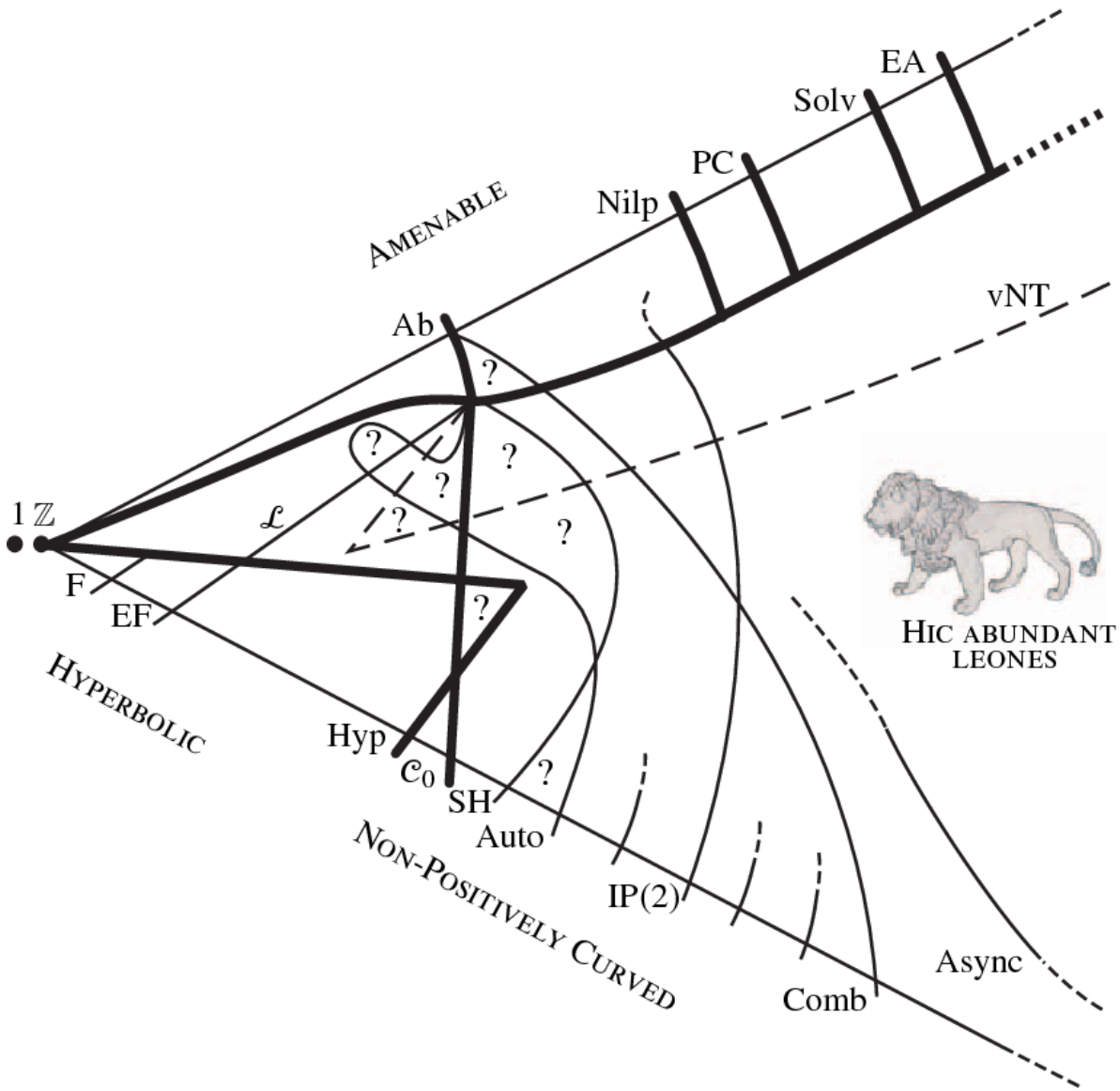
Certains groupes n'aiment pas les algorithmes !

Anthony Genevois, Université de Montpellier (CNRS)

21 février 2023

Carte des groupes (M. Bridson, ICM 2006)



Question principale

Question vague

Un problème de théorie des groupes peut-il toujours être résolu par un algorithme ?

Question principale

Question vague

Un problème de théorie des groupes peut-il toujours être résolu par un algorithme ?

Théorème (J. Belk et C. Bleak, 2017)

Le problème de l'ordre dans $2V$ ne peut pas être résolu par un algorithme.

Problème de l'ordre

Problème de l'ordre

Soit G un groupe. Existe-t-il un algorithme qui, pour chaque élément $g \in G$, calcul son ordre (i.e. le plus petit $n \in \mathbb{N}^*$ tel que $g^n = 1$) ?

Problème de l'ordre

Problème de l'ordre

Soit G un groupe. Existe-t-il un algorithme qui, pour chaque élément $g \in G$, calcul son ordre (i.e. le plus petit $n \in \mathbb{N}^*$ tel que $g^n = 1$) ?

Groupes finis

- L'ordre de $p \in \mathbb{Z}/n\mathbb{Z}$ est $n/\mathrm{pgcd}(p, n)$.

Problème de l'ordre

Problème de l'ordre

Soit G un groupe. Existe-t-il un algorithme qui, pour chaque élément $g \in G$, calcul son ordre (i.e. le plus petit $n \in \mathbb{N}^*$ tel que $g^n = 1$) ?

Groupes finis

- L'ordre de $p \in \mathbb{Z}/n\mathbb{Z}$ est $n/\mathrm{pgcd}(p, n)$.
- Soit $\sigma \in \mathfrak{S}_n$. Décomposons σ comme un produit de cycles à supports disjoints $\kappa_1 \cdots \kappa_r$. L'ordre de σ est le ppcm des longueurs des κ_i.

Problème de l'ordre

Problème de l'ordre

Soit G un groupe. Existe-t-il un algorithme qui, pour chaque élément $g \in G$, calcul son ordre (i.e. le plus petit $n \in \mathbb{N}^*$ tel que $g^n = 1$) ?

Groupes finis

- L'ordre de $p \in \mathbb{Z}/n\mathbb{Z}$ est $n/\mathrm{pgcd}(p, n)$.
- Soit $\sigma \in \mathfrak{S}_n$. Décomposons σ comme un produit de cycles à supports disjoints $\kappa_1 \cdots \kappa_r$. L'ordre de σ est le ppcm des longueurs des κ_i.
- Soit G un groupe fini. D'après le théorème de Lagrange, l'ordre d'un élément g est fini et divise $|G|$. Il suffit de calculer les puissances de g jusqu'à $|G|$ pour trouver son ordre.

Problème de l'ordre

Quelques groupes infinis

- L'ordre de $(a, b) \in \mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ vaut ∞ si $a \neq 0$, 2 si $a = 0$ et $b \neq 0$, et 1 sinon.

Problème de l'ordre

Quelques groupes infinis

- L'ordre de $(a, b) \in \mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ vaut ∞ si $a \neq 0$, 2 si $a = 0$ et $b \neq 0$, et 1 sinon.
- L'ordre de $r \in \mathbb{R}/\mathbb{Z}$ est ∞ si r est irrationnel et, si $r = p/q$ irréductible, $\mathrm{ord}(r) = q$.

Problème de l'ordre

Quelques groupes infinis

- L'ordre de $(a, b) \in \mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ vaut ∞ si $a \neq 0$, 2 si $a = 0$ et $b \neq 0$, et 1 sinon.
- L'ordre de $r \in \mathbb{R}/\mathbb{Z}$ est ∞ si r est irrationnel et, si $r = p/q$ irréductible, $\mathrm{ord}(r) = q$.

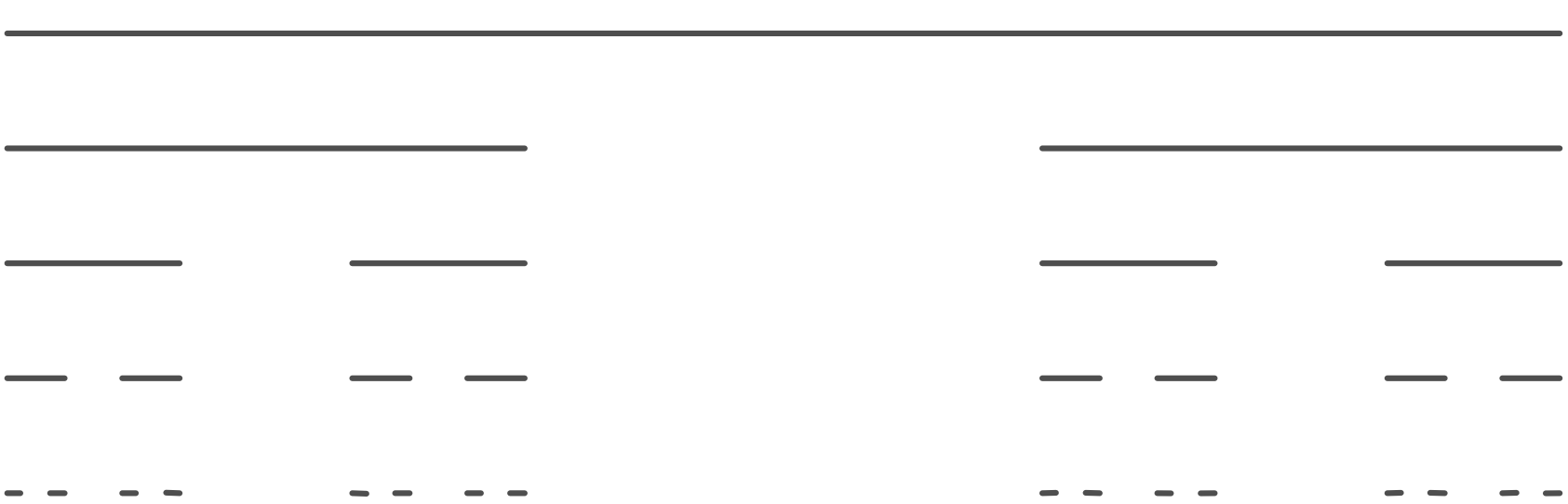
Remarque

Il peut être très difficile de déterminer si un nombre réel est rationel ou non. Par exemple, personne ne sait si $\pi + e$ est rationnel ou irrationnel.

Rappels sur l'espace de Cantor

Définition (Cantor, 1883)

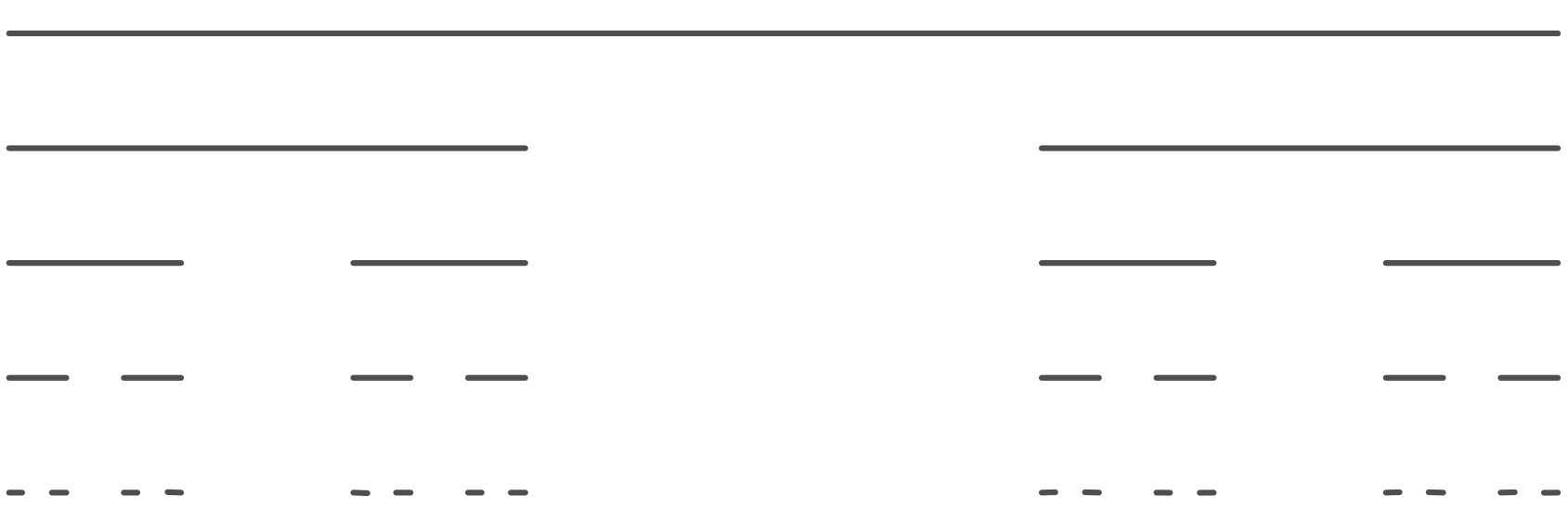
L'ensemble de Cantor est le sous-ensemble de $[0, 1]$ obtenu en itérant autant que possible l'opération "enlever le tier du milieu d'un segment".



Rappels sur l'espace de Cantor

Définition (Cantor, 1883)

L'ensemble de Cantor est le sous-ensemble de $[0, 1]$ obtenu en itérant autant que possible l'opération "enlever le tier du milieu d'un segment".



Remarque

Topologiquement, l'ensemble de Cantor peut être décrit comme $\{0, 1\}^{\mathbb{N}}$ muni de la topologie produit.

Rappels sur l'espace de Cantor

Théorème (Brouwer, 1910)

L'espace de Cantor $\mathfrak{C}$ est l'unique espace topologique qui est compact, totalement discontinu, sans point isolé et métrisable.

Rappels sur l'espace de Cantor

Théorème (Brouwer, 1910)

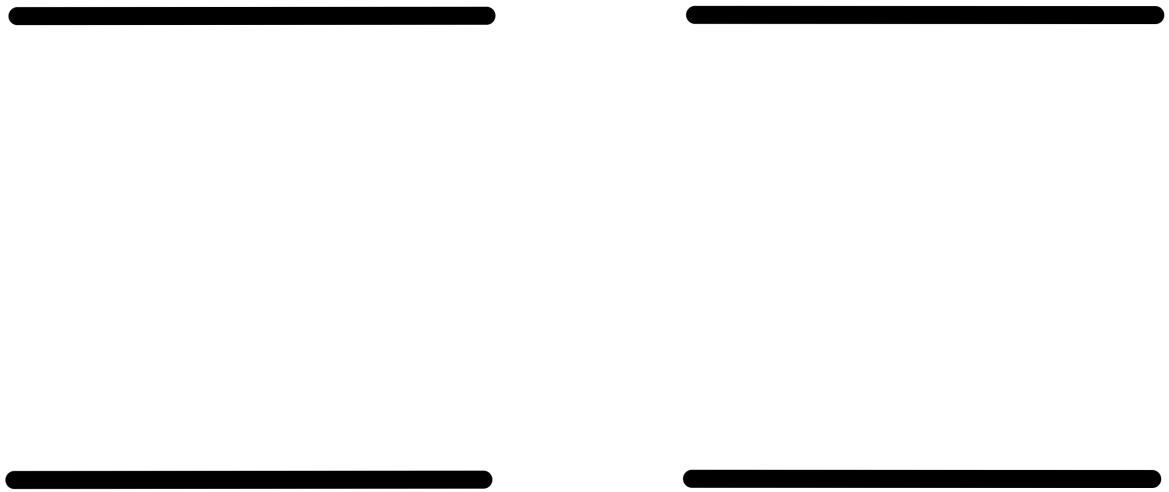
L'espace de Cantor $\mathfrak{C}$ est l'unique espace topologique qui est compact, totalement discontinu, sans point isolé et métrisable.

Corollaire

Pour tout $n \geq 1$, $\mathfrak{C}^n$ est homéomorphe à l'espace de Cantor.

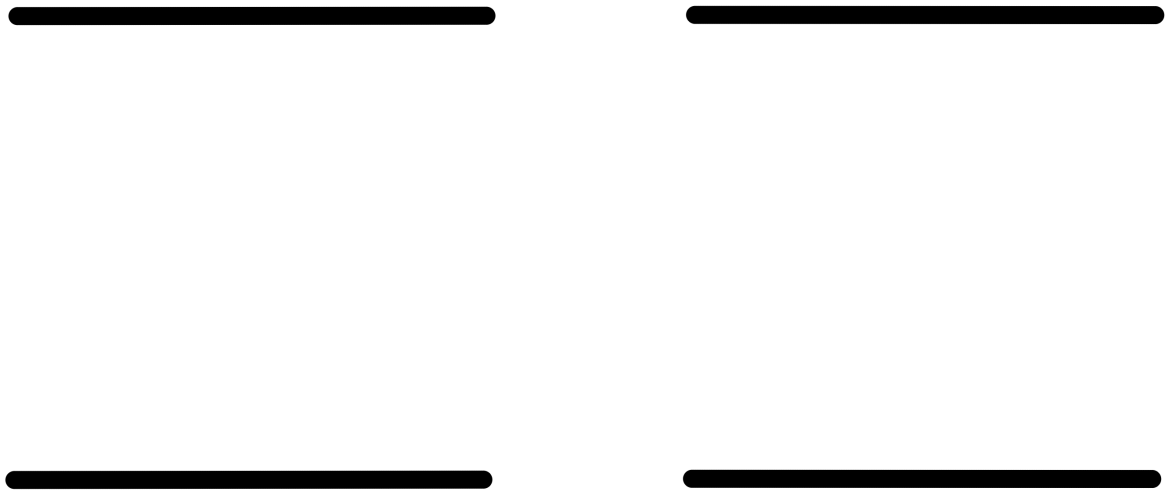
Groupe V (Thompson, 1965)

Le groupe V est un groupe d'homéomorphismes affines par morceaux de l'espace de Cantor $\mathfrak{C}$.



Groupe V (Thompson, 1965)

Le groupe V est un groupe d'homéomorphismes affines par morceaux de l'espace de Cantor $\mathfrak{C}$.

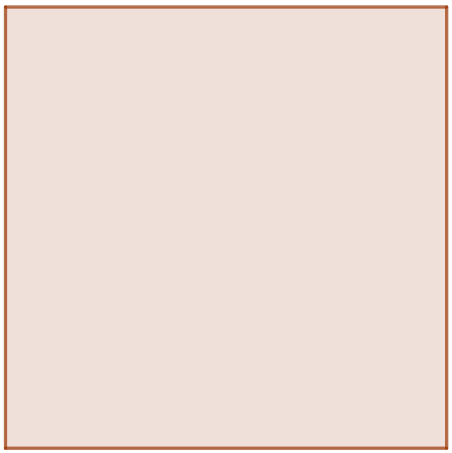


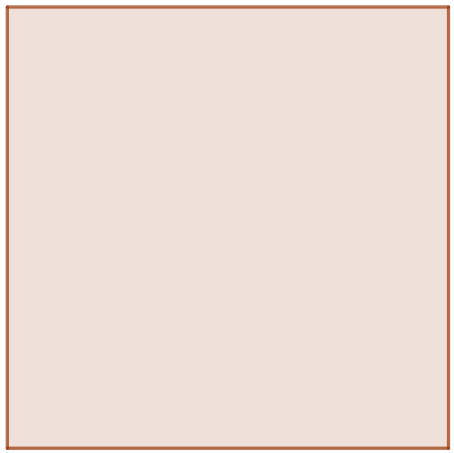
Théorème

Le problème de l'ordre dans V est résoluble par algorithme.

Groupe $2V$ (Brin, 2004)

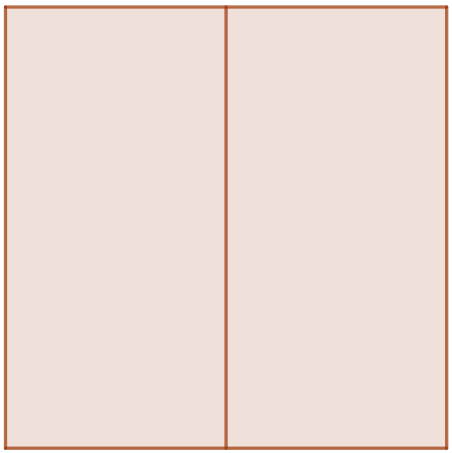
Le groupe $2V$ est un groupe d'homéomorphismes affines par morceaux de l'espace de Cantor $\mathfrak{C}^2$.

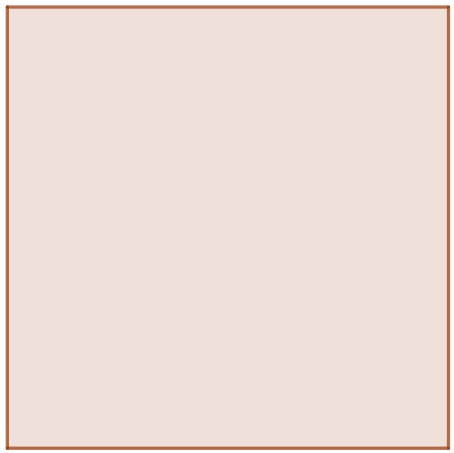




Groupe $2V$ (Brin, 2004)

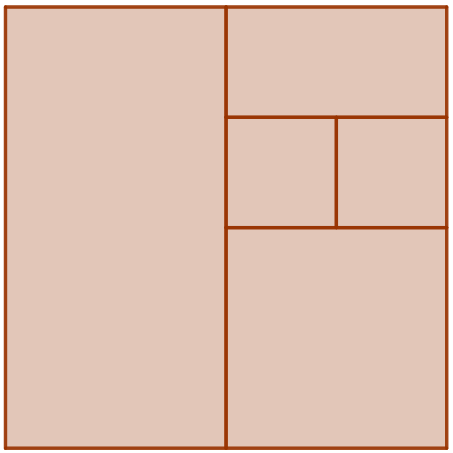
Le groupe $2V$ est un groupe d'homéomorphismes affines par morceaux de l'espace de Cantor $\mathfrak{C}^2$.

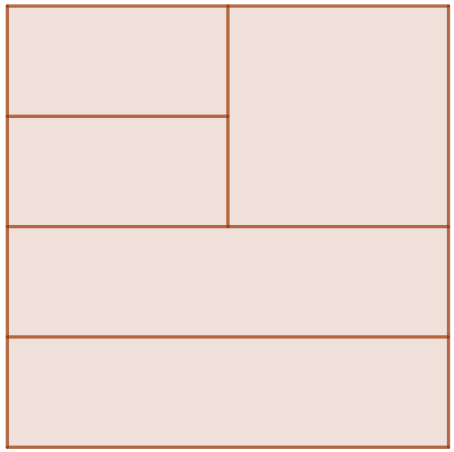




Groupe $2V$ (Brin, 2004)

Le groupe $2V$ est un groupe d'homéomorphismes affines par morceaux de l'espace de Cantor $\mathfrak{C}^2$.





Théorème principal

Théorème (J. Belk et C. Bleak, 2017)

Le problème de l'ordre dans $2V$ ne peut pas être résolu par un algorithme.

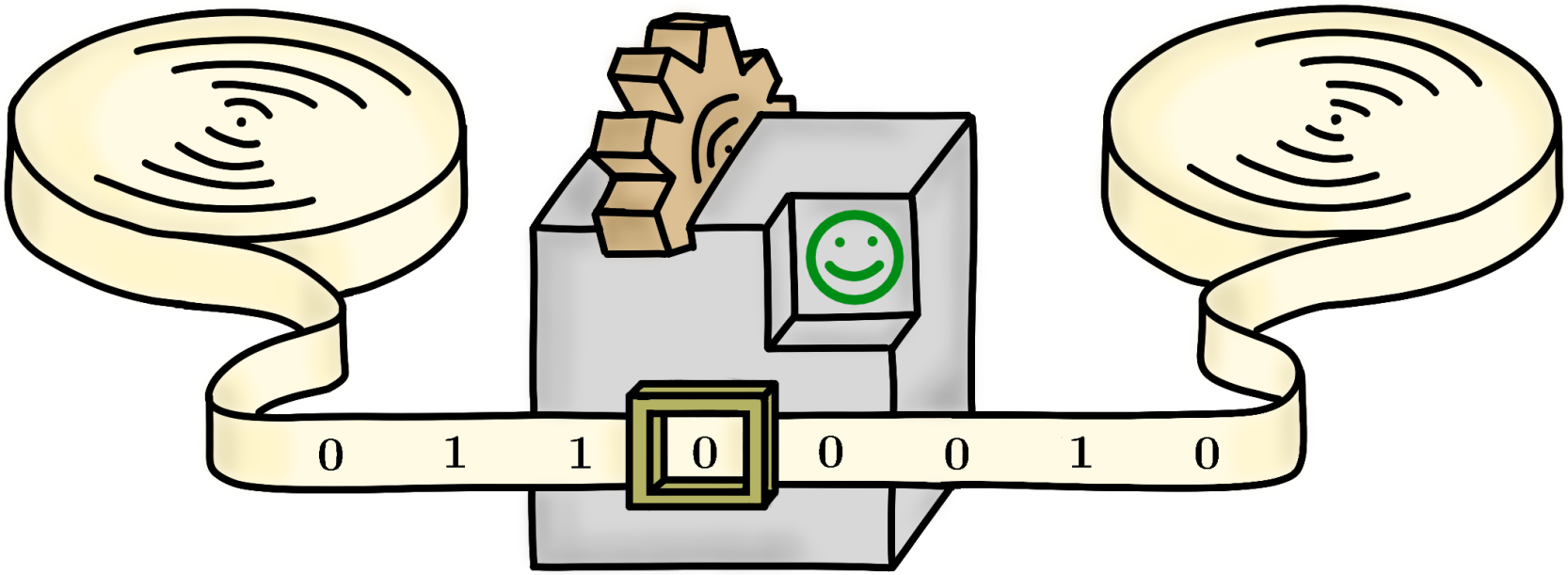
Définition

Une *machine de Turing* $(\mathscr{S}, \mathscr{A}, \mathscr{T})$ est la donnée de trois ensembles :

- un *alphabet* $\mathscr{A}$ fini,
- un *ensemble d'états* $\mathscr{S}$ fini,
- un ensemble d'instructions

$$\mathscr{T} \subset \mathscr{S} \times \mathscr{A} \times \mathscr{S} \times \mathscr{A} \cup \mathscr{S} \times \{\leftarrow, \rightarrow\} \times \mathscr{S}.$$

Machines de Turing



$$\mathscr{S} = \{\text{☺}, \text{☹}\} \qquad \mathscr{A} = \{0, 1\}$$

$$\mathscr{T} = \{(0, \text{☺}, 1, \text{☹}), (1, \text{☺}, 0, \text{☹}), (\text{☹}, \leftarrow, \text{☺})\}$$

Problème de l'arrêt

Théorème (Turing, 1936)

Il n'existe pas d'algorithme décidant, pour chaque machine de Turing et pour chaque entrée, si la machine s'arrêtera.

Problème de l'arrêt

Théorème (Turing, 1936)

Il n'existe pas d'algorithme décidant, pour chaque machine de Turing et pour chaque entrée, si la machine s'arrêtera.

Preuve. On peut supposer que l'alphabet de toute machine de Turing est inclus dans $\mathbb{N}$. Fixons une énumération des machines de Turing et une énumération des mots (finis) écrits sur $\mathbb{N}$. Si le problème de l'arrêt est décidable, il existe un algorithme A qui détermine si la n-ème machine de Turing s'arrête lorsqu'on lui donne le m-ème mot comme instruction. Le nouvel algorithme B défini par

$B(n)$ s'arrête si $A(n, n)$ dit non et boucle à l'infini sinon

vérifie la propriété : si m est le numéro de B, alors $B(m)$ s'arrête si, et seulement si, $B(m)$ ne s'arrête pas. □

Preuve du théorème

Objectif : Associer à certaines machines de Turing un élément de $2V$.

Preuve du théorème

Objectif : Associer à certaines machines de Turing un élément de $2V$.

Définition

Une machine de Turing est *CR* (pour *complète réversible*) si l'application naturelle associée

$$\mathscr{S} \times \mathscr{A}^{\mathbb{Z}} \to \mathscr{S} \times \mathscr{A}^{\mathbb{Z}}$$

est une bijection.

Preuve du théorème

Objectif : Associer à certaines machines de Turing un élément de $2V$.

Définition

Une machine de Turing est *CR* (pour *complète réversible*) si l'application naturelle associée

$$\mathscr{S} \times \mathscr{A}^{\mathbb{Z}} \to \mathscr{S} \times \mathscr{A}^{\mathbb{Z}}$$

est une bijection.

Par exemple, la machine de Turing précédente est CR :

$$\mathscr{S} = \{\text{☺}, \text{☹}\} \qquad \mathscr{A} = \{0, 1\}$$

$$\mathscr{T} = \{(0, \text{☺}, 1, \text{☹}), (1, \text{☺}, 0, \text{☹}), (\text{☹}, \leftarrow, \text{☺})\}$$

Preuve du théorème

En supposant que $\mathscr{S} = \mathscr{A} = \{0, 1\}$ pour simplifier, on va identifier $\mathscr{S} \times \mathscr{A}^{\mathbb{Z}}$ avec $\mathfrak{C}^2 \simeq (\{0, 1\}^{\mathbb{N}})^2$ via

$$\left\{ \begin{array}{ccc} \mathscr{S} \times \mathscr{A}^{\mathbb{Z}} & \to & \mathfrak{C}^2 \\ (s, \cdots a_{-1} a_0 a_1 \cdots) & \mapsto & (s a_1 a_2 \cdots, a_0 a_{-1} a_{-2} \cdots) \end{array} \right. .$$

Preuve du théorème

En supposant que $\mathscr{S} = \mathscr{A} = \{0, 1\}$ pour simplifier, on va identifier $\mathscr{S} \times \mathscr{A}^{\mathbb{Z}}$ avec $\mathfrak{C}^2 \simeq (\{0, 1\}^{\mathbb{N}})^2$ via

$$\left\{ \begin{array}{ccc} \mathscr{S} \times \mathscr{A}^{\mathbb{Z}} & \to & \mathfrak{C}^2 \\ (s, \cdots a_{-1} a_0 a_1 \cdots) & \mapsto & (s a_1 a_2 \cdots, a_0 a_{-1} a_{-2} \cdots) \end{array} \right. .$$

Ainsi, chaque machine de Turing CR induit un homéomorphisme $\mathfrak{C}^2 \to \mathfrak{C}^2$ via sa bijection associée $\mathscr{S} \times \mathscr{A}^{\mathbb{Z}} \to \mathscr{S} \times \mathscr{A}^{\mathbb{Z}}$.

Preuve du théorème

Par exemple,

$$(0u_1u_2\cdots, 0v_1v_2\cdots) \mapsto \left(☹, \cdots v_2v_1\overline{0}u_1u_2\cdots\right)$$

$$\mapsto \left(☺, \cdots v_2\overline{v_1}0u_1u_2\cdots\right) \mapsto (10u_1u_2\cdots, v_1v_2\cdots)$$

pour la machine de Turing

$$\mathscr{S} = \{☺, ☹\} \qquad \mathscr{A} = \{0, 1\}$$

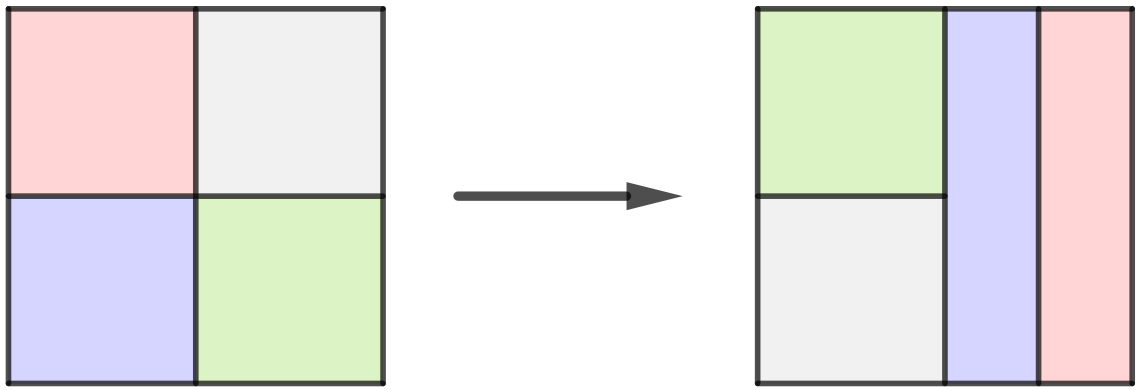
$$\mathscr{T} = \{(0, ☺, 1, ☹), (1, ☺, 0, ☹), (☹, \leftarrow, ☺)\}$$

Preuve du théorème

En étudiant tous les cas possibles, on trouve l'homéomorphisme $\mathfrak{C}^2 \to \mathfrak{C}^2$ suivant :

$$\begin{cases} (0u, 0v) & \mapsto & (10u, v) \\ (0u, 1v) & \mapsto & (11u, v) \\ (1u, 0v) & \mapsto & (0u, 1v) \\ (1u, 1v) & \mapsto & (0u, 0v) \end{cases}$$

La représentation graphique ci-dessous montre que cet homéomorphisme définit un élément de $2V$.



Preuve du théorème

Conséquence : S'il existe un algorithme résolvant le problème de l'ordre dans $2V$, alors cet algorithme résout aussi le problème de périodicité pour les machines de Turing CR.

Preuve du théorème

Conséquence : S'il existe un algorithme résolvant le problème de l'ordre dans $2V$, alors cet algorithme résout aussi le problème de périodicité pour les machines de Turing CR.

Définition

Une machine de Turing CR est *périodique* si sa bijection associée

$$\mathscr{S} \times \mathscr{A}^{\mathbb{Z}} \to \mathscr{S} \times \mathscr{A}^{\mathbb{Z}}$$

est d'ordre fini.

Preuve du théorème

Conséquence : S'il existe un algorithme résolvant le problème de l'ordre dans $2V$, alors cet algorithme résout aussi le problème de périodicité pour les machines de Turing CR.

Définition

Une machine de Turing CR est *périodique* si sa bijection associée

$$\mathscr{S} \times \mathscr{A}^{\mathbb{Z}} \to \mathscr{S} \times \mathscr{A}^{\mathbb{Z}}$$

est d'ordre fini.

Théorème (Kari et Ollinger, 2008)

Il n'existe pas d'algorithme résolvant le problème de périodicité des machines de Turing CR.