

Práctica Modelos de Computación: CYK Algorithm

Alumno: Giulio Francesco Tizzano

Grupo 101 Modelos de Computación

Índice

1. Introducción a la práctica
2. Decisiones de diseño tomadas en la práctica y cómo se ha implementado
3. Descripción de cada una de las pruebas de tests que el alumno ha definido
4. Completitud de los tests definidos en el proyecto y los definidos por el alumno
5. Conclusiones

Introducción a la práctica:

La práctica para realizar trata sobre la implementación de un algoritmo llamado CYK (en honor a Cocke, Younger y Kasami). Dicho algoritmo permite realizar la comprobación de la pertenencia de palabras para un lenguaje definido. Para ello el propio algoritmo toma como entrada una gramática independiente del contexto (tipo 2 – jerarquía de Chomsky) y una palabra no vacía en la Forma Normal de Chomsky. La idea de la práctica es programar en el lenguaje Java el algoritmo y el resto de los aspectos necesarios para la ejecución adecuada del algoritmo. En él, el alumno que realiza la implementación ha de entender cómo funciona el algoritmo y debe razonar cómo definir los elementos terminales y no terminales de una gramática, el axioma y el conjunto de producciones y de cómo guardar dichos elementos en memoria para la ejecución adecuada del algoritmo. Todo esto respaldado por unos tests básicos implementados en Junit que cada método debe de pasar con éxito.

Decisiones de diseño tomadas en la práctica y cómo se han implementado:

Dado que una gramática es una tupla de cuatro elementos $G = (E_N, E_T, S, P)$ conteniendo el conjunto de elementos no terminales, el conjunto de elementos terminales, el axioma y el conjunto de producciones, decidí definir un conjunto de elementos no terminales y terminales a través de la interfaz Set (del API java.util.*) con un HashSet<Character>. Luego, para el axioma de la gramática simplemente decidí guardarla dentro de una variable de tipo primitivo char (character) para que cuando en el método de 'setStartSymbol()' introducimos un elemento no terminal como parámetro ese valor se le asigne directamente a la variable axioma. Finalmente, para las producciones definí un HashMap que como primer parámetro acepta un carácter y como segundo un conjunto de elementos HashSet<String> para poder guardar las reglas de la gramática en la forma normal de Chomsky.

Descripción de cada una de las pruebas de tests que el alumno ha definido:

En su totalidad había que definir ocho tests a parte de los tests ya definidos en la plantilla de la práctica, es decir, dos tests adicionales por cada gramática proporcionada en la práctica para validar que una palabra pertenezca al lenguaje y otro para comprobar que no pertenece al lenguaje. A continuación, se muestran las fotos donde se ve el algoritmo aplicado a mano para cada una de las cadenas que también se han validado en el código de los tests adicionales:

$G_1 = (\{A, B, C\}, \{a, b\}, A, P)$, con $P =$

$$P = \begin{cases} A ::= BC \\ B ::= CA|a \\ C ::= AB|b \end{cases}$$

$A \rightarrow BC \rightarrow CAC \rightarrow bBCC \rightarrow bbbB \rightarrow bbbbb$

b	a	b	b
c	B	c	c
∅	A	∅	-
B	∅	-	-
A	-	-	-

→ Cadenas aceptadas

$b \notin \{a\}$

b	a	a	a	b
c	B	B	B	c
∅	∅	∅	A	-
∅	∅	∅	-	-
∅	∅	-	-	-
∅	-	-	-	-

→ Cadenas no válidas

$G_2 = (\{A, B, C\}, \{a, b\}, A, P)$, con $P =$

$$P = \begin{cases} A ::= BC|a \\ B ::= ~~CB~~ CD \\ C ::= ~~AC~~ BA|b \\ D ::= c \end{cases}$$

$A \rightarrow BC \rightarrow CD BA \rightarrow CD CDA \rightarrow bcbca$

b	c	b	c	a
c	D	c	D	A
B	∅	B	∅	-
A	∅	c	-	-
∅	∅	-	-	-
∅	A	-	-	-

→ Válido

b	c	b	c	a	a	b
c	D	c	D	A	A	c
B	∅	B	∅	∅	∅	-
A	∅	c	∅	∅	-	-
∅	∅	∅	∅	-	-	-
A	∅	∅	-	-	-	-
∅	∅	-	-	-	-	-
∅	-	-	-	-	-	-

→ No Válido

$G_3 = (\{S, A, B, C, D\}, \{a, b\}, S, P)$, con $P =$

$$P = \begin{cases} S ::= AB \\ A ::= BS|a \\ B ::= SA|b|DC \\ C ::= a \\ D ::= b \end{cases}$$

$S \rightarrow AB \rightarrow BS DC \rightarrow BABDC \rightarrow babba$

b	a	b	b	a
B,D	A,C	B,D	B,D	A,C
B	S	∅	B	-
A	∅	∅	-	-
S	∅	∅	-	-
S,B	-	-	-	-

→ Válido

a	b	b	a	a
A,C	B,D	B,D	A,C	A,C
S	∅	∅	∅	-
∅	∅	∅	-	-
A,C	B,D	B,D	A,C	A,C

→ No Válido

$G_4 = (\{S, A, B\}, \{a, b\}, S, P)$, con $P =$

$$P = \begin{cases} S ::= AB|BA|a \\ A ::= AA|BB \\ B ::= BB|SB|b \end{cases}$$

$S \rightarrow AB \rightarrow ABB \rightarrow BBBA \rightarrow BBBB \rightarrow bbbbbb$

$S \rightarrow BA \rightarrow BB AA \rightarrow BBBA \rightarrow BBBB \rightarrow bbbbbb$

b	b	b	b	b	b
B	B	B	B	B	B
A,B	A,B	A,B	A,B	A,B	∅
S,A,B	S,A,B	S,A,B	S,A,B	∅	∅
S,A,B	S,A,B	S,A,B	-	-	-
S,A,B	S,A,B	-	-	-	-
S,A,B	-	-	-	-	-

→ Válido

b	b	b	b	a	a
B	B	B	B	S	B
A,B	A,B	A,B	∅	B	∅
S,A,B	S,A,B	∅	A,B	∅	-
S,A,B	∅	S,A,B	∅	-	-
∅	S,A,B	∅	-	-	-
S,A,B	∅	-	-	-	-
∅	-	-	-	-	-

→ No Válido

Práctica CYK

Giulio Francesco Tizzano

En cuanto al código de los tests adicionales, la idea era definir las cuatro gramáticas usando los métodos implementados en la clase **CYKAlgorithm.java** como `addNonTerminal()`, `addTerminal()`, `setStartSymbol()` y `addProduction()` para definir todos los elementos necesarios y posteriormente utilizar un `assertTrue()` para ver si una cadena pertenecía al lenguaje o un `assertFalse()` para asegurarnos de que no pertenece al lenguaje.

Completitud de los tests definidos en el proyecto y los definidos por el alumno:

The image shows two screenshots of an IDE, likely IntelliJ, displaying test results for the CYK algorithm. The top screenshot shows the test `comprobarAniadirTerminalValido` with the following code:

```
@Test
public void comprobarAniadirTerminalValido() throws CYKAlgorithmException {
    cyk = new CYKAlgorithm();
    cyk.addTerminal(terminal: 'a');
}
```

The test result shows "Tests passed: 100.00 %" and "test passed. (0.085 s)".

The bottom screenshot shows the test `comprobarAniadirTerminalNoValido1` with the following code:

```
@Test
public void comprobarAniadirTerminalNoValido1() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addTerminal(terminal: '0');
}
```

The test result shows "Tests passed: 100.00 %" and "test passed. (0.08 s)".

Práctica CYK

Giulio Francesco Tizzano

```

    }

    @Test
    public void comprobarAniadirTerminalNoValido2() throws CYKAlgorithmException {
        thrown.expect(type: CYKAlgorithmException.class);
        cyk = new CYKAlgorithm();
        cyk.addTerminal(terminal: 'A');
    }
}

```

Previous Next Select

Results ×

eu.gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarAniadirTerminalNoValido2 ×

Tests passed: 100.00 %

1 test passed. (0.069 s)

```

    @Test
    public void comprobarAniadirTerminalNoValido3() throws CYKAlgorithmException {
        thrown.expect(type: CYKAlgorithmException.class);
        cyk = new CYKAlgorithm();
        cyk.addTerminal(terminal: 'a');
        cyk.addTerminal(terminal: 'a');
    }
}
// Arreglado
@Test

```

Previous Next Select

Results ×

eu.gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarAniadirTerminalNoValido3 ×

Tests passed: 100.00 %

The test passed. (0.071 s)

```

    }
// Arreglado
@Test
    public void comprobarAniadirNoTerminalValido() throws CYKAlgorithmException {
        cyk = new CYKAlgorithm();
        cyk.addNonTerminal(nonterminal: 'S');
    }

    @Test
    public void comprobarAniadirNoTerminalNoValido1() throws CYKAlgorithmException {
        thrown.expect(type: CYKAlgorithmException.class);
        cyk = new CYKAlgorithm();
    }
}

```

Previous Next Select

Results ×

eu.gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarAniadirNoTerminalValido ×

Tests passed: 100.00 %

1 test passed. (0.09 s)

Práctica CYK

Giulio Francesco Tizzano

```
@Test
public void comprobarAniadirNoTerminalNoValido1() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addNonTerminal(nonterminal: 'a');
}
```

Previous Next Select

Tests passed: 100.00 %

st passed. (0.099 s)

```
@Test
public void comprobarAniadirNoTerminalNoValido2() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addNonTerminal(nonterminal: '0');
}

// Este Test: arreglado
@Test
public void comprobarAniadirNoTerminalNoValido3() throws CYKAlgorithmException {
```

Previous Next Select

Tests passed: 100.00 %

e test passed. (0.075 s)

```
// Este Test: arreglado
@Test
public void comprobarAniadirNoTerminalNoValido3() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'S');
}
```

Previous Next Select

Tests passed: 100.00 %

est passed. (0.088 s)

Práctica CYK

Giulio Francesco Tizzano

```
@Test
public void comprobarEstablecerAxiomaValido() throws CYKAlgorithmException {
    cyk = new CYKAlgorithm();
    cyk.addNonTerminal(nonterminal: 'S');
    cyk.setStartSymbol(nonterminal: 'S');
}

@Test
public void comprobarEstablecerAxiomaNoValido1() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
}
```

Tests passed: 100.00 %

Test passed. (0.092 s)

```
@Test
public void comprobarEstablecerAxiomaNoValido1() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.setStartSymbol(nonterminal: 'S');
}
```

Tests passed: 100.00 %

Test passed. (0.088 s)

Práctica CYK

Giulio Francesco Tizzano

```
@Test
public void comprobarEstablecerAxiomaNoValido2() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addTerminal(terminal: 'a');
    cyk.setStartSymbol(nonterminal: 'S');
}
```

Results ×

gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarEstablecerAxiomaNoValido2 ×

Tests passed: 100.00 %

test passed. (0.085 s)

```
@Test
public void comprobarEstablecerAxiomaNoValido3() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addTerminal(terminal: 'a');
    cyk.setStartSymbol(nonterminal: 'a');
}
```

Results ×

gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarEstablecerAxiomaNoValido3 ×

Tests passed: 100.00 %

test passed. (0.084 s)

Práctica CYK

Giulio Francesco Tizzano

```
@Test
public void comprobarAniadirProduccionValida() throws CYKAlgorithmException {
    cyk = new CYKAlgorithm();
    cyk.addTerminal(terminal: 'a');
    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addProduction(nonterminal: 'S', production: "AS");
    cyk.addProduction(nonterminal: 'S', production: "a");
}
```

```
@Test
public void comprobarAniadirProduccionNoValida1() throws CYKAlgorithmException {
```

Previous Next Select

Tests passed: 100.00 %

st passed. (0.071 s)

```
@Test
public void comprobarAniadirProduccionNoValida1() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addTerminal(terminal: 'a');

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');

    cyk.addProduction(nonterminal: 'S', production: "b");
}

// Arreglado
@Test
public void comprobarAniadirProduccionNoValida2() throws CYKAlgorithmException {
```

Previous Next Select

Tests passed: 100.00 %

test passed. (0.077 s)

Práctica CYK

Giulio Francesco Tizzano

```
// Arreglado
@Test
public void comprobarAniadirProduccionNoValida2() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addTerminal(terminal: 'a');

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');

    cyk.addProduction(nonterminal: 'C', production: "a");
}

@Test
public void comprobarAniadirProduccionNoValida3() throws CYKAlgorithmException {
```

Results X

u.gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarAniadirProduccionNoValida2 X

Tests passed: 100.00 %

The test passed. (0.084 s)

```
@Test
public void comprobarAniadirProduccionNoValida3() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addTerminal(terminal: 'a');

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addProduction(nonterminal: 'S', production: "AB");
}

Results X
```

u.gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarAniadirProduccionNoValida3 X

Tests passed: 100.00 %

test passed. (0.079 s)

Práctica CYK

Giulio Francesco Tizzano

```
@Test
public void comprobarAniadirProduccionNoValida4() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addTerminal (terminal: 'a');

    cyk.addNonTerminal (nonterminal: 'S');
    cyk.addNonTerminal (nonterminal: 'A');

    cyk.addProduction (nonterminal: 'S', production: "A");
}
```

Previous Next Select

Tests passed: 100.00 %

st passed. (0.084 s)

```
@Test
public void comprobarAniadirProduccionNoValida5() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addTerminal (terminal: 'a');

    cyk.addNonTerminal (nonterminal: 'S');
    cyk.addNonTerminal (nonterminal: 'A');
    cyk.addProduction (nonterminal: 'A', production: "aA");
}

@Test
public void comprobarAniadirProduccionNoValida6() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
}
```

Previous Next Select

Results ×

u.gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarAniadirProduccionNoValida5 ×

Tests passed: 100.00 %

he test passed. (0.086 s)

Práctica CYK

Giulio Francesco Tizzano

```
@Test
public void comprobarAniadirProduccionNoValida6() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addTerminal(terminal: 'a');

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');

    cyk.addProduction(nonterminal: 'A', production: "aa");
}
```

```
@Test
public void comprobarAniadirProduccionNoValida7() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
}
```

Results ×

eu.gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarAniadirProduccionNoValida6 ×

Tests passed: 100.00 %

The test passed. (0.095 s)

```
@Test
public void comprobarAniadirProduccionNoValida7() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addTerminal(terminal: 'a');

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');

    cyk.addProduction(nonterminal: 'A', production: "SAA");
}
```

Arreglado

```
@Test
```

odcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarAniadirProduccionNoValida7 ×

Tests passed: 100.00 %

passed. (0.075 s)

Práctica CYK

Giulio Francesco Tizzano

```
@Test
public void comprobarAniadirProduccionNoValida8() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);
    cyk = new CYKAlgorithm();
    cyk.addTerminal(terminal: 'a');

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');

    cyk.addProduction(nonterminal: 'A', production: "SA");
    cyk.addProduction(nonterminal: 'A', production: "SA");
}

@Test
public void comprobarRecuperarProducciones() throws CYKAlgorithmException {
    cyk = new CYKAlgorithm();
```

Previous Next Select

Results ×

gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarAniadirProduccionNoValida8 ×

Tests passed: 100.00 %

test passed. (0.082 s)

```
@Test
public void comprobarRecuperarProducciones() throws CYKAlgorithmException {
    cyk = new CYKAlgorithm();
    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addNonTerminal(nonterminal: 'B');
    cyk.addNonTerminal(nonterminal: 'C');

    cyk.addTerminal(terminal: 'a');
    cyk.addTerminal(terminal: 'b');

    cyk.setStartSymbol(nonterminal: 'S');

    cyk.addProduction(nonterminal: 'S', production: "AB");
    cyk.addProduction(nonterminal: 'S', production: "BC");
```

Previous Next Select

Results ×

gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarRecuperarProducciones ×

Tests passed: 0.00 %

0 test passed, 1 test failed. (0.087 s)

- es.ceu.gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest Failed
- > comprobarRecuperarProducciones Failed: expected:<A::=[BA|a]> but was:<A::=[a|BA]>

Práctica CYK

Giulio Francesco Tizzano

```
// Arreglado
@Test
public void comprobarEliminarGramaticaValido() throws CYKAlgorithmException {
    cyk = new CYKAlgorithm();
    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addNonTerminal(nonterminal: 'B');
    cyk.addNonTerminal(nonterminal: 'C');

    cyk.addTerminal(terminal: 'a');
    cyk.addTerminal(terminal: 'b');

    cyk.setStartSymbol(nonterminal: 'S');

    cyk.addProduction(nonterminal: 'S', production: "AB");
    cyk.addProduction(nonterminal: 'A', production: "a");
    cyk.addProduction(nonterminal: 'B', production: "b");
    cyk.addProduction(nonterminal: 'C', production: "a");
}

Tests passed: 100.00 %
test passed. (0.081 s)
```

```
@Test
1 public void comprobarDerivacionNoValido1() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);

    cyk = new CYKAlgorithm();

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addNonTerminal(nonterminal: 'B');
    cyk.addNonTerminal(nonterminal: 'C');

    cyk.addTerminal(terminal: 'a');
    cyk.addTerminal(terminal: 'b');
    cyk.addTerminal(terminal: 'c');
}

Tests passed: 100.00 %
test passed. (0.078 s)
```

Práctica CYK

Giulio Francesco Tizzano

```
}

@Test
public void comprobarDerivacionNoValido2() throws CYKAlgorithmException {
    thrown.expect(type: CYKAlgorithmException.class);

    cyk = new CYKAlgorithm();

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addNonTerminal(nonterminal: 'B');
    cyk.addNonTerminal(nonterminal: 'C');
}
```

Previous Next Select

ts ×

si.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarDerivacionNoValido2 ×

Tests passed: 100.00 %

est passed. (0.078 s)

```
@Test
public void comprobarDerivacionValido1() throws CYKAlgorithmException {

    cyk = new CYKAlgorithm();

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addNonTerminal(nonterminal: 'B');
    cyk.addNonTerminal(nonterminal: 'C');

    cyk.addTerminal(terminal: 'a');
    cyk.addTerminal(terminal: 'b');
}
```

Previous Next Select

×

modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarDerivacionValido1 ×

Tests passed: 100.00 %

it passed. (0.091 s)

Práctica CYK

Giulio Francesco Tizzano

```
@Test
public void comprobarDerivacionValido2() throws CYKAlgorithmException {

    cyk = new CYKAlgorithm();

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addNonTerminal(nonterminal: 'B');
    cyk.addNonTerminal(nonterminal: 'C');

    cyk.addTerminal(terminal: 'a');
    cyk.addTerminal(terminal: 'b');

    cyk.setStartSymbol(nonterminal: 'S');

    cyk.addProduction(nonterminal: 'S', production: "AB");
    cyk.addProduction(nonterminal: 'S', production: "BC");
}
```

Previous Next Select

results ×

u.gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.comprobarDerivacionValido2 ×

Tests passed: 100.00 %

The test passed. (0.07 s)

```
// *** PRIMER GRAMATICA ***
// Primera gramática
@Test
public void miTestPrimeraGramaticaAcepta() throws CYKAlgorithmException{

    cyk = new CYKAlgorithm();

    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addNonTerminal(nonterminal: 'B');
    cyk.addNonTerminal(nonterminal: 'C');

    cyk.addTerminal(terminal: 'a');
    cyk.addTerminal(terminal: 'b');

    cyk.setStartSymbol(nonterminal: 'A');
}
```

Previous Next Select

results ×

.gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.miTestPrimeraGramaticaAcepta ×

Tests passed: 100.00 %

The test passed. (0.086 s)

Práctica CYK

Giulio Francesco Tizzano

```
    ,
    @Test
    public void miTestPrimeraGramaticaNoAcepta() throws CYKAlgorithmException{
        cyk = new CYKAlgorithm();

        cyk.addNonTerminal(nonterminal: 'A');
        cyk.addNonTerminal(nonterminal: 'B');
        cyk.addNonTerminal(nonterminal: 'C');

        cyk.addTerminal(terminal: 'a');
        cyk.addTerminal(terminal: 'b');

        cyk.setStartSymbol(nonterminal: 'A');

        cyk.addProduction(nonterminal: 'A', production: "BC");

        cyk.addProduction(nonterminal: 'B', production: "CA");
    }
```

ults ×

gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.miTestPrimeraGramaticaNoAcepta ×

Tests passed: 100.00 %

test passed. (0.079 s)

```
    // Segunda gramática:
    @Test
    public void miTestSegundaGramaticaAcepta() throws CYKAlgorithmException{
        cyk = new CYKAlgorithm();

        cyk.addNonTerminal(nonterminal: 'A');
        cyk.addNonTerminal(nonterminal: 'B');
        cyk.addNonTerminal(nonterminal: 'C');
        cyk.addNonTerminal(nonterminal: 'D');

        cyk.addTerminal(terminal: 'a');
        cyk.addTerminal(terminal: 'b');
    }
```

ults ×

gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.miTestSegundaGramaticaAcepta ×

Tests passed: 100.00 %

test passed. (0.082 s)

Práctica CYK

Giulio Francesco Tizzano

```
@Test
public void miTestSegundaGramaticaNoAcepta() throws CYKAlgorithmException{
    cyk = new CYKAlgorithm();

    cyk.addNonTerminal(nonterminal:'A');
    cyk.addNonTerminal(nonterminal:'B');
    cyk.addNonTerminal(nonterminal:'C');
    cyk.addNonTerminal(nonterminal:'D');

    cyk.addTerminal(terminal:'a');
    cyk.addTerminal(terminal:'b');
    cyk.addTerminal(terminal:'c');

    cyk.setStartSymbol(nonterminal:'A');
```

Results ×

eu.gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.miTestSegundaGramaticaNoAcepta ×

Tests passed: 100.00 %

The test passed. (0.082 s)

```
// Tercera gramática:
@Test
public void miTestTerceraGramaticaAcepta() throws CYKAlgorithmException{
    cyk = new CYKAlgorithm();

    cyk.addNonTerminal(nonterminal:'S');
    cyk.addNonTerminal(nonterminal:'A');
    cyk.addNonTerminal(nonterminal:'B');
    cyk.addNonTerminal(nonterminal:'C');
    cyk.addNonTerminal(nonterminal:'D');

    cyk.addTerminal(terminal:'a');
    cyk.addTerminal(terminal:'b');

    cyk.setStartSymbol(nonterminal:'S');
```

Results ×

eu.gisi.modcomp.cyk_algorithm.algorithm.test.BasicTest.miTestTerceraGramaticaAcepta ×

Tests passed: 100.00 %

The test passed. (0.072 s)

Práctica CYK

Giulio Francesco Tizzano

```
@Test
public void miTestTerceraGramaticaNoAcepta() throws CYKAlgorithmException{
    cyk = new CYKAlgorithm();

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addNonTerminal(nonterminal: 'B');
    cyk.addNonTerminal(nonterminal: 'C');
    cyk.addNonTerminal(nonterminal: 'D');

    cyk.addTerminal(terminal: 'a');
    cyk.addTerminal(terminal: 'b');

    cyk.setStartSymbol(nonterminal: 'S');
```

Tests passed: 100.00 %

test passed. (0.071 s)

```
@Test
public void miTestCuartaGramaticaAcepta() throws CYKAlgorithmException{
    cyk = new CYKAlgorithm();

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addNonTerminal(nonterminal: 'B');

    cyk.addTerminal(terminal: 'a');
    cyk.addTerminal(terminal: 'b');

    cyk.setStartSymbol(nonterminal: 'S');

    cyk.addProduction(nonterminal: 'S', production: "AB");
```

Tests passed: 100.00 %

test passed. (0.075 s)

Práctica CYK

Giulio Francesco Tizzano

```
}
@Test
public void miTestCuartaGramaticaNoAcepta() throws CYKAlgorithmException{
    cyk = new CYKAlgorithm();

    cyk.addNonTerminal(nonterminal: 'S');
    cyk.addNonTerminal(nonterminal: 'A');
    cyk.addNonTerminal(nonterminal: 'B');

    cyk.addTerminal(terminal: 'a');
    cyk.addTerminal(terminal: 'b');

    cyk.setStartSymbol(nonterminal: 'S');

    cyk.addProduction(terminal: 'S', production: "AB");
}
```

nodcomp.cyk_algorithm.algorithm.test.BasicTest.miTestCuartaGramaticaNoAcepta X

Tests passed: 100.00 %

passed. (0.095 s)

Conclusiones:

Tras haber completado la práctica, puedo confirmar que me ha ayudado a reforzar la comprensión del algoritmo CYK ya que la primera vez que te es presentado no resulta intuitivo por la rigurosidad de la presentación del algoritmo. En cuanto a la práctica, he pasado más tiempo investigando y pensando en cómo aplicar el algoritmo en el lenguaje Java más que en entender el propio algoritmo. Lo que más tiempo ha consumido ha sido hacer debugging y hacer funcionar cada método de la práctica. Por lo que, lo que más tiempo ha consumido ha sido definitivamente la parte de la programación. Estimo que habré pasado unas 30 y pico horas en su totalidad para completar la práctica repartido entre varios días.