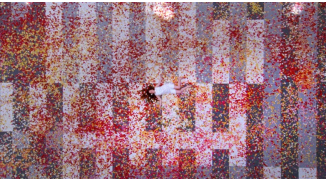


### Boucles étranges « Alter Ego »

Processus contrôlé, Point d'attention, Séquentialité  
Système de codages, Théorèmes d'itération  
Théorème de récursion de Kleene et Théorème du point fixe.  
Attention volontaire, Test Stroop, Cône de conscience,  
Ensemble sémantique, Décidabilité, Théorème de Rice



Sensibilité : Cloud of Petals (2018), de Sarah Meyohas Olivier Raynaud

1

---

---

---

---

---

---

---

---

### Venu des montagnes

□ Boucle étrange : ces murs qui nous parlent...



2

---

---

---

---

---

---

---

---

### Approche Méthodologique

□ Quelles approches pour des études dont le sujet est la « conscience » ?  
✓ L'intelligence artificielle est née en 1955, elle peut être un excellent terrain pour l'étude de la conscience.

**Approche méthodologique en deux étapes :**  
Démarche inspirée d'autres sujets d'études et domaines scientifiques (Physique, chimie, ...)

- 1) Identification et formalisation des grandes lois scientifiques « de base » des phénomènes naturels (analyse, expérimentation, ...)
- 2) Exploitation de ces lois par des processus d'ingénierie pour la réalisation d'applications utiles.

✓ L'objectif pourra être la recherche de principes de base, de mécanismes de la conscience sans pour cela vouloir prendre en compte toutes les spécificités de sa déclinaison chez l'homme.

[Extrait de « Une approche de la conscience... » par F. Anceau]

3

---

---

---

---

---

---

---

---

**Processus automatiques VS contrôlés**

	Un traitement (ou processus) automatique	Un traitement (ou processus) contrôlé
1)	Est d'exécution rapide	S'effectue relativement lentement
2)	Se réalise sans intervention de la conscience  Est non intentionnel Ne peut être interrompu une fois qu'il a été amorcé : il est irrépressible Est incontrôlable en cours d'exécution	Est contrôlé consciemment pour son déclenchement, son exécution et son interruption  Est intentionnel  Contrôlé et contrôlable tout au long de son exécution
3)	Consomme peu de ressources cognitives en MDT  Nécessite un faible effort cognitif	Consomme une grande partie des ressources en MDT ;  Nécessite un effort cognitif important
4)	Permet à d'autres traitements d'être réalisés en parallèle, dans la limite des ressources disponibles	Ne permet pas l'exécution en parallèle d'autres traitements, s'il est trop coûteux

Tableau 1: Principales caractéristiques des traitements automatiques et contrôlés

4

---

---

---

---

---

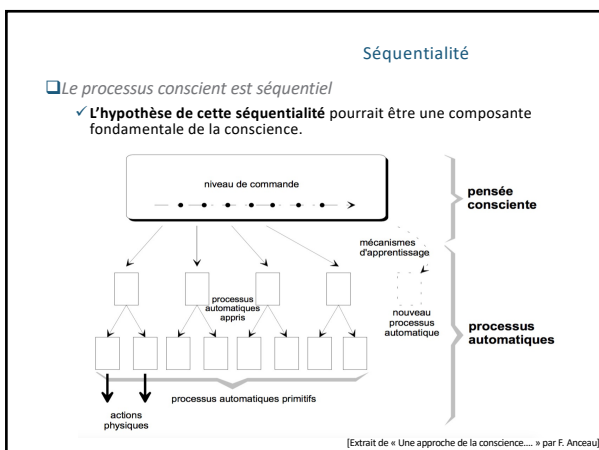
---

---

---

---

---



5

---

---

---

---

---

---

---

---

---

---

**Conscience réflexive**

☐ *L'existence*

- ✓ **Indépendant des fonctions de hauts niveaux**, raison, mémoire,...;
- ✓ **Capacité de penser**, je suis et je ressens le monde qui m'entoure;
- ✓ **Localisation dans le temps et l'espace**; je suis là, et maintenant;

☐ *Hypothèse de travail*

- ✓ **La conscience d'exister est binaire**, un être vivant en est doté ou non.
- ✓ **Exister et penser que nous existons peut être du même niveau abstrait.** Passer de l'existence à la constatation de son existence peut se faire par déduction ou agrégations de concepts.

6

---

---

---

---

---

---

---

---

---

---

**Attention volontaire**

☐ ..... Je porte mon attention donc je suis .....

« **L'attention** est la prise de possession par l'esprit, sous une forme claire et vive, d'un objet ou d'une suite de pensées parmi plusieurs qui sont présents simultanément [...]. Elle implique le retrait de certains objets afin de traiter plus efficacement les autres [...] »

*The Principles of Psychology* (James, 1890)

<https://www.elsevier.com/fr-fr/connect/psw/lattention-definition-et-quelques-concepts-les-tests-dattention-spatiale>

---

---

---

---

---

---

---

---

7

**Attention volontaire**

☐ *Test de Stroop*

✓ D'après les expériences de Stroop, lorsque le nom d'une couleur est écrit dans une couleur différente (mot incongruent), il est plus difficile de nommer la couleur dans laquelle il est écrit, que de le lire.

Ⓐ vert jaune rouge bleu jaune bleu vert rouge bleu vert

Ⓑ bleu rouge jaune vert rouge vert bleu jaune rouge jaune

Ⓒ vert bleu jaune rouge vert jaune rouge jaune vert bleu

rouge vert jaune vert bleu rouge bleu vert jaune bleu

Extrait de [[https://fr.wikipedia.org/wiki/Effet\\_Stroop](https://fr.wikipedia.org/wiki/Effet_Stroop)]

---

---

---

---

---

---

---

---

8

**Point d'attention**

☐ *Espace mental interprété comme une surface dynamique...*

F. Anceau

---

---

---

---

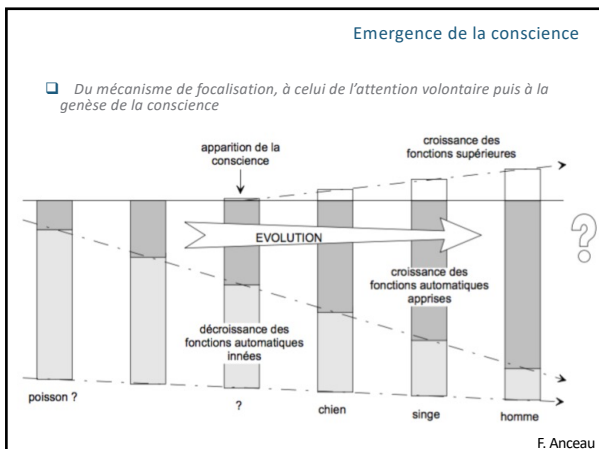
---

---

---

---

9



10

---

---

---

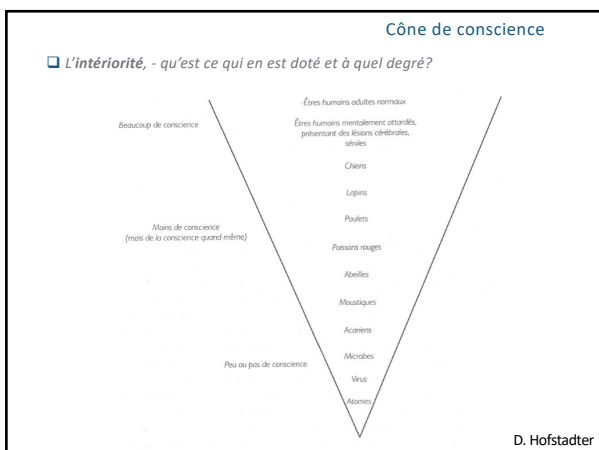
---

---

---

---

---



11

---

---

---

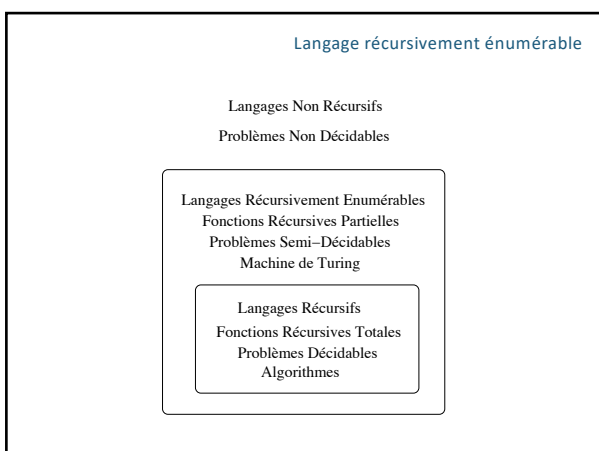
---

---

---

---

---



12

---

---

---

---

---

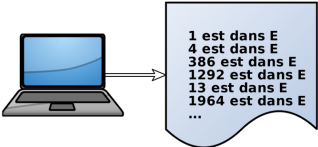
---

---

---

### Langage récursivement énumérable

**Définition (Langage récursivement énumérable)**  
 Un ensemble  $E$  d'entiers naturels est dit **récursivement énumérable** si :  
 - Il existe un algorithme prenant en entrée un entier naturel et qui s'**arrête** exactement sur les entiers de  $E$  ou de manière équivalente  
 - Il existe un procédé algorithmique qui au cours de son exécution **énumère** en sortie tous les entiers de  $E$  et seulement ceux-ci.



[https://fr.wikipedia.org/wiki/R%C3%A9cursivement\\_%C3%A9num%C3%A9rable](https://fr.wikipedia.org/wiki/R%C3%A9cursivement_%C3%A9num%C3%A9rable)

13

---

---

---

---

---

---

---

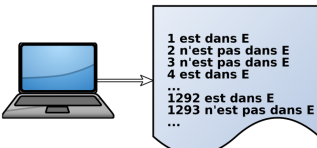
---

---

---

### Langage récursif

**Définition (Langage récursif)**  
 Un langage  $L$  est dit récursif **si et seulement si** il existe une machine de Turing  $M$  avec  $L = L(M)$  et telle que :  
 - si  $w \in L$  alors  $M$  accepte  $w$  (s'arrête);  
 - si  $w \notin L$   $M$  s'arrête dans un état non final.



[https://fr.wikipedia.org/wiki/Ensemble\\_r%C3%A9cursif](https://fr.wikipedia.org/wiki/Ensemble_r%C3%A9cursif)

14

---

---

---

---

---

---

---

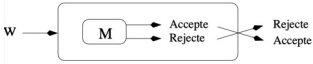
---

---


---

### Propriétés

**Propriété**  
 Soit  $L$  un langage, si  $L$  est récursif alors son complémentaire  $\bar{L}$  est aussi récursif.



**Propriété**  
 Soit  $L$  un langage, si un langage  $L$  et son complémentaire  $\bar{L}$  sont récursivement énumérables alors  $L$  et  $\bar{L}$  sont récursifs.



15

---

---

---

---

---

---

---

---

---

---

### Langage Diagonal

On supposera qu'on dispose d'un codage pour les machines de Turing.  
Ainsi  $w_i$  codera pour la machine  $M_i$ .

**Définition**  
Le langage Diagonal  $L_d$  est l'ensemble des chaînes  $w_i$  telles que  $w_i \notin L(M_i)$ .

**Propriété**  
Le langage  $L_d$  n'est pas récursivement énumérable.

**Éléments de démonstration**  
Supposons l'existence d'une machine  $M_i$  de codage  $w_i$  telle que  $L(M_i) = L_d$ .  
Posons nous alors la question : " $w_i \in L_d$ ?" :  
 - si  $w_i \in L_d$ , alors  $w_i \in L(M_i)$  et donc, par définition de  $L_d$ ,  $w_i \notin L_d$ .  
   Absurde ;  
 - si  $w_i \notin L_d$ , alors  $w_i \notin L(M_i)$  et donc, par définition de  $L_d$ ,  $w_i \in L_d$ .  
   Absurde.  
 Donc la machine  $M_i$  n'existe pas.

16

---

---

---

---

---

---

---

---

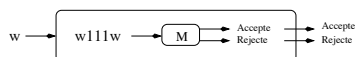
### Langage Universel

**Définition**  
Le langage Universel  $L_u$  est l'ensemble des chaînes  $(M, w)$  codant une machine de Turing  $M$  et une entrée  $w$  reconnue par la machine  $M$ .

**Propriété**  
Le langage  $L_u$  est récursivement énumérable mais pas récursif.

**Éléments de démonstration**

- Construire la machine  $U$  telle que  $L_u = L(U)$ .  
 Pour cela il suffit de décoder  $M$  et faire en sorte que  $U$  simule  $M$  pour l'entrée  $w$ . Ainsi  $U$  accepte  $(M, w)$  si et seulement si  $M$  accepte  $w$ .
- Montrons que  $L_u$  n'est pas récursif.  
 Supposons  $L_u$  récursif, alors  $L_u$  est aussi récursif.  
 Donc il existe une machine  $M$  qui accepte  $(w_i, w)$  si et seulement si la machine codée par  $w_i$  n'accepte pas  $w$ .



17

---

---

---

---

---

---

---

---

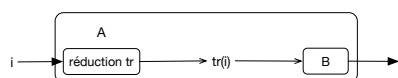
### Réduction de problème

**Définition**  
Une réduction d'un problème  $A$  à un problème  $B$  est une fonction  $tr()$  calculable telle que pour toute instance  $i$  de  $A$ ,  $i$  est une instance positive de  $A$  si et seulement si  $tr(i)$  est instance positive de  $B$ .

On dit que  $A$  se réduit à  $B$  s'il existe une réduction de  $A$  à  $B$ .

**Propriété**  
Soient  $A$  et  $B$  deux problèmes tel que  $A$  se réduit à  $B$ , si  $A$  n'est pas décidable alors  $B$  n'est pas décidable.

**Éléments de démonstration**



18

---

---

---

---

---

---


---

---

### Nombres de Godël

☐ Existence d'un codage

**Définition** (Code d'une fonction)  
 Nous appelons **code** d'une fonction l'image  $e$  de cette fonction par un système de codage donné  $\varphi$ , nous notons  $\varphi_e$  la fonction ayant pour code  $e$  dans ce système de codage.

  
Kurt Gödel (1906-1978)

**Théorème** (Numérotation de Godël)  
 Il existe une bijection primitive-réursive entre la classe *REC* et un sous ensemble  $\mathbb{G} \subset \mathbb{N}$ .

[https://en.wikipedia.org/wiki/Kurt\\_Gödel](https://en.wikipedia.org/wiki/Kurt_Gödel)

---

---

---

---

---

---

---

---

19

### Théorème d'itération

☐ Théorème d'itération (ou théorème  $S_{mn}$ ) de S. Kleene

**Théorème d'itération :**  
 Soit  $\varphi$  un système de codage des fonctions récursives, alors il existe une fonction calculable  $s$  telle que pour tout  $e$  et tous nombres  $x$  et  $y$  on a

$$\varphi_{s(e,x)}(y) = \varphi_e(x, y)$$

La fonction  $s$  est appelée fonction d'itération ou fonction s-m-n dans sa forme paramétrisée

$\varphi_e$   

$x, y$
ins. 1
ins. 2
...
ins. n

$\langle M_{e,x} \rangle$

$M_{e,x}$   

$y$
- recopie $e$
- remplace la tierce variable par $x$ dans $e$
- exécute le code avec $y$
(c.a.d. simule $\varphi_e(x, y)$ )

$\langle s \rangle$

$s$   

$e, x$
- retourne $\langle M_{e,x} \rangle$

$$\varphi_{s(e,x)}(y) = \varphi_{\langle M_{e,x} \rangle}(y) = \varphi_e(x, y)$$


---

---

---

---

---

---

---

---

20

### Théorème d'itération

☐ Exemple en LISP

**Macro Lisp:**  

```
(defun S11 (f x)
  (let ((y (gensym)))
    (list `lambda (list y) (list f x y))))
```

**Cas d'usage:**  
 $S_{11} \cdot (\text{lambda } (x \ y) \ (+ \ x \ y)) \ 3$   
 sera évalué  
 $(\text{lambda } (g_{42}) ((\text{lambda } (x \ y) \ (+ \ x \ y)) \ 3 \ g_{42}))$

---

---

---

---

---

---

---

---

21

**Théorème de récursion**

*Théorème de récursion de S. Kleen*

**Théorème de récursion :**  
 Soit  $\varphi$  un système de codage des fonctions récursives, et  $f$  une fonction partielle récursive alors il existe un indice  $e$  tel que  

$$\varphi_e(x) = f(e, x)$$

**Démonstration:** Soit la fonction  $g(y, x) = f(s(y, y), x)$ ,  $g$  est calculable donc il existe  $a$  tel que  $\varphi_a(y, x) = g(y, x)$ .

**G**

$y, x$

<G>

- lancer  $f(s(y, y), x)$

**$\varphi_a$**

$y, x$

a

- lancer  $f(s(y, y), x)$

Pour  $y = a$ , on obtient  $\varphi_a(a, x) = f(s(a, a), x)$   
 par ailleurs  $\varphi_a(a, x) = \varphi_{s(a, a)}(x)$   
 En posant  $e = s(a, a)$  on a bien trouvé  $e$  tel que  $\varphi_e(x) = f(e, x)$ .

---

---

---

---

---

---

---

---

22

**Application du théorème de récursion**

*Problème de l'arrêt*

**Problème de l'arrêt** (intuition) : le problème de l'arrêt est le problème de décision qui détermine **pour** un programme **et** une entrée donnée, **si** ce programme s'arrête sur cette entrée ou non.

**Programme halt() :**  
 Données: prog, m;  
 début  
 si prog(m) s'arrête alors accepter;  
 sinon refuser  
 fin

**procedure diagonale(x) :**  
 si halt accepte (x, x)  
     boucle infinie  
 sinon  
     accepter

Que se passe-t-il pour l'appel diagonal(diagonal) ?

---

---

---

---

---

---

---

---

23

**Application du théorème de récursion**

*Problème de programme le plus court*

**Problème du programme minimal** (intuition) : le problème du programme minimal consiste à décider s'il existe ou non un programme équivalent avec un code source plus court.

**Programme énumérateur() :**  
 Données: -  
 début  
 Énumérer les programmes les plus courts  
 fin

**Programme C() :**  
 Données: -  
 début  
 Récupérer son propre code <w>,  
 Lancer énumérateur pour trouver D tq  
     |code D| > |w|  
 Simuler D (avec w le cas échéant)  
 fin

Que peut-on dire du comportement de C ?

---

---

---

---

---

---

---

---

24



### Théorème du point fixe

*Théorème du point fixe de H. Rogers appelé aussi second théorème de récursion de Kleen.*

**Théorème du point fixe :**  
 Soit  $\varphi$  un système de codage des fonctions récursives, et  $f$  une fonction récursive totale alors il existe un indice  $e$  tel que

$$\varphi_e(x) = \varphi_{f(e)}(x)$$

<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;"> <p style="text-align: center; margin: 0;"><b>M</b></p> <p style="text-align: center; margin: 0;">x,y</p> </div> <p style="margin: 0;">m</p> <ul style="list-style-type: none"> <li>- récupération du code m</li> <li>- calcul de s(m,x)</li> <li>- calcul de a=f(s(m,x))</li> <li>- lancement de U(a,y)</li> </ul>	<p><math>\varphi_m(x,y) = \varphi_{f(s(m,x))}(y)</math> (par définition de la machine M créée.)</p> <p><math>\varphi_m(x,y) = \varphi_{s(m,x)}(y)</math> (par le théorème d'itération)</p> <p>En posant <math>e = s(m,x)</math>, on a <math>\varphi_e() = \varphi_{f(e)}()</math></p>
---	---

25

---

---

---

---

---

---

---

---

---

---

### Programmes QUINE

*Programme auto reproducteur*

Définition : Un **quine** est un programme informatique qui imprime son propre code source.

✓ *En langage naturel*  
 Recopier et recopier entre guillemets la phrase  
 « Recopier et recopier entre guillemet la phrase »

26

---

---

---

---

---

---

---

---

---

---

### Programmes QUINE

*Programme auto reproducteur*

**Propriété :** Il existe une machine de Turing **M** qui écrit son propre code, autrement dit qui produit en sortie sur son ruban le mot **<M>**.

*Démonstration*

	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> <p style="margin: 0; text-align: center;"><b>M<sub>x</sub></b></p> <p style="margin: 0; text-align: center;">y</p> </div>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> <p style="margin: 0; text-align: center;"><b>f</b></p> <p style="margin: 0; text-align: center;">x</p> </div>
<M>	- affiche x	- Ecrire le code du programme qui écrit x - retourner ce code <Mx>

La fonction  $f : x \rightarrow \langle M_x \rangle$  est calculable.

Il existe  $e$  tel que  $\varphi_e() = \varphi_{f(e)}()$  (par le théorème du point fixe)  
 $\varphi_e() = \varphi_{f(e)}() = \varphi_{\langle M_e \rangle}() = e$ , on obtient  $\varphi_e() = e$

27

---

---

---

---

---

---

---

---

---

---

### Programmes QUINE

☐ Programme auto reproducteur

**Propriété :** Il existe une machine de Turing  $M$  qui écrit son propre code, autrement dit qui produit en sortie sur son ruban le mot  $\langle M \rangle$ .

*Démonstration alternative*

- Soit la machine  $Print_{\langle X \rangle}()$  qui imprime  $\langle X \rangle$  quelque soit son entrée  $X$ .
- Soit la machine  $DescPrint(\langle M \rangle)$  qui imprime le code de  $Print_{\langle M \rangle}()$

$\langle M \rangle$  → lancer DescPrint( $\langle M \rangle$ )  
concaténer le résultat avec  $\langle M \rangle$

→  $\langle Print_{\langle M \rangle}() \langle M \rangle \rangle$

ma QUINE

lancer Print\_{\langle M \rangle}()  
lancer B(\langle B \rangle)

→  $\langle Print_{\langle B \rangle}() \langle B \rangle \rangle$

28

---

---

---

---

---

---

---

---

### Ensembles décidables

☐ Des fonctions récursives totales aux ensembles décidables

**Notation (Fonction caractéristique)**  
Etant donné un ensemble  $A$  inclus dans  $\mathbb{N}^k$ , nous notons  $\chi_A$  sa fonction caractéristique à valeur dans  $\{0, 1\}$  telle que :  
pour tout  $\vec{x}$  dans  $\mathbb{N}^k$ ,  $\chi_A(\vec{x}) = 1$  si et seulement si  $\vec{x} \in A$ .

**Définition (Ensemble décidable)**  
Un ensemble  $A$  inclus dans  $\mathbb{N}^k$  est **décidable** si  $\chi_A$  est  $\mu$ -récursive et totale.

29

---

---

---

---

---

---

---

---

### Ensemble sémantique

**Définition (Ensemble sémantique)**  
Un ensemble de codes  $A$  inclus dans  $\mathbb{G}$  est appelé **ensemble sémantique** si pour tout  $x, y$  de  $\mathbb{G}$ , tel que  $x \in A$  et  $\varphi_x \simeq \varphi_y$ , nous avons  $y \in A$ .

**Propriétés syntaxiques**  
Le code source contient une instruction conditionnelle si...alors »  
Le code source contient moins de 10 fonctions

**Propriétés sémantiques**  
« le programme ne rend jamais de valeurs nulles »  
« le programme ne termine jamais par une erreur d'exécution »  
« le programme calcule le même résultat qu'un programme donné »  
« le programme calcule un résultat correct par rapport à sa spécification »  
« le programme contient un virus »  
« le programme accède à un site Web »

Wikipédia

30

---

---

---

---

---

---

---

---

### Théorème de Rice

□ De la décidabilité de propriétés propres à des ensembles sémantiques.

**Théorème (Théorème de Rice)**  
 Soit  $A$  inclus dans  $\mathbb{G}$  un ensemble sémantique.  
 Alors  $A$  est décidable **si et seulement si**  $A = \emptyset$  ou  $A = \mathbb{G}$ .

**Corollaire**  
**Problème de l'arrêt :**  
 Soit  $\vec{x}$  dans  $\mathbb{N}^k$ , soit  $H = \{n \in \mathbb{G} \mid \varphi_n(\vec{x}) \downarrow\}$ , alors  $H$  est indécidable.

**Problème de la totalité :**  
 Soit  $T = \{n \in \mathbb{G} \mid \varphi_n(\vec{x}) \downarrow \forall \vec{x} \in \mathbb{N}^k\}$ , alors  $T$  est indécidable.

**Problème de l'équivalence:**  
 Soit  $m$  dans  $\mathbb{N}$ . Soit  $E = \{n \in \mathbb{G} \mid \varphi_n \simeq \varphi_m\}$ , alors  $E$  est indécidable.

31

---

---

---

---

---

---

---

---

### Méthodes de contournement

En pratique, on adopte des stratégies pour contourner les limitations du théorème de Rice

1. Vérification de programmes avec seulement un nombre fini d'états.
2. Méthode de vérification partiellement automatisée.
3. Méthodes à base d'approximations.
4. Méthodes restreignant l'ensemble des programmes sur lesquels on veut démontrer la propriété.
5. Restrictions raisonnables de l'ensemble des fonctions calculables<sup>4</sup>.

Wikipédia

32

---

---

---

---

---

---

---

---

### Art de la référence

□ Boucle étrange : ces murs qui nous parlent

Photo: <https://eriffillgallery.co.uk/usual-suspects-by-artist-catman-at-eriffill-gallery-london/>

33

---

---

---

---

---

---

---

---

Pour résumer

☐ *Quelques points essentiels de l'exposé*

- ✓ **Numérotation de Gödel** : à toute fonction REC peut être associé un code dans N.
- ✓ **Théorèmes de Kleene** d'itération, de récursion et du point fixe
- ✓ **Fonction caractéristique et ensemble sémantique;**
- ✓ **Théorème de Rice** et corollaires problème de l'arrêt, de la totalité et de l'équivalence.

---

---

---

---

---

---

---

---

34

Bibliographie notoire

☐ *Ouvrages et liens de références exploités pour la réalisation de l'exposé.*

- ✓ **Site wikipedia** pour les théorème d'itération, de récursion et du point fixe.
- ✓ **Une approche de la conscience** vue comme une extension du mécanisme de l'attention volontaire par F. Anceau
- ✓ **Je suis une boucle étrange** – Dunod Edition 2008 par D. Hofstadter
- ✓ **La notion de récursivité**, de la première cybernétique au connexionnisme par P. Livet

---

---

---

---

---

---

---

---

35