 s3, s> => <if b then (s1;s3,s2;(For Skip b s2 s3)) else Skip,  
103          s>  
104      -}  
105  
106      -- For  
107  sosStm (Inter (For s1 b s2 s3) s) = Inter (If b (Comp (Comp (Comp (s1) (s3)) (s2)) (For Skip  
108    b s2 s3)) (Skip)) s  
109  
110  
111
```

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos y tiene como una garantía de hasta 100.000 euros por depositante. Consulta más información en inglés.

13/1/25, 13:45

StructuralSemantics.hs

```
1  --
2  -- Structural Operational Semantics for WHILE 2020.
3  -- Examen de Lenguajes de Programación. UMA.
4  --
5  -- Apellidos, Nombre: Cardeñosa Borrego, Juan Manuel
6  --
7
8 module StructuralSemantics where
9
10
11 --
12 -- NO MODIFICAR EL CODIGO DE ABAJO
13 --
14
15 import Data.List.HT (takeUntil)
16
17 import While20
18
19 -- representation of configurations for While
20
21 data Config = Inter Stm State -- <S, s>
22           | Final State      -- s
23           | Stuck Stm State  -- <S, s>
24
25 isFinal :: Config -> Bool
26 isFinal (Final _) = True
27 isFinal _          = False
28
29 isInter :: Config -> Bool
30 isInter (Inter _) = True
31 isInter _          = False
32
33 isStuck :: Config -> Bool
34 isStuck (Stuck _) = True
35 isStuck _          = False
36
37 -- representation of the transition relation <S, s> -> s'
38
39 sosStm :: Config -> Config
40
41 -- x := a
42
43 sosStm (Inter (Ass x a) s) = Final (update s x (aVal a s))
44   where
45     update s x v y = if x == y then v else s y
46
47 -- skip
48
49 sosStm (Inter Skip s) = Final s
50
51 -- s1; s2
```

Consulta condiciones aquí



do your thing

WUOLAH

localhost:50364/4f34bec0-b08c-4ee4-9aa2-262692922db8/

1/3

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

```

52
53 sosStm (Inter (Comp ss1 ss2) s)
54   | isFinal next = Inter ss2 s'
55   where
56     next = sosStm (Inter ss1 s)
57     Final s' = next
58
59 sosStm (Inter (Comp ss1 ss2) s)
60   | isStuck next = Stuck (Comp stm ss2) s'
61   where
62     next = sosStm (Inter ss1 s)
63     Stuck stm s' = next
64
65 sosStm (Inter (Comp ss1 ss2) s)
66   | isInter next = Inter (Comp ss1' ss2) s'
67   where
68     next = sosStm (Inter ss1 s)
69     Inter ss1' s' = next
70
71 -- if b then s1 else s2
72
73 sosStm (Inter (If b ss1 ss2) s)
74   | bVal b s = Inter ss1 s
75
76 sosStm (Inter (If b ss1 ss2) s)
77   | not (bVal b s) = Inter ss2 s
78
79 -----
80 -- NO MODIFICAR EL CODIGO DE ARRIBA
81 -----
82
83 {-}
84
85 [swap sos] -----
86           <Swap x y, s> => <z:= y; y := x; x := z, s'>
87 -}
88
89 -- Sentencia Swap
90
91 sosStm (Inter (Swap x y) s) = Inter stm s''
92   where
93     Inter stm s' = sosStm (Inter ((Comp (Comp (Ass z (V "y")) (Ass y (V "z")))) (Ass x (V "z")))) s)
94     z = "z"
95     -- Para re establecer el valor de z, si fuese necesario, sino, sería la salida (Inter stm s')
96     Final s'' = sosStm (Inter (Ass z (N (s z))) s')
97
98 {-}
99
100 [for sos] -----

```

```
102          <For s1 b s2 s3, s> => <if b then (s1;s3,s2;(For Skip b s2 s3)) else Skip,  
103          s>  
104      -}  
105  
106      -- For  
107      sosStm (Inter (For s1 b s2 s3) s) = Inter (If b (Comp (Comp (Comp (s1) (s3)) (s2)) (For Skip  
108      b s2 s3)) (Skip)) s  
109  
110  
111
```

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos y tiene como una garantía de hasta 100.000 euros por depositante. Consulta más información en inglés.

13/1/25, 13:45

StructuralSemantics.hs

```
1  --
2  -- Structural Operational Semantics for WHILE 2020.
3  -- Examen de Lenguajes de Programación. UMA.
4  --
5  -- Apellidos, Nombre: Cardeñosa Borrego, Juan Manuel
6  --
7
8 module StructuralSemantics where
9
10
11 --
12 -- NO MODIFICAR EL CODIGO DE ABAJO
13 --
14
15 import Data.List.HT (takeUntil)
16
17 import While20
18
19 -- representation of configurations for While
20
21 data Config = Inter Stm State -- <S, s>
22           | Final State      -- s
23           | Stuck Stm State  -- <S, s>
24
25 isFinal :: Config -> Bool
26 isFinal (Final _) = True
27 isFinal _          = False
28
29 isInter :: Config -> Bool
30 isInter (Inter _) = True
31 isInter _          = False
32
33 isStuck :: Config -> Bool
34 isStuck (Stuck _) = True
35 isStuck _          = False
36
37 -- representation of the transition relation <S, s> -> s'
38
39 sosStm :: Config -> Config
40
41 -- x := a
42
43 sosStm (Inter (Ass x a) s) = Final (update s x (aVal a s))
44   where
45     update s x v y = if x == y then v else s y
46
47 -- skip
48
49 sosStm (Inter Skip s) = Final s
50
51 -- s1; s2
```

Consulta condiciones aquí



do your thing

WUOLAH

localhost:50364/4f34bec0-b08c-4ee4-9aa2-262692922db8/

1/3

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

```

52
53 sosStm (Inter (Comp ss1 ss2) s)
54   | isFinal next = Inter ss2 s'
55   where
56     next = sosStm (Inter ss1 s)
57     Final s' = next
58
59 sosStm (Inter (Comp ss1 ss2) s)
60   | isStuck next = Stuck (Comp stm ss2) s'
61   where
62     next = sosStm (Inter ss1 s)
63     Stuck stm s' = next
64
65 sosStm (Inter (Comp ss1 ss2) s)
66   | isInter next = Inter (Comp ss1' ss2) s'
67   where
68     next = sosStm (Inter ss1 s)
69     Inter ss1' s' = next
70
71 -- if b then s1 else s2
72
73 sosStm (Inter (If b ss1 ss2) s)
74   | bVal b s = Inter ss1 s
75
76 sosStm (Inter (If b ss1 ss2) s)
77   | not (bVal b s) = Inter ss2 s
78
79 -----
80 -- NO MODIFICAR EL CODIGO DE ARRIBA
81 -----
82
83 {-}
84
85 [swap sos] -----
86           <Swap x y, s> => <z:= y; y := x; x := z, s'>
87 -}
88
89 -- Sentencia Swap
90
91 sosStm (Inter (Swap x y) s) = Inter stm s''
92   where
93     Inter stm s' = sosStm (Inter ((Comp (Comp (Ass z (V "y")) (Ass y (V "z")))) (Ass x (V "z")))) s)
94     z = "z"
95     -- Para re establecer el valor de z, si fuese necesario, sino, sería la salida (Inter stm s')
96     Final s'' = sosStm (Inter (Ass z (N (s z))) s')
97
98 {-}
99
100 [for sos] -----

```

```
102          <For s1 b s2 s3, s> => <if b then (s1;s3,s2;(For Skip b s2 s3)) else Skip,  
103          s>  
104      -}  
105  
106      -- For  
107  sosStm (Inter (For s1 b s2 s3) s) = Inter (If b (Comp (Comp (Comp (s1) (s3)) (s2)) (For Skip  
108    b s2 s3)) (Skip)) s  
109  
110  
111
```

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos y tiene como una garantía de hasta 100.000 euros por depositista. Consulta más información en inglés.

13/1/25, 13:45

StructuralSemantics.hs

```
1  --
2  -- Structural Operational Semantics for WHILE 2020.
3  -- Examen de Lenguajes de Programación. UMA.
4  --
5  -- Apellidos, Nombre: Cardeñosa Borrego, Juan Manuel
6  --
7
8 module StructuralSemantics where
9
10
11 --
12 -- NO MODIFICAR EL CODIGO DE ABAJO
13 --
14
15 import Data.List.HT (takeUntil)
16
17 import While20
18
19 -- representation of configurations for While
20
21 data Config = Inter Stm State -- <S, s>
22           | Final State      -- s
23           | Stuck Stm State  -- <S, s>
24
25 isFinal :: Config -> Bool
26 isFinal (Final _) = True
27 isFinal _          = False
28
29 isInter :: Config -> Bool
30 isInter (Inter _) = True
31 isInter _          = False
32
33 isStuck :: Config -> Bool
34 isStuck (Stuck _) = True
35 isStuck _          = False
36
37 -- representation of the transition relation <S, s> -> s'
38
39 sosStm :: Config -> Config
40
41 -- x := a
42
43 sosStm (Inter (Ass x a) s) = Final (update s x (aVal a s))
44   where
45     update s x v y = if x == y then v else s y
46
47 -- skip
48
49 sosStm (Inter Skip s) = Final s
50
51 -- s1; s2
```

Consulta condiciones aquí



do your thing

WUOLAH

localhost:50364/4f34bec0-b08c-4ee4-9aa2-262692922db8/

1/3

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

```

52
53 sosStm (Inter (Comp ss1 ss2) s)
54   | isFinal next = Inter ss2 s'
55   where
56     next = sosStm (Inter ss1 s)
57     Final s' = next
58
59 sosStm (Inter (Comp ss1 ss2) s)
60   | isStuck next = Stuck (Comp stm ss2) s'
61   where
62     next = sosStm (Inter ss1 s)
63     Stuck stm s' = next
64
65 sosStm (Inter (Comp ss1 ss2) s)
66   | isInter next = Inter (Comp ss1' ss2) s'
67   where
68     next = sosStm (Inter ss1 s)
69     Inter ss1' s' = next
70
71 -- if b then s1 else s2
72
73 sosStm (Inter (If b ss1 ss2) s)
74   | bVal b s = Inter ss1 s
75
76 sosStm (Inter (If b ss1 ss2) s)
77   | not (bVal b s) = Inter ss2 s
78
79 -----
80 -- NO MODIFICAR EL CODIGO DE ARRIBA
81 -----
82
83 {-}
84
85 [swap sos] -----
86           <Swap x y, s> => <z:= y; y := x; x := z, s'>
87 -}
88
89 -- Sentencia Swap
90
91 sosStm (Inter (Swap x y) s) = Inter stm s''
92   where
93     Inter stm s' = sosStm (Inter ((Comp (Comp (Ass z (V "y")) (Ass y (V "z")))) (Ass x (V "z")))) s)
94     z = "z"
95     -- Para re establecer el valor de z, si fuese necesario, sino, sería la salida (Inter stm s')
96     Final s'' = sosStm (Inter (Ass z (N (s z))) s')
97
98 {-}
99
100 [for sos] -----

```

```
102          <For s1 b s2 s3, s> => <if b then (s1;s3,s2;(For Skip b s2 s3)) else Skip,
103          s>
104
105      -}
106
107  -- For
108  sosStm (Inter (For s1 b s2 s3) s) = Inter (If b (Comp (Comp (Comp (s1) (s3)) (s2)) (For Skip
109  b s2 s3)) (Skip)) s
110
111
```

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos y tiene como una garantía de hasta 100.000 euros por depositante. Consulta más información en inglés.

13/1/25, 13:45

StructuralSemantics.hs

```
1  --
2  -- Structural Operational Semantics for WHILE 2020.
3  -- Examen de Lenguajes de Programación. UMA.
4  --
5  -- Apellidos, Nombre: Cardeñosa Borrego, Juan Manuel
6  --
7
8 module StructuralSemantics where
9
10
11 --
12 -- NO MODIFICAR EL CODIGO DE ABAJO
13 --
14
15 import Data.List.HT (takeUntil)
16
17 import While20
18
19 -- representation of configurations for While
20
21 data Config = Inter Stm State -- <S, s>
22           | Final State      -- s
23           | Stuck Stm State  -- <S, s>
24
25 isFinal :: Config -> Bool
26 isFinal (Final _) = True
27 isFinal _          = False
28
29 isInter :: Config -> Bool
30 isInter (Inter _) = True
31 isInter _          = False
32
33 isStuck :: Config -> Bool
34 isStuck (Stuck _) = True
35 isStuck _          = False
36
37 -- representation of the transition relation <S, s> -> s'
38
39 sosStm :: Config -> Config
40
41 -- x := a
42
43 sosStm (Inter (Ass x a) s) = Final (update s x (aVal a s))
44   where
45     update s x v y = if x == y then v else s y
46
47 -- skip
48
49 sosStm (Inter Skip s) = Final s
50
51 -- s1; s2
```

Consulta condiciones aquí



do your thing

WUOLAH

localhost:50364/4f34bec0-b08c-4ee4-9aa2-262692922db8/

1/3

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

```

52
53 sosStm (Inter (Comp ss1 ss2) s)
54   | isFinal next = Inter ss2 s'
55   where
56     next = sosStm (Inter ss1 s)
57     Final s' = next
58
59 sosStm (Inter (Comp ss1 ss2) s)
60   | isStuck next = Stuck (Comp stm ss2) s'
61   where
62     next = sosStm (Inter ss1 s)
63     Stuck stm s' = next
64
65 sosStm (Inter (Comp ss1 ss2) s)
66   | isInter next = Inter (Comp ss1' ss2) s'
67   where
68     next = sosStm (Inter ss1 s)
69     Inter ss1' s' = next
70
71 -- if b then s1 else s2
72
73 sosStm (Inter (If b ss1 ss2) s)
74   | bVal b s = Inter ss1 s
75
76 sosStm (Inter (If b ss1 ss2) s)
77   | not (bVal b s) = Inter ss2 s
78
79 -----
80 -- NO MODIFICAR EL CODIGO DE ARRIBA
81 -----
82
83 {-}
84
85 [swap sos] -----
86           <Swap x y, s> => <z:= y; y := x; x := z, s'>
87 -}
88
89 -- Sentencia Swap
90
91 sosStm (Inter (Swap x y) s) = Inter stm s''
92   where
93     Inter stm s' = sosStm (Inter ((Comp (Comp (Ass z (V "y")) (Ass y (V "z")))) (Ass x (V "z")))) s)
94     z = "z"
95     -- Para reestablecer el valor de z, si fuese necesario, sino, sería la salida (Inter stm s')
96     Final s'' = sosStm (Inter (Ass z (N (s z))) s')
97
98 {-}
99
100 [for sos] -----

```

```
102      <For s1 b s2 s3, s> => <if b then (s1;s3,s2;(For Skip b s2 s3)) else Skip,  
103      s>  
104  
105  -}  
106  
107 -- For  
108 sosStm (Inter (For s1 b s2 s3) s) = Inter (If b (Comp (Comp (Comp (s1) (s3)) (s2)) (For Skip  
109 b s2 s3)) (Skip)) s  
110  
111
```

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos y tiene como una garantía de hasta 100.000 euros por depositante. Consulta más información en inglés.

13/1/25, 13:45

StructuralSemantics.hs

```
1  --
2  -- Structural Operational Semantics for WHILE 2020.
3  -- Examen de Lenguajes de Programación. UMA.
4  --
5  -- Apellidos, Nombre: Cardeñosa Borrego, Juan Manuel
6  --
7
8 module StructuralSemantics where
9
10
11 --
12 -- NO MODIFICAR EL CODIGO DE ABAJO
13 --
14
15 import Data.List.HT (takeUntil)
16
17 import While20
18
19 -- representation of configurations for While
20
21 data Config = Inter Stm State -- <S, s>
22           | Final State      -- s
23           | Stuck Stm State  -- <S, s>
24
25 isFinal :: Config -> Bool
26 isFinal (Final _) = True
27 isFinal _          = False
28
29 isInter :: Config -> Bool
30 isInter (Inter _) = True
31 isInter _          = False
32
33 isStuck :: Config -> Bool
34 isStuck (Stuck _) = True
35 isStuck _          = False
36
37 -- representation of the transition relation <S, s> -> s'
38
39 sosStm :: Config -> Config
40
41 -- x := a
42
43 sosStm (Inter (Ass x a) s) = Final (update s x (aVal a s))
44   where
45     update s x v y = if x == y then v else s y
46
47 -- skip
48
49 sosStm (Inter Skip s) = Final s
50
51 -- s1; s2
```

Consulta condiciones aquí



do your thing

localhost:50364/4f34bec0-b08c-4ee4-9aa2-262692922db8/

WUOLAH

1/3

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

```

52
53 sosStm (Inter (Comp ss1 ss2) s)
54   | isFinal next = Inter ss2 s'
55   where
56     next = sosStm (Inter ss1 s)
57     Final s' = next
58
59 sosStm (Inter (Comp ss1 ss2) s)
60   | isStuck next = Stuck (Comp stm ss2) s'
61   where
62     next = sosStm (Inter ss1 s)
63     Stuck stm s' = next
64
65 sosStm (Inter (Comp ss1 ss2) s)
66   | isInter next = Inter (Comp ss1' ss2) s'
67   where
68     next = sosStm (Inter ss1 s)
69     Inter ss1' s' = next
70
71 -- if b then s1 else s2
72
73 sosStm (Inter (If b ss1 ss2) s)
74   | bVal b s = Inter ss1 s
75
76 sosStm (Inter (If b ss1 ss2) s)
77   | not (bVal b s) = Inter ss2 s
78
79 -----
80 -- NO MODIFICAR EL CODIGO DE ARRIBA
81 -----
82
83 {-}
84
85 [swap sos] -----
86           <Swap x y, s> => <z:= y; y := x; x := z, s'>
87 -}
88
89 -- Sentencia Swap
90
91 sosStm (Inter (Swap x y) s) = Inter stm s''
92   where
93     Inter stm s' = sosStm (Inter ((Comp (Comp (Ass z (V "y")) (Ass y (V "z")))) (Ass x (V "z")))) s)
94     z = "z"
95     -- Para re establecer el valor de z, si fuese necesario, sino, sería la salida (Inter stm s')
96     Final s'' = sosStm (Inter (Ass z (N (s z))) s')
97
98 {-}
99
100 [for sos] -----

```

```
102          <For s1 b s2 s3, s> => <if b then (s1;s3,s2;(For Skip b s2 s3)) else Skip,  
103          s>  
104      -}  
105  
106      -- For  
107  sosStm (Inter (For s1 b s2 s3) s) = Inter (If b (Comp (Comp (Comp (s1) (s3)) (s2)) (For Skip  
108    b s2 s3)) (Skip)) s  
109  
110  
111
```

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos y tiene como una garantía de hasta 100.000 euros por depositante. Consulta más información en inglés.

13/1/25, 13:45

StructuralSemantics.hs

```
1  --
2  -- Structural Operational Semantics for WHILE 2020.
3  -- Examen de Lenguajes de Programación. UMA.
4  --
5  -- Apellidos, Nombre: Cardeñosa Borrego, Juan Manuel
6  --
7
8 module StructuralSemantics where
9
10
11 --
12 -- NO MODIFICAR EL CODIGO DE ABAJO
13 --
14
15 import Data.List.HT (takeUntil)
16
17 import While20
18
19 -- representation of configurations for While
20
21 data Config = Inter Stm State -- <S, s>
22           | Final State      -- s
23           | Stuck Stm State  -- <S, s>
24
25 isFinal :: Config -> Bool
26 isFinal (Final _) = True
27 isFinal _          = False
28
29 isInter :: Config -> Bool
30 isInter (Inter _) = True
31 isInter _          = False
32
33 isStuck :: Config -> Bool
34 isStuck (Stuck _) = True
35 isStuck _          = False
36
37 -- representation of the transition relation <S, s> -> s'
38
39 sosStm :: Config -> Config
40
41 -- x := a
42
43 sosStm (Inter (Ass x a) s) = Final (update s x (aVal a s))
44   where
45     update s x v y = if x == y then v else s y
46
47 -- skip
48
49 sosStm (Inter Skip s) = Final s
50
51 -- s1; s2
```

Consulta condiciones aquí



do your thing

WUOLAH

```

52
53 sosStm (Inter (Comp ss1 ss2) s)
54   | isFinal next = Inter ss2 s'
55   where
56     next = sosStm (Inter ss1 s)
57     Final s' = next
58
59 sosStm (Inter (Comp ss1 ss2) s)
60   | isStuck next = Stuck (Comp stm ss2) s'
61   where
62     next = sosStm (Inter ss1 s)
63     Stuck stm s' = next
64
65 sosStm (Inter (Comp ss1 ss2) s)
66   | isInter next = Inter (Comp ss1' ss2) s'
67   where
68     next = sosStm (Inter ss1 s)
69     Inter ss1' s' = next
70
71 -- if b then s1 else s2
72
73 sosStm (Inter (If b ss1 ss2) s)
74   | bVal b s = Inter ss1 s
75
76 sosStm (Inter (If b ss1 ss2) s)
77   | not (bVal b s) = Inter ss2 s
78
79 -----
80 -- NO MODIFICAR EL CODIGO DE ARRIBA
81 -----
82
83 {-}
84
85 [swap sos] -----
86           <Swap x y, s> => <z:= y; y := x; x := z, s'>
87 -}
88
89 -- Sentencia Swap
90
91 sosStm (Inter (Swap x y) s) = Inter stm s''
92   where
93     Inter stm s' = sosStm (Inter ((Comp (Comp (Ass z (V "y")) (Ass y (V "z")))) (Ass x (V "z")))) s)
94     z = "z"
95     -- Para re establecer el valor de z, si fuese necesario, sino, sería la salida (Inter stm s')
96     Final s'' = sosStm (Inter (Ass z (N (s z))) s')
97
98 {-}
99
100 [for sos] -----

```

```
102          <For s1 b s2 s3, s> => <if b then (s1;s3,s2;(For Skip b s2 s3)) else Skip,  
103          s>  
104      -}  
105  
106      -- For  
107      sosStm (Inter (For s1 b s2 s3) s) = Inter (If b (Comp (Comp (Comp (s1) (s3)) (s2)) (For Skip  
108      b s2 s3)) (Skip)) s  
109  
110  
111
```