



Fonctions,  
Computations,  
Algorithmes : Niveaux  
d'abstraction et  
niveaux d'explication

---

PHI 6385 A23 SÉANCE 2  
JONATHAN SIMON

---

# Apercu

- 1) Niveaux d'abstraction et niveaux d'explication
- 2) Behaviorisme, physicalisme, fonctionnalisme (un premier passage)
- 3) Fonctionnalisme vs computationnalisme
- 4) Calcul analogue vs calcul numérique (analog and digital computation)
- 5) Calcul numérique / digital computation: (machines de Turing)
- 6) Qu'est-ce que c'est que de mettre en œuvre un calcul?



Niveaux d'explication

---

# Marr

- 3 Niveaux :
- Le niveau computationnel
- Le niveau algorithmique
- Le niveau d'implémentation

---

# Marr

- Le niveau computationnel
- Le niveau général, en précisant le problème à résoudre :  
p. ex.: «faire un gâteau»

---

# Marr

- Le niveau algorithmique
- Le niveau spécifiant les éléments cruciaux du plan pour résoudre le problème, par exemple : «La recette de ma grand-mère pour faire un gâteau». (Mélange une poignée de farine, une cuillère de sucre...).

---

# Marr

- Le niveau d'implémentation
- Le niveau spécifiant tous les détails de la façon dont on suit exactement la recette. Quelle quantité de farine exactement, et quelle sorte (raffinée ou non) ? Quelle quantité de sucre exactement, et quelle sorte (brun, blanc, canne) ? Quel type de bol à mélanger, en bois ou en métal ? Etc...

---

# Marr

Note que cette distinction en 3 niveaux est arbitraire : en réalité, nous pouvons parler de degrés de spécification, de la quantité de zoom que nous faisons....

(malgré tout, la règle des 3 est convaincante, par exemple dans le domaine de la gestion, la même distinction apparaît : stratégique vs tactique vs opérationnel).

---

# Niveau d'explication

Les théories de la conscience (et les théories de la cognition) peuvent être en désaccord sur le niveau d'explication approprié :

...Une chose est-elle consciente en vertu

1) du problème qu'elle résout (peu importe comment), ou

2) en vertu de la stratégie générale qu'elle utilise pour résoudre un problème, ou

3) en raison de détails très spécifiques de la façon dont elle met en œuvre cette stratégie ?

---

## Niveau d'explication

- 1) du problème qu'elle résout (peu importe comment), ou
- 2) en vertu de la stratégie générale qu'elle utilise pour résoudre un problème, ou
- 3) en raison de détails très spécifiques de la façon dont elle met en œuvre cette stratégie ?

---

## Niveau d'explication

- 1) du problème qu'elle résout (peu importe comment), (behaviorisme)
- 2) en vertu de la stratégie générale qu'elle utilise pour résoudre un problème, (fonctionnalisme)
- 3) en raison de détails très spécifiques de la façon dont elle met en œuvre cette stratégie (physicalisme)

---

## Niveau d'explication

*Remarque : il s'agit d'une dimension de variation distincte de la question "métaphysique" du dualisme par rapport au matérialisme. Il s'agit de 2 questions indépendantes (plus ou moins) sur la nature de la conscience.*

---

# Niveau d'explication

Compréhension / Discussion :

- 1) Comment peut-on faire la différence entre les niveaux ?
- 2) Pourquoi y a-t-il trois niveaux ?
- 3) Pourquoi supposons-nous que la conscience (ou la cognition) sont en quelque sorte des aspects des stratégies de résolution de problèmes ?

---

2) Behaviorisme, physicalisme,  
fonctionnalisme (computationnel)

---

Comportementalisme / Behaviorisme: les états mentaux sont des états comportementaux

Physicalisme: les états mentaux sont des états neurobiologiques (le matériel du cerveau)

Fonctionnalisme: les états mentaux sont des états fonctionnels (p. ex. computationnels -- le logiciel du cerveau)

---

# Un distinction metaphysique

- états mentaux: mon état mental de ressentir une douleur (un événement ou un processus auquel mon cerveau participe)
- propriétés mentales: ma propriété de ressentir une douleur / d'être dans la douleur (un type / universel / caractéristique qui me caractérise) (ou, la propriété de mon état...)

- 
- « The Type-Token Distinction »: La distinction entre les types et les individus
  - Type Physicalism = les propriétés mentales sont des propriétés physiques
  - Token Physicalism = les choses (objets, états) ayant des propriétés mentales sont des choses ayant des propriétés physiques
  - (Le physicalisme de type est plus fort en tant que théorie réductrice)

---

Comportementalisme: les états/propriétés mentaux sont des états/propriétés comportementaux

Physicalisme: les états/propriétés mentaux sont des états/propriétés neurobiologiques (le matériel du cerveau)

Fonctionnalisme: les états/propriétés mentaux sont des états/propriétés computationnels (le logiciel du cerveau)

---

Comportementalisme: un état mental est un ensemble de dispositions/tendances pour se comporter de certaines manières, en fonction de certaines stimuli

- sentir une douleur dans la main, c'est retirer la main si on la touche, dire que ça fait mal si on le demande, etc.

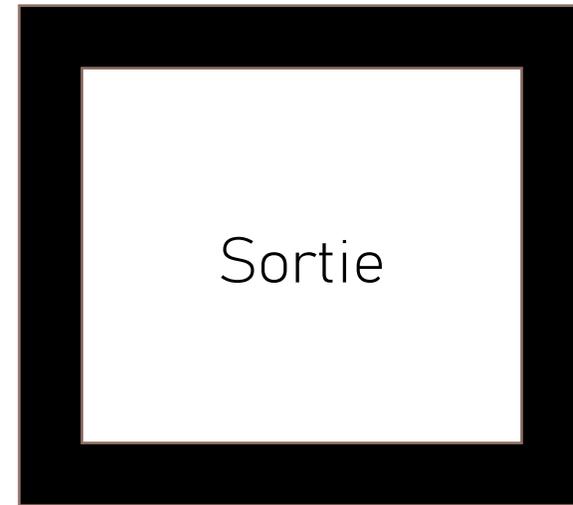
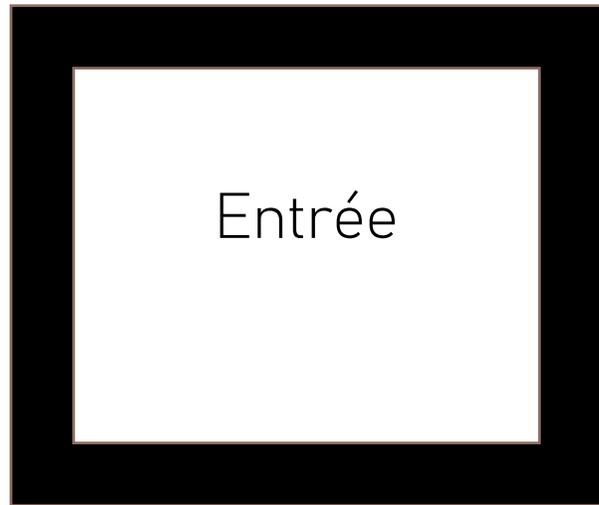


## Comportementalisme:

- des fonctions d'entrée-sortie

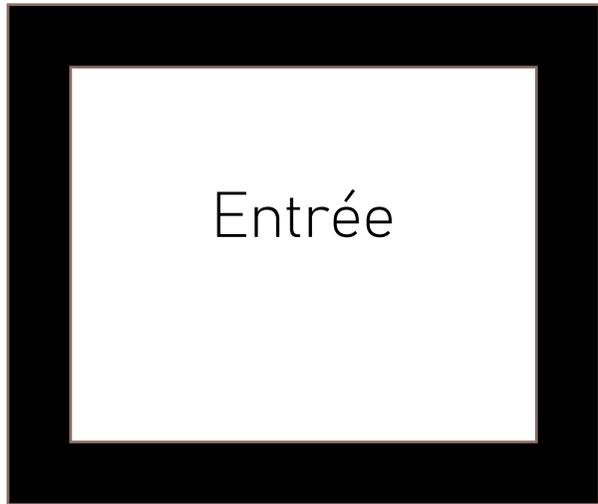
---

# comportementalisme

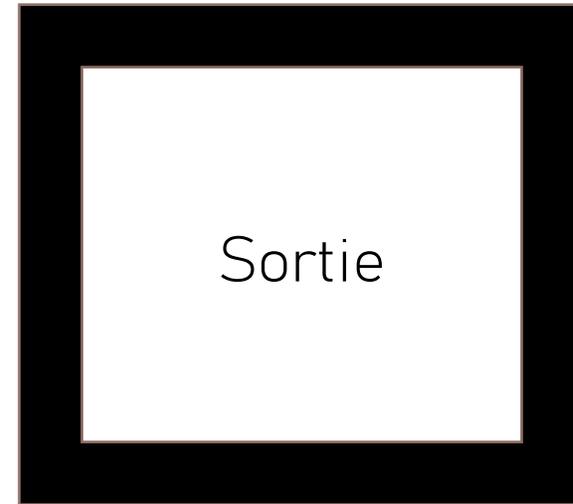


---

# comportementalisme



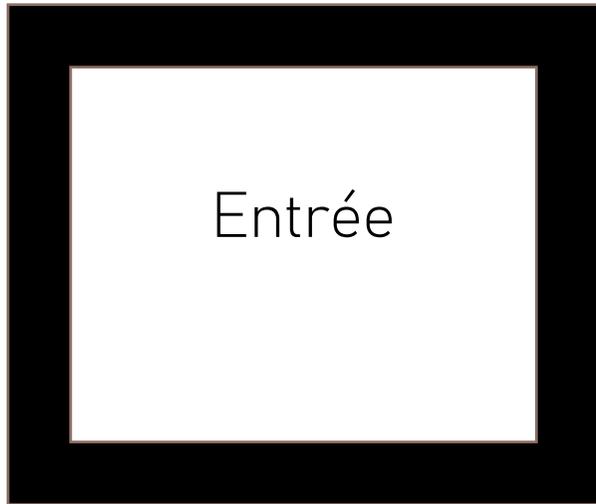
On touche votre main



Vous retirez votre main

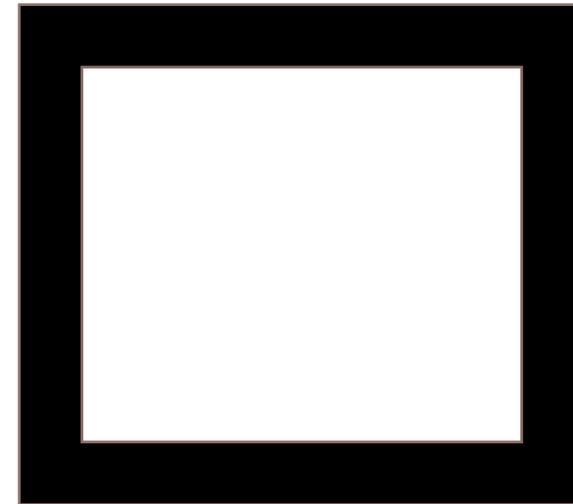
---

# comportementalisme



On touche votre main

« Comment vous sentez-vous ? »



Vous retirez votre main

« Ça fait mal ! »

---

Pour contraster:

Le behaviorisme philosophique : les états internes sont des dispositions à se comporter...

Le behaviorisme méthodologique : les états internes sont impossibles à étudier, on doit plutôt étudier le comportement.

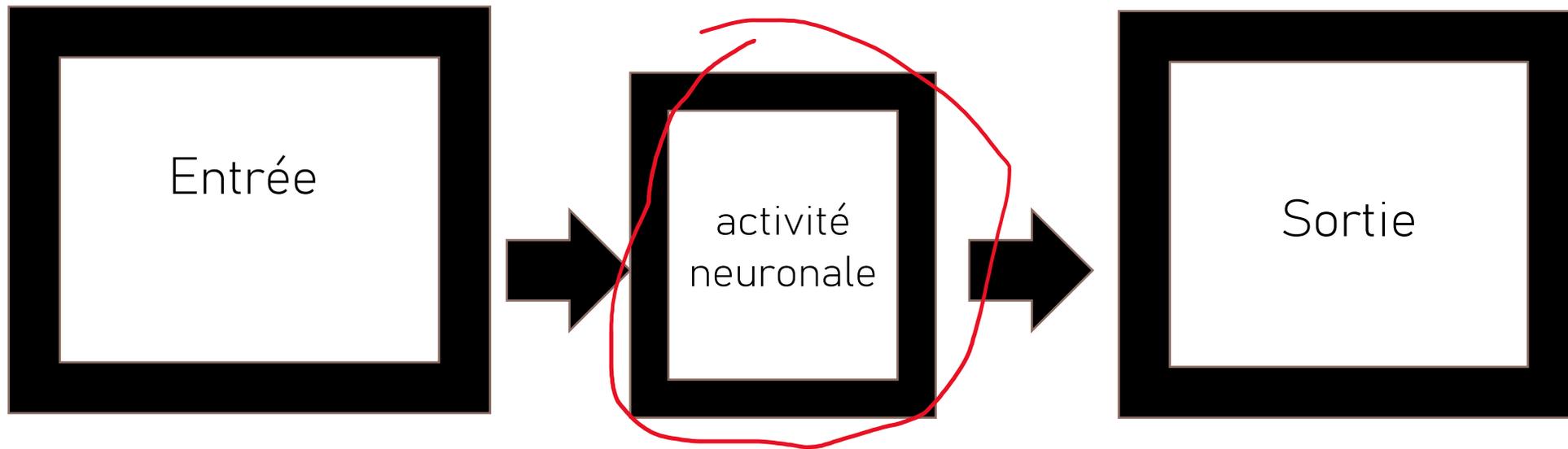
---

Physicalisme: un état mental est un état neurobiologique

- sentir une douleur dans la main est pour un certain de vos assemblages neuronaux de tirer selon un certain modèle (le tir des fibres C)

---

# Physicalisme



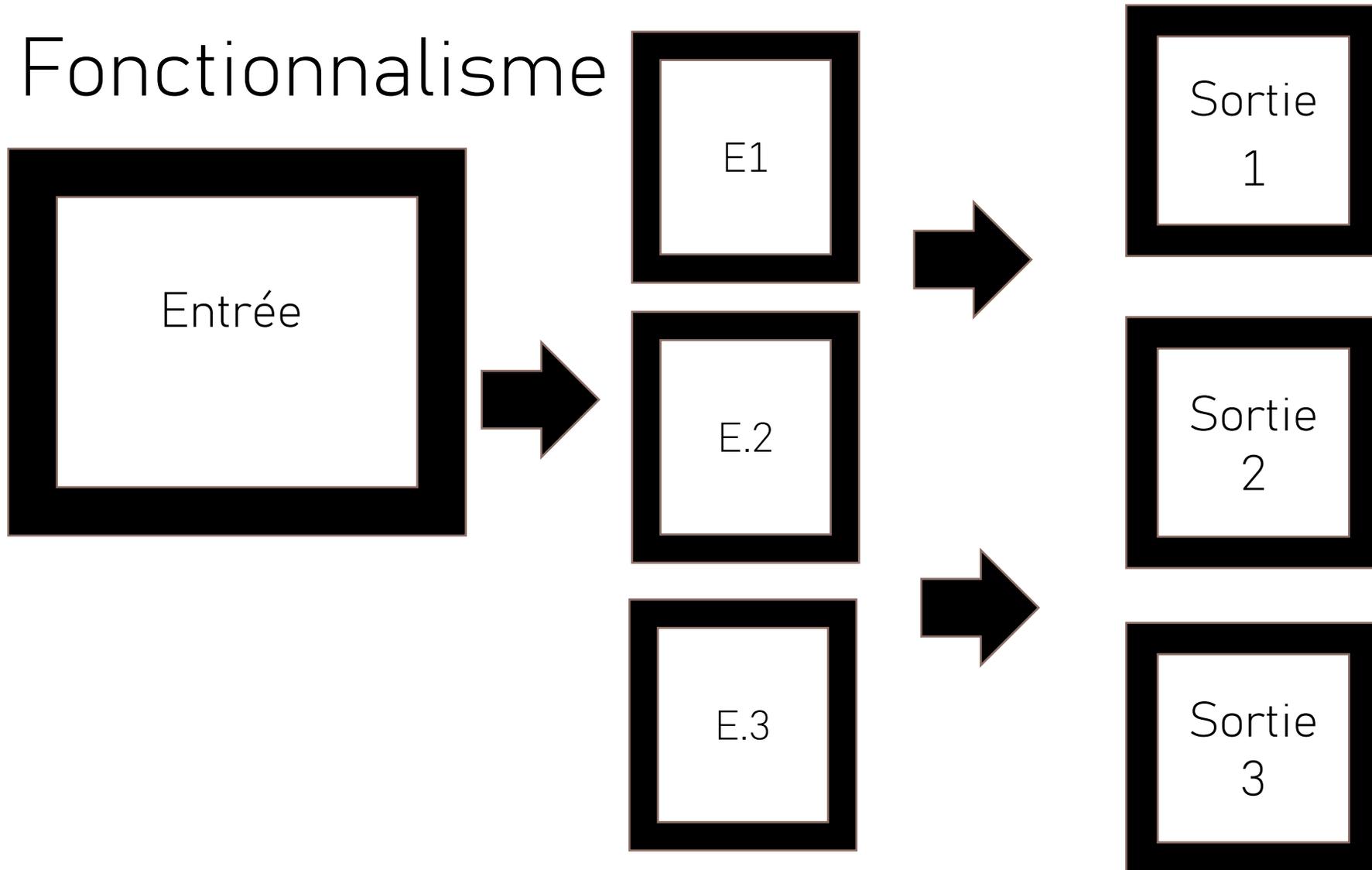
---

Fonctionnalisme: un état mental est un état fonctionnel (p.ex. computationnel)

- sentir la douleur dans votre main est pour vous d'être dans un certain état interne qui, *étant donné les autres états mentaux dans lesquels vous êtes*, a tendance à produire un comportement douloureux avec des stimuli appropriés

---

# Fonctionnalisme



- 
- Pour chaque équivalence théorique, trois degrés de force (logique):
  - 1) Version Type-A/Analytique/Logique/A Priori: Il n'y a pas de fossé épistémique : l'analyse conceptuelle révèle la vérité de l'équivalence
  - 2) Version Type-B/Synthétique/A Posteriori: Il y'a un fossé épistémique : l'équivalence est un vrai équivalence d'identité, mais il faut le découvrir empiriquement
  - 3) Version Type-D,E,F/Corrélatif: l'équivalence n'est pas un équivalence d'identité, c'est juste une correspondance : les deux choses se présentent ensemble, bien qu'elles ne soient pas les mêmes

---

# Le fonctionnalisme

- Block (dans « Troubles with Functionalism») définit le fonctionnalisme ainsi:
- Fonctionnalisme: Chaque type d'état mental est un état qui, étant donné certaines entrées/stimuli sensoriels et certains autres états mentaux, consiste dans une disposition à agir d'une certaine façon et à avoir certains états mentaux

---

# Le fonctionnalisme

- Pour contraster :
- Comportementalisme: Chaque type d'état mental est un état qui, étant donné certains inputs sensoriels et certains états mentaux, consiste dans une disposition à agir d'une certaine façon et à avoir certains états mentaux

---

# Le fonctionnalisme

- Un premier défi pour le fonctionnalisme : la circularité
- Le fonctionnalisme devrait offrir une analyse des états mentaux (au moins la version de type A doit le faire). Mais une analyse ne devrait pas faire appel à ce qu'elle analyse (les états mentaux)

---

# Le fonctionnalisme

- Deux approches à ce défi :
- 1) Le fonctionnalisme machinique / computationnel (Putnam)
- 2) Fonctionnalisme causal-théorique (Lewis)

---

# Le fonctionnalisme machinique

- Le fonctionnalisme machinique comprend les états mentaux comme des états d'un système de calcul / computation (typiquement : une machine de Turing).
- On y revient.

---

# Le fonctionnalisme causal-théorique

- A.k.a. *le fonctionnalisme Ramsey-Lewis*

---

# Le fonctionnalisme Ramsey-lewis

Une formulation alternative du fonctionnalisme qui évite également l'objection de la circularité.

L'idée : vous prenez une théorie du domaine, puis vous quantifiez tous les termes que vous voulez définir.

Cela vous donne la « phrase de Ramsey » de la théorie. Vous l'utilisez pour définir chaque terme de manière non circulaire

---

# Le fonctionnalisme Ramsey-lewis

Exemple:

La théorie de la douleur: La douleur est un état qui est causé par une lésion des tissus et qui conduit à pleurer, à moins que vous ne vouliez être stoïque, auquel cas elle conduit à vous mordre la langue.

---

# Le fonctionnalisme Ramsey-lewis

Exemple:

Énoncé de Ramsey correspondante: Il existe  $P1, P2$  tel que:  $P1$  est un état qui est causé par une lésion des tissus et qui conduit à pleurer, à moins que vous n'est pas dans  $P2$ , auquel cas elle conduit à vous mordre la langue.

---

# Le fonctionnalisme Ramsey-lewis

Ensuite, " de ressentir la douleur" est analysé comme suit : être dans un état qui correspond à la description de P1

« X ressent la douleur » = « Il existe P1,P2 tel que: P1 est un état qui est causé par une lésion des tissus et qui conduit à pleurer, à moins que vous n'est pas dans P2, auquel cas elle conduit à vous mordre la langue et x et dans P1 »

---

# Le fonctionnalisme Ramsey-lewis

Observations :

- 1) Les états internes sont définis implicitement
- 3) Holisme : les états ne sont définis qu'implicitement en référence les uns aux autres. Si vous modifiez l'un des états, vous modifiez chacun d'eux

# Compréhension / discussion

- 1) Les physicalistes sont-ils également confrontés à un problème de holisme / d'interdépendance ?
- 2) Le fonctionnalisme et le "type A" sont-ils mieux assortis ? Le physicalisme et le "type B" correspondent-ils mieux ?
- 3) Le fonctionnalisme de Ramsey-Lewis est-il plus général que le fonctionnalisme de la machine ? Devrions-nous vouloir cette généralité supplémentaire ? Est-ce que cela a de l'importance (puisque les deux ne font que spécifier des *familles de théories* de toute façon) ?

---

### 3) Fonctionnalisme vs computationalisme

---

# Fonctionnalisme vs computationnalisme

- En ce qui nous concerne, il est important de savoir si les ordinateurs numériques peuvent être conscients.
- Ici, il faut être prudent : il ne suffit pas de motiver **le fonctionnalisme**, il faut aussi motiver **le computationnalisme.....**

---

# Fonctionnalisme vs computationnalisme

- Fonctionnalisme (à propos de la conscience) : pour qu'un système soit conscient, il suffit qu'il ait une certaine organisation fonctionnelle.
- (Cela signifie, comme indiqué ci-dessus, une organisation régie par des mécanismes, de sorte que l'on peut dire si le système a mal fonctionné ou s'il a fonctionné correctement.)

---

# Fonctionnalisme vs computationnalisme

- Beaucoup de choses peuvent avoir une organisation fonctionnelle sans être des ordinateurs : un réfrigérateur, ton cœur...
- En principe, la définition de Ramsey-Lewis englobe cette notion très générale

---

# Fonctionnalisme vs computationnalisme

- *Le calcul (computation)* est un type spécifique de fonction
- -- quel type, exactement?

---

# Fonctionnalisme vs computationnalisme

- Piccinnini:
- Le calcul (computation) au sens générique est **le traitement de véhicules** (définis comme des entités ou des variables qui peuvent changer d'état) **conformément à des règles** qui sont sensibles à certaines propriétés des véhicules et, en particulier, aux différences entre les différentes parties (c'est-à-dire les parties spatio-temporelles) des véhicules. Le traitement est effectué par un mécanisme fonctionnellement organisé, c'est-à-dire un mécanisme dont les composants sont fonctionnellement organisés pour traiter leurs véhicules conformément aux règles pertinentes. Ainsi, si le mécanisme fonctionne mal, un calcul erroné se produit.

---

# Fonctionnalisme vs computationnalisme

- Piccinnini:
- Comme les calculs concrets et leurs véhicules sont définis indépendamment des supports physiques qui les mettent en œuvre, nous les appelons «indépendants du support» (medium independent)

---

# Fonctionnalisme vs computationnalisme

- Piccinnini:
- En d'autres termes, les véhicules sont indépendants du support si et seulement si la règle (c'est-à-dire la carte entrée-sortie) qui définit un calcul n'est sensible qu'aux différences entre les parties des véhicules selon des dimensions de variation spécifiques - elle est insensible à toute autre propriété physique plus concrète des véhicules.

---

# Computationalisme

- Argument #1 de Piccinnini en faveur d'un computationnalisme générique :
- 1) Les processus neuronaux qui expliquent la cognition sont des manipulations de véhicules indépendants du support.
- 2) Les manipulations de véhicules moyennement indépendants sont des calculs au sens générique (conformément à la section 2.1).
- 3) Par conséquent, les processus neuronaux qui expliquent la cognition sont des calculs au sens générique.

---

# Computationalisme

1) Les processus neuronaux qui expliquent la cognition sont des manipulations de véhicules indépendants du support.

argument pour la prémisse 1 :

Les aspects fonctionnels des trains de pointes, tels que les taux de pointes et la synchronisation des pointes, sont similaires dans tout le système nerveux, quelles que soient les propriétés physiques des stimuli (auditifs, visuels et somatosensoriels) et peuvent être mis en œuvre soit par le tissu neural, soit par un autre support physique, tel qu'un circuit à base de silicium (cf. Craver, 2010). Ainsi, les trains de pointes – des séquences de pointes telles que celles produites par les neurones en temps réel – semblent être des véhicules indépendants du support, se qualifiant ainsi comme des véhicules appropriés pour le calcul générique. Des considérations analogues s'appliquent à d'autres véhicules manipulés par les neurones, tels que les changements de tension dans les dendrites, les neurotransmetteurs et les hormones. En supposant que les cerveaux traitent des véhicules indépendants du milieu, il s'ensuit que les cerveaux effectuent des calculs au sens générique du terme.

---

# Computationalisme

A quoi ressemblerait un bon défi à cette prémisse, et comment Piccinini pourrait-il y répondre ?

--- Tu pourrais peut-être trouver une caractéristique de la synchronisation des pointes, par exemple, et affirmer qu'elle n'est pas vraiment indépendante du support.

---Tu pourrais alors t'attendre à ce que Piccinini réponde à ton argument en essayant de montrer que l'aspect en question n'a rien à voir avec la cognition ou qu'il est effectivement indépendant du support.

## Compréhension / discussion

- 1) Est-ce que tous les véhicules au sens ci-dessus doivent être des véhicules de représentation, c'est-à-dire semblables à des mots ou à des lettres ?
- 2) Existe-t-il des processus qui sont considérés comme informatiques dans ce sens mais qui ne pourraient pas être mis en œuvre par un ordinateur numérique ?

---

4) Calcul analogue vs calcul  
numérique (analog and digital  
computation)

---

# Numérique

- Le calcul numérique abstrait est la manipulation de chaînes d'éléments discrets, c'est-à-dire de chaînes de lettres d'un alphabet fini, ou de *chiffres* (digits)
- Le calcul numérique concret est le traitement de séquences de chiffres selon des règles générales définies sur les chiffres.

---

# Numérique

- un chiffre est un état macroscopique (d'un composant du système) dont le type peut être distingué de manière fiable et non ambiguë par le système des autres types macroscopiques.
- Les chiffres peuvent être concaténés (c'est-à-dire ordonnés) pour former des séquences ou des chaînes, qui servent de véhicules aux calculs numériques. Un calcul numérique consiste à traiter des chaînes de chiffres conformément à une règle, qui est simplement une carte allant des chaînes d'entrée, plus éventuellement des états internes, aux chaînes de sortie. L'addition, la multiplication, l'identité et le tri sont des exemples de règles qui peuvent figurer dans un calcul numérique.

---

# Numérique

- un système physique est un système informatique numérique si et seulement si c'est un système fonctionnellement organisé pour manipuler des chaînes de chiffres en entrée, en fonction du type des chiffres et de leur emplacement sur la chaîne, conformément à une règle définie sur les chaînes (et éventuellement certains états internes du système).

---

# Analogue

- Les ordinateurs analogiques abstraits sont des systèmes dont la fonction est de manipuler des variables continues – variables qui peuvent varier continuellement dans le temps et prendre n'importe quelle valeur réelle dans certains intervalles – spécifiées par des équations différentielles, afin d'instancier des relations fonctionnelles appropriées entre les variables.

---

# Analogue

- L'une des principales fonctions des ordinateurs analogiques est de résoudre certains systèmes d'équations différentielles en instanciant les relations entre les variables continues spécifiées dans les équations.

---

# Analogue

- Les variables continues physiquement implémentées sont des véhicules très différents des chaînes de chiffres. Alors qu'un système informatique numérique peut toujours distinguer sans ambiguïté les chiffres les uns des autres, un système informatique analogique physiquement implémenté ne peut pas faire de même avec les valeurs exactes des variables continues, car ces variables ne peuvent être mesurées qu'avec une marge d'erreur. C'est principalement pour cette raison que les calculs analogiques (au sens strict du terme) sont un autre type de processus que les calculs numériques. Néanmoins, comme les ordinateurs numériques, les ordinateurs analogiques fonctionnent sur des véhicules indépendants du milieu.

---

# Analogue vs numérique

- Les ordinateurs analogiques peuvent-ils faire des choses que les ordinateurs numériques ne peuvent pas faire ?
- La conscience pourrait-elle être une question de calcul analogique mais pas de calcul numérique ?
- Si les IA peuvent un jour être conscientes, est-ce plus probable parce que le computationalisme numérique est vrai, ou parce que le computationalisme générique est vrai ?



5) Turing

---

# Le fonctionnalisme machinique

- Le fonctionnalisme machinique comprend les états mentaux comme des états d'un système de calcul numérique (typiquement : une machine de Turing: De nombreuses définitions de la computation s'avèrent être équivalentes aux machines de Turing.).



## State 2

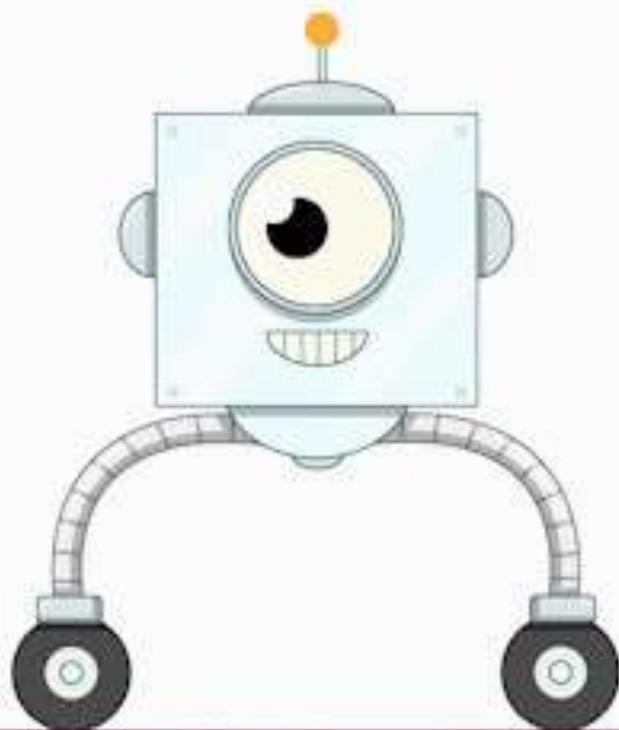
if 0

then erase

write 1

move right

go to state 5



---

# Le fonctionnalisme machinique

- Une machine de Turing : une longueur de bande / papier (aussi longue que nécessaire) divisée en cellules.
- Chaque cellule porte un numéro écrit dessus.
- Un lecteur/enregistreur fonctionne sur une cellule à la fois.
- Il lit le numéro, puis le modifie selon son *état*, et selon un ensemble de règles, puis se déplace vers une autre cellule (et modifie son état) selon un ensemble de règles.

---

# Le fonctionnalisme machinique

- Il est essentiel que les règles indiquent au scanner ce qu'il doit écrire étant donné ce qu'il lit *et dans quel état il se trouve*. Les règles disent aussi *dans quel état il doit aller*.

---

# Le fonctionnalisme machinique

La liste des règles est appelée *une table de machine*. Voici un exemple de table de machine pour une machine à coke (prochaine page). Notez que la «signification» des états est définie implicitement (les étiquettes ne sont ici qu'à des fins heuristiques)

# Example: Machine à Coke (Coke coute 10c)

TABLE 1

	$S_1$	$S_2$
Input : pièce de 5 cents	N'émettre aucun output Passer en $S_2$	Emettre un coca-cola Passer en $S_1$
Input : pièce de 10 cents	Emettre un coca-cola Rester en $S_1$	Emettre un coca-cola et une pièce de 5 cents Passer en $S_1$

---

## Example: Machine à Coke (coke coute 15c)

Input	État	Prochain État	Output
5c	Désire 15c	Désire 10c	rien
5c	Désire 10c	Désire 5c	rien
5c	Désire 5c	Désire 15c	Coke!
10c	Désire 15c	Désire 5c	rien
10c	Désire 10c	Désire 15c	Coke!
10d	Désire 5c	Désire 10c	Coke!

---

## Example: Machine à Coke (coke coute 15c)

Input	État	Prochain État	Output
5c	$S_0$	$S_1$	rien
5c	$S_1$	$S_2$	rien
5c	$S_2$	$S_0$	Coke!
10c	$S_0$	$S_2$	rien
10c	$S_1$	$S_0$	Coke!
10d	$S_2$	$S_1$	Coke!

---

# Le fonctionnalisme machinique

Observations :

1) Seules les entrées et les sorties sont définies explicitement, les états internes sont définis implicitement

2) Holisme : les états ne sont définis qu'implicitement en référence les uns aux autres. Si vous modifiez l'un des états, vous modifiez chacun d'eux

---

# Le Test de Turing

- 1) Si un système numérique peut passer le test du jeu d'imitation suffisamment souvent, il est intelligent
- 2) GPT4 réussit assez souvent le test du jeu d'imitation
- 3) il est donc intelligent

---

# Compréhension / consolidation

- Est-il logique d'utiliser un modèle comme celui-ci pour tester la conscience plutôt que l'intelligence ?
- Ce test soutient-il vraiment le computationnisme, même s'il est en fin de compte comportemental ?
- Comment faire la différence entre un ordinateur qui fonctionne mal et quelque chose qui n'est pas du tout un ordinateur ?

---

Qu'est-ce que c'est que de mettre  
en œuvre un calcul?

---

Chalmers, Maudlin, Klein, Bostrom:

Bonnes et mauvaises abstractions

---

# Bonnes et mauvaises abstractions

- Si les machines de Turing (en tant que descriptions abstraites ou mathématiques) spécifient des algorithmes, comment dire qu'un système donné «est» (ou, implémente / met en oeuvre) une machine de Turing ?

---

# Bonnes et mauvaises abstractions

- Premier problème : les descriptions des machines de Turing sont trop abstraites - même un rocher pourrait être «*interprété*» comme une machine de Turing
- Deuxième problème : les descriptions des machines de Turing sont trop exigeantes : même les ordinateurs ont des problèmes. Cela signifie-t-il que ce ne sont pas vraiment des ordinateurs ?

---

# Bonnes et mauvaises abstractions

- Problème: les descriptions des machines de Turing sont trop abstraites - même un rocher pourrait être «interprété» comme une machine de Turing

---

## Example: Machine à Coke (coke coute 15c)

Input	État	Prochain État	Output
5c	Désire 15c	Désire 10c	rien
5c	Désire 10c	Désire 5c	rien
5c	Désire 5c	Désire 15c	Coke!
10c	Désire 15c	Désire 5c	rien
10c	Désire 10c	Désire 15c	Coke!
10d	Désire 5c	Désire 10c	Coke!

---

## Example: Machine à Coke (coke coute 15c)

Input	État	Prochain État	Output
5c	$S_0$	$S_1$	rien
5c	$S_1$	$S_2$	rien
5c	$S_2$	$S_0$	Coke!
10c	$S_0$	$S_2$	rien
10c	$S_1$	$S_0$	Coke!
10d	$S_2$	$S_1$	Coke!

---

## Example: Machine à Coke (coke coute 15c)

Input	État	Prochain État	Output
$l_1$	$S_0$	$S_1$	$O_1$
$l_1$	$S_1$	$S_2$	$O_1$
$l_1$	$S_2$	$S_0$	$O_2$
$l_2$	$S_0$	$S_2$	$O_1$
$l_2$	$S_1$	$S_0$	$O_2$
$l_2$	$S_2$	$S_1$	$O_2$

---

# Bonnes et mauvaises abstractions

- Problème: les descriptions des machines de Turing sont trop abstraites - même un rocher pourrait être «interprété» comme une machine de Turing

---

# Bonnes et mauvaises abstractions

- Problème:

Étiquette l'environnement initial de la roche comme I1, sa configuration initiale S1, sa configuration un moment plus tard S2, l'environnement comme O1 : alors il correspond à une ligne sur ce tableau !

---

# Bonnes et mauvaises abstractions

- Résolution:
- Nous avons besoin d'une sorte de contrainte contrefactuelle, ou d'isomorphisme entre table de machine et mécanisme, c'est-à-dire que si le système avait été dans un état initial différent ou avait reçu une entrée différente, il aurait quand même satisfait à la description (de table de machine) en question...

---

# Bonnes et mauvaises abstractions

- problèmes résiduels :
- 1) Est-ce trop fort ? Nous devons permettre que, par exemple, les ordinateurs puissent avoir des problèmes (tout en restant des ordinateurs). Quel degré de défaillance pouvons-nous autoriser ? À quel moment un système passe-t-il du statut d'ordinateur très glitchy ou peu fiable, à celui de ne plus être un ordinateur (Bostrom).

---

# Bonnes et mauvaises abstractions

- problèmes résiduels :
- 2) Cause par omission : le fait de se fier à un critère contrefactuel peut impliquer que cela fait une différence, pour quelque chose qui ne se produit pas, si cela aurait pu se produire (l'alto silencieux ajoute à l'esthétique de la pièce, Klein).

---

# Bonnes et mauvaises abstractions

- problèmes résiduels :
- 3) Ce n'est pas assez fort ? Nous pouvons décrire des systèmes qui respectent la condition contrefactuelle, mais qui intuitivement ne semblent toujours pas mettre en œuvre l'algorithme (Maudlin, Klein).

---

# Bonnes et mauvaises abstractions

- problèmes résiduels :
- 3) Ce n'est pas assez fort ?
- Simon vs Theodore vs Alvin:

---

# Bonnes et mauvaises abstractions

- Simon est une machine de Turing qui, étant donné un nombre, produit ses facteurs. Pour le nombre 20, elle produit (1,2,4,5). Étant donné le nombre 21, elle produit (1,3,7)...

---

# Bonnes et mauvaises abstractions

- Théodore ne peut résoudre qu'un seul problème (par exemple, trouver les facteurs de 20 : 1,2,4,5). Quel que soit le nombre que tu lui donnes, il exécute la recette pour factoriser 20 et obtient (1,2,4,5).

---

# Bonnes et mauvaises abstractions

- Alvin est composé de Théodore et de Simon : par défaut, il exécute le programme le plus simple de Théodore. Mais il possède un interrupteur qui vérifie si l'entrée est 20 ou non. Si l'entrée n'est pas 20, l'interrupteur désactive le mécanisme de Théodore et active celui de Simon.
- Alvin est contrefactuellement sensible.... mais si on lui donne l'entrée 20, il fait la même chose que Simon.