


IFT1015 Programmation 1

Énoncé switch; Automate

Marc Feeley

L'énoncé switch

- L' énoncé **switch** permet de faire une **exécution conditionnelle** d'un de plusieurs énoncés en fonction de la valeur d'une expression
 - Syntaxe de base : **switch** ($\langle expression \rangle_0$) {
 case $\langle expression \rangle_1$: $\langle \text{énoncés} \rangle_1$
 case $\langle expression \rangle_2$: $\langle \text{énoncés} \rangle_2$
 case $\langle expression \rangle_3$: $\langle \text{énoncés} \rangle_3$
 ...
}
- 
- corps du **switch**

Exemples

```
// Fichier: choix1.js

// Ce programme illustre l'énoncé switch.

var choix = function (x) {

    switch (x/10) {
        case 1: print("a");
        case 2: print("b");
        case 3: print("c");
        case 4: print("d");
        case 5: print("e");
    }
};

choix(50); // imprime: e

choix(30); // imprime: c d e

choix(0); // imprime rien
choix(90); // imprime rien
```

Exemples

```
// Fichier: choix2.js

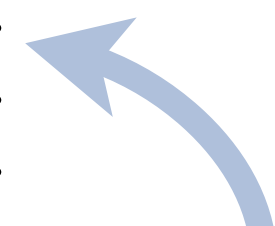
// Ce programme illustre les énoncés switch et break.

var choix = function (x) {

    switch (x/10) {
        case 1: print("a"); break;
        case 2: print("b"); break;
        case 3: print("c"); break;
        case 4: print("d"); break;
        case 5: print("e"); break;
    }
};

choix(50); // imprime: e
choix(30); // imprime: c

choix(0); // imprime rien
choix(90); // imprime rien
```



Un **break** exécuté dans un **switch** sort immédiatement du **switch**

Exemples


```
// Fichier: choix3.js

// Ce programme illustre le switch avec default.

var choix = function (x) {

    switch (x/10) {
        case 1: print("a"); break;
        case 2: print("b"); break;
        case 3: print("c"); break;
        case 4: print("d"); break;
        default: print("e"); break;
    }
};

choix(50); // imprime: e
choix(30); // imprime: c
choix(0);  // imprime: e
choix(90); // imprime: e
```



Le cas **default**
est exécuté
lorsqu'aucun
autre cas
concorde

Exemples

```
// Fichier: choix3-avec-if.js
```

```
// Ce programme est équivalent à choix3.js .
```

```
var choix = function (x) {  
    var n = x/10;  
    if (n === 1) {  
        print("a");  
    } else if (n === 2) {  
        print("b");  
    } else if (n === 3) {  
        print("c");  
    } else if (n === 4) {  
        print("d");  
    } else {  
        print("e");  
    }  
};
```

```
choix(50); // imprime: e  
choix(30); // imprime: c  
choix(0);  // imprime: e  
choix(90); // imprime: e
```

Un **switch** peut
être remplacé par
une cascade de
ifs

Exemples

```
// Fichier: choix4.js

// Ce programme illustre le switch avec des cas texte.

var traduire = function (x) {

    switch (x) {
        case "un":      return 1;
        case "deux":    return 2;
        case "trois":   return 3;
        case "quatre":  return 4;
        default:        return -1;
    }
};

print( traduire("trois") ); // imprime: 3
print( traduire("dix") );  // imprime: -1
```

Les cas sont des expressions de n'importe quel type

Exemples

```
// Fichier: choix5.js

// Ce programme illustre le switch avec des cas texte.

var traduire = function (x) {

    switch (x) {
        case "un":      return "one";
        case "deux":    return "two";
        case "trois":    return "three";
        case "quatre":   return "four";
        default:        return "other";
    }
};

print( traduire("trois") ); // imprime: three
print( traduire("dix") );  // imprime: other
```

traduire du
français à
l'anglais

Exemples

```
// Fichier: choix6.js

// Ce programme illustre le switch avec des cas texte.

var traduire = function (x) {

    switch (x) {
        case "uno":
        case "un":      return "one";
        case "dos":
        case "deux":    return "two";
        case "tres":
        case "trois":   return "three";
        case "cuatro":
        case "quatre":  return "four";
        default:        return "other";
    }
};

print( traduire("trois") ); // imprime: three
print( traduire("dos") );  // imprime: two
```

traduire du français et
espagnol à l'anglais

Automates

```
// Fichier: automate1.js

// Utilisation de switch pour réaliser un automate.

var automate = function (etat) {

    while (true) {
        switch (etat) {
            case 1: print("un");      etat = 3; break;
            case 2: print("deux");    etat = 4; break;
            case 3: print("trois");   etat = 2; break;
            case 4: print("quatre");  return;
        }
    }
};

automate(1); // imprime?
```

imprime :
un trois deux quatre

Automates

```
// Fichier: automate1.js

// Utilisation de switch pour réaliser un automate.

var automate = function (etat) {

    while (true) {
        switch (etat) {
            case 1: print("un");      etat = 3; break;
            case 2: print("deux");    etat = 4; break;
            case 3: print("trois");   etat = 2; break;
            case 4: print("quatre");  return;
        }
    }
};

automate(1); // imprime?
```



Exemple : valider un texte

- Specification : on désire une fonction **valide** qui prend un texte en paramètre et qui retourne un booléen indiquant si le texte est conforme au format suivant : *il contient aucun autre caractère que 0 et 1 et un nombre pair de 1*
- Tests unitaires :

```
var testValide = function () { // tests unitaires
  assert( valide("10010") == true );
  assert( valide("1 1") == false );
  assert( valide("1101") == false );
  assert( valide("1") == false );
  assert( valide("11") == true );
  assert( valide("") == true );
};
```


Exemple : valider un texte

```
// Fichier: valide1.js

// Validation d'un texte binaire à parité paire.

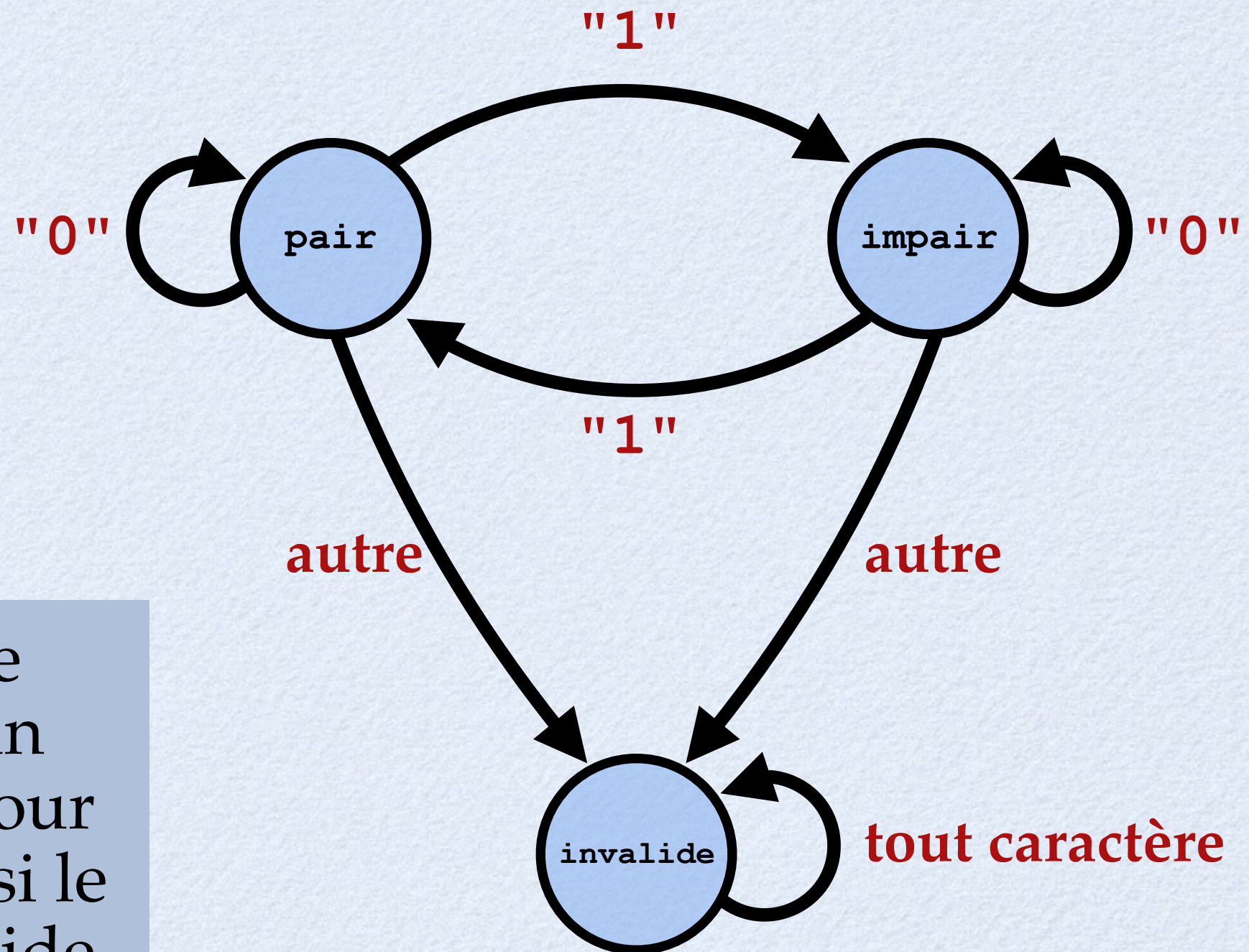
var valide = function (t) {

    var n = 0;

    for (var i=0; i<t.length; i++) {
        switch (t.charAt(i)) {
            case "0": break;
            case "1": n++; break;
            default: return false;
        }
    }

    return (n % 2) == 0;
};
```


Exemple : valider un texte



Approche
utilisant un
automate pour
déterminer si le
texte est valide

Exemple : valider un texte

```
// Fichier: valide2.js

// Validation d'un texte binaire à parité paire.

var valide = function (t) {

    var pair = true;

    for (var i=0; i<t.length; i++) {
        switch (t.charAt(i)) {
            case "0": break;
            case "1": switch (pair) {
                case true: pair = false; break;
                case false: pair = true; break;
            }
            break;
            default: return false;
        }
    }

    return pair;
};
```


Exemple : valider un texte

```
// Fichier: valide3.js

// Validation d'un texte binaire à parité paire.

var valide = function (t) {

    var pair = true;

    for (var i=0; i<t.length; i++) {
        switch (t.charAt(i)) {
            case "0": break;
            case "1": pair = !pair; break;
            default: return false;
        }
    }

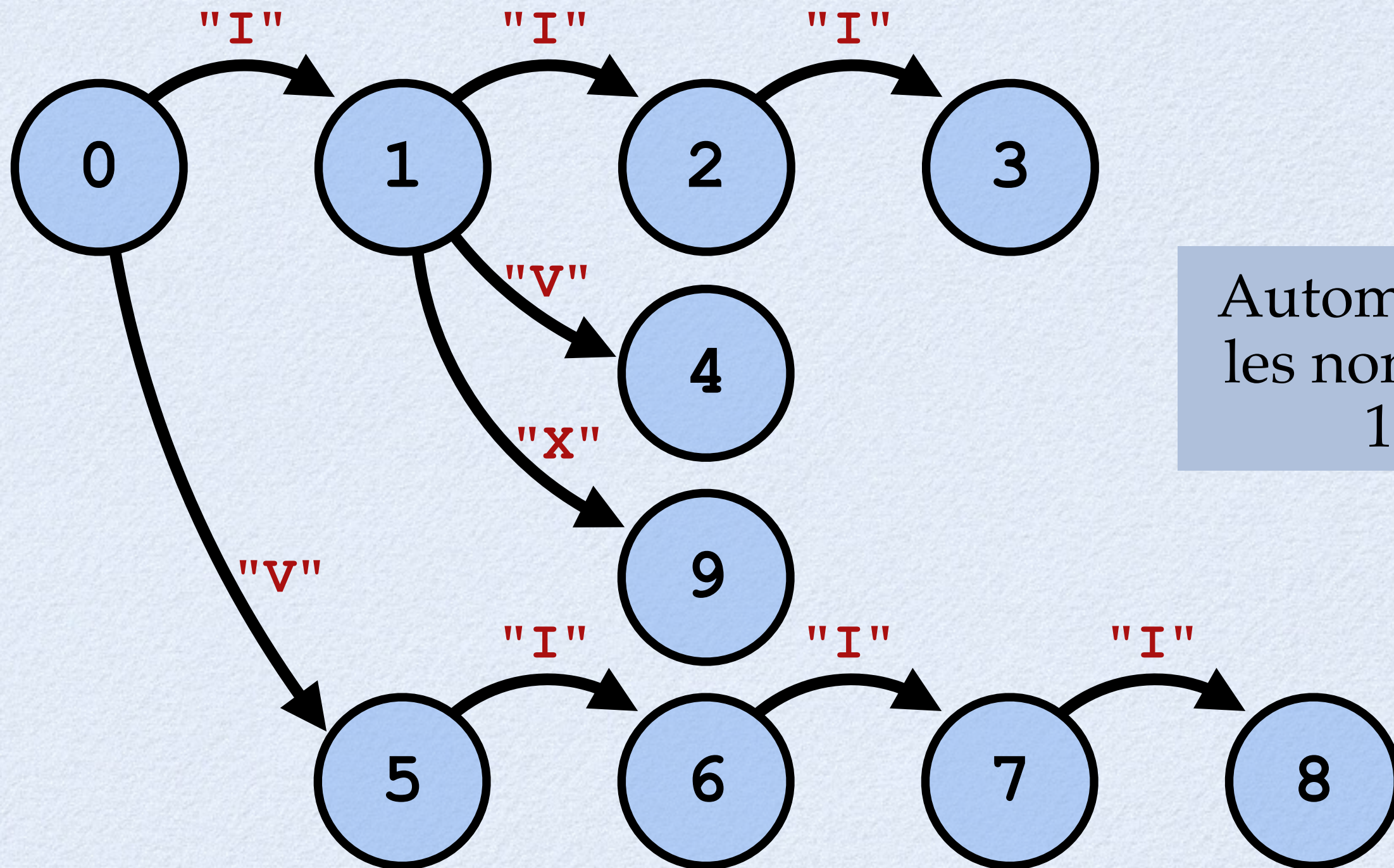
    return pair;
};
```


Exemple : nombres romains

- Specification : on désire une fonction **romain** qui prend un texte en paramètre et qui retourne un entier. Cet entier est -1 si le texte n'est pas un nombre romain entre 1 et 3999, sinon l'entier retourné est la valeur du nombre romain.
- Tests unitaires :

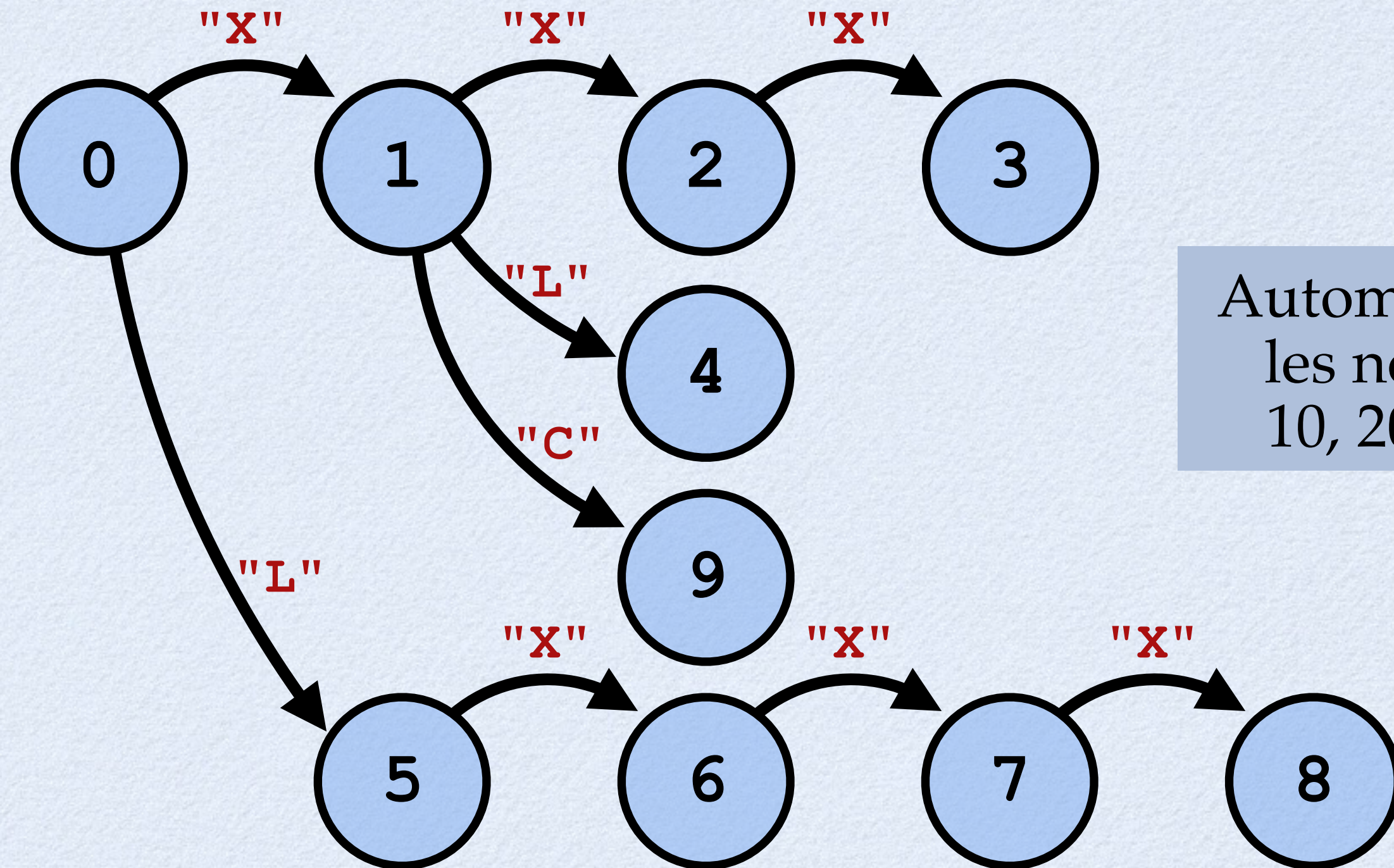
```
var testRomain = function () { // tests unitaires
    assert( romain("I")         == 1      );
    assert( romain("XX")        == 20     );
    assert( romain("CIV")       == 104    );
    assert( romain("CCCLXXX")   == 380    );
    assert( romain("MCMVII")    == 1907   );
    assert( romain("MMMCMXCIX") == 3999   );
    assert( romain("CIL")       == -1     );
    assert( romain("Z")         == -1     );
    assert( romain("")          == -1     );
};
```


Exemple : nombres romains



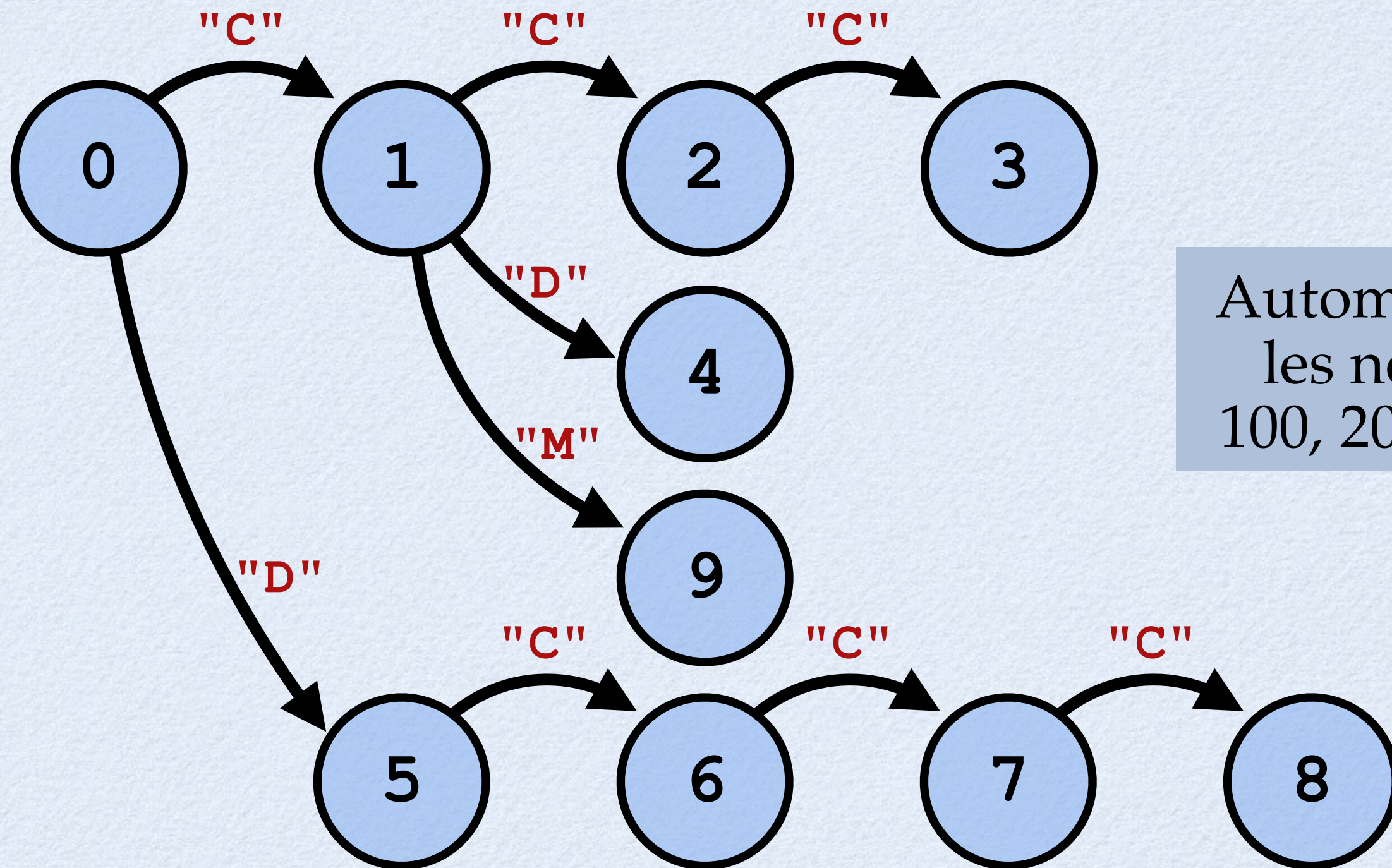
Automate pour
les nombres de
1 à 9

Exemple : nombres romains



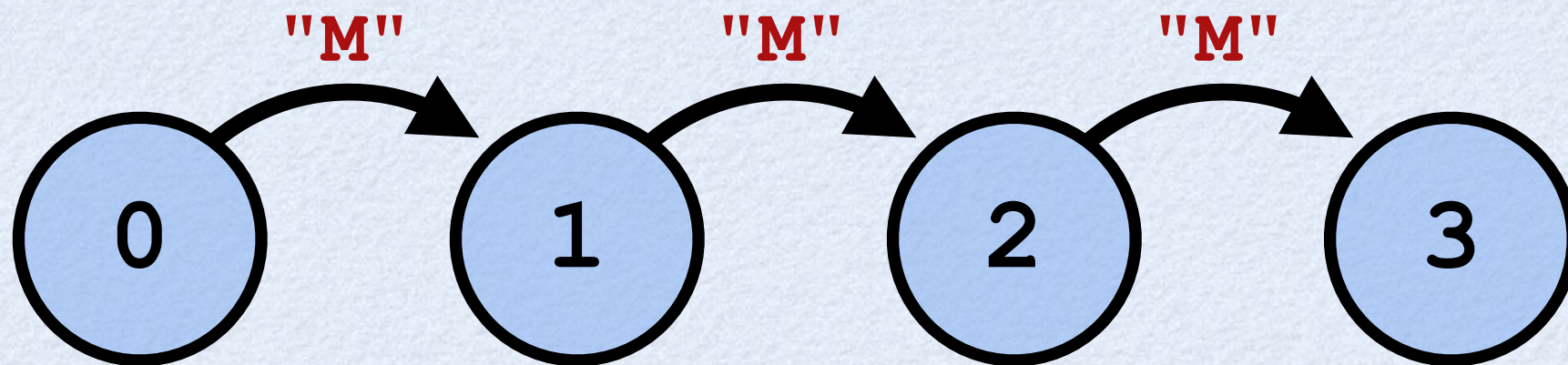
Automate pour
les nombres
10, 20, ... 90

Exemple : nombres romains



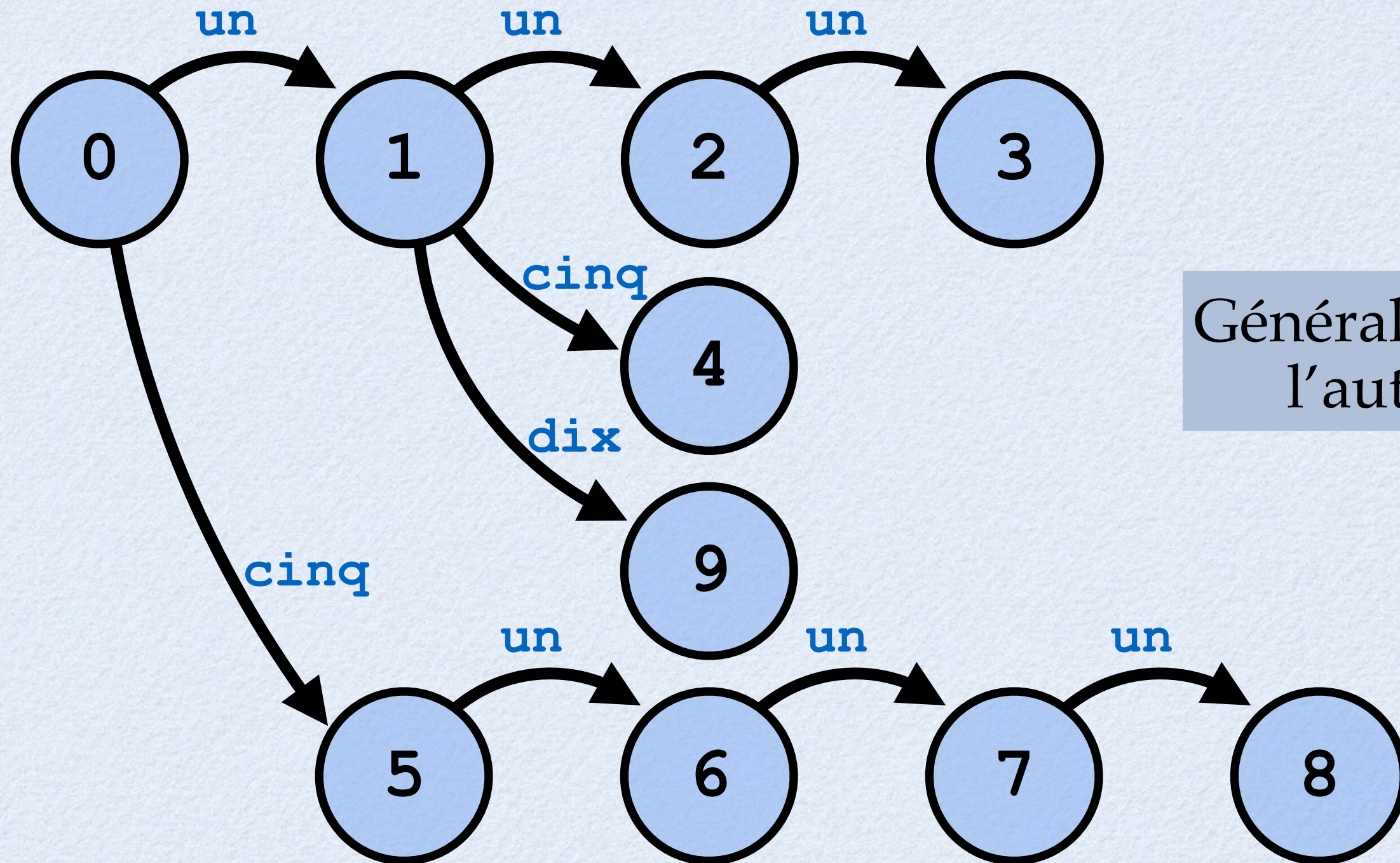
Automate pour
les nombres
100, 200, ... 900

Exemple : nombres romains



Automate pour
les nombres
1000, 2000, 3000

Exemple : nombres romains



Généralisation de
l'automate


```

var romain = function (t) {

    var pos = 0;

    var prochain = function () {
        if (pos < t.length) {
            return t.charAt(pos);
        } else {
            return " ";
        }
    };

    var chiffre = function (un, cinq, dix) {

        var val = 0;

        while (true) {

            switch (prochain()) {

                case un:    if (val <= 2 || (val >= 5 && val <= 7)) {
                            val += 1;
                            pos++;
                        } else {
                            return val;
                        }
                        break;

                case cinq: if (val <= 1) {
                            val = 5 - val;
                            pos++;
                        } else {
                            return val;
                        }
                        break;

                case dix:  if (val == 1) {
                            val = 9;
                            pos++;
                        } else {
                            return val;
                        }
                        break;

                default:   return val;
            }
        }
    };

    var m = chiffre("M", "", " ");
    var c = chiffre("C", "D", "M");
    var d = chiffre("X", "L", "C");
    var u = chiffre("I", "V", "X");

    if (pos == t.length && pos > 0) {
        return m*1000 + c*100 + d*10 + u;
    } else {
        return -1;
    }
};

```



```

var romain = function (t) {

    var pos = 0;

    var prochain = function () {
        if (pos < t.length) {
            return t.charAt(pos);
        } else {
            return " ";
        }
    };

    var chiffre = function (un, cinq, dix) { ... };

    var m = chiffre("M", "", "");
    var c = chiffre("C", "D", "M");
    var d = chiffre("X", "L", "C");
    var u = chiffre("I", "V", "X");

    if (pos == t.length && pos > 0) {
        return m*1000 + c*100 + d*10 + u;
    } else {
        return -1;
    }

};

```



```
var chiffre = function (un, cinq, dix) {  
  
    var val = 0;  
  
    while (true) {  
  
        switch (prochain()) {  
  
            case un:    if (val <= 2 || (val >= 5 && val <= 7)) {  
                        val += 1;  
                        pos++;  
                    } else {  
                        return val;  
                    }  
                    break;  
  
            case cinq: if (val <= 1) {  
                        val = 5 - val;  
                        pos++;  
                    } else {  
                        return val;  
                    }  
                    break;  
  
            case dix:  if (val == 1) {  
                        val = 9;  
                        pos++;  
                    } else {  
                        return val;  
                    }  
                    break;  
  
            default:   return val;  
        }  
    }  
};
```