

<http://irem.univ-reunion.fr/spip.php?article595>



Machines de Turing avec CSS3

- Lycée et post-bac
- Informatique et sciences du numérique



Date de mise en ligne : mardi 11 septembre 2012

Copyright © IREM de la Réunion - Tous droits réservés

La machine de Turing est simulée avec un tableau à deux lignes : La première est formée de cases initialement vides que la machine de Turing colorie (la bande) et la seconde ligne du tableau est formée de cases toutes invisibles sauf une, représentant la position actuelle de la machine de Turing (et affichant au passage l'état actuel de celle-ci). Du code CSS en JavaScript permet de modifier l'aspect du tableau et donc de simuler les déplacements de la machine de Turing.

Exemple 1

La plus vieille machine de Turing connue a été inventée par Turing lui-même : Elle colorie alternativement les cases parcourues en bleu et rouge [1]. Pour la tester, il suffit de cliquer quelques fois sur le bouton idoine.

```
var position1=0; var etat1="A"; var Turing1={"A":"B Blue R","B":"A Red R"}; function creer1(){ for(var n=0;n<=24;n++){ cell=document.createElement('td'); cell.style.border="1px Black solid"; cell.style.width="16px"; cell.style.height="16px"; cell.style.background="White"; cell.style.color="rgba(0,0,0,0)"; cell.innerHTML="."; cell.id="a"+n.toString(); document.getElementById('bande1').appendChild(cell); cell=document.createElement('td'); cell.id="m"+n.toString(); cell.style.border="0px White solid"; cell.style.width="16px"; cell.style.height="16px"; document.getElementById('machine1').appendChild(cell); } robot=document.getElementById('m'+position1.toString()); robot.style.border="3px Brown outset"; robot.style.background="Yellow"; robot.innerHTML=etat1; } function jouer1(){ ancien=document.getElementById("m"+position1.toString()); ancien.style.background="White"; ancien.style.border="0px White solid"; actuel=document.getElementById("a"+position1.toString()); actions=Turing1[actuel.style.background.split(" ")[5]][ancien.innerHTML.split(" ")]; ancien.innerHTML=""; if (actions[2]=="R"){position1++;}else{position1--}; actuel.style.background=actions[1]; etat1=actions[0]; nouveau=document.getElementById("m"+position1.toString()); nouveau.style.border="3px Brown outset"; nouveau.style.background="Yellow"; nouveau.innerHTML=etat1; } Machine de Turing
```

Si elle est dans l'état A

- En voyant du blanc, elle passe à l'état B, peint du bleu et va vers la droite;

Si elle est dans l'état B

- En voyant du blanc, elle passe à l'état A, peint du rouge et va vers la droite;

Circulez, y'a plein à voir !

Algorithmes

Donc une machine de Turing peut être représentée en html par un tableau de deux lignes, dont la première rangée est formée de cellules d'identifiants respectifs a0, a1, a2 etc. (jusqu'à a24) et dont la seconde ligne est formée de cellules d'identifiants respectifs m0, m1, etc. Parmi les cellules de la seconde ligne, une seule est visible, celle où est censée se trouver la machine de Turing.

Secrets de fabrication

Pour fabriquer en html un tableau de 50 cellules munies de leurs identifiants, mieux vaut utiliser JavaScript. Dans le document le tableau dont les deux lignes ont pour identifiants respectifs « bande » et « machine » est juste décrit au minimum par le code html suivant :

Les cellules sont ajoutées l'une après l'autre dans ce tableau avec une fonction JavaScript lancée par la méthode « onLoad » de la page html. Dans une boucle sur n (allant de 0 à 24), on fait les opérations suivantes :

- création d'une cellule (élément de type « td ») avec `cell=document.createElement('td')` (stockée dans une variable « cell ») ;
- style de la bordure de cette cellule sur « trait plein, noir, de 1 pixel d'épaisseur », avec `cell.style.border='1px Black solid'` ;
- dimensions de la cellule fixées à 16 pixels avec `cell.style.width='16px'` et `cell.style.height='16px'` ;
- couleur de la cellule sur blanc avec `cell.style.background='White'` ;
- affichage d'un caractère invisible (les cellules vides ne s'affichant pas sous Firefox), avec une couleur transparente `cell.style.color='rgba(0,0,0,0)'` et l'affichage dans cette couleur transparente, d'un point avec `cell.innerHTML='.'` ;
- Attribution d'un identifiant variable à la cellule avec `cell.id='a'+n.toString()` ;
- Placement de cette cellule dans la première ligne du tableau avec `document.getElementById('bande').appendChild(cell)`.

Ensuite, une suite d'opérations analogue est effectuée sur la cellule m0-m1-etc. avec

```
cell=document.createElement('td');
cell.id='m'+n.toString();
cell.style.border='0px White solid';
cell.style.width='16px';
cell.style.height='16px';
document.getElementById('machine').appendChild(cell);<div class='code_download' style='text-align: right;'> <a
href='http://irem.univ-reunion.fr/local/cache-code/cf0c42dcb421d4e95295040287429e5c.txt' style='font-family:
verdana, arial, sans; font-weight: bold; font-style: normal;'>Télécharger
```

Cette partie de la boucle crée 25 cellules invisibles dans la seconde ligne du tableau. Pour finir, l'une de ces cellules doit être rappelée à la réalité en lui donnant des propriétés CSS particulières : fond jaune, bord marron en relief de 3

pixels d'épaisseur, et affichage de l'état de la machine de Turing dans le cadre :

```
robot=document.getElementById('m'+position.toString());
robot.style.border='3px Brown outset';
robot.style.background='Yellow';
robot.innerHTML=etat;<div class='code_download' style='text-align: right;'> <a
href='http://irem.univ-reunion.fr/local/cache-code/4222a57abcd45bff29dae2774a91e7d.txt' style='font-family:
verdana, arial, sans; font-weight: bold; font-style: normal;'>Télécharger
```

La variable `etat` est une chaîne de caractères précédemment initialisée à `A A` ; de même que la variable `position` est un entier préalablement initialisé dans le script.

Enfin la machine de Turing est décrite par un graphe pondéré, en fait un tableau associatif en JavaScript, comme par exemple pour la machine de Turing universelle du dernier onglet :

```
var Turing={"White" : {"A":"B Blue R","B":"A Red L"},
"Blue" : {"A":"A Red L","B":"B Red R"},
"Red" : {"A":"A Blue L","B":"A White R"}};<div class='code_download' style='text-align: right;'> <a
href='http://irem.univ-reunion.fr/local/cache-code/227075230e1279202ac6c6c516eaf1ea.txt' style='font-family:
verdana, arial, sans; font-weight: bold; font-style: normal;'>Télécharger
```

La machine de Turing lit la couleur de la cellule qui est au-dessus d'elle (une chaîne de caractères en CSS, portant le nom de la couleur en anglais) et, selon cette couleur et son état actuel, reçoit une chaîne de caractères portant trois mots : Son prochain état, la couleur qu'elle va peindre à la place de celle qu'elle vient de lire, et le sens de déplacement qu'elle va adopter (R ou L selon le sens).

Chaque changement d'état de la machine de Turing se fait lors du clic sur un bouton html, lequel est muni d'une méthode `onClick` (chaque fois qu'on clique le bouton, la fonction `jouer` est appelée) :

```
<button id="go" onclick="jouer();">Allez !button>
```

Les opérations suivantes sont alors effectuées :

- Si par exemple la variable `position` est égale à 8, le texte `m'+position.toString()` est égal à `m8`, ce qui est l'identifiant de la cellule où se trouve la machine de Turing ; cette cellule est récupérée dans la variable `ancien` avec `ancien=document.getElementById('m'+position.toString())` ;
- La machine de Turing est alors rendue invisible avec `ancien.style.background='White'` et `ancien.style.border='0px White solid'` (fond blanc, bord blanc d'épaisseur nulle) ;
- La cellule à lire (couleur déterminant l'évolution de la machine de Turing) est stockée dans la variable `actuel` avec `actuel=document.getElementById('a'+position.toString())` ;
- La couleur portée par cette cellule est récupérée comme cinquième élément de son style CSS (`actuel.style.background.split(' ')[5]`) ; par exemple, si la cellule est blanche on aura le mot `White` ;
- L'entrée correspondant à cette couleur dans la description de la machine de Turing (tableau associatif) est un tableau associatif, dont l'entrée correspondant à l'état actuel de la machine de Turing (`ancien.innerHTML`) est lue. Par exemple, `actions=Turing[actuel.style.background.split(' ')[5]][ancien.innerHTML].split(' ')` transforme la chaîne de caractères `A Red L` en le tableau dont les trois entrées sont `A`, `Red` et `L` ;
- Maintenant qu'on n'a plus besoin de l'ancien état, on l'efface avec `ancien.innerHTML=""` ; `actions` contient alors un tableau comme `['A','Red','L']` ;
- Le mouvement de la machine de Turing se fait en incrémentant ou décrémentant sa position selon que `actions[2]` soit égal à `R` ou `L` : `if (actions[2]=='R') position++ ; else position-- ;` ; par exemple si la direction est

- `Â« L Â»` (comme `Â« left Â»`) la variable "position devient égale à 7 ;
- On peint la cellule devant laquelle se trouve la machine de Turing, avec la couleur `actions[1]` : `actuel.style.background=actions[1]` (par exemple, en rouge) ;
- Le changement d'état se fait avec `etat=actions[0]` (la variable `Â« etat Â»` est une variable globale) ;
- on récupère la cellule devant laquelle se trouve actuellement la machine de Turing avec `nouveau=document.getElementById('m'+position.toString())` ; par exemple la cellule d'identifiant `Â« m7 Â»` est adressée ;
- on modifie l'aspect de cette cellule en jouant sur ses propriétés CSS : `nouveau.style.border='3px Brown outset'` pour le cadre, `nouveau.style.background='Yellow'` pour la couleur de fond,
- et `nouveau.innerHTML=etat` pour afficher (au milieu du cadre) l'état de la machine de Turing.

C'est en affichant une cellule auparavant invisible et en cachant l'ancienne cellule, qu'on simule le mouvement de la machine de Turing.

Le problème de l'arrêt

Les machines de Turing présentées ici sont atypiques en ce que leur bande est initialement vide, et qu'elles ne s'arrêtent jamais. Pour simuler l'arrêt d'une machine de Turing, il suffit de rajouter un état `Â« H Â»` qui ne possède aucune entrée dans le tableau associatif, ce qui, au moment où la machine s'arrête, va bloquer le programme JavaScript, avec un message d'erreur dans la console d'erreur (que personne ne regarde jamais de toute façon).

Une méthode plus `Â« propre Â»` mais plus longue consiste à mettre toute la fonction `Â« jouer Â»` dans un test comme `if(etat !='H')...`

Exemple 2

Un compteur en binaire. Chaque fois que la machine de Turing arrive tout à droite (retour à la case départ), un nombre entier est codé en binaire sur la bande (une case bleue représentant 0 et une case rouge représentant 1) :

```
var position2=24; var etat2="A"; var Turing2={"White" : {"A":"B White L","B":"A Red R"}, "Blue" : {"A":"A Blue R","B":"A Red R"}, "Red" : {"A":"A Red R","B":"B Blue Left"}}; function creer2(){ for(var n =0;n<=24;n++){ cell=document.createElement('td'); cell.style.border="1px Black solid"; cell.style.width="16px"; cell.style.height="16px"; cell.style.background="White"; cell.style.color="rgba(0,0,0,0)"; cell.innerHTML="."; cell.id="b"+n.toString(); document.getElementById('bande2').appendChild(cell); cell=document.createElement('td'); cell.id="n"+n.toString(); cell.style.border="0px White solid"; cell.style.width="16px"; cell.style.height="16px"; document.getElementById('machine2').appendChild(cell); } robot=document.getElementById('n'+position2.toString()); robot.style.border="3px Brown outset"; robot.style.background="Yellow"; robot.innerHTML=etat2; } function jouer2(){ ancien=document.getElementById("n"+position2.toString()); ancien.style.background="White"; ancien.style.border="0px White solid"; actuel=document.getElementById("b"+position2.toString()); couleur=actuel.style.background.split(" ")[5]; actions=Turing2[couleur][ancien.innerHTML].split(" "); ancien.innerHTML=""; if (actions[2]=="R"){position2++;}else{position2--}; actuel.style.background=actions[1]; etat2=actions[0]; nouveau=document.getElementById("n"+position2.toString()); nouveau.style.border="3px Brown
```

outset"; nouveau.style.background="Yellow"; nouveau.innerHTML=etat2; } Machine de Turing qui compte

Si on code les bleus par des 0 et les rouges par des 1, à chaque passage par la case tout à droite, la machine affiche un nombre en binaire, qui est un de plus que le précédent (1, 10, 11, 100, 101, etc).

Si elle est dans l'état A

- En voyant du blanc, elle passe à l'état B, laisse du blanc et va vers la gauche;
- En voyant du bleu, elle reste à l'état A, laisse du bleu et va vers la droite;
- En voyant du rouge, elle reste à l'état A, laisse du rouge et va vers la droite;

Si elle est dans l'état B

- En voyant du blanc, elle passe à l'état A, peint du rouge et va vers la droite;
- En voyant du bleu, elle passe à l'état A, peint du rouge et va vers la droite;
- En voyant du rouge, elle reste à l'état B, peint du bleu et va vers la gauche;

Allez allez !

Exemple 3

Cet exemple était inconnu de Turing : La plus petite [machine de Turing](#) universelle [2], qui même sur une bande initialement vide, admet un comportement chaotique :

```
var position3=10; var etat3="A"; var Turing3={"White" : {"A":"B Blue R","B":"A Red L"}, "Blue" : {"A":"A Red L","B":"B Red R"}, "Red" : {"A":"A Blue L","B":"A White R"}}; function creer3(){ for(var n=0;n<=24;n++){
cell=document.createElement('td'); cell.style.border="1px Black solid"; cell.style.width="16px";
cell.style.height="16px"; cell.style.background="White"; cell.style.color="rgba(0,0,0,0)"; cell.innerHTML=".";
cell.id="c"+n.toString(); document.getElementById('bande3').appendChild(cell); cell=document.createElement('td');
cell.id="o"+n.toString(); cell.style.border="0px White solid"; cell.style.width="16px"; cell.style.height="16px";
document.getElementById('machine3').appendChild(cell); }
robot=document.getElementById('o'+position3.toString()); robot.style.border="3px Brown outset";
robot.style.background="Yellow"; robot.innerHTML=etat3; } function jouer3(){
ancien=document.getElementById("o"+position3.toString()); ancien.style.background="White";
ancien.style.border="0px White solid"; actuel=document.getElementById("c"+position3.toString());
couleur=actuel.style.background.split(" ")[5]; actions=Turing3[couleur][ancien.innerHTML.split(" ")]
ancien.innerHTML=""; if (actions[2]=="R"){position3++;}else{position3--;} actuel.style.background=actions[1];
etat3=actions[0]; nouveau=document.getElementById("o"+position3.toString()); nouveau.style.border="3px Brown outset";
nouveau.style.background="Yellow"; nouveau.innerHTML=etat3; } La machine de Turing (2,3) de Wolfram
```

C'est la plus petite des machines de Turing universelles (théorème d'Alex Smith). Ses deux états sont notés A et B, et la bande porte trois couleurs (le blanc initial, le bleu et le rouge).

Si elle est dans l'état A

- En voyant du blanc, elle passe à l'état B, peint du bleu et va vers la droite;
- En voyant du bleu, elle reste à l'état A, peint du rouge et va vers la gauche;
- En voyant du rouge, elle reste à l'état A, peint du bleu et va vers la gauche;

Si elle est dans l'état B

- En voyant du blanc, elle passe à l'état A, peint du rouge et va vers la gauche;
- En voyant du bleu, elle reste à l'état B, peint du rouge et va vers la droite;
- En voyant du rouge, elle passe à l'état A, peint du blanc (efface) et va vers la droite;

Allez allez !

[1] Dans son article, Turing lui faisait écrire des 0 et des 1 en alternance, ce qu'on peut aisément faire en remplaçant le CSS par des *innerHTML*.

[2] c'est-à-dire capable de simuler toute autre machine de Turing convenablement codée au préalable sur sa bande