Modèle probabiliste de perception

Objectif

Le but de ce TP est l'étude de la perception tactile de l'image des lettres en se basant sur des modèles probabilistes et sur la programmation sur ordinateur. Le modèle probabiliste de perception que l'on va décrire dans ce qui suit, montrera l'importance et la simplicité des méthodes mathématiques utilisées et constituera un exemple de la construction d'un modèle mathématique d'un phénomène psychologique, dans le but de l'appliquer à la construction d'automates doués de qualités intellectuelles (scanneur).

1. Principe de la perception

Un procédé de traduction des images des lettres commence par encadrer la lettre H, par exemple, par 25 petits carrés (figure 1). En indiquant un sens pour lire les carrés qui constituent le cadre de la lettre H, sens indiqué par les flèches, on obtient un déroulement dans le temps (figure 2). Les carrés, éléments du cadre, sont de deux sortes : occupé (noir) et libre (blanc). La lecture des éléments du cadre de la lettre peut être faite, par exemple, par une cellule photoélectrique qui se déplace suivant les colonnes dans le sens indiqué par les flèches, en comptant le nombre d'éléments noirs et blancs rencontrés et enregistrant le résultat obtenu. Cette méthode permet à la machine d'enregistrer toutes les lettres possibles de la langue considérée. Donc reconnaître une lette d'un texte reviendra à identifier l'image obtenue à son enregistrement en mémoire que nous appelons *référence*, sous hypothèse que la référence ne varie pas pendant l'expérience.

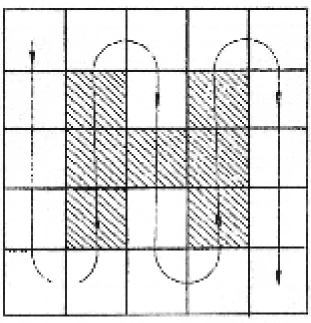

Figure 1

Figure 2

L'opération que doit faire la machine consiste donc à compter les deux types d'éléments du cadre et à comparer le résultat avec les références. Pour cela, on calcule le coefficient de corrélation entre le cadre considéré et chaque référence ; ce coefficient est donné par :

$$\rho(i,j) = \frac{p(i \cap j) - p(i)p(j)}{\sqrt{p(i)p(i_0)p(j)p(j_0)}}$$

- $p(i)$ et $p(i_0)$ sont respectivement les probabilités (fréquences) d'apparition de l'élément noir et blanc pour l'image perçue
- $p(j)$ et $p(j_0)$ sont respectivement les probabilités d'apparition de l'élément noir et blanc pour la référence
- $p(i \cap j)$ est la probabilité de coïncidence de l'élément noir de l'image perçue avec l'élément noir de la référence

Lorsque l'image coïncide avec la référence, on a $p(i) = p(j)$, $p(i_0) = p(j_0)$, $p(i \cap j) = p(i)$; le coefficient de corrélation devient :

$$\rho(i,j) = \frac{p(i) - p^2(i)}{p(i)(1-p(i))} = 1$$

Du fait de la possibilité de certains défauts dans l'impression du cadre, de déformations optiques, etc., l'image perçue peut ne pas correspondre exactement avec la référence ; on choisit donc l'image caractérisée par le coefficient de corrélation maximal.

Considérons l'exemple des lettres H, Π, Λ et l'espace :

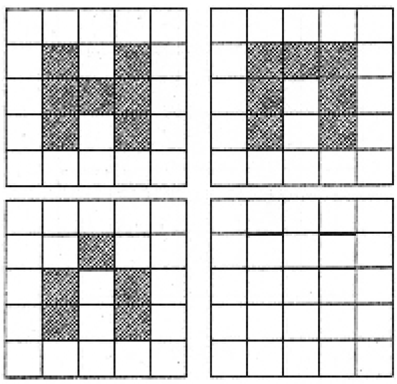

Figure 3

Question 1.1 : *écrire une fonction permettant d'afficher ces 4 lettres en introduisant les probabilités d'apparition des éléments du cadre p(i) et p(i_0) de l'image de chaque lettre.*

Algorithme de la fonction (réalisée sous Matlab puis en C)

Données
lettre, h, pi, lambda → tableaux de 25 booléens à une dimension (VRAI ≡ noir)
p(j), p(j_0) → réels correspondant aux probabilités d'apparition de l'élément noir/blanc dans la lettre de référence

Traitement
DEBUT
h = [0,0,0,0,0,0,1,1,1,0,0,0,1,0,0,0,1,1,1,0,0,0,0,0,0] *// lecture de chaque référence*
pi = [0,0,0,0,0,0,1,1,1,0,0,1,0,0,0,0,1,1,1,0,0,0,0,0,0]
lambda = [0,0,0,0,0,0,1,1,0,0,0,1,0,0,0,0,1,1,0,0,0,0,0,0,0]
h(j) = 7/25 ; h(j_0) = 18/25 *// probabilités d'apparition dans la référence*
pi(j) = 7/25 ; pi(j_0) = 18/25
lambda(j) = 5/25 ; lambda(j_0) = 20/25
p(i) = 0 ; p(i_0) = 0 *// probabilités d'apparition dans l'image perçue*
h(i∩j) = 0 ; pi(i∩j) = 0 ; lambda(i∩j) = 0 *// probabilités de coïncidence*

```
pour k de 0 à 24 faire
    si lettre[k] = VRAI alors
        p(i) = p(i) + 1
        si h[k] = VRAI alors h(i∩j) = h(i∩j) + 1 finsi
        si pi[k] = VRAI alors pi(i∩j) = pi(i∩j) + 1 finsi
        si lambda[k] = VRAI alors lambda(i∩j) = lambda(i∩j) + 1 finsi
    finsi
finpour
```

p(i) = p(i) / 25 ; p(i_0) = p(i_0) / 25
h(i∩j) = h(i∩j) / 25 ; pi(i∩j) = pi(i∩j) / 25 ; lambda(i∩j) = lambda(i∩j) / 25

// coefficients de corrélation pour chaque lettre référence

$$\rho(h) = \frac{h(i \cap j) - p(i) * h(j)}{\sqrt{p(i) * p(i_0) * h(j) * h(j_0)}}$$

$$\rho(pi) = \frac{pi(i \cap j) - p(i) * pi(j)}{\sqrt{p(i) * p(i_0) * pi(j) * pi(j_0)}}$$

$$\rho(lambda) = \frac{lambda(i \cap j) - p(i) * lambda(j)}{\sqrt{p(i) * p(i_0) * lambda(j) * lambda(j_0)}}$$

resultat = max (ρ(h) , ρ(pi) , ρ(lambda))

// affichage des résultats
si resultat = ρ(h) alors écrire « la lettre la plus proche est H » finsi
si resultat = ρ(pi) alors écrire « la lettre la plus proche est PI » finsi
si resultat = ρ(lambda) alors écrire « la lettre la plus proche est lambda » finsi

FIN

➔ *code source en annexe.*

Question 1.2 : *que peut-on en conclure ?*

Certaines images perçues peuvent correspondre à plusieurs références ; en effet la vérification se base uniquement sur un « comptage », ce qui peut amener à des ambiguïtés notamment dans le cas de H et PI qui ont le même nombre de cases noires, mais placées en des endroits différents.

Le calcul du coefficient de corrélation est impossible dans le cas de la lettre référence « espace » car il engendre un dénominateur nul ; c'est pour cela que la lettre espace n'apparaît pas dans l'algorithme.

2. Lecture d'un texte

Considérons maintenant un système complet d'événements (hypothèses) $A_1,\ldots,A_n$ pouvant être mis en liaison avec l'image perçue et correspondant aux références. Supposons que l'on connaisse les probabilités d'apparition des références $p(A_1),\ldots p(A_n)$. Considérons maintenant l'événement K_r « le $r^{ième}$ élément du cadre est noir », $1 \leq r \leq M$ avec M le nombre d'éléments composant chaque référence et K_r^c l'événement contraire. Bien entendu les événements $A_1,\ldots,A_n$ ne sont pas indépendants des événements K_r^c. Si l'on examine le $r^{ième}$ carré du cadre et si l'on constate que c'est l'événement K_r qui a lieu, alors les probabilités des hypothèses se modifiant, elles deviennent des *probabilités conditionnelles* $p(A_k/K_r)$, $1 \leq k \leq n$ données par la relation suivante :

$$p(A_k / K_r) = \frac{p(A_k)p(K_r / A_k)}{p(K_r)}$$

Sous hypothèse que $A_1,\ldots,A_n$ constituent un système complet d'événements, car on déchiffre des textes écrits seulement à l'aide de ces lettres, on en déduit :

$$p(A_1 / K_r) + \ldots + p(A_n / K_r) = 1$$

Notons que les probabilités des hypothèses (références) s'obtiennent par une analyse statistique préalable.

Supposons que nous voulions déchiffrer un texte écrit à l'aide de quatre lettres seulement : H, Π, Λ et l'espace (n = 4). Dans ce cas nous avons :

$A_1 \equiv H$	$p(A_1) = \frac{1}{2}$
$A_2 \equiv \Pi$	$p(A_2) = \frac{3}{10}$
$A_3 \equiv \Lambda$	$p(A_3) = \frac{1}{10}$
$A_4 \equiv$ espace	$p(A_4) = \frac{1}{10}$

Question 2.1 : *après perception des $p(K_1^c/A_k)$, calculer $p(K_1^c)$*

Perception des $p(K_1^c/A_k)$:

- $p(K_1^c/A_1) = 1$ (« la probabilité que le premier élément de la lettre H soit blanc »)
- $p(K_1^c/A_2) = 1$
- $p(K_1^c/A_3) = 1$
- $p(K_1^c/A_4) = 1$

On sait que $p(A \cap B) = p(B)*p(A/B)$ d'où $p(A/B) = p(A \cap B) / p(B)$

Il vient :

$p(K_1^c) = p(K_1^c \cap A_1) + p(K_1^c \cap A_2) + p(K_1^c \cap A_3) + p(K_1^c \cap A_4)$
$p(K_1^c) = p(K_1^c/A_1)*p(A_1) + p(K_1^c/A_2)*p(A_2) + p(K_1^c/A_3)*p(A_3) + p(K_1^c/A_4)*p(A_4)$

$p(K_1^c) = p(A_1) + p(A_2) + p(A_3) + p(A_4)$

Or le système d'événements est complet, donc $p(A_1) + p(A_2) + p(A_3) + p(A_4) = 1$, soit

$$\boxed{p(K_1^c) = 1}$$

Question 2.2 : *déduire $p(A_k/K_1^c)$, que peut-on conclure ?*

$$p(A_k / K_1^c) = \frac{p(A_k) * p(K_1^c / A_k)}{p(K_1^c)} = \frac{p(A_k) * 1}{1} = p(A_k)$$

➔ le fait de savoir que la première case est blanche n'apporte aucune information sur la lettre recherchée ; en effet la première case de chaque référence est blanche.

Question 2.3 : *que constate-t-on par rapport à $K_2^c, \ldots K_6^c$?*

Le phénomène observé précédemment est identique pour les 6 premières cases de chaque référence, donc $p(A_k/K_r^c) = p(A_k)$ pour $1 \leq r \leq 6$.

Question 2.4 : *calculer $p(K_7)$ et déduire $P(A_k/K_7)$*

$p(K_7) = p(A_1) + p(A_2) + p(A_3) = 9/10$, d'où :

$$p(A_1 / K_7) = \frac{p(A_1) * p(K_7 / A_1)}{p(K_7)} = \frac{p(A_1)}{p(K_7)} = \frac{1}{2} * \frac{10}{9} = \frac{5}{9}$$
$$p(A_2 / K_7) = \frac{p(A_2) * p(K_7 / A_2)}{p(K_7)} = \frac{p(A_2)}{p(K_7)} = \frac{3}{10} * \frac{10}{9} = \frac{1}{3}$$
$$p(A_3 / K_7) = \frac{p(A_3) * p(K_7 / A_3)}{p(K_7)} = \frac{p(A_3)}{p(K_7)} = \frac{1}{10} * \frac{10}{9} = \frac{1}{9}$$

Question 2.5 : *conclure*

$p(A_1/K_7) + p(A_2/K_7) + p(A_3/K_7) = 1$, d'où $p(A_4/K_7) = 0$

➔ Connaître la couleur de la case 7 nous permet d'éliminer ou non la référence 4 (espace) ; dans 9/10 des cas, cette lettre n'apparaît pas.

Question 2.6 : *répondre aux questions 2.3, 2.4 et 2.5 pour l'élément K_8*

$p(K_8) = p(K_7) = 9/10$ d'où :

$$p(A_1 / K_8) = \frac{p(A_1) * p(K_8 / A_1)}{p(K_8)} = \frac{p(A_1)}{p(K_8)} = \frac{1}{2} * \frac{10}{9} = \frac{5}{9}$$

$$p(A_2/K_8) = \frac{p(A_2)*p(K_8/A_2)}{p(K_8)} = \frac{p(A_2)}{p(K_8)} = \frac{3}{10}*\frac{10}{9} = \frac{1}{3}$$
$$p(A_3/K_8) = \frac{p(A_3)*p(K_8/A_3)}{p(K_8)} = \frac{p(A_3)}{p(K_8)} = \frac{1}{10}*\frac{10}{9} = \frac{1}{9}$$

➔ on obtient des résultats similaires à ceux de K_7.

Considérons maintenant les trois références A_1, A_2, A_3 ayant respectivement les probabilités 5/9, 3/9, 1/9 et qui joueront les rôles de $p(A_1)$, $p(A_2)$, $p(A_3)$.

Question 2.7 : *calculer $p(K_9)$ et déduire $p(A_k/K_9)$*

$p(K_9) = p(A_1) + p(A_2) = 8/9$ d'où :

$$p(A_1/K_9) = \frac{p(A_1)*p(K_9/A_1)}{p(K_9)} = \frac{p(A_1)}{p(K_9)} = \frac{5}{9}*\frac{9}{8} = \frac{5}{8}$$
$$p(A_2/K_9) = \frac{p(A_2)*p(K_9/A_2)}{p(K_9)} = \frac{p(A_2)}{p(K_9)} = \frac{3}{9}*\frac{9}{8} = \frac{3}{8}$$

➔ $p(A_1/K_9) + p(A_2/K_9) = 1$ donc $p(A_3/K_9) = 0$.

Question 2.8 : *conclure*

Si la case 9 est noire, on n'a aucune chance d'avoir la lettre lambda ni l'espace.

Question 2.9 : *que constate-t-on par rapport à K_{10} et K_{11} ?*

Les cas K_{10} et K_{11} sont les mêmes que $K_1,\ldots,K_6$: la couleur de la case est identique pour chacune des références.

Question 2.10 : *calculer $p(K_{12}^c)$ et déduire les probabilités $p(A_k/K_{12}^c)$*

$p(K_{12}^c) = p(A_1) = 5/9$ d'où :

$$p(A_1/K_{12}^c) = \frac{p(A_1)*p(K_{12}^c/A_1)}{p(K_{12}^c)} = \frac{p(A_1)}{p(K_{12}^c)} = \frac{5}{9}*\frac{9}{5} = 1$$

➔ $p(A_1/K_{12}^c) = 1$ donc $p(A_2/K_{12}^c) = p(A_3/K_{12}^c) = 0$.

Question 2.11 : *déduire la référence la plus probable à réaliser*

Avec une probabilité d'apparition de 5/9, la lettre H semble être la référence la plus probable à réaliser.

3. Les éléments critiques

L'examen du 7^{ième}, 9^{ième} et 12^{ième} élément montre que ces éléments apportent quelque chose d'essentiel dans le processus de perception de l'image perspective, ce qui permet de les qualifier de *critiques*. Dans l'examen de toute l'image, il y a de tels éléments critiques. Il en résulte que le processus de perception pourrait être écourté considérablement si l'on savait au préalable quels sont les éléments critiques, c'est-à-dire les éléments qui ne coïncident pas dans toutes les références. Calculons la probabilité de l'événement K_r, $1 \leq r \leq M$:

$$p(K_r) = \frac{m_r}{n}$$

m_r indiquant le nombre de fois où le r^{ième} est noir dans les n références. Les éléments critiques seront les éléments pour lesquels la probabilité $p(K_r)$ est différente de zéro ou un.

La vérification d'un tel élément du cadre nous apporte donc une certaine quantité d'information égale à :

$$H_r = -p(K_r)\log p(K_r) - (1 - p(K_r))\log(1 - p(K_r))$$

Cette quantité d'information est maximale pour $p(K_r) = 1/2$. Parmi les éléments composant les références, il y en a certains pour lesquels la probabilité de l'événement K_r est différente de zéro ou un ; ce sont les éléments critiques dont la vérification fournit une certaine quantité d'information. Les éléments critiques pour lesquels la probabilité de l'événement K_r est 1/2 nous donnent la plus grande quantité d'information, ce seront donc les éléments critiques *optimaux*. Du point de vue pratique, lorsque l'on a n références, chacune étant formée par M éléments, on calcule d'abord la probabilité de l'événement K_r et on choisira les éléments pour lesquels la probabilité K_r est 1/2, ce seront les éléments critiques optimaux. La vérification d'un point critique optimal éliminant la moitié des hypothèses, on choisit le nombre d'éléments critiques optimaux qui soit suffisant pour l'élimination de toutes les hypothèses, à l'exception d'une seule.

Question 3.1 : *calculer les probabilités critiques des éléments du cadre par rapport aux quatre références A_1, A_2, A_3 et A_4*

Calculs d'après la formule donnée $p(K_r) = \frac{m_r}{n}$:

$p(K_1) = 0$	$p(K_6) = 0$	$p(K_{11}) = 0$	$p(K_{16}) = 0$	$p(K_{21}) = 0$
$p(K_2) = 0$	$p(K_7) = 3/4$	$p(K_{12}) = 1/2$	$p(K_{17}) = 3/4$	$p(K_{22}) = 0$
$p(K_3) = 0$	$p(K_8) = 3/4$	$p(K_{13}) = 1/4$	$p(K_{18}) = 3/4$	$p(K_{23}) = 0$
$p(K_4) = 0$	$p(K_9) = 1/2$	$p(K_{14}) = 0$	$p(K_{19}) = 1/2$	$p(K_{24}) = 0$
$p(K_5) = 0$	$p(K_{10}) = 0$	$p(K_{15}) = 0$	$p(K_{20}) = 0$	$p(K_{25}) = 0$

➔ les éléments critiques sont : K_7, K_8, K_9, K_{12}, K_{13}, K_{17}, K_{18} et K_{19}.

Question 3.2 : *déduire les éléments critiques optimaux*

D'après les calculs précédents, les éléments critiques optimaux sont : K_9, K_{12} et K_{19}.

Question 3.3 : *calculer p(K_9) et déduire les références possibles à réaliser*

$$p(K_9) = \frac{m_9}{4} = \frac{2}{4} = \frac{1}{2}$$

➔ les références possibles à réaliser sont A_1 et A_2.

Considérons maintenant les deux références A_1 et A_2 avec les probabilités 5/8 et 3/8.

Question 3.4 : *calculer de nouveau les probabilités critiques des éléments du cadre par rapport à A_1 et A_2*

On sait que p(A_1) = 5/8, or la case 12 détermine complètement la référence (A_1 si blanc, A_2 si noir), donc p(K_{12}^c) = p(A_1) = 5/8.
De même la case 13 détermine complètement la référence (A_1 si noir, A_2 si blanc), donc p(K_{13}) = p(A_2) = 3/8.

Question 3.5 : *calculer p(A_1 / K_{12}^c) et déduire l'image perçue*

On sait que p(A1) = p(K12c) = 5/8 et p(K12c/A1) = 1 (la case 12 est forcément blanche si la lettre est A1), d'où :

$$p(A_1 / K_{12}^c) = \frac{p(A_1)p(K_{12}^c / A_1)}{p(K_{12}^c)} = \frac{5/8*1}{5/8} = 1$$

➔ sachant que la case 12 est blanche, on a la référence A1 comme lettre dans tous les cas, soit H.

Question 3.6 : *proposer un algorithme permettant de percevoir l'image à l'aide de ces éléments critiques*

Dans notre cas, avec les lettres H, PI, LAMBDA et ESPACE nous avons trouvé les éléments critiques optimaux K_9, K_{12}, K_{19}. Nous allons donc réaliser des tests sur ces cases pour déterminer la lettre qui se rapproche le plus en terme de probabilité de celle entrée par l'utilisateur.

Algorithme (réalisé sous Flash en ActionScript)

Données

cases cochées par l'utilisateur → tableau de 25 cases qui prennent les valeurs VRAI ou FAUX selon qu'elles soient cochées ou décochées

Traitement

DEBUT

```
    test_case = 0                // variable de test

    // on teste si les cases critiques sont cochées

    si (case[19] est cochée) alors test_case = test_case + 1
    finsi

    si (case[12] est cochée) alors test_case = test_case + 2
    finsi

    si (case[9] est cochée) alors test_case = test_case + 4
    finsi

    // suivant la valeur de test_case, on est capable de déterminer la lettre la plus proche

    cas où test_case = 0 : le caractère est ESPACE
    cas où test_case = 2 : le caractère est LAMBDA
    cas où test_case = 5 : le caractère est H
    cas où test_case = 7 : le caractère est PI
    autres cas : le caractère est inconnu
```

FIN

➔ *code source en annexe ; le choix de Flash nous a semblé le plus simple étant donnée la nature de l'application à réaliser.*

4. Conclusion

Ce TP nous a permis d'effectuer une reconnaissance de caractères de deux manières différentes en utilisant les probabilités.

La première consiste à calculer les probabilités pour chaque case sur chaque lettre de l'alphabet mémorisé, puis de regarder le plus fort coefficient de corrélation pour trouver la lettre qui se rapproche le plus de la lettre donnée par l'utilisateur.

La seconde est plus optimisée ; elle nécessite moins de calculs du fait de l'utilisation des éléments critiques optimaux. Elle semble mieux adaptée au traitement d'alphabets plus importants (par exemple nos 26 lettres) mais nécessite dans ce cas un nouveau calcul des éléments critiques qui diffèrent de ceux d'un alphabet de 4 cacractères.

ANNEXES

Code Matlab (fichier tp1_partie1.m)

```
function tp1_partie1(lettre)

clear;

% lettre à tester
lettre=input('Entrer la séquence correspondant à la lettre à tester :')

% lettres à comparer
a=[0,0,0,0,0,0,1,1,1,0,0,0,1,0,0,0,1,1,1,0,0,0,0,0,0];  % H
b=[0,0,0,0,0,0,1,1,1,0,0,1,0,0,0,0,1,1,1,0,0,0,0,0,0];  % PI
c=[0,0,0,0,0,0,1,1,0,0,0,1,0,0,0,0,1,1,0,0,0,0,0,0,0];  % LAMBDA

% probabilités des lettres à comparer
aj=7/25;    % fréquence d'apparition de l'élément noir pour H
aj0=18/25;  % fréquence d'apparition de l'élément blanc pour H
bj=7/25;    % fréquence d'apparition de l'élément noir pour PI
bj0=18/25;  % fréquence d'apparition de l'élément blanc pour PI
cj=1/5;     % fréquence d'apparition de l'élément noir pour LAMBDA
cj0=4/5;    % fréquence d'apparition de l'élément blanc pour LAMBDA

% initialisation des variables
p_i=0;  % fréquence d'apparition de l'élément noir pour la lettre donnée
p_i0=0; % fréquence d'apparition de l'élément blanc pour la lettre donnée
aij=0;  % probabilité de coincidence de l'élément noir de la lettre donnée avec H
bij=0;  % probabilité de coincidence de l'élément noir de la lettre donnée avec PI
cij=0;  % probabilité de coincidence de l'élément noir de la lettre donnée avec LAMBDA

% calcul des p_i, aij, bij, cij
for k=1:25
   if lettre(k)==1
      p_i=p_i+1;
      if a(k)==1
         aij=aij+1;
      end
      if b(k)==1
         bij=bij+1;
      end
      if c(k)==1
         cij=cij+1;
      end
   end
end

p_i=p_i/25;
p_i0=1-p_i;
aij=aij/25;
```

```
bij=bij/25;
cij=cij/25;

% calcul du coefficient de corrélation de chaque lettre
rho_a=(aij-p_i*aj)/(sqrt(p_i*p_i0*aj*aj0));
rho_b=(bij-p_i*bj)/(sqrt(p_i*p_i0*bj*bj0));
rho_c=(cij-p_i*cj)/(sqrt(p_i*p_i0*cj*cj0));

% recherche du coefficient le plus élevé
rho=[rho_a,rho_b,rho_c];
resultat=max(rho);

% affichage du résultat
if resultat==rho_a
  'la lettre la plus proche est H'
   precision=resultat
end
if resultat==rho_b
   'la lettre la plus proche est pi'
   precision=resultat
end
if resultat==rho_c
   'la lettre la plus proche est lambda'
   precision=resultat
end
```

Code Flash (fichier projet.fla)

<u>bouton de validation</u> :

```
on (release){
        test_case = 0;
        if (cb19.getValue() == true){
                test_case += 1;
        }
        if (cb12.getValue() == true){
                test_case += 2;
        }
        if (cb9.getValue() == true){
                test_case += 4;
        }

        switch (test_case){
                case 0: X = "ESPACE";
                                break;
                case 2: X = "LAMBDA";
                                break;
                case 5: X = "H";
                                break;
                case 7: X = "PI";
                                break;
                default: X = "CARACTERE INCONNU !";
        }
}
```