

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software).

<<http://www.ati.es/novatica/>>
<<http://www.ati.es/reicis/>>

ATI es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AIZ**, **ASTIC**, **RITSI** e **HispaniX**, junto a la que participa en **Prolnnova**.

Consejo Editorial

Ignacio Aguillo Sousa, Guillem Alsina González, María José Escalona Cuaserna, Rafael Fernández Calvo (presidente del Consejo), Jaime Fernández Martínez, Luis Fernández Sanz, Didac Lopez Vilas, Celestino Martín Alonso, José Onofre Montesa Andrés, Francisco Noguera Puig, Ignacio Pérez Martínez, Andrés Pérez Payares, Viktu Pons i Colomer, Juan Carlos Vigo López

Coordinación Editorial

Llorenç Pagés Casas <pages@ati.es>

Composición y autedición

Jorge Lácer Gil de Rames

Traediciones

Grupo de Lengua e Informática de ATI <<http://www.ati.es/gt/lengua-informatica/>>

Administración

Tomás Brunete, María José Fernández, Enric Camarero, Felicidad Lopez

Secciones Técnicas - Coordinadores

Acceso y recuperación de la información

José María Gómez Hidalgo (Qinetel) <jmgomez@yahoo.es>

Manuel J. Maña López (Universidad de Huelva) <manuel.mana@di.esia.uhu.es>

Administración Pública electrónica

Francisco López Crespo (IAIE) <flc@ati.es>

Sebastià Justicia Pérez (Diputació de Barcelona) <sjusticia@ati.es>

Arquitecturas

Enrique F. Torres Moreno (Universidad de Zaragoza) <enrique.torres@unizar.es>

José Filichardo (Universidad Politécnica de Valencia) <jfilich@disca.upv.es>

Auditoría SITIC

Marina Touriño Troilito <marinatourino@marinatourino.com>

Manuel Palao García-Suelto (ATI) <manuel@palao.com>

Derecho e tecnologías

Isabel Hernando Collazos (Fac. Derecho de Donostia, UPV) <isabel.hernando@ehu.es>

Elena Davara Fernández de Marcos (Davara & Davara) <edavara@davara.com>

Enseñanza Universitaria de la Informática

Cristóbal Pareja Flores (DSIC-UCM) <cpareja@sis.ucm.es>

J. Angel Velázquez Irujo (DLSI, URJC) <angel.velazquez@urjc.es>

Entorno digital personal

Andrés Marín López (Univ. Carlos III) <amarin@it.uc3m.es>

Diego Gachet Páez (Universidad Europea de Madrid) <gachet@uem.es>

Estándares Web

Encarna Quesada Ruiz (Virati) <encarna.quesada@virati.com>

José Carlos del Arco Prieto (TCP Sistemas e Ingeniería) <jcarco@gmail.com>

Gestión del Conocimiento

Juan Baiget Solé (Cap Gemini Ernst & Young) <juan.baiget@cei.es>

Informática y Filosofía

José Angel Olivás Varela (Escuela Superior de Informática, UCLM) <joseangel.olivas@uclm.es>

Roberto Feltrero Diego (UNED) <rfeltrero@gmail.com>

Informática Gráfica

Miguel Chover Sellés (Universitat Jaume I de Castellón) <chover@lsi.uji.es>

Roberto Vivó Hernández (Eurographics, sección española) <rvivo@dsic.upv.es>

Ingeniería del Software

Javier Dolado Cosin (DLSI-UPV) <dolado@si.ehu.es>

Daniel Rodríguez García (Universidad de Alcalá) <daniel.rodriguez@uah.es>

Inteligencia Artificial

Vicente Botti Navarro, Vicente Julián Inglada (DSIC-UPV) <vbotti.vinglada@dsic.upv.es>

Interacción Persona-Computador

Pedro M. Latorre Andrés (Universidad de Zaragoza, AIPO) <platorre@unizar.es>

Francisco L. Gutiérrez Vela (Universidad de Granada, AIPO) <fgutierrez@ugr.es>

Lengua e Informática

M. del Carmen Ugarte García (ATI) <cugarte@ati.es>

Lenguajes Informáticos

Oscar Belmonte Fernández (Univ. Jaime I de Castellón) <obelmonte@lsi.uji.es>

Inmaculada Coma Taty (Univ. de Valencia) <inmaculada.coma@uv.es>

Lingüística computacional

Xavier Gómez Guinovart (Univ. de Vigo) <xggo@uvigo.es>

Manuel Palomar (Univ. de Alicante) <mpalomar@disi.ua.es>

Mundo estudiantil y jóvenes profesionales

Federico G. Mon Trotti (RITSI) <gnu.fede@gmail.com>

Mikel Salazar Peña (Área de Jóvenes Profesionales, Junta de ATI Madrid) <mikelbo_uni@yahoo.es>

Profesión Informática

Rafael Fernández Calvo (ATI) <rfcalvo@ati.es>

Miquel Sarries Grifó (ATI) <miquel@sarries.net>

Redes y servicios telemáticos

José Luis Marzo Lázaro (Univ. de Girona) <joseluis.marzo@udg.es>

Juan Carlos López López (UCLM) <juancarlos.lopez@uclm.es>

Robótica

José Cortés Arenas (Sopra Group) <joscortare@gmail.com>

Juan González Gómez (Universidad Carlos III) <juan@bearobotics.com>

Seguridad

Javier Arellano Bertolin (Univ. de Deusto) <jarellino@deusto.es>

Javier López Muñoz (ETSI Informática-UMA) <jlm@cc.uma.es>

Sistemas de Tiempo Real

Alejandro Alonso Muñoz (Univ. Antonio de la Puente Alvaro (DIT-UPM) <aaalonso.ijpuente@dit.upm.es>

Software Libre

Jesus M. González Barahona (GSYC-URJC) <jgib@gsyc.es>

Israel Herriz Tabernero (Universidad Politécnica de Madrid) <isra@herriz.org>

Tecnología de Objetos

Jesus Garcia Molina (DIS-UM) <jmolina@um.es>

Gustavo Rossi (LPIA-UNLP Argentina) <gustavo@sol.unlp.edu.ar>

Tecnologías para la Educación

Juan Manuel Dodero Beardo (UC3M) <dodero@inf.uc3m.es>

César Pablo Córcoles Briongo (UOC) <ccorcoles@uoc.edu>

Tecnologías y Empresa

Didac Lopez Vilas (Universitat de Girona) <didac.lopez@ati.es>

Francisco Javier Cantais Sánchez (Indra Sistemas) <fjcantais@gmail.com>

Tendencias tecnológicas

Alonso Álvarez García (TD) <aaad@td.es>

Gabriel Martí Fuentes (Interbits) <gabi@atinet.es>

TIC y Turismo

Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga) <aguayo.guevara@lcc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. **Novática** permite la reproducción, sin ánimo de lucro, de todos los artículos, a menos que lo impida la modalidad de © o copyright elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid

Padilla 66, 3º, dcha., 28006 Madrid

Tel. 91 4029391 - fax 91 9309385 - novatica@ati.es

Composición, Edición y Redacción ATI Valencia

Av. del Peño de Valencia 23, 46005 Valencia

Tel. 963740173 - novatica_prod@ati.es

Administración y Redacción ATI Cataluña

Via Laietana 48, 1º, 08003 Barcelona

Tel. 934125235 - fax 934127713 - secregen@ati.es

Redacción ATI Aragón

Lagasca 9, 3-B, 50006 Zaragoza

Tel. fax 976235161 - secreara@ati.es

Redacción ATI Andalucía

Sección de Sevilla - secreand@ati.es

Redacción ATI Galicia

Sección de Santiago - secregal@ati.es

Suscripción y Ventas

novatica.subscripciones@atinet.es

Publicidad

Padilla 66, 3º, dcha., 28006 Madrid

Tel. 91 4029391 - fax 91 9309385 - novatica@ati.es

Imprenta: Derra S.A., Juan de Austria 68, 08005 Barcelona

Depósito legal: B 15.154-1975 - ISSN: 0211-2124; CODEN NOVACQ

Portada: Corredor de hierba - Concha Arias Pérez / © ATI

Diseño: Fernando Agresta / © ATI 2003

editorial

Hasta el infinito y más allá > 02

en resumen

El futuro ya está aquí y se hace compatible con el presente > 02

Llorenç Pagés Casas

Noticias de IFIP y CLEI

Últimas actividades del IFIP TC13: Human-Computer Interaction > 03

Julio Abascal González

monografía

Internet IPv6: una revolución silenciosa

Editores invitados: Jordi Domingo Pascual, Eduardo Jacob y Carlos Ralli Ucendo

Presentación. IPv6: Un nuevo espacio para la innovación > 05

Jordi Domingo Pascual, Eduardo Jacob, Carlos Ralli Ucendo

Estado del IPv6. World IPv6 Day (8/6/2011), IPv6 Launch Day (6/6/2012) > 08

João Luis Silva Damas

Internet6: Impacto en los productos y servicios digitales > 11

Carlos Ralli Ucendo

Ecosistema IPv6: Tecnologías utilizadas > 17

Octavio Alfamega

Internet6: Alcanzando la masa crítica de usuarios y tráfico > 23

Juan Pedro Cerezo Martín, Javier Benítez, Norberto Ojinaga Goitia, Antonio Hernández Armenteros, Carlos Ralli Ucendo, Óscar Pantoja García

Despliegue en las empresas y redes corporativas: La visión de un integrador > 29

Miguel González Fernández

IPv6: Internet Society y la visión de los usuarios > 35

Josu Aramberri

Internet IPv6 en las redes académicas y de investigación: REDIRIS - Géant > 40

Tomás P. de Miguel, Miguel Angel Sotos, Francisco Monserrat, Esther Robles

Actividades del IETF al respecto de IPv6 > 44

Jordi Palet Martínez

Redes Definidas por Software e IPv6: Situación actual > 47

Eduardo Jacob

secciones técnicas

Administración Pública electrónica

Interoperabilidad en los sistemas de información públicos > 50

Sebastià Justicia Pérez

Estándares web

Guías para el modelado de procesos de negocio > 56

Laura Sánchez-González, Francisco Ruiz González, Félix García Rubio

SOA4All Integrated Ranking:

Una herramienta holística basada en preferencias > 62

José María García, David Ruiz, Antonio Ruiz-Cortés

Referencias autorizadas

> 65

sociedad de la información

Ética profesional

Enseñanza de la Seguridad Computacional como instrumento de la ética profesional > 72

Wilmer Pereira

Programar es crear

El problema del Buscaminas Cuadrado en 3D > 78

(Competencia UTN-FRC 2012, problema F, enunciado)

Julio Javier Castillo, Diego Javier Serrano, Marina Elizabeth Cárdenas

El problema de los paréntesis y los corchetes > 79

(Competencia UTN-FRC 2011, problema C, solución)

Julio Javier Castillo, Diego Javier Serrano, Marina Elizabeth Cárdenas

asuntos interiores

Coordinación editorial / Programación de Novática / Socios Institucionales > 81

El problema de los paréntesis y los corchetes

Julio Javier Castillo, Diego Javier Serrano, Marina Elizabeth Cárdenas
 Laboratorio de Investigación de Software MsLabs, Dpto. Ing. en Sistemas de Información, Facultad Regional Córdoba - Universidad Tecnológica Nacional (Argentina)

El enunciado de este problema apareció en el número 217 de *Novática* (mayo-junio 2012, p.59)

<jotacastillo@gmail.com>, <diegojserrano@gmail.com>, <ing.marinacardenas@gmail.com>

El problema planteado consiste en reconocer expresiones matemáticas que contienen paréntesis y corchetes balanceados. Por ejemplo, la expresión “[(a*b) + (c/2)] * [(3*v)+2]” tiene que ser reconocida por el programa, mientras que la expresión “((y*x) + z)” no tiene que ser reconocida, pues contiene un corchete de cierre sin su correspondiente corchete de apertura, y falta un paréntesis de apertura.

Ya que el problema consiste en determinar si los paréntesis y corchetes están balanceados, se pueden ignorar los demás caracteres de la expresión y analizar la cadena equivalente “[()] [()]”.

Desde el punto de vista de la Teoría Formal de Lenguajes Formales y Autómatas, el problema se puede plantear como “reconocer el lenguaje que genere paréntesis y corchetes balanceados”, el cual se corresponde con la siguiente gramática G en BNF (*Backus Naur Form*).

$$G = (\{ (,), [,] \}, \{S\}, S, \{S := SS \mid (S) \mid [S] \mid \lambda\})$$

donde G es una 4-upla formada por los símbolos terminales, no terminales, el símbolo distinguido (o axioma) y el conjunto de reglas de producción de la gramática G.

Esta gramática produce un conjunto de cadenas, y al tratarse de una gramática independiente de contexto, sabemos que existe un autómata pila que las reconoce.

Una propuesta para el autómata a pila (AP) que reconoce el lenguaje anterior es :

$$AP = (\Sigma_E, \Gamma, Q, q_0, A_0, F, f)$$

$$\Sigma_E = \{ (,), [,] \}, \Gamma = \{ A_0, (,), [,] \}, Q = \{ q_0, q_1, q_2, q_3, q_4, q_f, t \}, q_0 = q_0, A_0 = A_0, F = \{ q_f \}$$

Donde Σ_E indica el conjunto de símbolos terminales de entrada, Γ indica el conjunto de símbolos de la pila, Q es el conjunto de estados del autómata, q_0 es el estado inicial del autómata, A_0 es el símbolo base de la pila, y F es el conjunto de estados finales. La

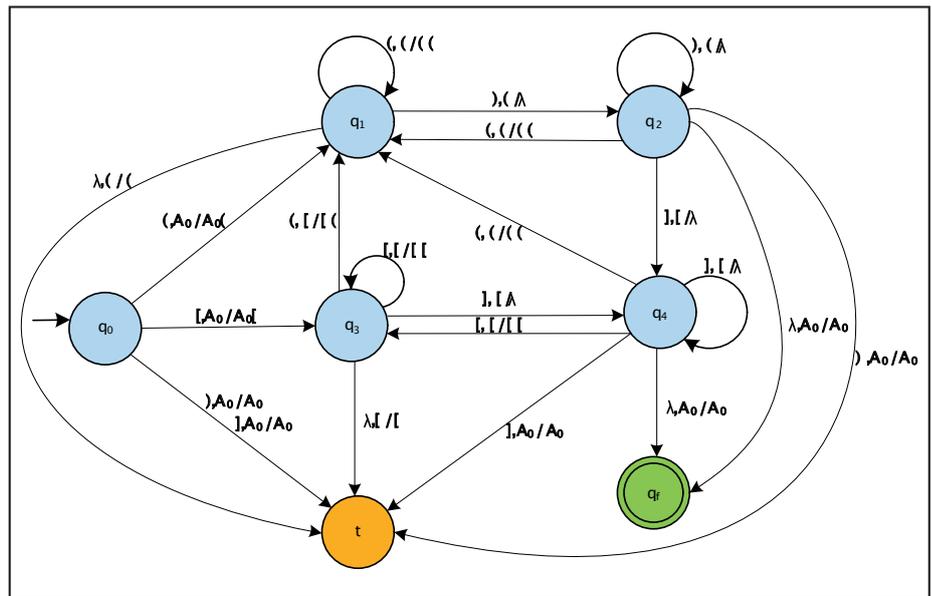


Figura 1. AP que acepta cadenas con paréntesis y corchetes balanceados.

función de transición f se puede deducir del autómata de la figura 1.

Esta propuesta asume que el AP aceptará cadenas por vaciado de pila y por alcance de estado final.

Asimismo, nótese que el AP incorpora un estado t (estado de trampa), que describe una condición de error de la cual el autómata no puede salir una vez que acontece, es decir que una vez que llega al estado de trampa, esa cadena no será aceptada. Las transiciones representan: “símbolo leído de la cadena de entrada, símbolo leído del tope de la pila / símbolos que se escriben sobre la pila”. Por ejemplo, $(A_0/A_0$ significa que se lee un paréntesis de la cinta de entrada, que se lee el símbolo base de la pila A_0 , y que se apilan sobre la pila A_0 y (.

El AP comienza apilando paréntesis o corchetes, representados por los estados q_1 y q_3 , y desapilando paréntesis o corchetes en los estados q_2 y q_4 . Si al desapilar se encuentra que los paréntesis o corchetes no están balanceados se transita del estado q_2 o del estado q_4 al estado trampa t. Por otra parte, si al apilar paréntesis o corchetes se termina de leer la cadena de entrada, esto indica que

han faltado paréntesis o corchetes de cierre, por lo cual ante esta situación se transita del estado q_1 o del estado q_3 al estado trampa t. Caso contrario, la cadena estará balanceada y el AP terminará en el estado final q_f , y con la pila solamente conteniendo el símbolo de base de la pila (condición de aceptación).

Este modelo formal puede ser fácilmente implementado si se utilizan bibliotecas estándar del lenguaje de programación de su elección. En efecto, la estructura de datos Stack (o pila) se encuentra como parte de la biblioteca estándar del lenguaje Java en el paquete `java.util.Stack`.

En otros lenguajes como Python, es posible utilizar los métodos `append()` y `pop()` para apilar y desapilar elementos de una lista. De esta manera, es posible utilizar una pila (estructura de datos de tipo LIFO) implementada con una lista y dos operaciones que trabajan sobre el tope de la pila.

Respecto a la solución en Java, se utiliza la colección `Stack` para apilar los caracteres ‘(’, y ‘[’, y se desapilan los caracteres ‘)’ y ‘]’. Además, se puede observar que cualquier otro carácter leído de la entrada estándar es ignorado por el programa.

A continuación se exponen dos soluciones a este problema, una planteada en lenguaje Java y la otra en el lenguaje de programación Python.

```
//Solución propuesta en Java
import java.util.Scanner;
import java.util.Stack;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int c = sc.nextInt();
        sc.nextLine();

        for (int i = 0; i < c; i++) {
            String linea = sc.next();
            Stack<Character> abiertos = new Stack<>();
            boolean resultadoOK = true;
            for (int j = 0; resultadoOK && j < linea.length(); j++) {
                switch(linea.charAt(j)) {
                    case '(': abiertos.push('('); break;
                    case '[': abiertos.push('['); break;
                    case ')': if (abiertos.empty() || abiertos.pop() != '(') resultadoOK = false;
                    break;
                    case ']': if (abiertos.empty() || abiertos.pop() != '[') resultadoOK = false;
                    break;
                }
            }
            if (resultadoOK) resultadoOK = abiertos.empty();
            System.out.println(resultadoOK ? "SI": "NO");
        }
    }
}
```

```
#!/Solución propuesta en Python
def balanceada(linea):
    abiertos = []
    for c in linea:
        if c == '(' or c == '[':
            abiertos.append(c)
        if c == ')' and (len(abiertos) == 0 or abiertos.pop() != '('):
            return False
        if c == ']' and (len(abiertos) == 0 or abiertos.pop() != '['):
            return False
    return len(abiertos) == 0

c = int(input())

for i in range(c):
    if balanceada(input()):
        print('SI')
    else:
        print('NO')
```