



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

## Taller #3

### Ejecución Simbólica Dinámica usando Z3

---

Ingeniería de Software II  
Primer Cuatrimestre de 2021

Integrante	LU	Correo electrónico
Sebastián Garbi	179/05	garbyseba@gmail.com
Brian Goldstein	27/14	brai.goldstein@gmail.com



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

## 1. Parte 1: Z3 Solver

### 1.1. Ejercicio 1

a.  $\neg(x \vee y) \equiv (\neg x \wedge \neg y)$

```
sat
(
  (define-fun y () Bool
    false)
  (define-fun x () Bool
    false)
)
```

Se satisface con  $x = y = false$

b.  $(x \wedge y) \equiv \neg(\neg x \vee \neg y)$

```
sat
(
  (define-fun y () Bool
    false)
  (define-fun x () Bool
    false)
)
```

Se satisface con  $x = y = false$

c.  $\neg(x \wedge y) \equiv \neg(\neg x \wedge \neg y)$

```
sat
(
  (define-fun y () Bool
    true)
  (define-fun x () Bool
    false)
)
```

Se satisface con  $x = false, y = true$

### 1.2. Ejercicio 2

a.  $3x + 2y = 36$

```
sat
(
  (define-fun y () Int
    0)
  (define-fun x () Int
    12)
)
```

Se satisface con  $x = 12, y = 0$

b.  $5x + 4y = 64$

```
sat
(
  (define-fun y () Int
    21)
  (define-fun x () Int
    (- 4))
)
```

Se satisface con  $x = -4, y = 21$

c.  $x * y = 64$

```
sat
(
  (define-fun x () Int
    64)
  (define-fun y () Int
    1)
)
```

Se satisface con  $x = 64, y = 1$

### 1.3. Ejercicio 3

- a1: 16 mod 2
- a2: 16 dividido por 4
- a3: El resto de la división entera de 16 por 5.

```
sat
(
  (define-fun a2 () Real
    4.0)
  (define-fun a1 () Real
    0.0)
  (define-fun a3 () Real
    1.0)
)
```

Se satisface con  $a1 = 0,0, a2 = 4,0, a3 = 1,0$

## 2. Parte 2: Dynamic Symbolic Execution (DSE)

### 2.1. Ejercicio 4

a.

Iteración	Input Concreto	Condición de ruta	Especificación para Z3	Resultado Z3
1	a=0,b=0,c=0	$C1$	AlgunLadoMayorACero NoTodaSumaDosLadosMayorATercero	a=1,b=1,c=2
2	a=1,b=1,c=2	$\neg C1 \wedge C2$	AlgunLadoMayorACero SumaDosLadosMayorATercero TodosLadosIguales	a=1,b=1,c=1
3	a=1,b=1,c=1	$\neg C1 \wedge \neg C2 \wedge C3$	AlgunLadoMayorACero SumaDosLadosMayorATercero NoTodosLadosIguales DosLadosIguales	a=2,b=1,c=2
4	a=2,b=1,c=2	$\neg C1 \wedge \neg C2 \wedge \neg C3 \wedge C4$	AlgunLadoMayorACero SumaDosLadosMayorATercero NoTodosLadosIguales TodosLadosDistintos	a=2,b=3,c=4
5	a=2,b=3,c=4	$\neg C1 \wedge \neg C2 \wedge \neg C3 \wedge \neg C4$	<i>END</i>	<i>END</i>

Donde

- $C1: a \leq 0 \vee b \leq 0 \vee c \leq 0$
- $C2: \neg(a + b > c \wedge a + c > b \wedge b + c > a)$
- $C3: a = b \wedge b = c$
- $C4: a = b \vee b = c \vee a = c$
- AlgunLadoMayorACero:  
(assert (not (or (<= a 0) (or (<= b 0)(<= c 0)))))
- NoTodaSumaDosLadosMayorATercero:  
(assert (not(and (>(+ a b) c)(and (>(+ a c) b)(>(+ b c) a)))))
- SumaDosLadosMayorATercero:  
(assert (and (>(+ a b) c)(and (>(+ a c) b)(>(+ b c) a)))))
- TodosLadosIguales:  
(assert (and (= a b)(= b c)))
- NoTodosLadosIguales:  
(assert (not(and (= a b)(= b c))))
- DosLadosIguales:  
(assert (or (= a b)(or (= b c)(= a c))))
- TodosLadosDistintos:  
(assert (not(or (= a b)(or (= b c)(= a c)))))

b.

BRANCH COVERAGE de EJECUCION SIMBOLICA

c.

DIBUJAR ARBOL

### 2.2. Ejercicio 5

a.

Iteración	Input Concreto	Condición de ruta	Especificación para Z3	Resultado Z3
1	k=0.0	$\neg C1$	Suma0Con1o3o5	k=-3.0
2	k=-3.0	$C1$	<i>END</i>	<i>END</i>

Donde

- C1:  $(5, 0 + k == 0 \vee 1, 0 + k == 0 \vee 3, 0 + k == 0)$
- Suma0Con1o3o5:  
`(assert (or (= (+ k 5.0) 0.0)(or (= (+ k 1.0) 0.0)(= (+ k 3.0) 0.0))))`

b.

BRANCH COVERAGE de EJECUCION SIMBOLICA

c.

DIBUJAR ARBOL