Логика распознавания последовательностей состояний SegTL (Segment Temporal Logic) и примеры её применения для формализации темпоральных требований к управляющим программ

Ануреев Игорь Сергеевич

Институт автоматики и электрометрии СО РАН

Институт систем информатики СО РАН

Новосибирский государственный университет

Новосибирск

Базовые понятия и определения

S – множество состояний

$\pi = s1, s2, s3, s4, \ldots$

s_1	s_2	s_3	s_4	…

П – множество бесконечных последовательностей состояний из S.

$\Delta \subseteq \Pi$

Пусть A, B (возможно с индексами и штрихами) – формулы SegTL.

$\Delta \vDash A \Leftrightarrow \pi \vDash A$ для любого $\pi \in \Delta$

Состояние имеет атрибуты.

Простой атрибут: имя и значение.

Параметрический атрибут: имя, параметры и значение.

Базовые понятия и определения

Имена атрибутов и параметров атрибутов – идентификаторы.

Формулы логики SegTL строятся над формулами некоторой базовой логикой L, которая может меняться, т. е. L является параметром логики SegTL.

В примерах ниже в качестве формул базовой логики L используется бесквантворные формулы типизированной логики первого порядка с равенством с базовым набором интерпретированных типов (bool, int, real и т. п.) и набором интерпретированных функций и предикатов, задающих операции над этими типами (+, *, <, <= и т. п.).

Пусть C, D, E (возможно с индексами и штрихами) – формулы L.

Истинность формул базовой логики L на состояниях на примере

$s \vDash x \leq f(x) \Leftrightarrow s(x) \leq s(f)(s(x))$

x – простой атрибут, f – параметрический атрибут с одним параметром.

s(x) и s(f) – значения этих атрибутов (простое значение и функция) в состоянии s.

Формулы SegTL и их семантика на синтаксических примерах

Операция (…) создания SegTL-формулы из базовой формулы

Запись: (D)

s_1	…
D	

s_1 – сопоставляемый сегмент (конечный префикс π)

Нульместная операция ( ) создания пустой формулы

Запись: ( ) ≡ (true)

s_1	…
true	

Операция (…) (…) последовательной комбинации базовых формул:

Запись: (D_1) (D_2) (D_3)

s_1	s_2	s_3	…
D_1	D_2	D_3	

s_1 s_2 s_3 – сопоставляемый сегмент (конечный префикс π)

Формулы SegTL и их семантика на синтаксических примерах

Операция and-then

Запись: A_1 and-then A_2

$(D_1)\ (D_2)\ (D_3)$ and-then $(E_1)\ (E_2)$

s_1	s_2	s_3	s_4	…
D_1	D_2	D_3		
		E_1	E_2	

(D1) (D2) ≡ (D1) ( ) and-then (D2)

Спецификаторы сдвига сегмента :b (begin) и :e (end)

Пример 1

$(D_1)\ (D_2)$:b (D_3) and-then $(E_1)\ (E_2)\ (E_3)$

$(D_1)\ (D_2)\ (D_3)$ and-then $(E_1)\ (E_2)$:e (E_3)

s_1	s_2	s_3	s_4	…
D_1	D_2	D_3		
	E_1	E_2	E_3	

Формулы SegTL и их семантика на синтаксических примерах

Пример 2

(D_1) :b (D_2) (D_3) and-then (E_1) (E_2) (E_3)

(D_1) (D_2) (D_3) and-then (E_1) (E_2) (E_3) :e

s_1	s_2	s_3	…
D_1	D_2	D_3	
E_1	E_2	E_3	

Пример 3

( ) :b (D_1) (D_2) (D_3) and-then (E_1) (E_2) (E_3)

(D_1) (D_2) (D_3) and-then (E_1) (E_2) (E_3) ( ) :e

s_1	s_2	s_3	s_4	…
( )	D_1	D_2	D_3	
E_1	E_2	E_3	( )	

Формулы SegTL и их семантика на синтаксических примерах

Пример 4

(D_1) :b (D_2) (D_3) and-then (E_1) (E_2) :e (E_3)

s_1	s_2	s_3	s_4	…
	D_1	D_2	D_3	
E_1	E_2	E_3		

Спецификатор :(…) модификации значений атрибутов состояний

(x > 0) :(z = x, u = z) (z > 0 && u > 0)

(x = 0) :(z = x, u = z) (z != u)

(x > 0) :(z = x + 1, m(u, v) = max(u, v) + 1) (m(x, z) > 2)

(x > 0) :(z = x + 1, m(x, z) = max(x, z) + 1) (m(x, z) > 2)

(x > 0) :(z = x + 1, “m = λ x z. max(x, z) + 1”) (m(x, z) > 2)

Формулы SegTL и их семантика на синтаксических примерах

Структурные скобки { }

Запись: {A} ≡ A

Операция always

Запись: always A

Пример 1

always (D)

s_1	s_2	s_3	s_4	…
D	D	D	D	D

Пример 2

always (D) (E)

s_1	s_2	s_3	s_4	…
D	E			
	D	E		
		D	E	

Формулы SegTL и их семантика на синтаксических примерах

Пример 3

{always (D)} and-then (E)

s_1	s_2	s_3	s_4	…
D	D	D	D	D
E				

Пример 4

{always (D)} and-then {always () (E)}

s_1	s_2	s_3	s_4	…
D	D	D	D	D
( )	E	E	E	E

Формулы SegTL и их семантика на синтаксических примерах

Операция or

Запись: A or B

(D_1) (D_2) or (E_1) (E_2) (E_3)

s_1	s_2	…
D_1	D_2	

или

s_1	s_2	s_3	…
E_1	E_2	E_3	

Формулы SegTL и их семантика на синтаксических примерах

Операция until

Запись: A until B

Пример 1

(D) until (E)

s_1	…
E	

или

s_1	s_2	s_3	…
D	D	D	
×	×	E	

Формулы SegTL и их семантика на синтаксических примерах

Пример 2

(D_1) (D_2) until (E_1) (E_2)

s_1	s_2	…
E_1	E_2	

или

s_1	s_2	s_3	s_4	…
D_1	D_2			
	D_1	D_2		
		D_1	D_2	
×	×			
	×	×		
		E_1	E_2	

Формулы SegTL и их семантика на синтаксических примерах

Пример 3

(D1) (D2) :b until (E1) (E2)

s_1	s_2	s_3	s_4	s_5	…
	E_1	E_2			

или

s_1	s_2	s_3	s_4	s_5	…
D_1	D_2				
	D_1	D_2			
		D_1	D_2		
	×	×			
		×	×		
			E_1	E_2	

Формулы SegTL и их семантика на синтаксических примерах

Пример 4

{(D) until (E)} and-then (C)

s_1	…
E	
C	

или

s_1	s_2	s_3	…
D	D	D	
×	×	E	
		C	

Формулы SegTL и их семантика на синтаксических примерах

Пример 5 {(D) until (E_1)} and-then {(C) until (E_2)}

s_1	…
E_1	
E_2	

или

s_1	s_2	s_3	…
E_1			
	C	C	
	×	E_2	

или

s_1	s_2	s_3	s_4	s_5	…
D	D	D			
×	×	E_1			
		E_2			

или

Формулы SegTL и их семантика на синтаксических примерах

Пример 5 {(D) until (E_1)} and-then {(C) until (E_2)}

или

S_1	S_2	S_3	S_4	…
D	D	D		
×	×	E_1		
		C	C	
		×	E_2	

Формулы SegTL и их семантика на синтаксических примерах

Операция *until

Запись: A *until B

Пример 1

(D) *until (E)

s_1	…
E	

или

s_1	s_2	s_3	…
D	D	D	
×	×	E	

Формулы SegTL и их семантика на синтаксических примерах

Пример 2

(D_1) (D_2) *until (E_1) (E_2)

s_1	s_2	…
E_1	E_2	

или

s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	…
D_1	D_2							
		D_1	D_2					
				D_1	D_2			
×	×	×	×	E_1	E_2			

Формулы SegTL и их семантика на синтаксических примерах

Операция if-then

Запись: if A_1 then A_2

Пример 1

if (D_1) (D_2) (D_3) then (E_1) (E_2)

s_1	s_2	s_3	…
D_1	D_2	D_3	
E_1	E_2		

Методология применения SegTL к формализации требований к управляющим программам с циклом управления

Встроенные атрибуты состояния

in:bool = true, если s – состояние на входе в контроллер после означивания входных переменных.

out:bool = true, если s – состояние на выходе из контроллера.

time:real – время глобальных часов, связанное с текущим состоянием.

(D):in – сокращение для (in = true && D)

(D):out – сокращение для (out = true && D)

Ограничения на последовательности состояний

s_1	s_2	s_3	s_4	…
in	out	in	out	

Обозначения в требованиях на ЕЯ

x:in:t означает, что x – входная переменная типа t.

x:out:t означает, что x – выходная переменная типа t.

Пример 1. Турникет

1. Турникет должен быть открыт (open:out:bool) не более 10 с.

```
if (open = false):in (open = true):b:(t = time)
```

```
then {(open = true) until ( ):in (open = false && time - t <= 10c)}
```

2. Если турникет был закрыт (open:out:bool) и оплата не выполнена (paid:in:bool), то он не откроется.

```
if (open = false && paid = false):in ( ):b then (open = false)
```

3. Если турникет был закрыт (open:out:bool) и оплата не выполнена (paid:in:bool), то он не откроется, пока не будет выполнена оплата.

```
if (open = false && paid = false):in then (open = false) until (paid = true):in
```

Пример 1. Турникет

4. После появления с монетоприемника сигнала получения оплаты (paid:in:bool) немедленно должен быть сформирован сигнал открытия турникета (open:out:bool).

```
if (open = false && paid = true):in ():b then (open = true)
```

5. После получения сигнала о проходе пользователя (PdOut:in:bool) турникет должен быть закрыт (open:out:bool) не более, чем через 1 с.

```
if (open = true && PdOut = true):in:(t = time)
then (open = true) until ( ) (open = false && time - t <= 1c)
```

6. Турникет должен быть открыт (open:out:bool) не менее 1 с.

```
if (open = false):in (open = true):b:(t = time)
then (open = true) until (time - t >= 1c)
```

Пример 1. Турникет

7. Если турникет открыт (opened:in:bool), должен гореть светодиод (enter:out:bool).

```
if (opened = true) then (enter = true)
```

8. После запрета прохода (enter:in:bool) должен быть разрешена работа монетоприемника (reset:out:bool).

```
if (enter = true):out (enter = false) ():b then (reset = true)
```

9. Если турникет только что открылся (open:out:bool), он будет открыт в течение 10 секунд или пока не пройдет пользователь (PdOut:in:bool).

```
if (open = false):in (open = true && PdOut = false):b:(t = time)
then {
    {(open = true && PdOut = false) until () (time - t > 10c)} or
    {(open = true) until (PdOut = true && time - t <= 10c):in ()} }
```

Пример 2. Требования, непривязанные к конкретным объектам управления

1. Если включился нагрев (heat:out:bool) и продолжался не менее T, то показания датчика температуры (temp:in:real) увеличились.

```
if (heat = false):in (heat = true):b:(t = time, temp0 = temp) and-then
    {(heat = true) until (time - t >= T)}
then (temp0 < temp)
```

Пример 2. Требования, непривязанные к конкретным объектам управления

2. Если объект не был кристаллизованным, и произошел его нагрев (heating:out:bool), а затем охлаждение (freezing:out:bool), то объект кристаллизовался (crystallized:in:bool).

```
if (crystallized = false && heating = false && freezing = false):in () and-then
        {(heating = true && freezing = false)
             until () (heating = false && freezing = true)} and-then
        {(heating = false && freezing = true)
             until {() (heating = false && freezing = false)}
then (crystallized = true)
```

Пример 2. Требования, непривязанные к конкретным объектам управления

3. Если нагрев (heating:out:bool) и охлаждение (freezing:out:bool) попеременно выполнялись не менее K раз, то после этого объект будет разрушен (destroyed:in:bool).

```
if {(destroyed = false && heating = false && freezing = false):in ():(k = 0)}
    and-then
        {{(heating = true && freezing = false)
            until () (heating = false && freezing = true)} and-then
         {(heating = false && freezing = true)
            until () (heating = true && freezing = false):(k = k + 1)}
        } *until (k >= K)
}
then (destroyed = true)
```