**Analyse descendante**

Exercice 1 :

public class ParseurDescendant {

private String texte;

private int position;

public ParseurDescendant(String texte) {

this.texte = texte;

this.position = 0;

}

public boolean parse() {

return S();

}

private boolean S() {

if (position < texte.length() && texte.charAt(position) == 'c') {

position++;

return true;

}

if (position < texte.length() && texte.charAt(position) == 'b') {

if (S()) {

position++;

return true;

}

}

return false;

}

public static void main(String[] args) {

String texte1 = "cbb";

ParseurDescendant parseur1 = new ParseurDescendant(texte1);

if (parseur1.parse()) {

System.out.println("La chaîne '" + texte1 + "' est valide selon la grammaire.");

} else {

System.out.println("La chaîne '" + texte1 + "' n'est pas valide selon la grammaire.");

}

String texte2 = "cbbb";

ParseurDescendant parseur2 = new ParseurDescendant(texte2);

if (parseur2.parse()) {

System.out.println("La chaîne '" + texte2 + "' est valide selon la grammaire.");

} else {

System.out.println("La chaîne '" + texte2 + "' n'est pas valide selon la grammaire.");

}

String texte3 = "b";

ParseurDescendant parseur3 = new ParseurDescendant(texte3);

if (parseur3.parse()) {

System.out.println("La chaîne '" + texte3 + "' est valide selon la grammaire.");

} else {

System.out.println("La chaîne '" + texte3 + "' n'est pas valide selon la grammaire.");

}

}

}

Exercice 2 :

public class ParseurDescendant {

private String texte;

private int position;

public ParseurDescendant(String texte) {

this.texte = texte;

this.position = 0;

}

public boolean parse() {

return S();

}

private boolean S() {

if (position + 3 <= texte.length() && texte.charAt(position) == 'b' && texte.charAt(position + 1) == 'c') {

position += 2;

if (S()) {

if (position < texte.length() && texte.charAt(position) == 'b') {

position++;

if (S()) {

if (S()) {

return true;

}

}

}

}

}

if (position < texte.length() && texte.charAt(position) == 'c') {

position++;

return true;

}

return false;

}

public static void main(String[] args) {

String texte1 = "bcbcc";

ParseurDescendant parseur1 = new ParseurDescendant(texte1);

if (parseur1.parse()) {

System.out.println("La chaîne '" + texte1 + "' est valide selon la grammaire.");

} else {

System.out.println("La chaîne '" + texte1 + "' n'est pas valide selon la grammaire.");

}

String texte2 = "bcbbbccc";

ParseurDescendant parseur2 = new ParseurDescendant(texte2);

if (parseur2.parse()) {

System.out.println("La chaîne '" + texte2 + "' est valide selon la grammaire.");

} else {

System.out.println("La chaîne '" + texte2 + "' n'est pas valide selon la grammaire.");

}

String texte3 = "c";

ParseurDescendant parseur3 = new ParseurDescendant(texte3);

if (parseur3.parse()) {

System.out.println("La chaîne '" + texte3 + "' est valide selon la grammaire.");

} else {

System.out.println("La chaîne '" + texte3 + "' n'est pas valide selon la grammaire.");

}

String texte4 = "bcbcb";

ParseurDescendant parseur4 = new ParseurDescendant(texte4);

if (parseur4.parse()) {

System.out.println("La chaîne '" + texte4 + "' est valide selon la grammaire.");

} else {

System.out.println("La chaîne '" + texte4 + "' n'est pas valide selon la grammaire.");

}

}

}

Exercice 3 :

public class ParseurDescendant {

private String texte;

private int position;

public ParseurDescendant(String texte) {

this.texte = texte;

this.position = 0;

}

public boolean parse() {

return S();

}

private boolean S() {

if (position < texte.length() && texte.charAt(position) == 'b') {

position++;

if (S()) {

if (position < texte.length() && texte.charAt(position) == 'b') {

position++;

return true;

}

}

}

if (position < texte.length() && texte.charAt(position) == 'c') {

position++;

if (A()) {

if (position < texte.length() && texte.charAt(position) == 'c') {

position++;

return true;

}

}

}

return false;

}

private boolean A() {

if (position < texte.length() && texte.charAt(position) == 'b') {

position++;

if (A()) {

if (A()) {

return true;

}

}

}

if (position < texte.length() && texte.charAt(position) == 'c') {

position++;

if (A()) {

if (S()) {

if (A()) {

if (position < texte.length() && texte.charAt(position) == 'b') {

position++;

return true;

}

}

}

}

}

if (position + 2 < texte.length() && texte.substring(position, position + 3).equals("dcb")) {

position += 3;

return true;

}

return false;

}

public static void main(String[] args) {

String[] chaines = {

"cdcbc",

"bcdcbcb",

"cbdcbdcbc",

"ccdcbcdcbcdcbbcr",

"cdcbbb",

"cdcb",

};

for (String texte : chaines) {

ParseurDescendant parseur = new ParseurDescendant(texte);

if (parseur.parse()) {

System.out.println("La chaîne '" + texte + "' est valide selon la grammaire.");

} else {

System.out.println("La chaîne '" + texte + "' n'est pas valide selon la grammaire.");

}

}

}

}

Exercice 4 :

public static void main(String[] args) {

String[] chaines = {

"d",

"bdc",

"bbdcc",

"b",

"c",

"bbcd",

"bcdd"

};

for (String texte : chaines) {

ParseurDescendant parseur = new ParseurDescendant(texte);

if (parseur.parse()) {

System.out.println("La chaîne '" + texte + "' est valide selon la grammaire.");

} else {

System.out.println("La chaîne '" + texte + "' n'est pas valide selon la grammaire.");

}

}

}