



MAISON DES MATHÉMATIQUES
ET DE L'INFORMATIQUE

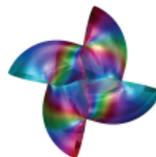


PAVAGES, APÉRIODICITÉ ET CALCULABILITÉ

Nathalie Aubrun

Lundi 12/12/2016 à 20h

à l'ENS de Lyon
Site Descartes



SOIRÉES MATHÉMATIQUES DE LYON

TRAILERS ET VIDÉOS
SUR
WWW.MMI-LYON.FR

Public : Étudiants post-bac
Entrée libre et gratuite



T 04 72 43 11 80
contact@mmi-lyon.fr

MMI portée par



Tutelles

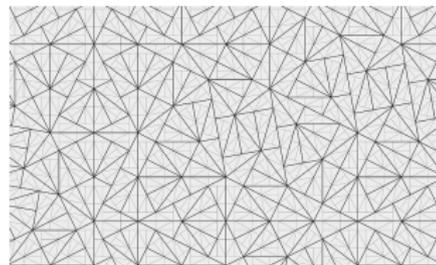
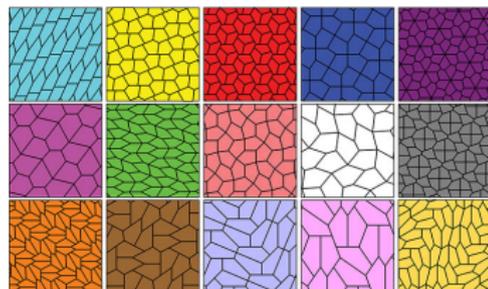
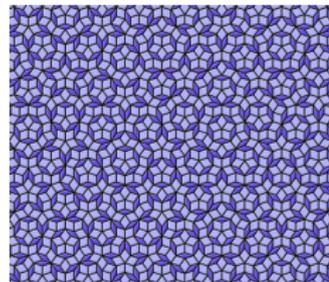
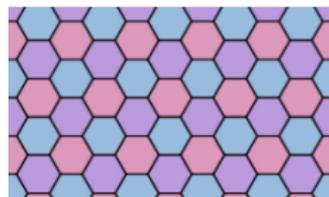


Partenaires SML

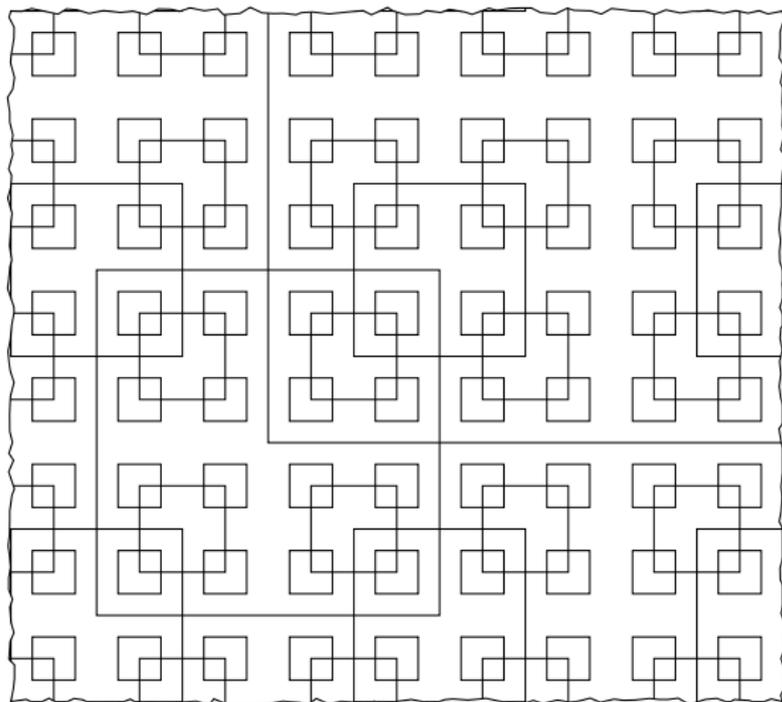


Les pavages, qu'est-ce que c'est ?

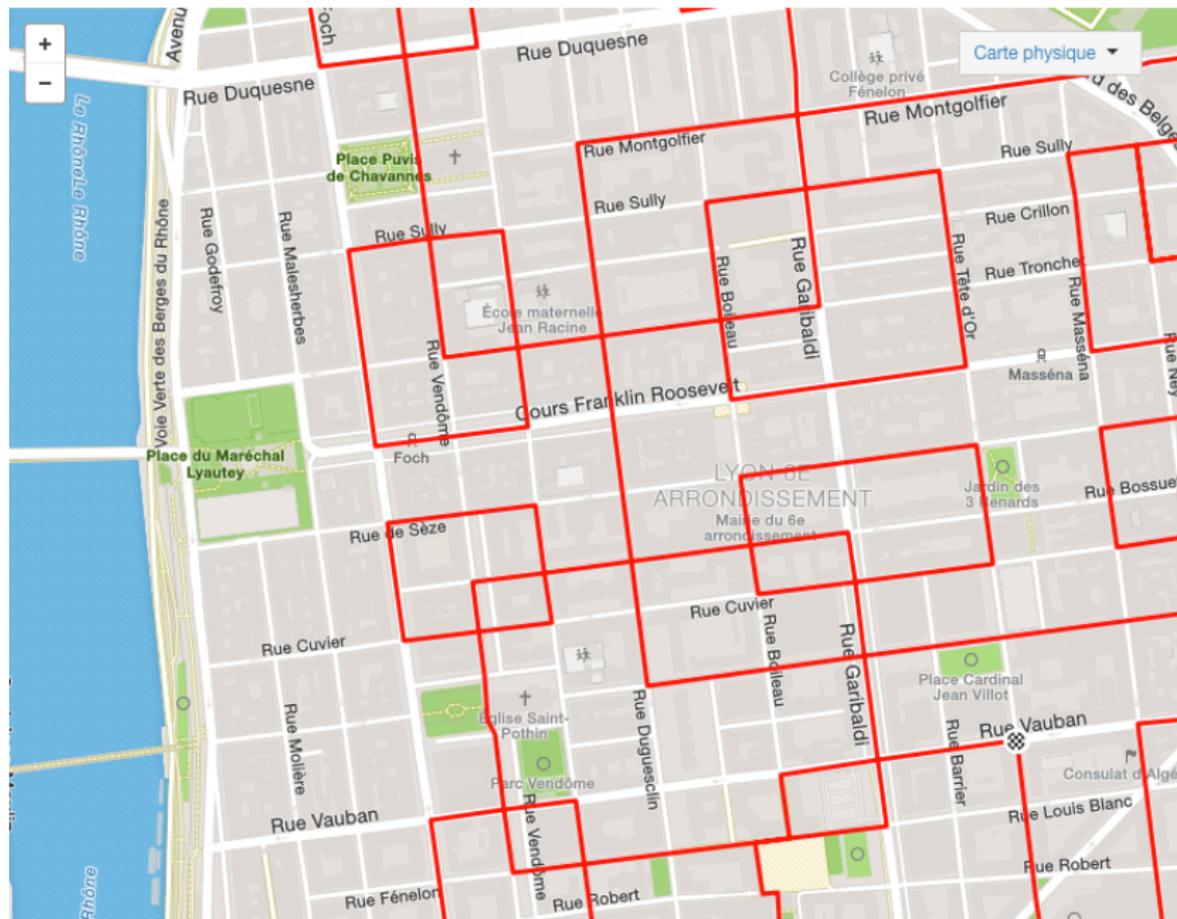
Recouvrir le plan (ou l'espace) avec des copies de certaines tuiles de base.



Comment construire des pavages compliqués avec peu de tuiles ?



Comment faire de jolis dessins dans les rues de Lyon ?



Tuiles de Wang (1961)

Dans cet exposé, on se restreint à des pavages

- du plan,
- par des tuiles carrées portant une couleur par arête,
- avec nombre fini de couleurs,



- où les tuiles sont placées sommet contre sommet,
- où deux tuiles voisines portent la même couleur sur leur arête commune.



Tuiles de Wang (1961)

Dans cet exposé, on se restreint à des pavages

- du plan,
- par des tuiles carrées portant une couleur par arête,
- avec nombre fini de couleurs,



- où les tuiles sont placées sommet contre sommet,
- où deux tuiles voisines portent la même couleur sur leur arête commune.



Interdiction de tourner les tuiles !!

Tuiles de Wang (1961)

Dans cet exposé, on se restreint à des pavages

- du plan,
- par des tuiles carrées portant une couleur par arête,
- avec nombre fini de couleurs,



- où les tuiles sont placées sommet contre sommet,
- où deux tuiles voisines portent la même couleur sur leur arête commune.



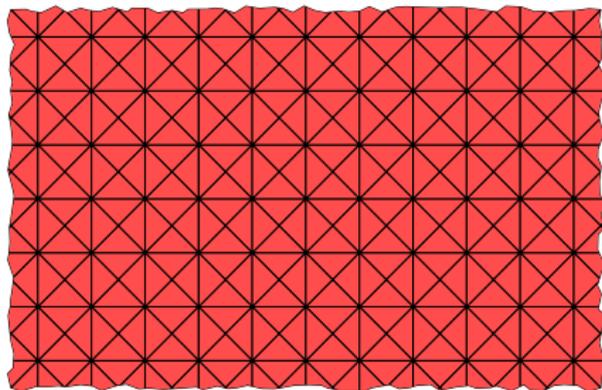
Interdiction de tourner les tuiles !!

Étant donné un jeu de tuiles τ , on s'intéresse à **tous les pavages valides** par τ .

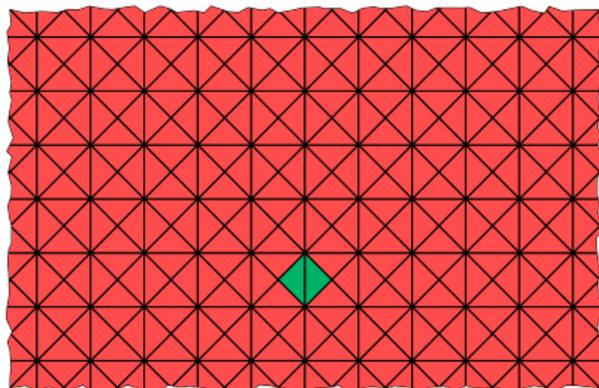
Premier exemple



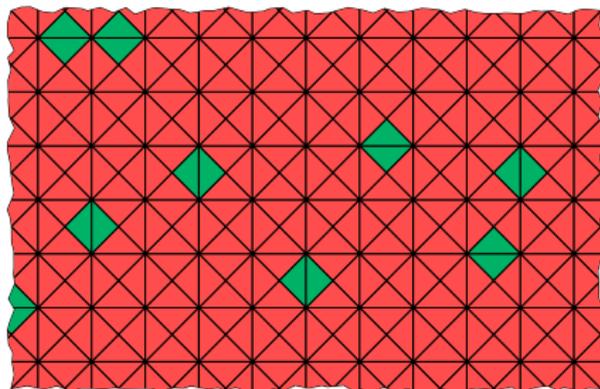
Premier exemple



Premier exemple



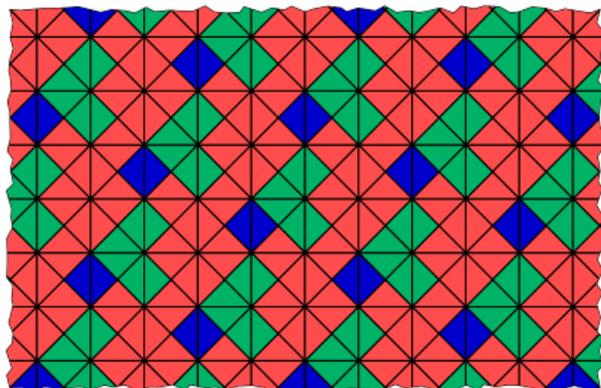
Premier exemple



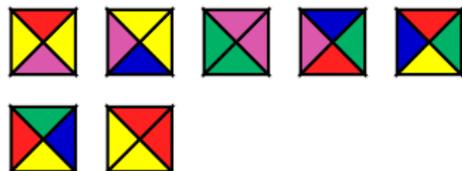
Deuxième exemple



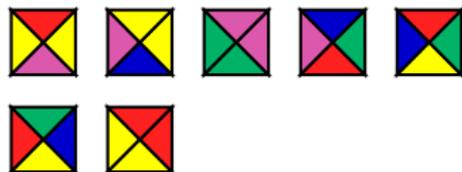
Deuxième exemple



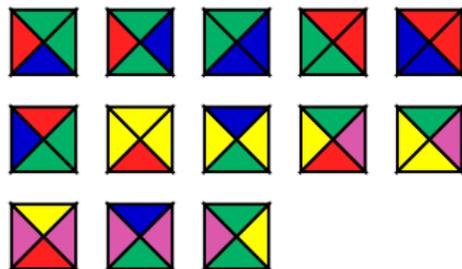
Et celui-ci ?



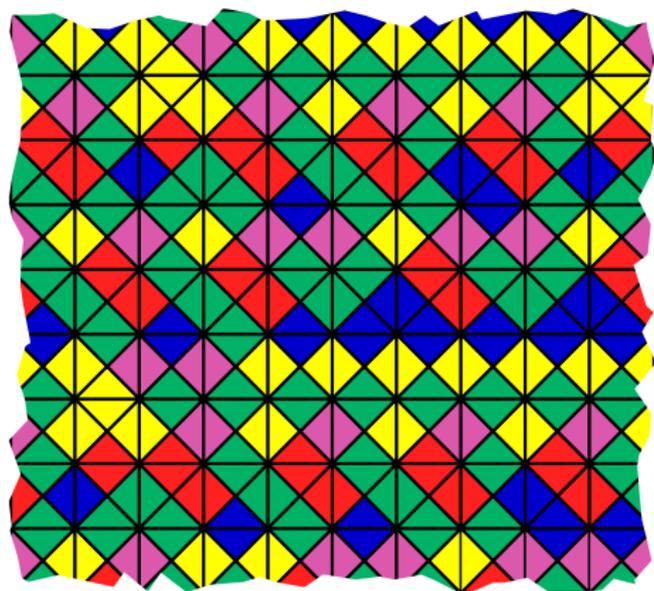
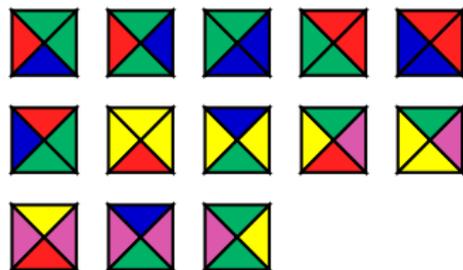
Et celui-ci ?



Et celui-là ?

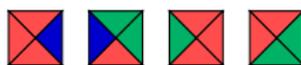


Et celui-là ?



Le modèle des tuiles de Wang

Ensemble fini de tuiles de Wang τ (une infinité de copies de chaque tuile)

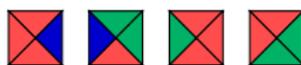


Règle de voisinage



Le modèle des tuiles de Wang

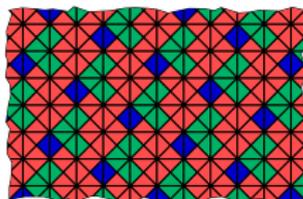
Ensemble fini de tuiles de Wang τ (une infinité de copies de chaque tuile)



Règle de voisinage



Exemple de pavage par τ (X_τ l'ensemble de tous les pavages valides)



Le modèle des tuiles de Wang

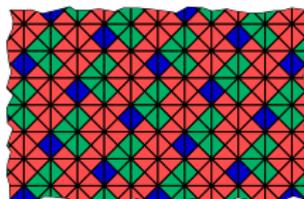
Ensemble fini de tuiles de Wang τ (une infinité de copies de chaque tuile)



Règle de voisinage



Exemple de pavage par τ (X_τ l'ensemble de tous les pavages valides)



Problème du Domino

Entrée : Un jeu fini de tuiles de Wang τ

Sortie : **Oui** s'il existe un pavage valide par τ ($X_\tau \neq \emptyset$), **Non** sinon ($X_\tau = \emptyset$).

Le problème du Domino : remarque

Problème du Domino

Entrée : Un jeu fini de tuiles de Wang τ

Sortie : **Oui** s'il existe un pavage valide par τ ($X_\tau \neq \emptyset$), **Non** sinon ($X_\tau = \emptyset$).

Le problème du Domino : remarque

Problème du Domino

Entrée : Un jeu fini de tuiles de Wang τ

Sortie : **Oui** s'il existe un pavage valide par τ ($X_\tau \neq \emptyset$), **Non** sinon ($X_\tau = \emptyset$).

Si le jeu de tuiles τ ne permet pas de paver le plan, je suis capable de le détecter.
Comment ?

Le problème du Domino : remarque

Problème du Domino

Entrée : Un jeu fini de tuiles de Wang τ

Sortie : **Oui** s'il existe un pavage valide par τ ($X_\tau \neq \emptyset$), **Non** sinon ($X_\tau = \emptyset$).

Si le jeu de tuiles τ ne permet pas de paver le plan, je suis capable de le détecter.
Comment ?

⇒ Méthode par **essai-erreur**.

Le problème du Domino : remarque

Problème du Domino

Entrée : Un jeu fini de tuiles de Wang τ

Sortie : **Oui** s'il existe un pavage valide par τ ($X_\tau \neq \emptyset$), **Non** sinon ($X_\tau = \emptyset$).

Si le jeu de tuiles τ ne permet pas de paver le plan, je suis capable de le détecter.
Comment ?

⇒ Méthode par **essai-erreur**.

Partie difficile du problème : comment savoir qu'un jeu de tuiles permet de paver le plan ?

Le problème du Domino : remarque

Problème du Domino

Entrée : Un jeu fini de tuiles de Wang τ

Sortie : **Oui** s'il existe un pavage valide par τ ($X_\tau \neq \emptyset$), **Non** sinon ($X_\tau = \emptyset$).

Si le jeu de tuiles τ ne permet pas de paver le plan, je suis capable de le détecter.
Comment ?

⇒ Méthode par **essai-erreur**.

Partie difficile du problème : comment savoir qu'un jeu de tuiles permet de paver le plan ?

Conjecture (Wang, 1961)

Si un jeu de tuiles de Wang permet de paver le plan, alors il peut toujours le faire de manière périodique.

Pavages périodiques (I)

On se donne un jeu de tuiles τ .

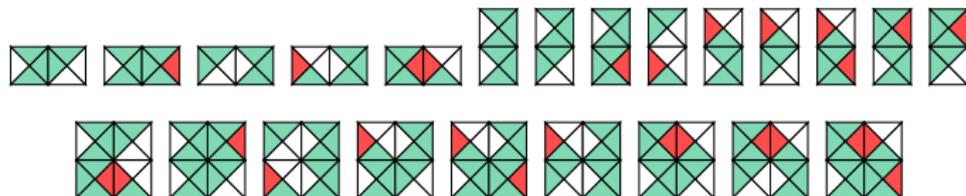


Pavages périodiques (I)

On se donne un jeu de tuiles τ .



On regarde les motifs finis rectangulaires que l'on peut construire.

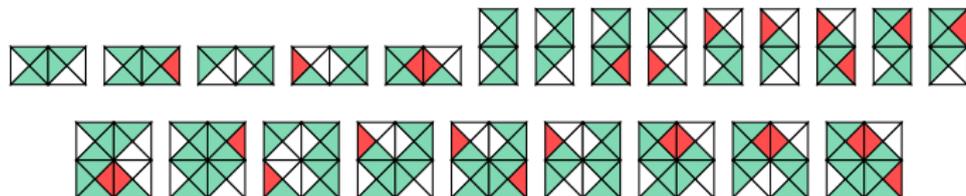


Pavages périodiques (I)

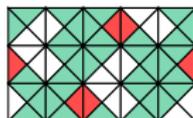
On se donne un jeu de tuiles τ .



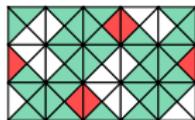
On regarde les motifs finis rectangulaires que l'on peut construire.



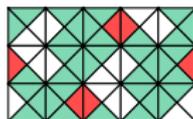
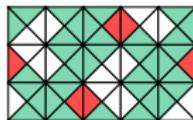
Parmi ceux-ci, on cherche les motifs qui peuvent être « collés à eux-mêmes ».



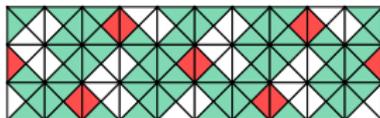
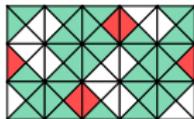
Pavages périodiques (II)



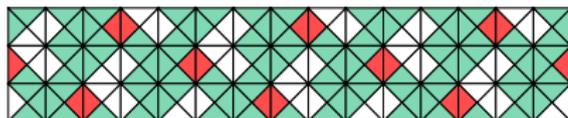
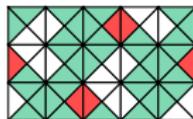
Pavages périodiques (II)



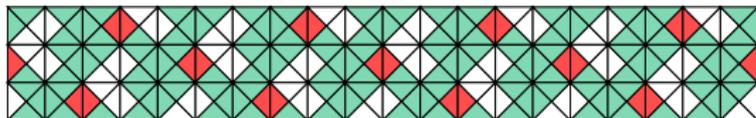
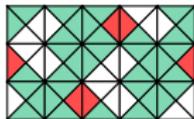
Pavages périodiques (II)



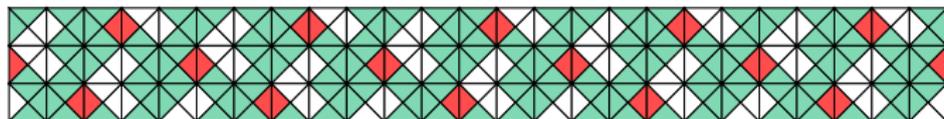
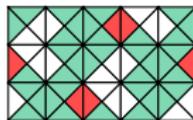
Pavages périodiques (II)



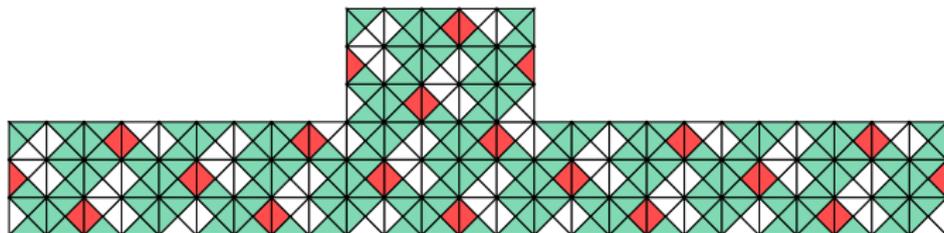
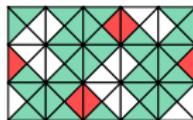
Pavages périodiques (II)



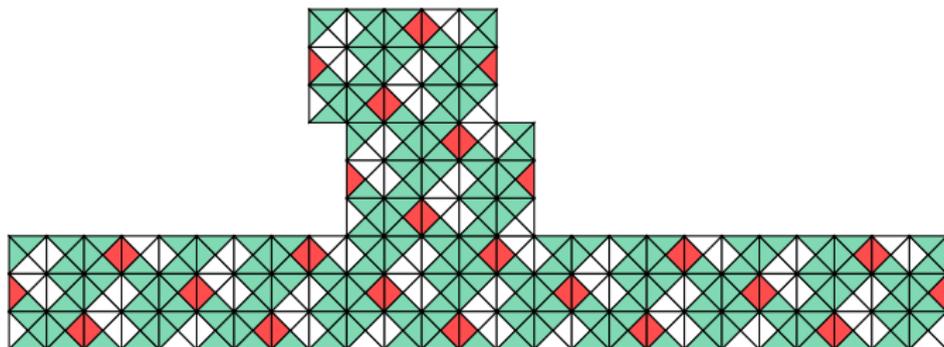
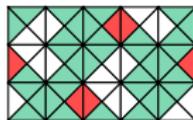
Pavages périodiques (II)



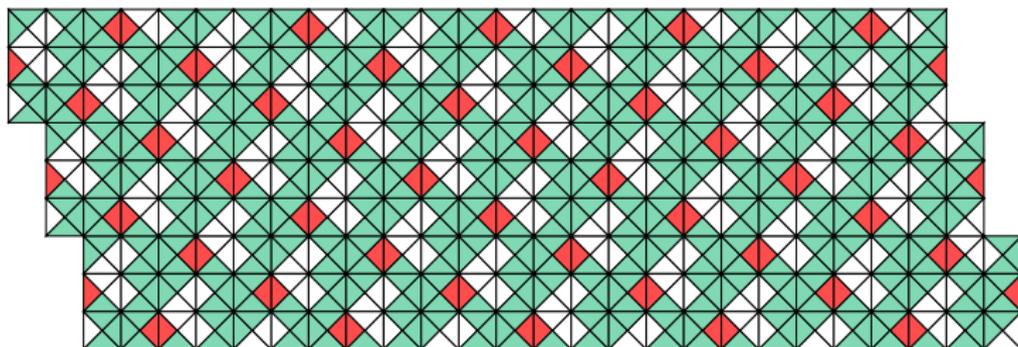
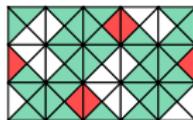
Pavages périodiques (II)



Pavages périodiques (II)



Pavages périodiques (II)



Premier exemple

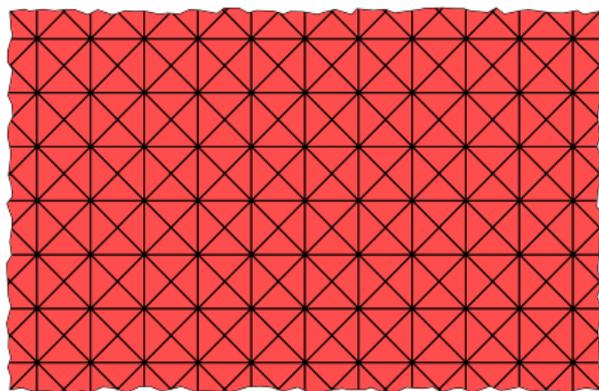


Peut-on construire un pavage périodique ?

Premier exemple



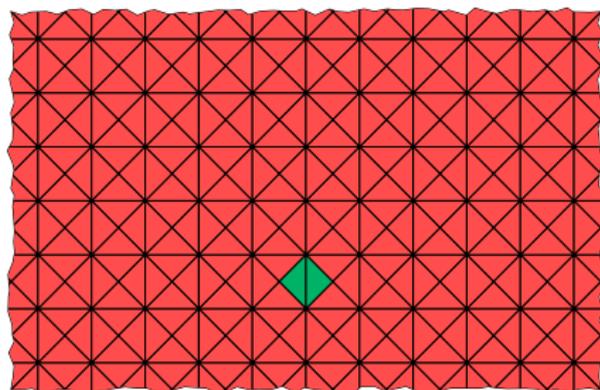
Peut-on construire un pavage périodique ?



Premier exemple



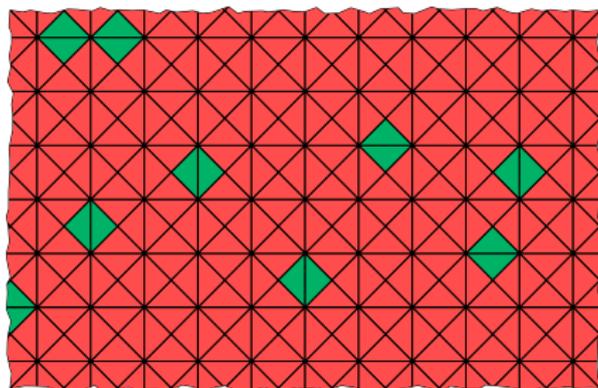
Peut-on construire un pavage périodique ?



Premier exemple



Peut-on construire un pavage périodique ?



Deuxième exemple

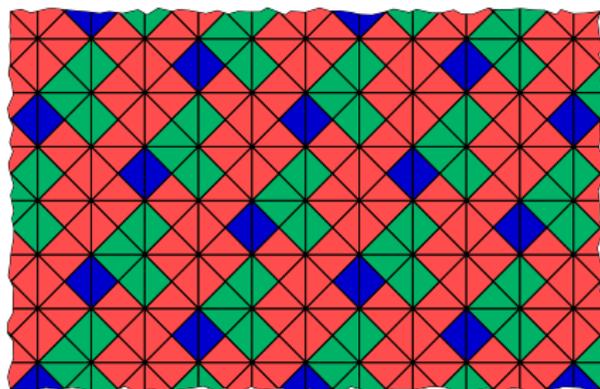


Peut-on construire un pavage périodique ?

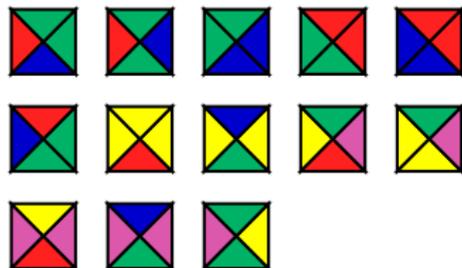
Deuxième exemple



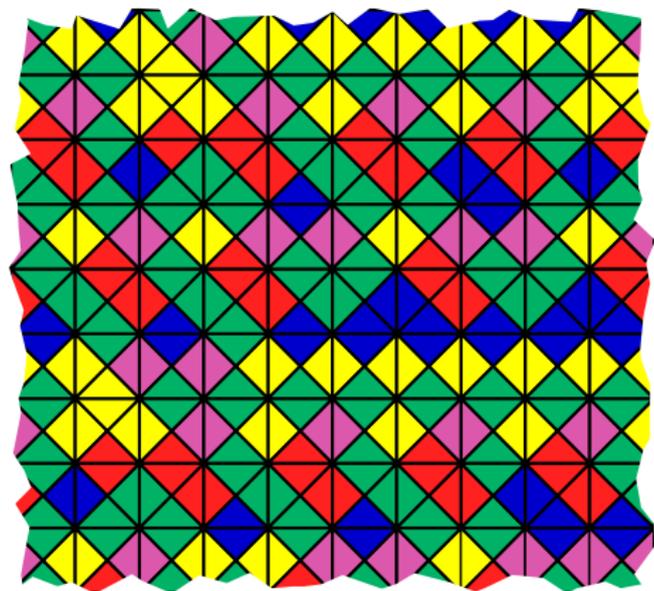
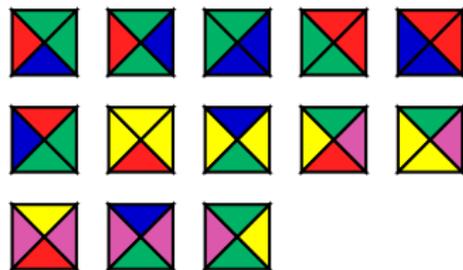
Peut-on construire un pavage périodique ?



Et celui-là ?



Et celui-là ?



Existence de jeux de tuiles apériodiques ?

Conjecture (Wang, 1961)

Si un jeu de tuiles de Wang permet de paver le plan, alors il peut toujours le faire de manière périodique.

Existence de jeux de tuiles apériodiques ?

Conjecture (Wang, 1961)

Si un jeu de tuiles de Wang permet de paver le plan, alors il peut toujours le faire de manière périodique.

Si la conjecture de Wang est vraie, on peut décider le problème du Domino !

Existence de jeux de tuiles apériodiques ?

Conjecture (Wang, 1961)

Si un jeu de tuiles de Wang permet de paver le plan, alors il peut toujours le faire de manière périodique.

Si la conjecture de Wang est vraie, on peut décider le problème du Domino !

Semi-algorithme 1 :

- 1 renvoie un entier n tel que le carré $[1; n] \times [1; n]$ ne peut pas être pavé, s'il existe
- 2 boucle sinon

Semi-algorithme 2 :

- 1 renvoie un entier n tel que le carré $[1; n] \times [1; n]$ ne peut pas être pavé, s'il existe
- 2 boucle sinon

Existence de jeux de tuiles apériodiques ?

Conjecture (Wang, 1961)

Si un jeu de tuiles de Wang permet de paver le plan, alors il peut toujours le faire de manière périodique.

Si la conjecture de Wang est vraie, on peut décider le problème du Domino !

Semi-algorithme 1 :

- 1 renvoie un entier n tel que le carré $[1; n] \times [1; n]$ ne peut pas être pavé, s'il existe
- 2 boucle sinon

Semi-algorithme 2 :

- 1 renvoie un entier n tel que le carré $[1; n] \times [1; n]$ ne peut pas être pavé, s'il existe
- 2 boucle sinon

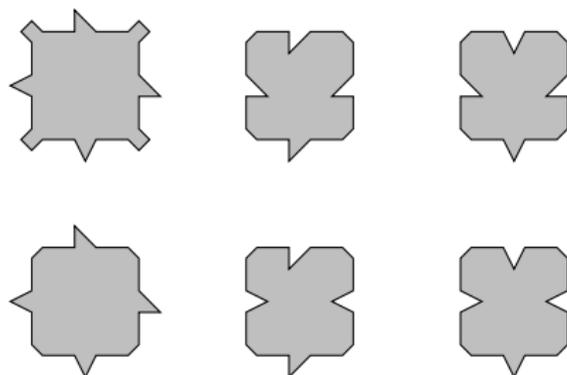
Un jeu de tuiles est **apériodique** s'il permet de paver le plan, et si aucun des pavages valides n'est périodique.

Question

Peut-on on construire un jeu de tuiles de Wang apériodique ?

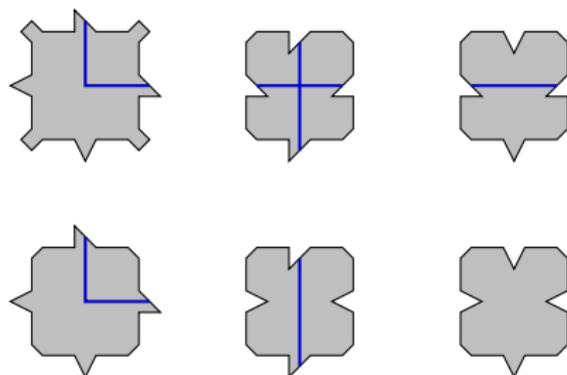
Le jeu de tuiles de Robinson

On s'autorise les rotations et réflexions des 6 tuiles suivantes :

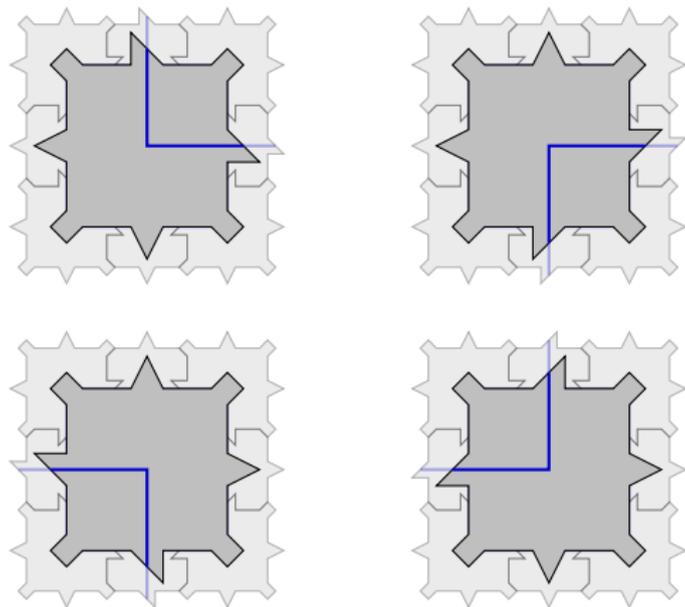


Le jeu de tuiles de Robinson

On s'autorise les rotations et réflexions des 6 tuiles suivantes :

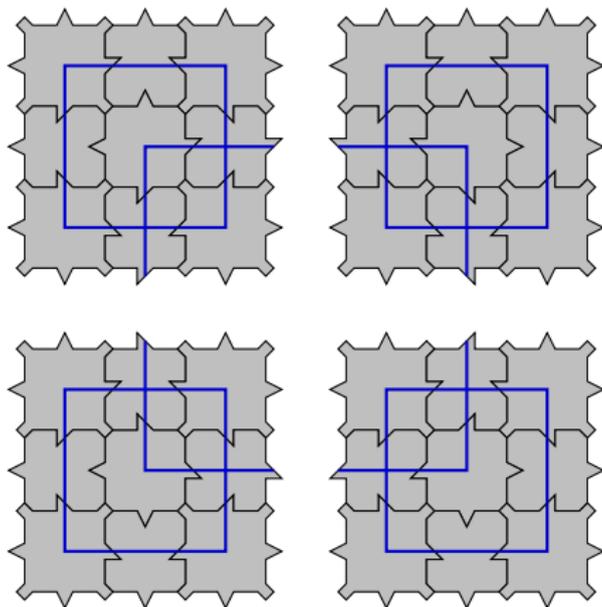


Pavage par le jeu de Robinson (I)

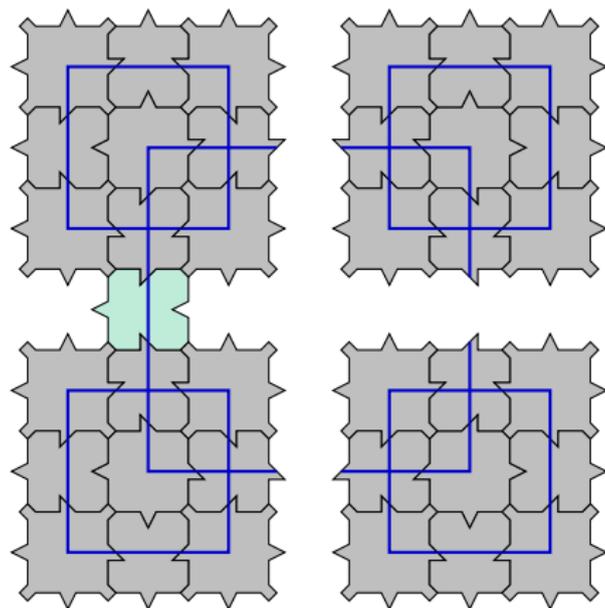


Ils se comportent comme de grands .

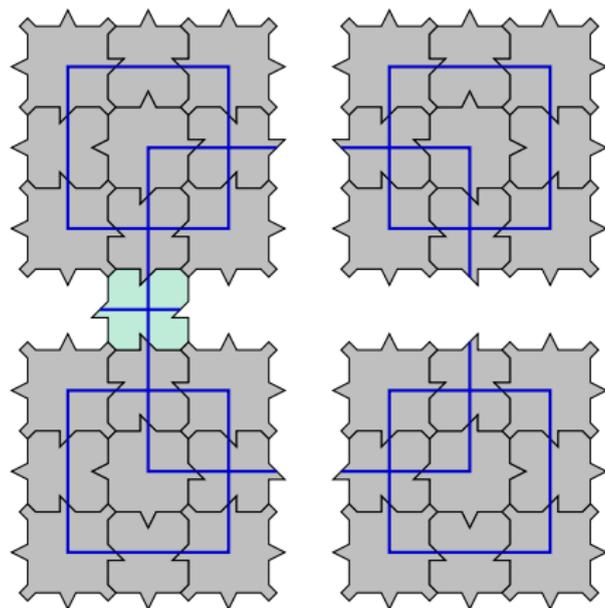
Pavage par le jeu de Robinson (II)



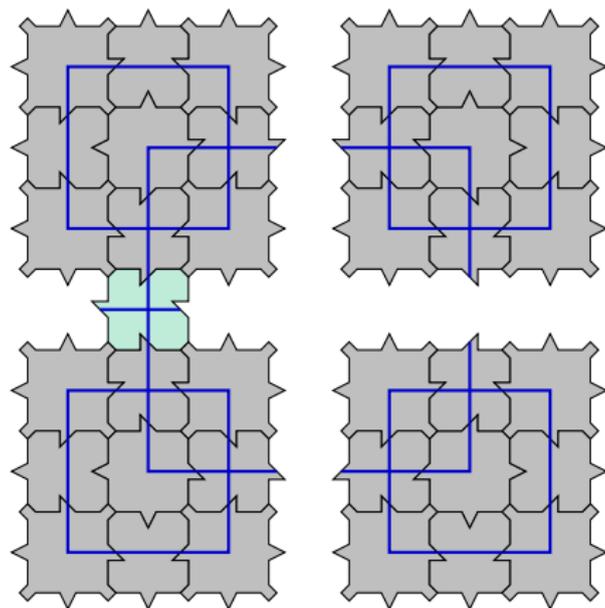
Pavage par le jeu de Robinson (II)



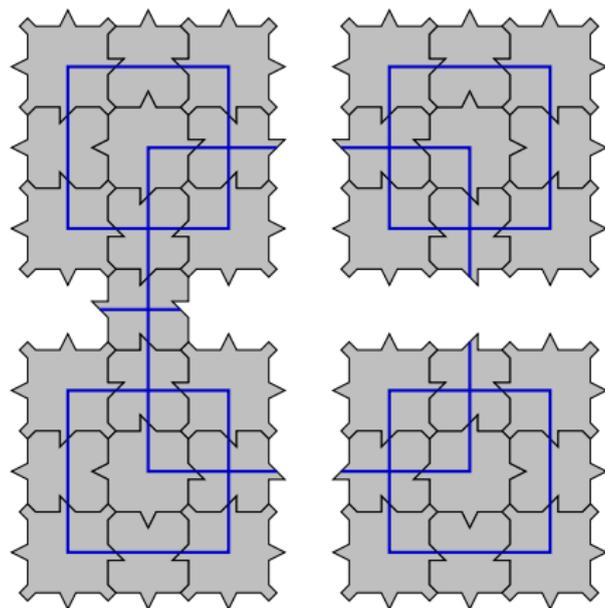
Pavage par le jeu de Robinson (II)



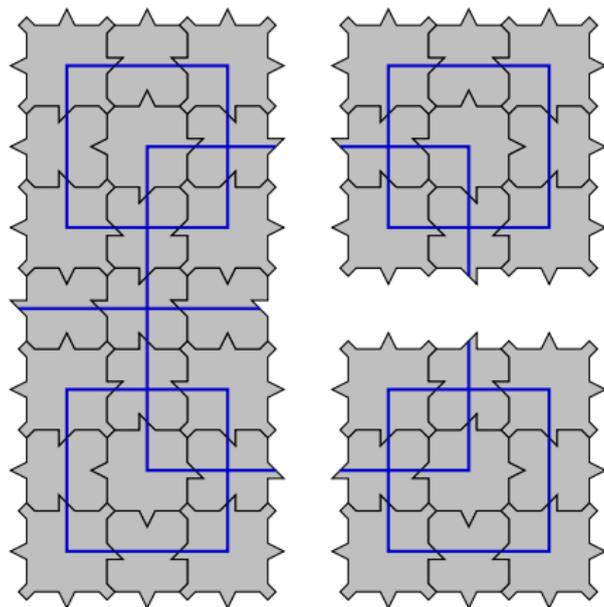
Pavage par le jeu de Robinson (II)



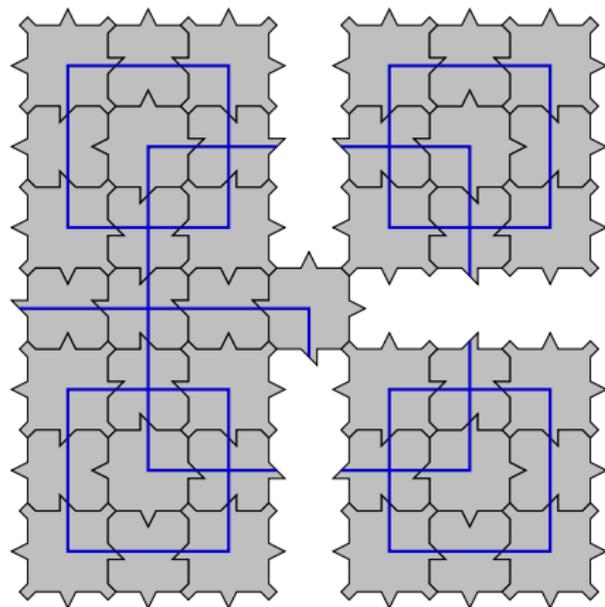
Pavage par le jeu de Robinson (II)



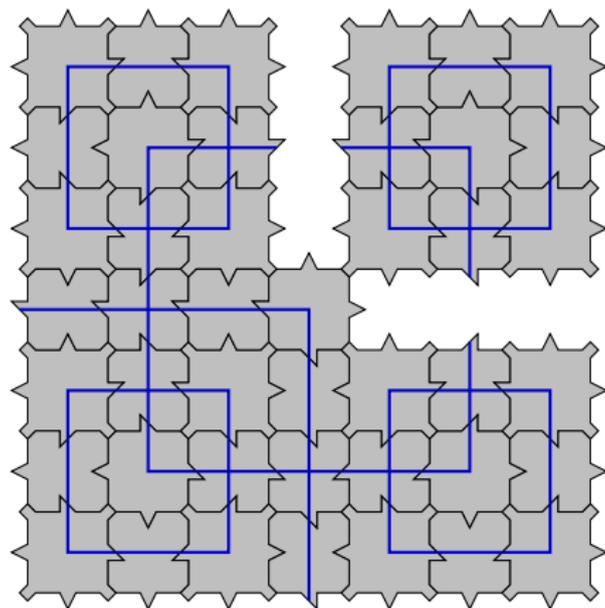
Pavage par le jeu de Robinson (II)



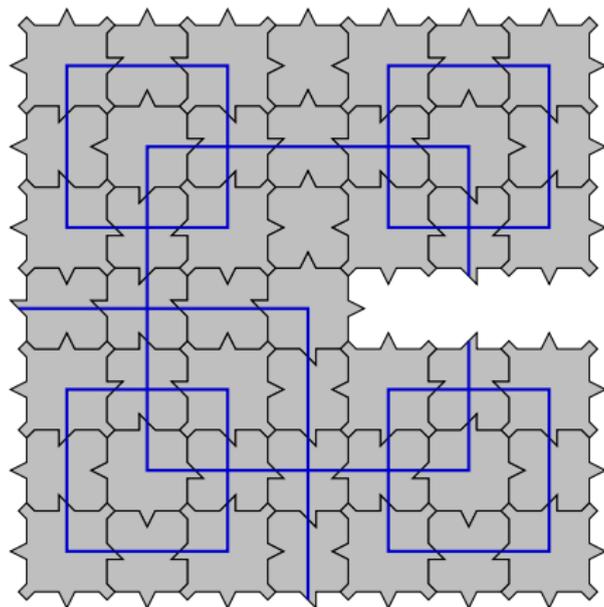
Pavage par le jeu de Robinson (II)



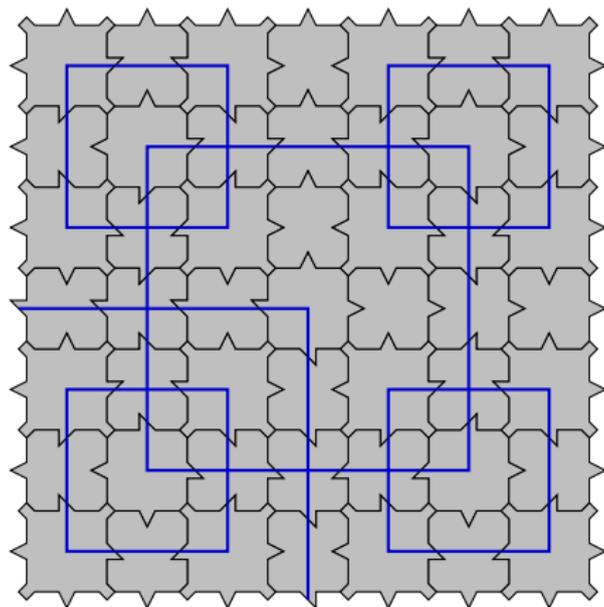
Pavage par le jeu de Robinson (II)



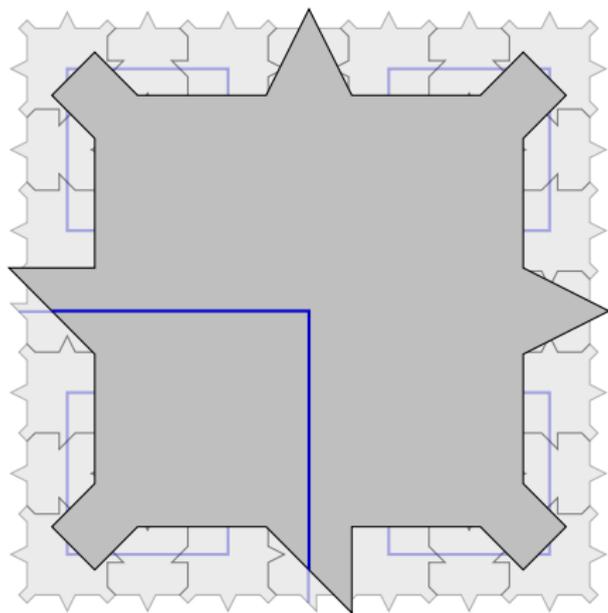
Pavage par le jeu de Robinson (II)



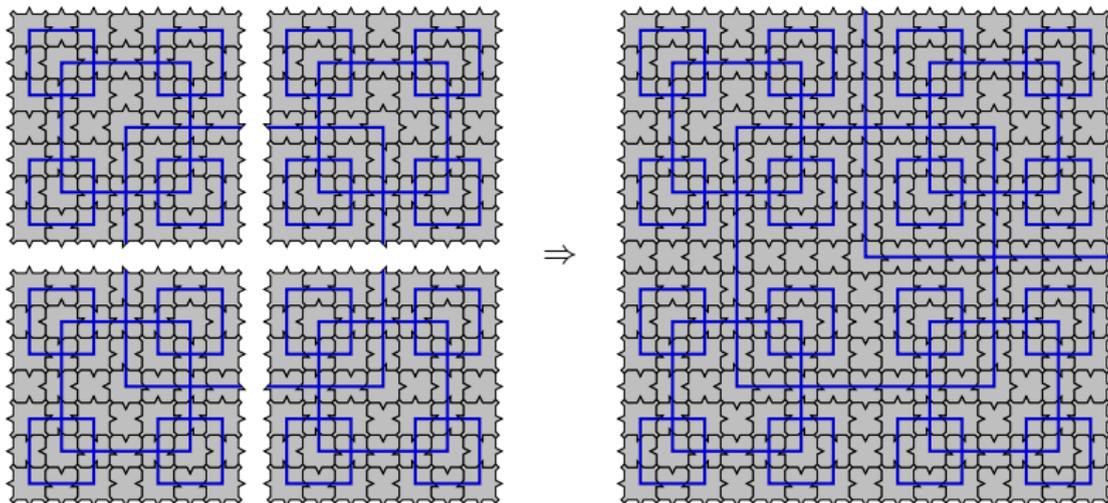
Pavage par le jeu de Robinson (II)



Pavage par le jeu de Robinson (II)

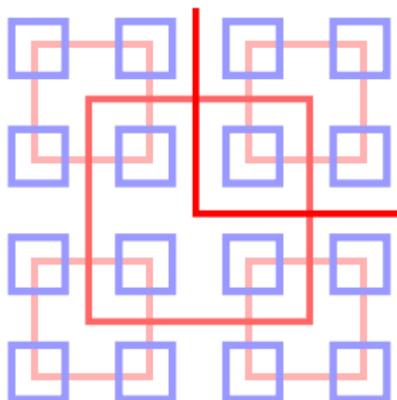


Pavage par le jeu de Robinson (III)



Sur la structure des pavages

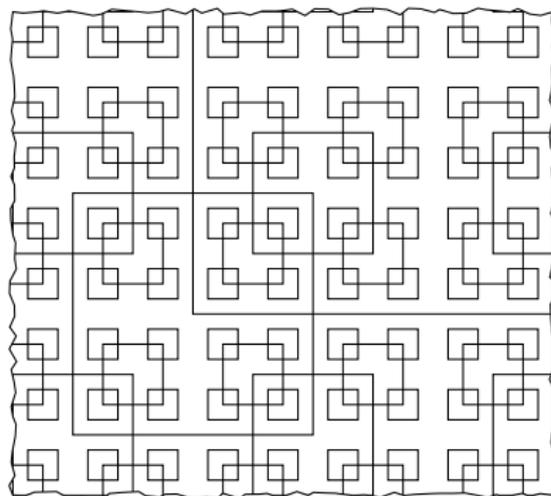
Hierarchie de carrés : les carrés de niveau n sont regroupés par 4 pour former les carrés de niveau $n + 1$



Apériodicité du jeu de tuiles de Robinson

Proposition

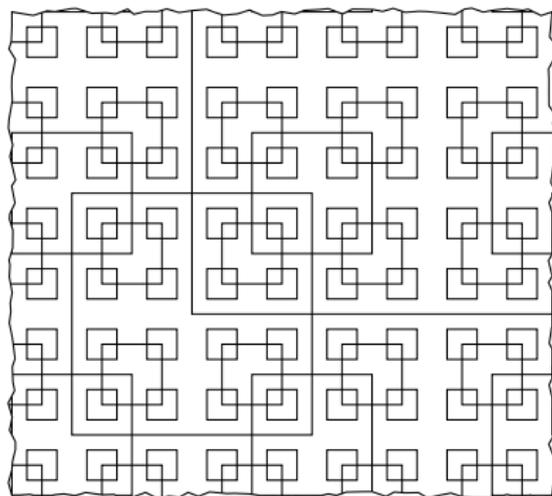
Le jeu de tuiles de Robinson est apériodique.



Apériodicité du jeu de tuiles de Robinson

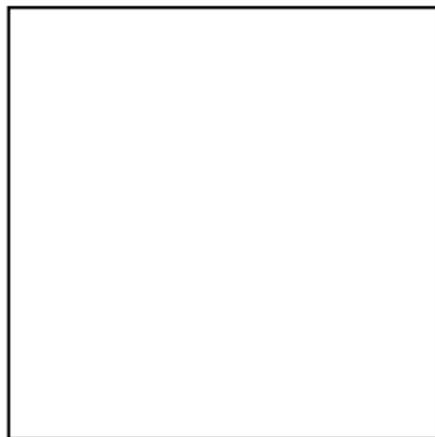
Proposition

Le jeu de tuiles de Robinson est apériodique.

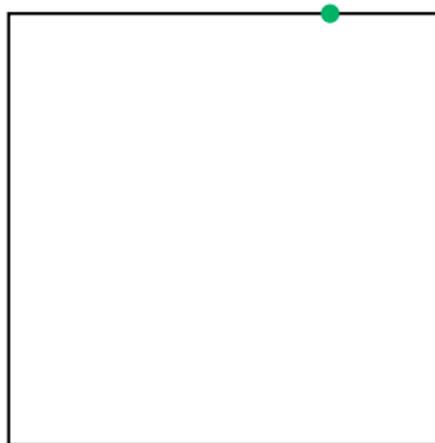


⇒ la conjecture formulée par Wang est fausse !

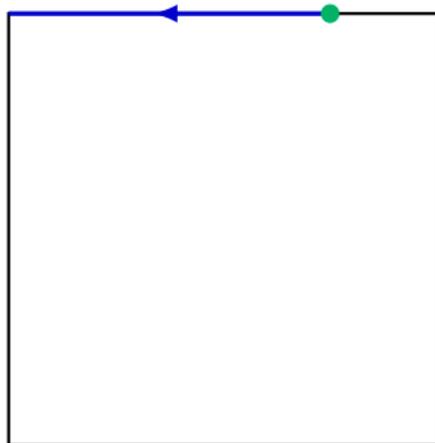
Préparation de l'itinéraire



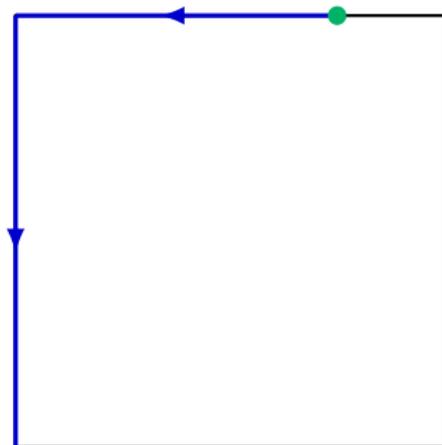
Préparation de l'itinéraire



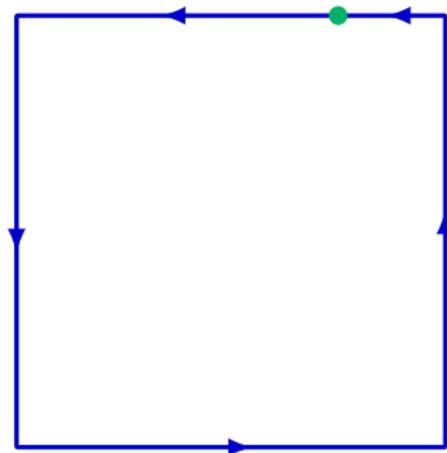
Préparation de l'itinéraire



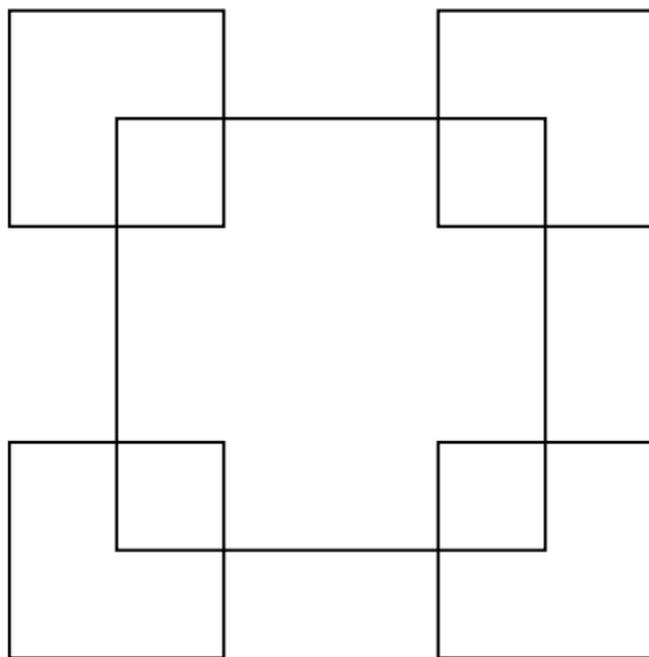
Préparation de l'itinéraire



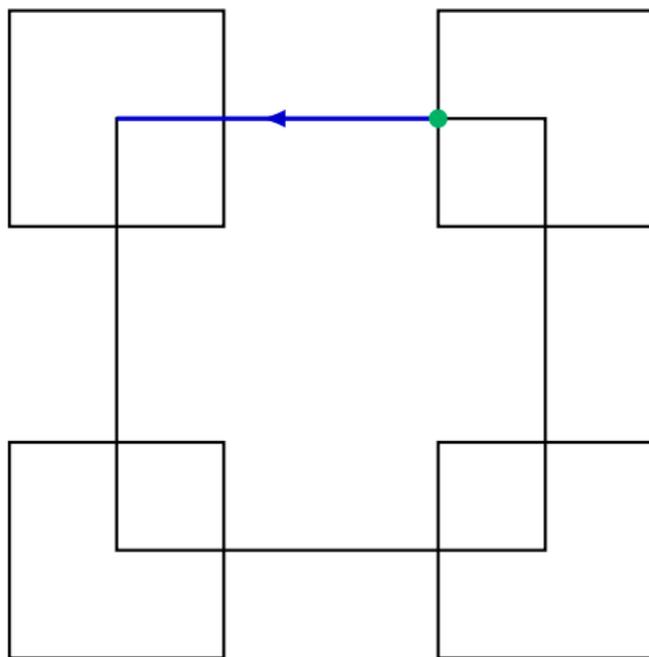
Préparation de l'itinéraire



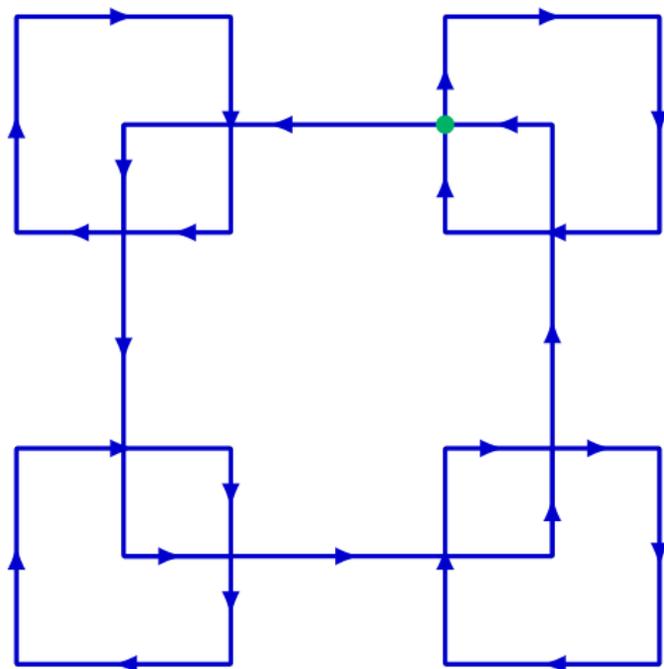
Préparation de l'itinéraire



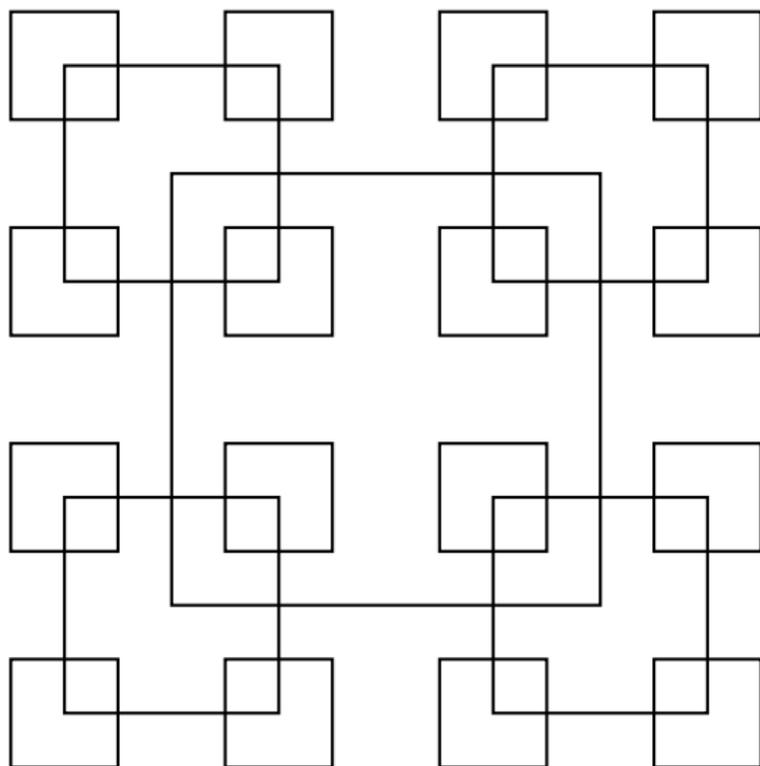
Préparation de l'itinéraire



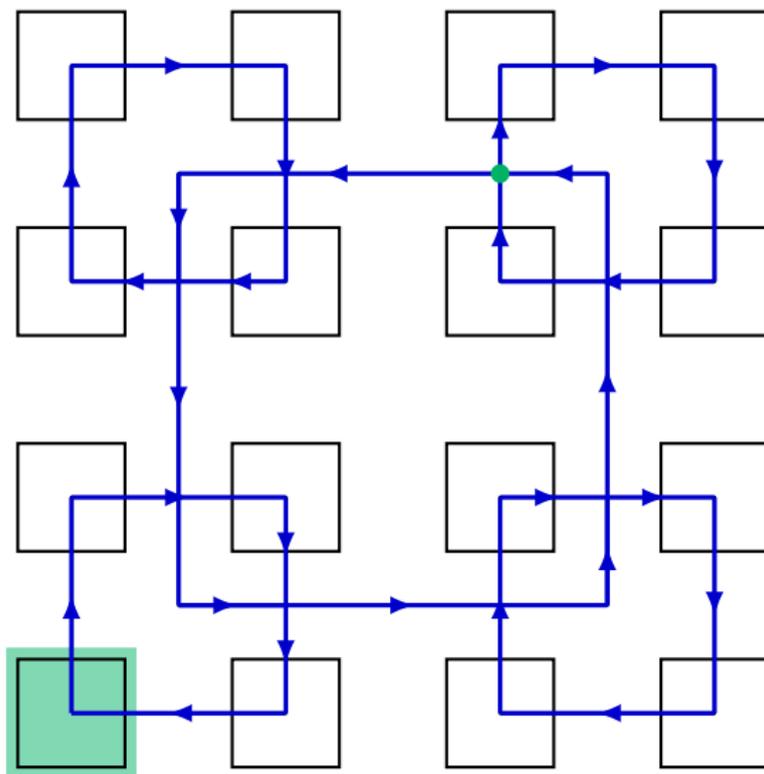
Préparation de l'itinéraire



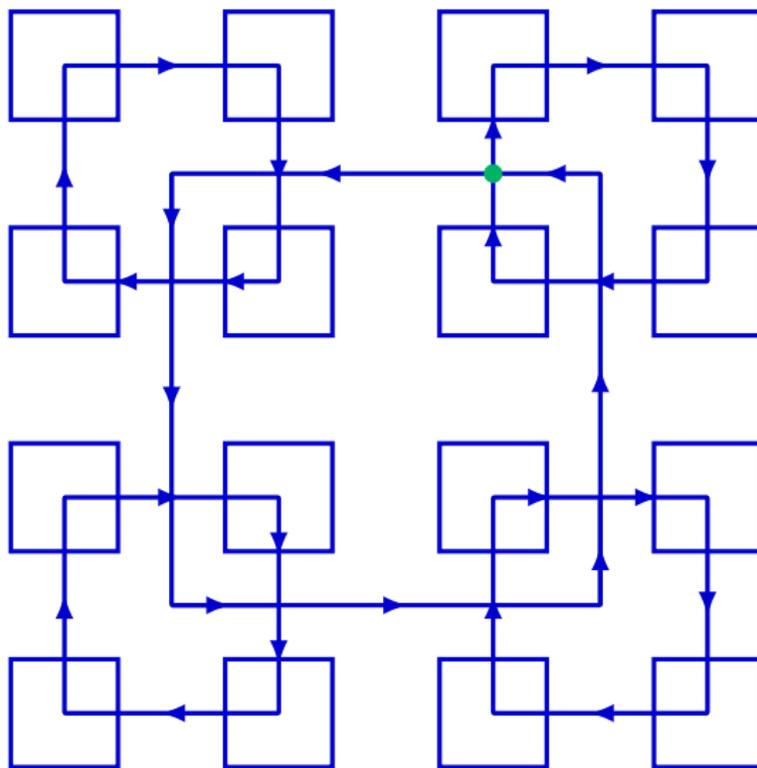
Préparation de l'itinéraire



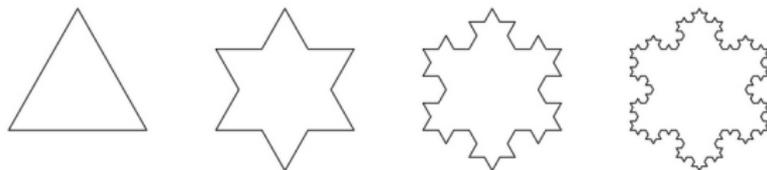
Préparation de l'itinéraire



Préparation de l'itinéraire

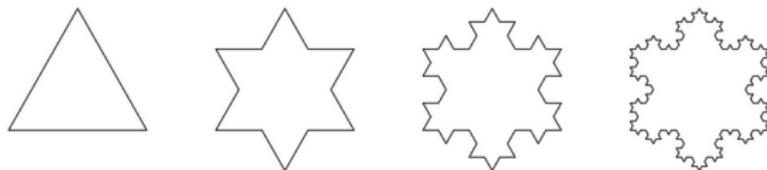


Flocon de Koch :

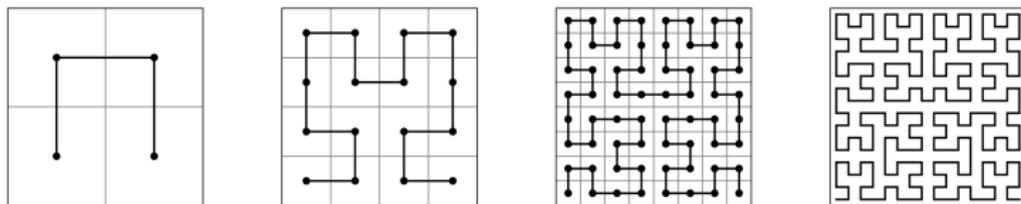


Fractales, substitutions

Flocon de Koch :

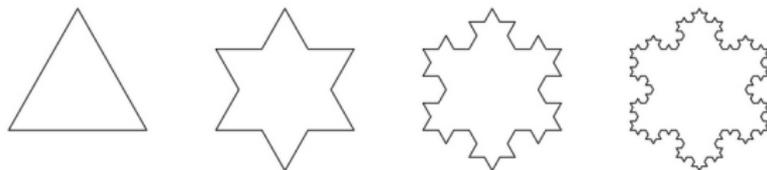


Courbe de Hilbert :

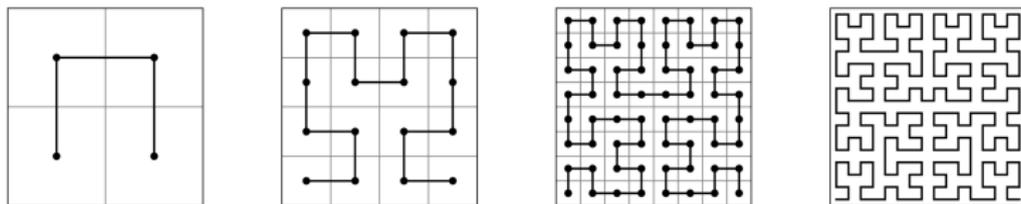


Fractales, substitutions

Flocon de Koch :



Courbe de Hilbert :



Théorème (Kari, 1994)

Il existe un jeu de tuiles de Wang dont les pavages valides reproduisent la courbe de Hilbert.

Le Robinson avec une substitution

4

Franz Gähler

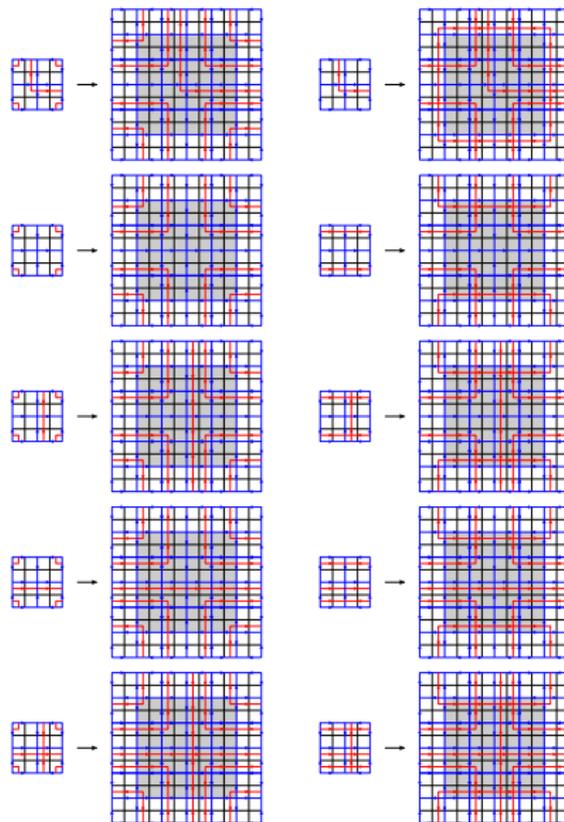
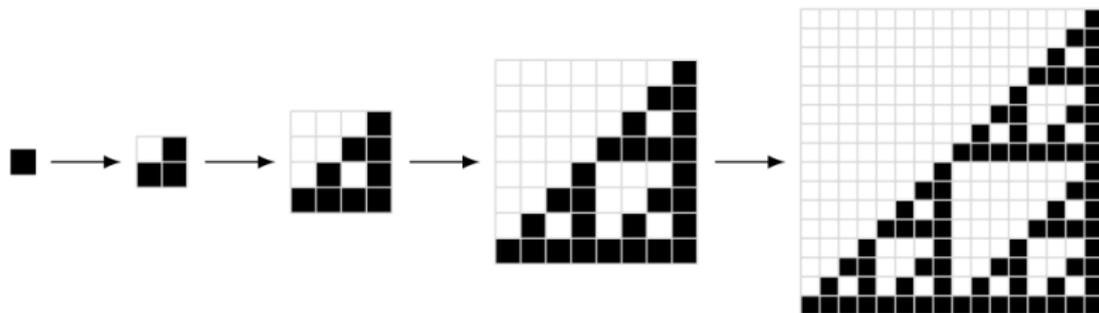
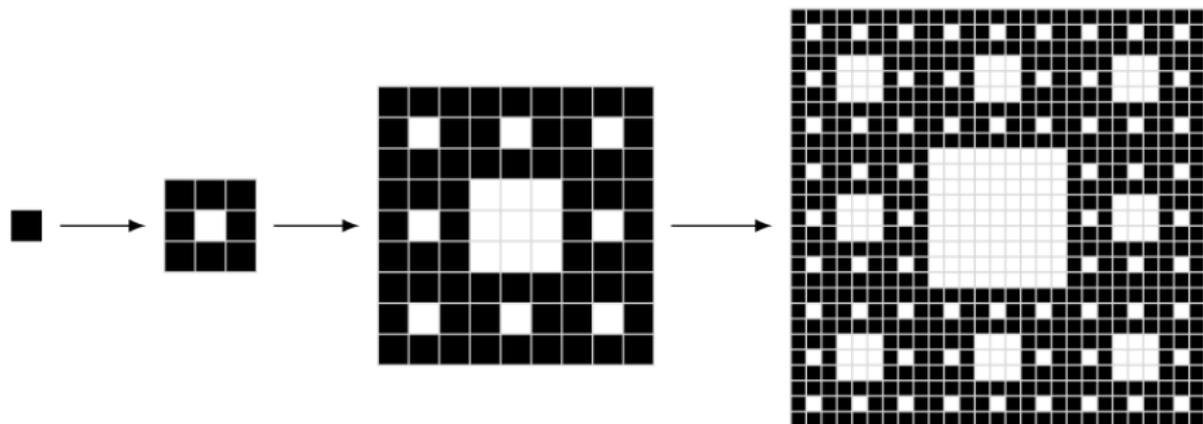


Fig. 3 Overlapping substitution for the Robinson tiling. Each tile is replaced by a 3×3 -patch of tiles. Rotated/reflected tiles are substituted by the corresponding rotated/reflected patches. The inflated tiles cover only the area shaded in gray. The substitutions of neighboring tiles have thus an overlap, on which they agree.

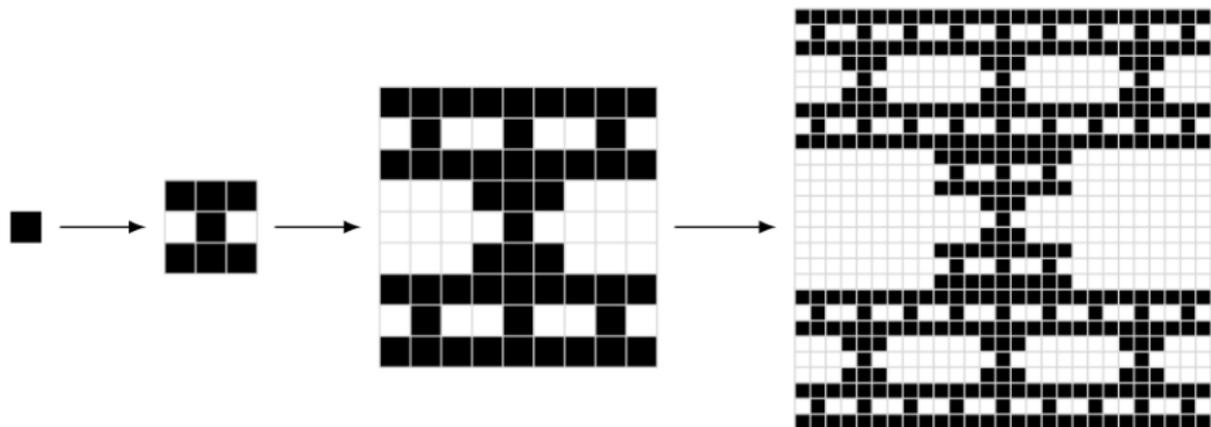
Des substitutions avec des tuiles de Wang



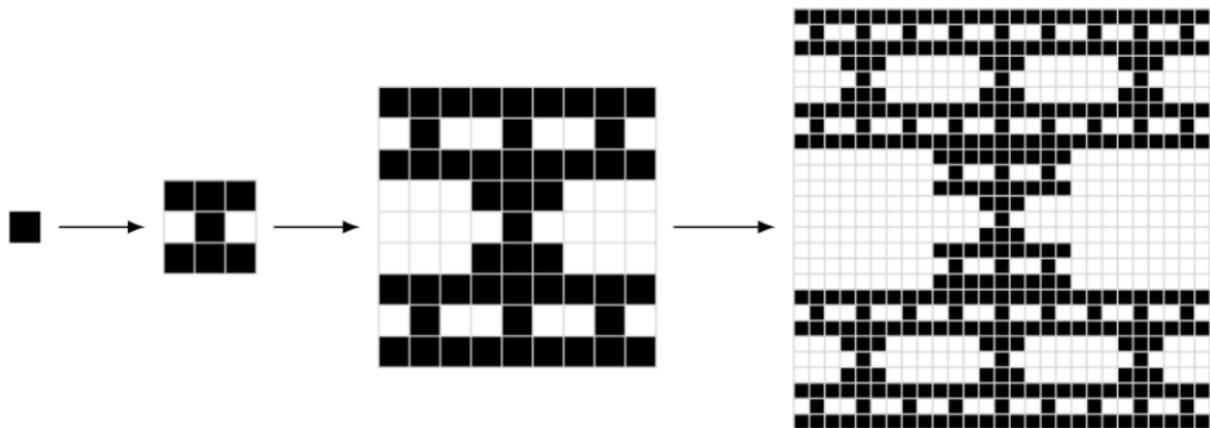
Des substitutions avec des tuiles de Wang



Des substitutions avec des tuiles de Wang



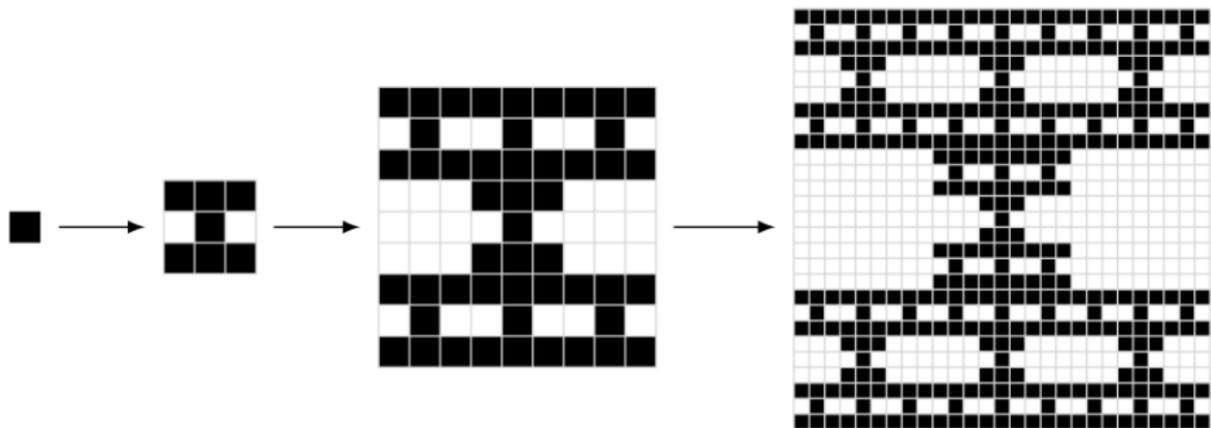
Des substitutions avec des tuiles de Wang



Théorème (Mozes, 1989)

Si un pavage est obtenu par substitution, alors il existe un jeu de tuiles de Wang qui le reproduit.

Des substitutions avec des tuiles de Wang



Théorème (Mozes, 1989)

Si un pavage est obtenu par substitution, alors il existe un jeu de tuiles de Wang qui le reproduit.

⇒ **Beaucoup** de jeux de tuiles apériodiques !

Et maintenant ?

La stratégie de Wang pour résoudre le problème du Domino a échoué. . .

Et maintenant ?

La stratégie de Wang pour résoudre le problème du Domino a échoué. . .

Et si en fait ce problème du Domino était *vraiment* difficile ?

Et maintenant ?

La stratégie de Wang pour résoudre le problème du Domino a échoué. . .

Et si en fait ce problème du Domino était *vraiment* difficile ?

Pour certains problèmes, il est **impossible** de trouver un algorithme qui les résout. On peut montrer qu'il **ne peut pas exister** de tel algorithme.

Et maintenant ?

La stratégie de Wang pour résoudre le problème du Domino a échoué. . .

Et si en fait ce problème du Domino était *vraiment* difficile ?

Pour certains problèmes, il est **impossible** de trouver un algorithme qui les résout. On peut montrer qu'il **ne peut pas exister** de tel algorithme.

Ces problèmes sont dits **indécidables**.

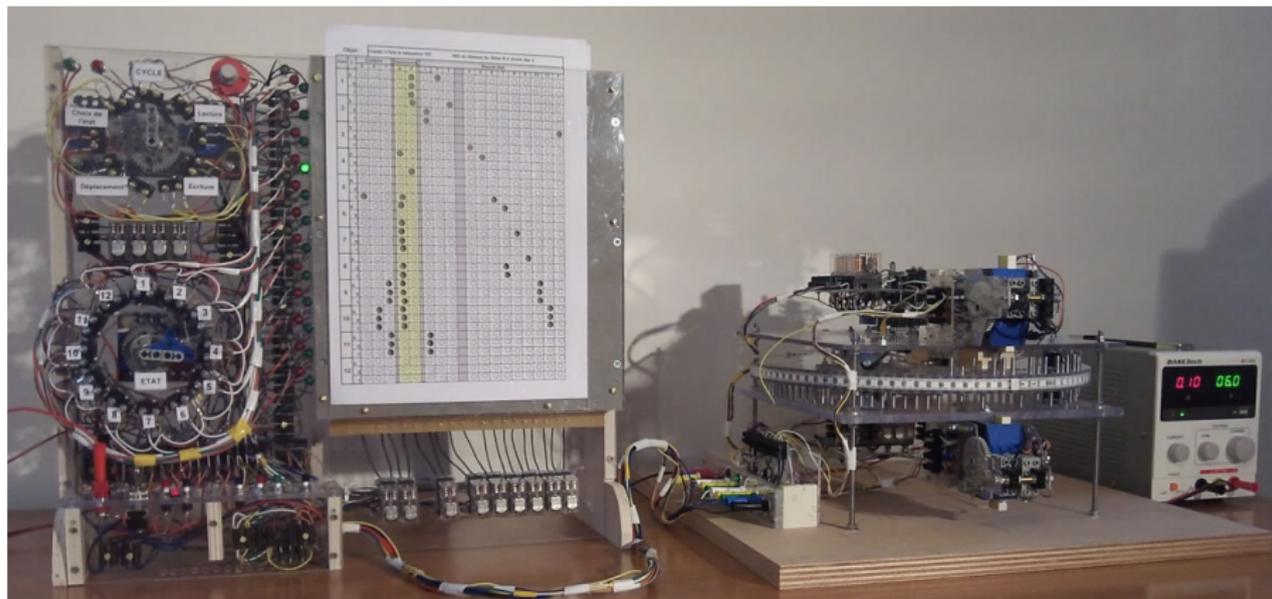
Machines de Turing (I)

Imaginées par Alan Turing en 1936, pour délimiter ce qu'un d'algorithme est capable de faire ou non.



Machines de Turing (I)

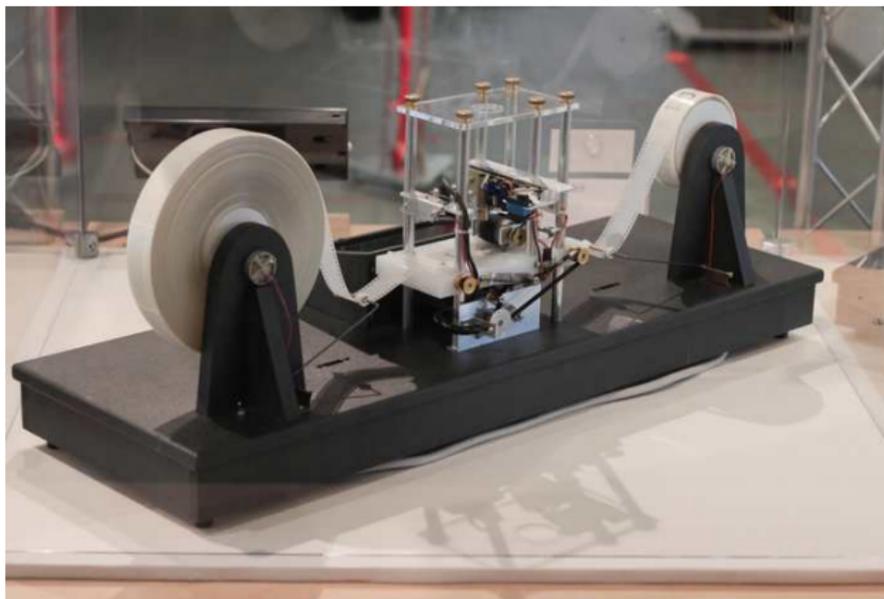
Imaginées par Alan Turing en 1936, pour délimiter ce qu'un d'algorithme est capable de faire ou non.



Modèle théorique : pas de limite ni en temps ni en espace de calcul.

Machines de Turing (I)

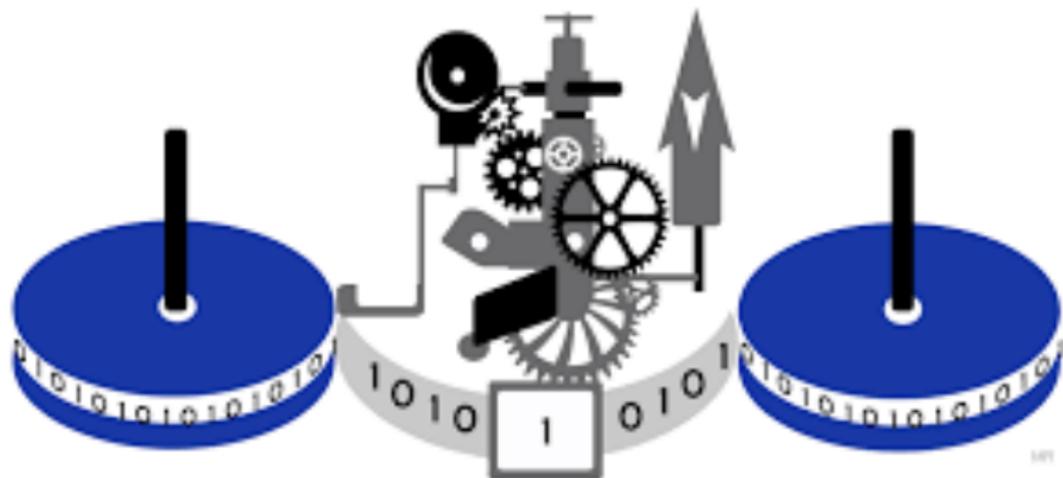
Imaginées par Alan Turing en 1936, pour délimiter ce qu'un d'algorithme est capable de faire ou non.



Modèle théorique : pas de limite ni en temps ni en espace de calcul.

Machines de Turing (I)

Imaginées par Alan Turing en 1936, pour délimiter ce qu'un d'algorithme est capable de faire ou non.



Modèle théorique : pas de limite ni en temps ni en espace de calcul.

Machines de Turing (I)

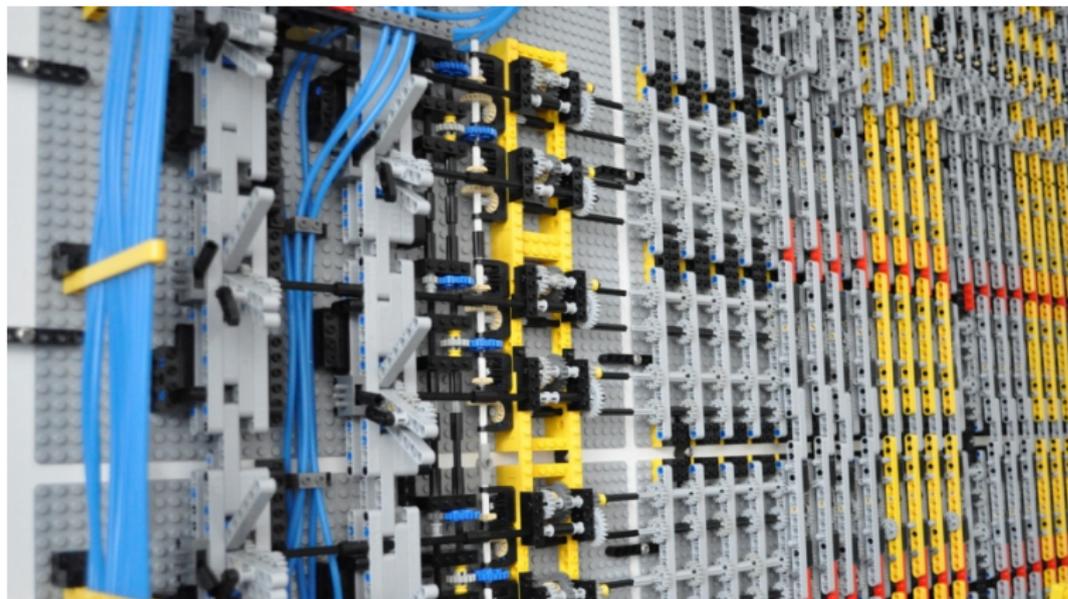
Imaginées par Alan Turing en 1936, pour délimiter ce qu'un d'algorithme est capable de faire ou non.



Modèle théorique : pas de limite ni en temps ni en espace de calcul.

Machines de Turing (I)

Imaginées par Alan Turing en 1936, pour délimiter ce qu'un d'algorithme est capable de faire ou non.

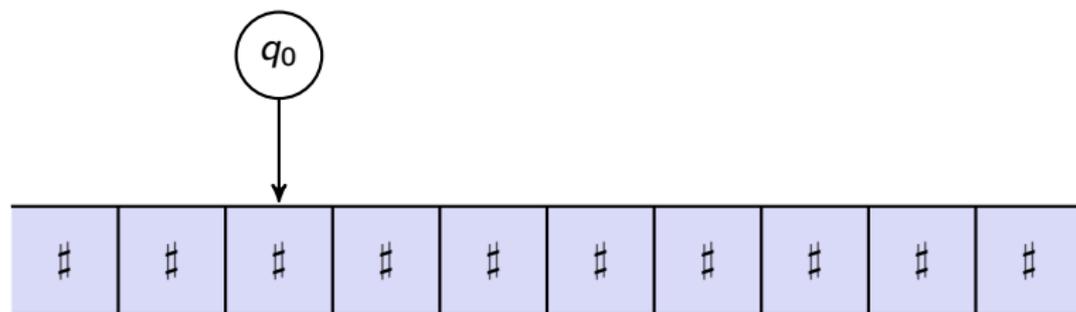


Modèle théorique : pas de limite ni en temps ni en espace de calcul.

Machines de Turing

Un exemple :

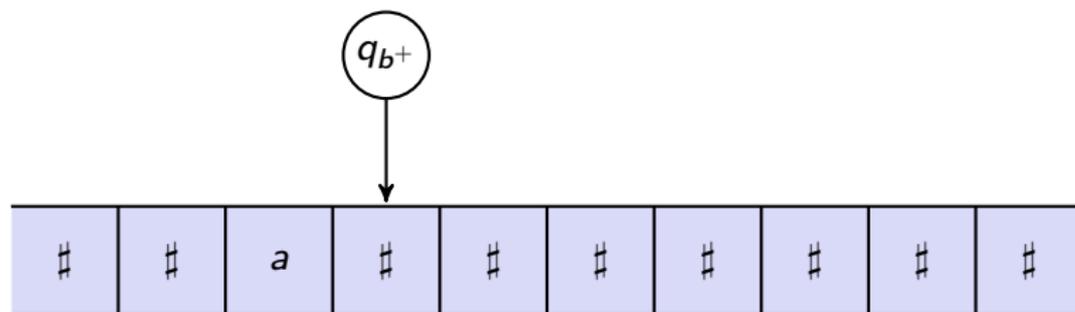
$\delta(q, x)$		Symbole x			
		a	b	\parallel	$\#$
État q	q_0	\perp	\perp	\perp	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	\perp	$(q_{b^{++}}, a, \rightarrow)$	\perp	\perp
	q_{b^+}	\perp	\perp	\perp	$(q_{\parallel}, b, \rightarrow)$
	$q_{b^{++}}$	\perp	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	\perp
	q_{\parallel}	$(q_{a^+}, a, \rightarrow)$	$(q_{\parallel}, b, \leftarrow)$	$(q_{\parallel}, \parallel, \leftarrow)$	$(q_{\parallel}, \parallel, \cdot)$



Machines de Turing

Un exemple :

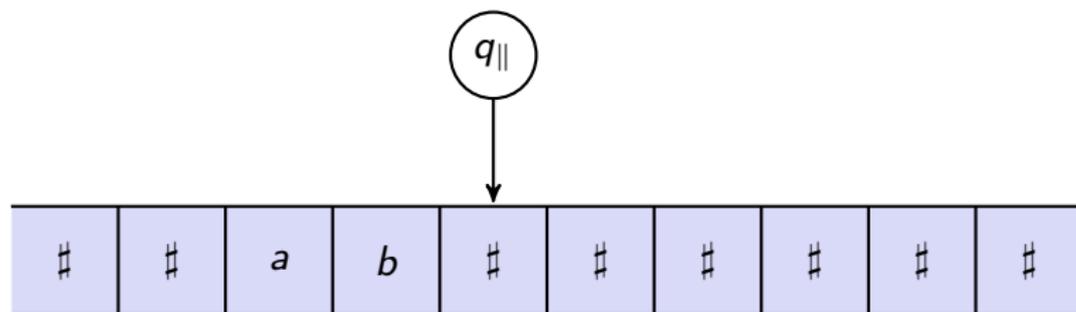
$\delta(q, x)$		Symbole x			
		a	b	\parallel	$\#$
État q	q_0	\perp	\perp	\perp	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	\perp	$(q_{b^{++}}, a, \rightarrow)$	\perp	\perp
	q_{b^+}	\perp	\perp	\perp	$(q_{\parallel}, b, \rightarrow)$
	$q_{b^{++}}$	\perp	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	\perp
	q_{\parallel}	$(q_{a^+}, a, \rightarrow)$	$(q_{\parallel}, b, \leftarrow)$	$(q_{\parallel}, \parallel, \leftarrow)$	$(q_{\parallel}, \parallel, \cdot)$



Machines de Turing

Un exemple :

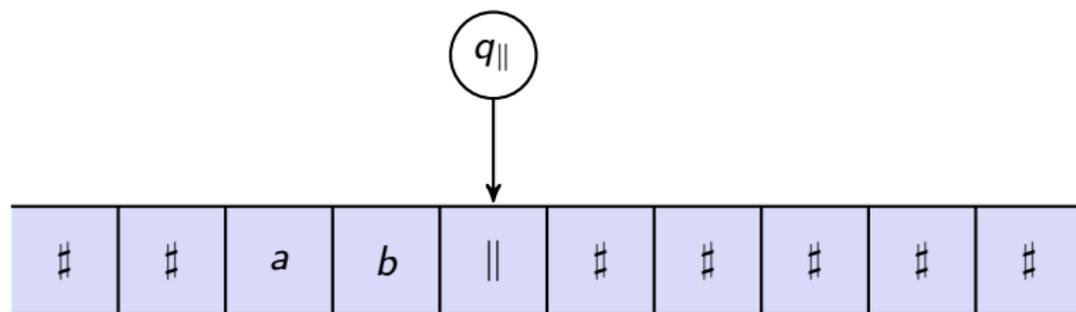
$\delta(q, x)$		Symbole x			
		a	b	\parallel	$\#$
État q	q_0	\perp	\perp	\perp	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	\perp	$(q_{b^{++}}, a, \rightarrow)$	\perp	\perp
	q_{b^+}	\perp	\perp	\perp	$(q_{\parallel}, b, \rightarrow)$
	$q_{b^{++}}$	\perp	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	\perp
	q_{\parallel}	$(q_{a^+}, a, \rightarrow)$	$(q_{\parallel}, b, \leftarrow)$	$(q_{\parallel}, \parallel, \leftarrow)$	$(q_{\parallel}, \parallel, \cdot)$



Machines de Turing

Un exemple :

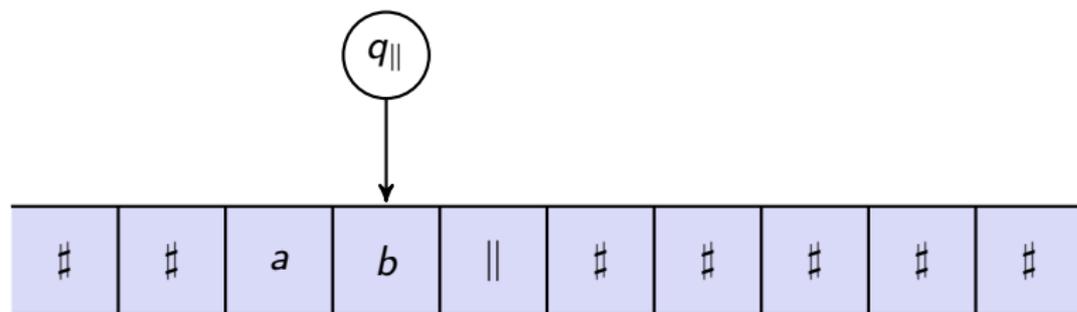
$\delta(q, x)$		Symbole x			
		a	b	\parallel	$\#$
État q	q_0	\perp	\perp	\perp	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	\perp	$(q_{b^{++}}, a, \rightarrow)$	\perp	\perp
	q_{b^+}	\perp	\perp	\perp	$(q_{\parallel}, b, \rightarrow)$
	$q_{b^{++}}$	\perp	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	\perp
	q_{\parallel}	$(q_{a^+}, a, \rightarrow)$	$(q_{\parallel}, b, \leftarrow)$	$(q_{\parallel}, \parallel, \leftarrow)$	$(q_{\parallel}, \parallel, \cdot)$



Machines de Turing

Un exemple :

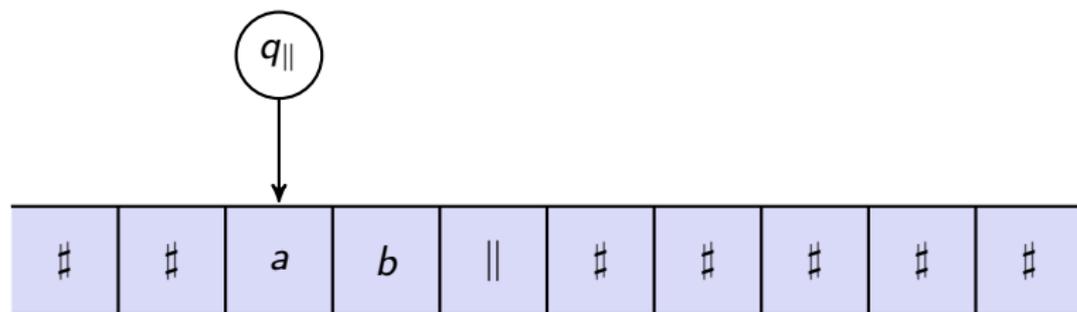
$\delta(q, x)$		Symbole x			
		a	b	\parallel	$\#$
État q	q_0	\perp	\perp	\perp	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	\perp	$(q_{b^{++}}, a, \rightarrow)$	\perp	\perp
	q_{b^+}	\perp	\perp	\perp	$(q_{\parallel}, b, \rightarrow)$
	$q_{b^{++}}$	\perp	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	\perp
	q_{\parallel}	$(q_{a^+}, a, \rightarrow)$	$(q_{\parallel}, b, \leftarrow)$	$(q_{\parallel}, \parallel, \leftarrow)$	$(q_{\parallel}, \parallel, \cdot)$



Machines de Turing

Un exemple :

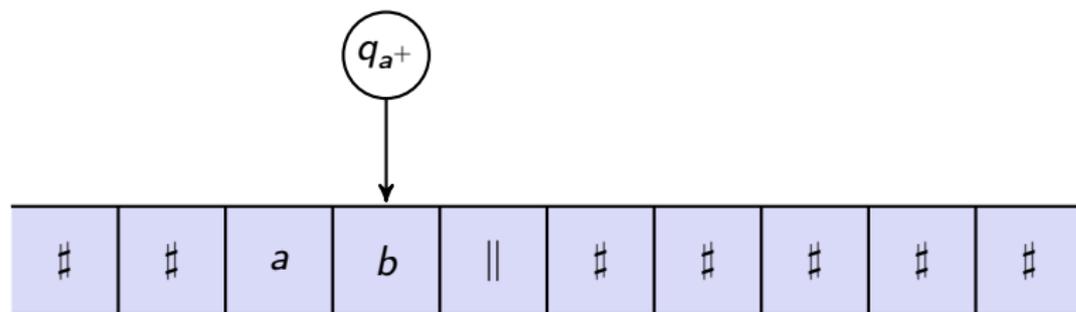
$\delta(q, x)$		Symbole x			
		a	b	\parallel	$\#$
État q	q_0	\perp	\perp	\perp	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	\perp	$(q_{b^{++}}, a, \rightarrow)$	\perp	\perp
	q_{b^+}	\perp	\perp	\perp	$(q_{\parallel}, b, \rightarrow)$
	$q_{b^{++}}$	\perp	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	\perp
	q_{\parallel}	$(q_{a^+}, a, \rightarrow)$	$(q_{\parallel}, b, \leftarrow)$	$(q_{\parallel}, \parallel, \leftarrow)$	$(q_{\parallel}, \parallel, \cdot)$



Machines de Turing

Un exemple :

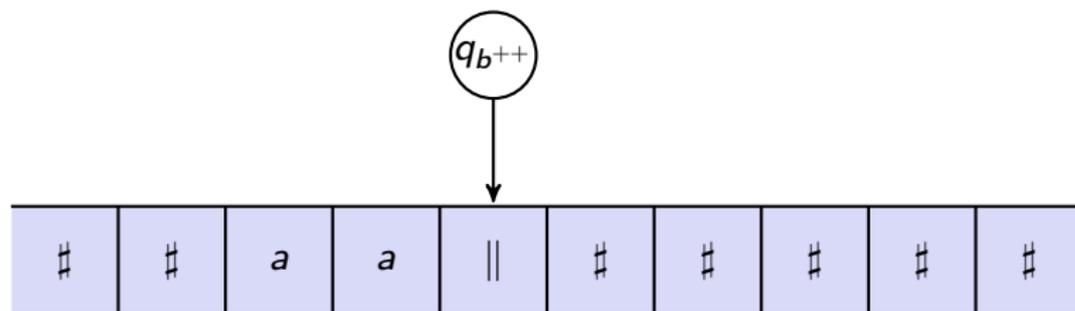
$\delta(q, x)$		Symbole x			
		a	b	\parallel	$\#$
État q	q_0	\perp	\perp	\perp	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	\perp	$(q_{b^{++}}, a, \rightarrow)$	\perp	\perp
	q_{b^+}	\perp	\perp	\perp	$(q_{\parallel}, b, \rightarrow)$
	$q_{b^{++}}$	\perp	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	\perp
	q_{\parallel}	$(q_{a^+}, a, \rightarrow)$	$(q_{\parallel}, b, \leftarrow)$	$(q_{\parallel}, \parallel, \leftarrow)$	$(q_{\parallel}, \parallel, \cdot)$



Machines de Turing

Un exemple :

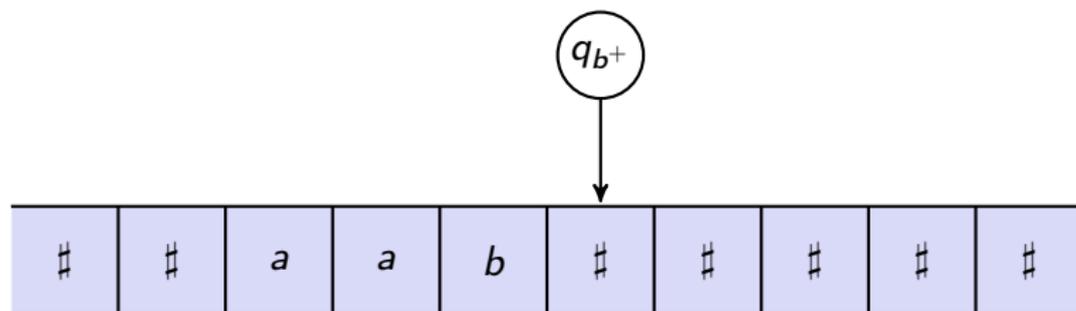
$\delta(q, x)$		Symbole x			
		a	b	\parallel	$\#$
État q	q_0	\perp	\perp	\perp	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	\perp	$(q_{b^{++}}, a, \rightarrow)$	\perp	\perp
	q_{b^+}	\perp	\perp	\perp	$(q_{\parallel}, b, \rightarrow)$
	$q_{b^{++}}$	\perp	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	\perp
	q_{\parallel}	$(q_{a^+}, a, \rightarrow)$	$(q_{\parallel}, b, \leftarrow)$	$(q_{\parallel}, \parallel, \leftarrow)$	$(q_{\parallel}, \parallel, \cdot)$



Machines de Turing

Un exemple :

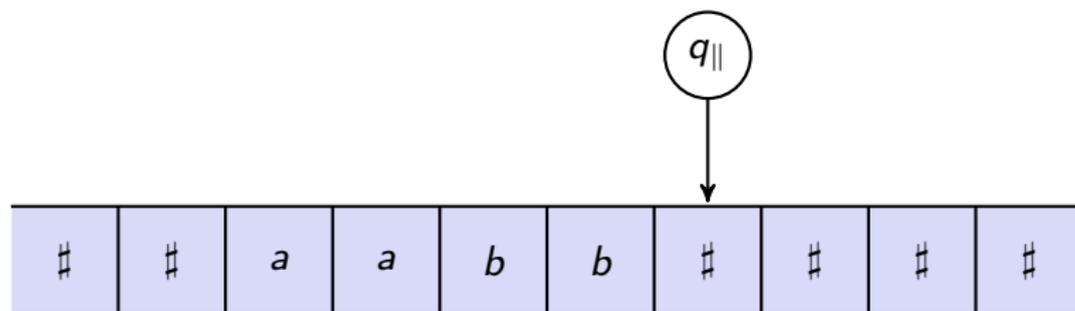
$\delta(q, x)$		Symbole x			
		a	b	\parallel	$\#$
État q	q_0	\perp	\perp	\perp	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	\perp	$(q_{b^{++}}, a, \rightarrow)$	\perp	\perp
	q_{b^+}	\perp	\perp	\perp	$(q_{\parallel}, b, \rightarrow)$
	$q_{b^{++}}$	\perp	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	\perp
	q_{\parallel}	$(q_{a^+}, a, \rightarrow)$	$(q_{\parallel}, b, \leftarrow)$	$(q_{\parallel}, \parallel, \leftarrow)$	$(q_{\parallel}, \parallel, \cdot)$



Machines de Turing

Un exemple :

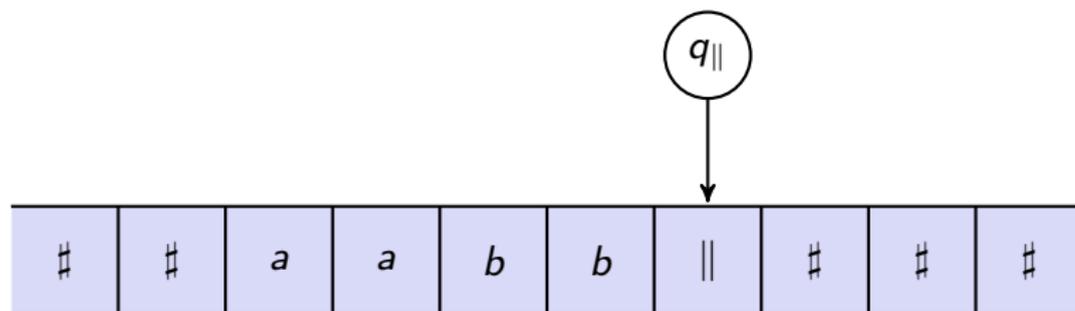
$\delta(q, x)$		Symbole x			
		a	b	\parallel	$\#$
État q	q_0	\perp	\perp	\perp	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	\perp	$(q_{b^{++}}, a, \rightarrow)$	\perp	\perp
	q_{b^+}	\perp	\perp	\perp	$(q_{\parallel}, b, \rightarrow)$
	$q_{b^{++}}$	\perp	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	\perp
	q_{\parallel}	$(q_{a^+}, a, \rightarrow)$	$(q_{\parallel}, b, \leftarrow)$	$(q_{\parallel}, \parallel, \leftarrow)$	$(q_{\parallel}, \parallel, \cdot)$



Machines de Turing

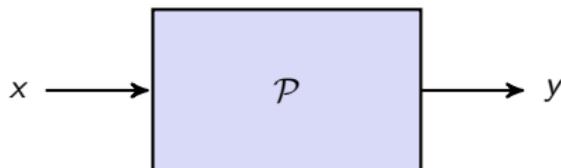
Un exemple :

$\delta(q, x)$		Symbole x			
		a	b	\parallel	$\#$
État q	q_0	\perp	\perp	\perp	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	\perp	$(q_{b^{++}}, a, \rightarrow)$	\perp	\perp
	q_{b^+}	\perp	\perp	\perp	$(q_{\parallel}, b, \rightarrow)$
	$q_{b^{++}}$	\perp	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	\perp
	q_{\parallel}	$(q_{a^+}, a, \rightarrow)$	$(q_{\parallel}, b, \leftarrow)$	$(q_{\parallel}, \parallel, \leftarrow)$	$(q_{\parallel}, \parallel, \cdot)$



Le problème de l'arrêt (I)

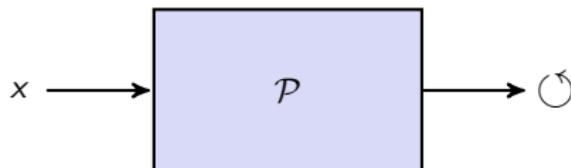
On oublie les machines de Turing, leur ruban, leurs états et leurs transitions. On parle maintenant de **programme**.



Un programme reçoit en **entrée** une information codée sous forme de suite de nombres x , et soit renvoie en **sortie** un résultat lui aussi codé par une suite de nombres y , soit ne s'arrête jamais \emptyset .

Le problème de l'arrêt (I)

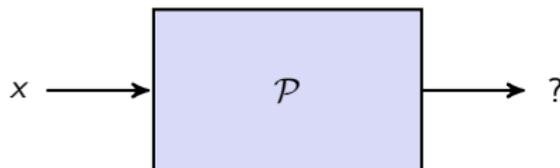
On oublie les machines de Turing, leur ruban, leurs états et leurs transitions. On parle maintenant de **programme**.



Un programme reçoit en **entrée** une information codée sous forme de suite de nombres x , et soit renvoie en **sortie** un résultat lui aussi codé par une suite de nombres y , soit ne s'arrête jamais \circlearrowright .

Le problème de l'arrêt (I)

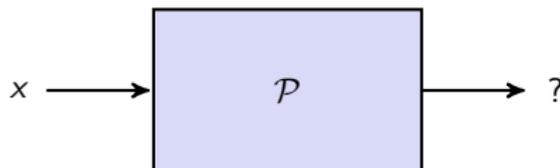
On oublie les machines de Turing, leur ruban, leurs états et leurs transitions. On parle maintenant de **programme**.



Un programme reçoit en **entrée** une information codée sous forme de suite de nombres x , et soit renvoie en **sortie** un résultat lui aussi codé par une suite de nombres y , soit ne s'arrête jamais \emptyset .

Le problème de l'arrêt (I)

On oublie les machines de Turing, leur ruban, leurs états et leurs transitions. On parle maintenant de **programme**.



Un programme reçoit en **entrée** une information codée sous forme de suite de nombres x , et soit renvoie en **sortie** un résultat lui aussi codé par une suite de nombres y , soit ne s'arrête jamais \circlearrowleft .

Remarque : On peut coder sous forme de suite de nombres n'importe quelle information finie. En particulier, x peut coder un entier... **ou un programme!!**

Le problème de l'arrêt (II)

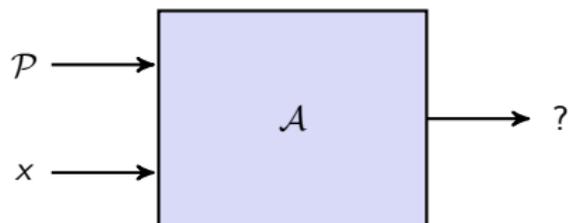
Question

Est-ce qu'il existe un super programme \mathcal{A} qui peut me dire si un programme \mathcal{P} s'arrête sur une entrée x ?

Le problème de l'arrêt (II)

Question

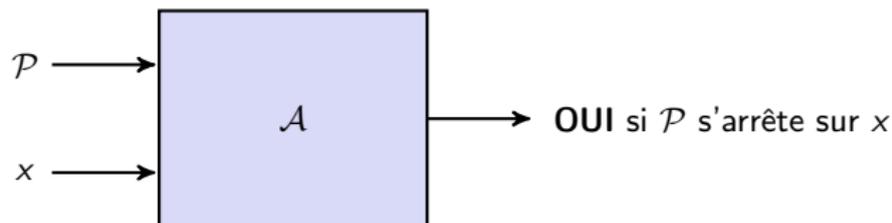
Est-ce qu'il existe un super programme \mathcal{A} qui peut me dire si un programme \mathcal{P} s'arrête sur une entrée x ?



Le problème de l'arrêt (II)

Question

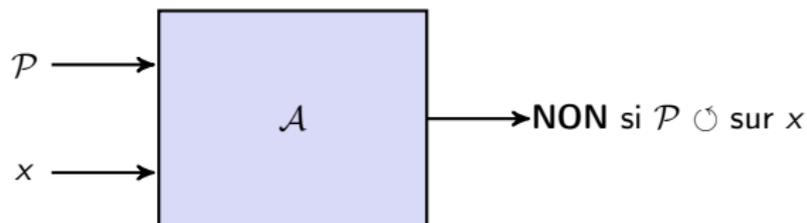
Est-ce qu'il existe un super programme \mathcal{A} qui peut me dire si un programme \mathcal{P} s'arrête sur une entrée x ?



Le problème de l'arrêt (II)

Question

Est-ce qu'il existe un super programme \mathcal{A} qui peut me dire si un programme \mathcal{P} s'arrête sur une entrée x ?

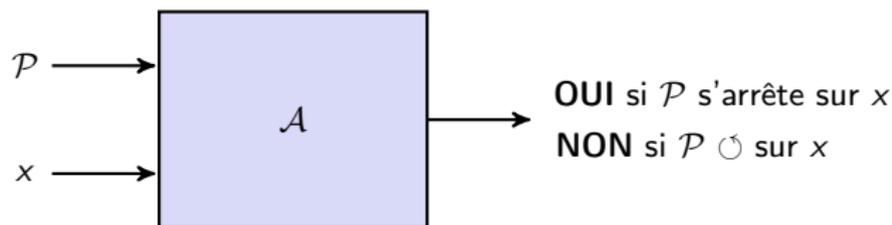


Le problème de l'arrêt (III)

On va montrer que le super-programme \mathcal{A} n'existe pas, en raisonnant par **l'absurde**. On suppose que \mathcal{A} existe,...

Le problème de l'arrêt (III)

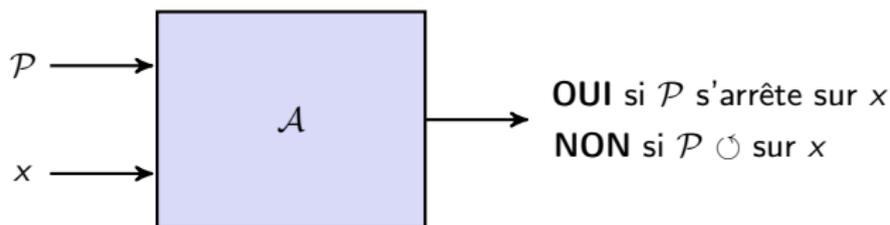
On va montrer que le super-programme \mathcal{A} n'existe pas, en raisonnant par **l'absurde**. On suppose que \mathcal{A} existe,...



et que \mathcal{A} est bien un super-programme, c'est-à-dire qu'il ne se trompe jamais !

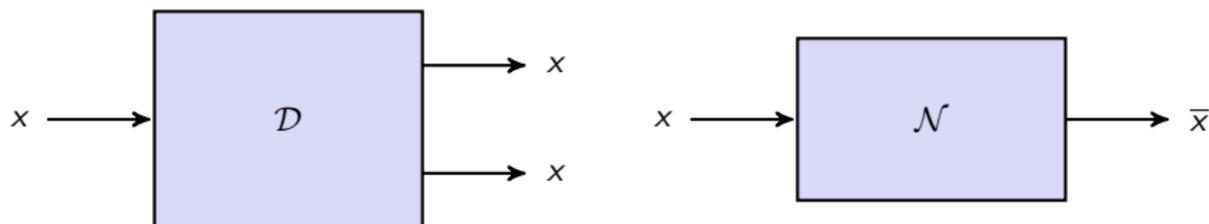
Le problème de l'arrêt (III)

On va montrer que le super-programme \mathcal{A} n'existe pas, en raisonnant par **l'absurde**. On suppose que \mathcal{A} existe,...



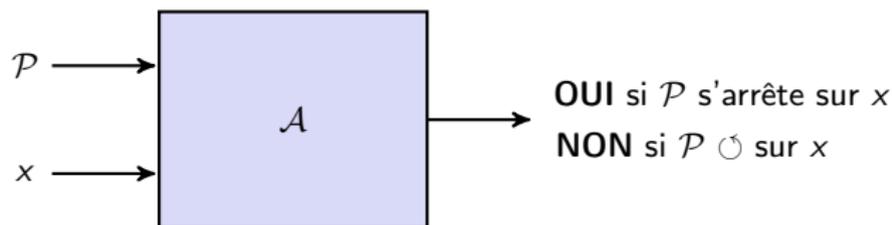
et que \mathcal{A} est bien un super-programme, c'est-à-dire qu'il ne se trompe jamais !

On utilise deux programmes auxiliaires \mathcal{D} et \mathcal{N}



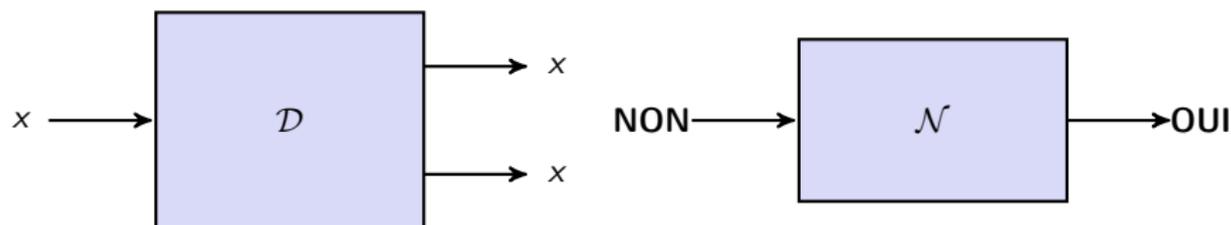
Le problème de l'arrêt (III)

On va montrer que le super-programme \mathcal{A} n'existe pas, en raisonnant par **l'absurde**. On suppose que \mathcal{A} existe,...



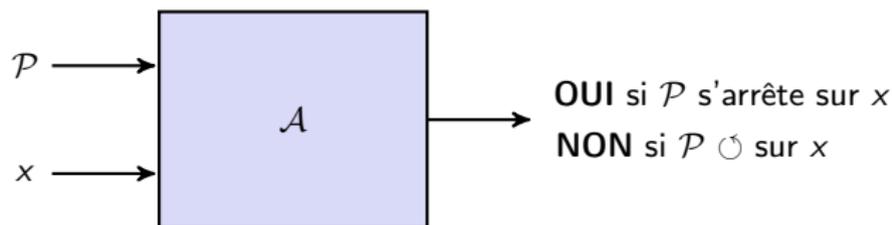
et que \mathcal{A} est bien un super-programme, c'est-à-dire qu'il ne se trompe jamais!

On utilise deux programmes auxiliaires \mathcal{D} et \mathcal{N}



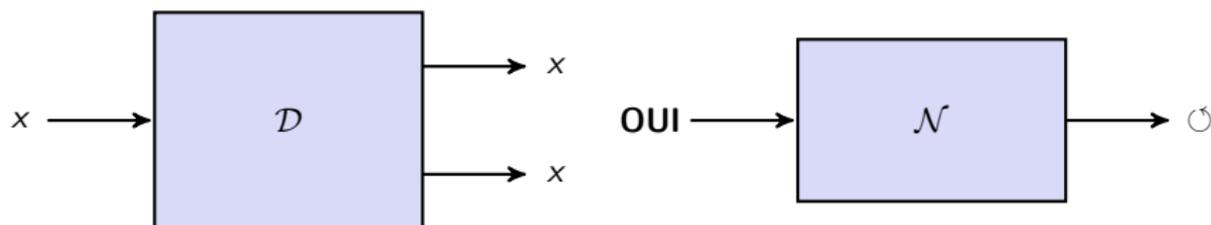
Le problème de l'arrêt (III)

On va montrer que le super-programme \mathcal{A} n'existe pas, en raisonnant par **l'absurde**. On suppose que \mathcal{A} existe,...



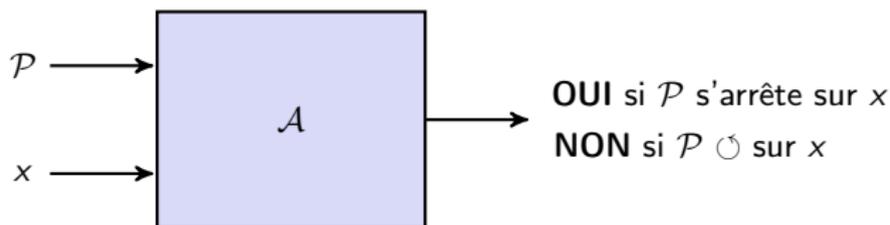
et que \mathcal{A} est bien un super-programme, c'est-à-dire qu'il ne se trompe jamais!

On utilise deux programmes auxiliaires \mathcal{D} et \mathcal{N}



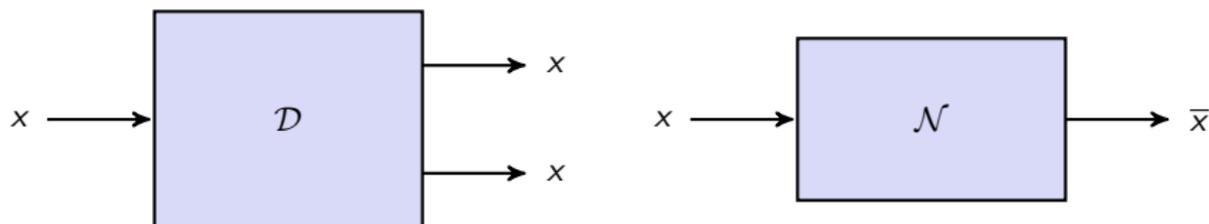
Le problème de l'arrêt (III)

On va montrer que le super-programme \mathcal{A} n'existe pas, en raisonnant par **l'absurde**. On suppose que \mathcal{A} existe,...



et que \mathcal{A} est bien un super-programme, c'est-à-dire qu'il ne se trompe jamais!

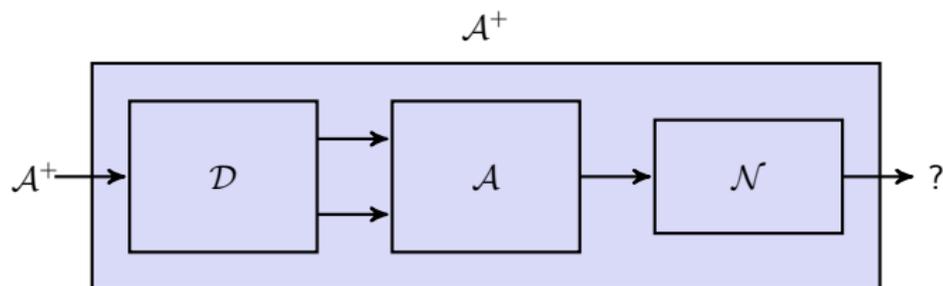
On utilise deux programmes auxiliaires \mathcal{D} et \mathcal{N}



que l'on combine avec \mathcal{A} pour construire le programme \mathcal{A}^+ .

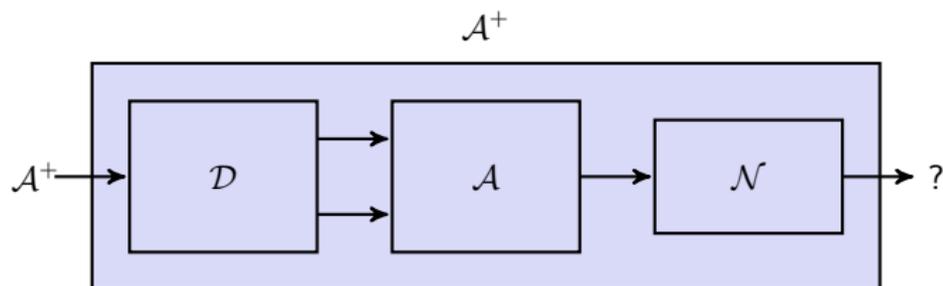
Le problème de l'arrêt (IV)

On donne en entrée \mathcal{A}^+ à notre programme \mathcal{A}^+ . Que se passe-t-il ?



Le problème de l'arrêt (IV)

On donne en entrée \mathcal{A}^+ à notre programme \mathcal{A}^+ . Que se passe-t-il ?

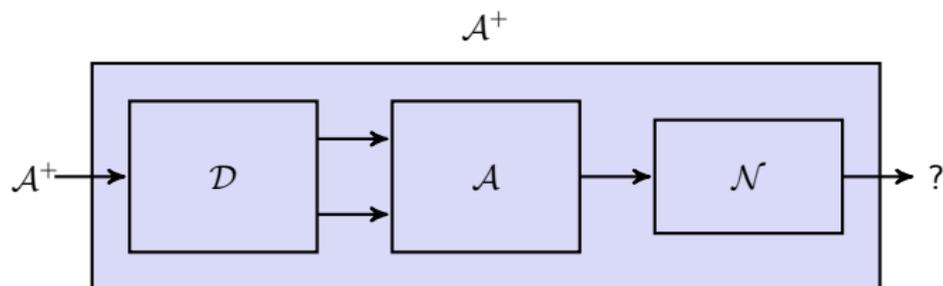


Conclusion :

- Si le programme \mathcal{A}^+ répond **OUI**, cela signifie que le programme \mathcal{A}^+ boucle.
- Mais si le programme \mathcal{A}^+ boucle, cela signifie le programme \mathcal{A}^+ répond **OUI**.
- ...

Le problème de l'arrêt (IV)

On donne en entrée \mathcal{A}^+ à notre programme \mathcal{A}^+ . Que se passe-t-il ?



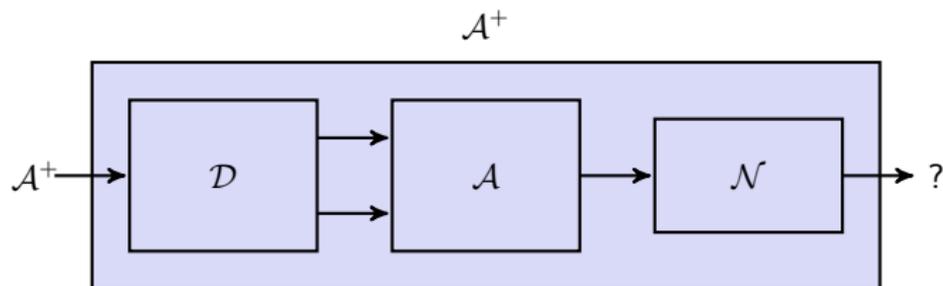
Conclusion :

- Si le programme \mathcal{A}^+ répond **OUI**, cela signifie que le programme \mathcal{A}^+ boucle.
- Mais si le programme \mathcal{A}^+ boucle, cela signifie le programme \mathcal{A}^+ répond **OUI**.
- ...

⇒ contradiction !! le super-programme \mathcal{A} ne peut pas exister.

Le problème de l'arrêt (IV)

On donne en entrée \mathcal{A}^+ à notre programme \mathcal{A}^+ . Que se passe-t-il ?



Conclusion :

- Si le programme \mathcal{A}^+ répond **OUI**, cela signifie que le programme \mathcal{A}^+ boucle.
- Mais si le programme \mathcal{A}^+ boucle, cela signifie le programme \mathcal{A}^+ répond **OUI**.
- ...

\Rightarrow contradiction !! le super-programme \mathcal{A} ne peut pas exister.

Théorème (Turing, 1936)

Il n'existe pas de programme/d'algorithme qui résout le problème de l'arrêt. On dit que le problème de l'arrêt est **indécidable**.

Théorème

Il n'existe pas de programme/d'algorithme qui décide si une machine de Turing machine \mathcal{M} s'arrête sur l'entrée vide.

Comment montrer qu'un problème est indécidable ?

Valable pour les problèmes de **décision** (ceux pour lesquels la réponse ne peut être que **OUI** ou **NON**).

Un problème est **indécidable** s'il ne peut pas exister d'algorithme/programme qui le résout (qui donne la bonne réponse quelle que soit l'entrée).

Comment montrer qu'un problème est indécidable ?

Valable pour les problèmes de **décision** (ceux pour lesquels la réponse ne peut être que **OUI** ou **NON**).

Un problème est **indécidable** s'il ne peut pas exister d'algorithme/programme qui le résout (qui donne la bonne réponse quelle que soit l'entrée).

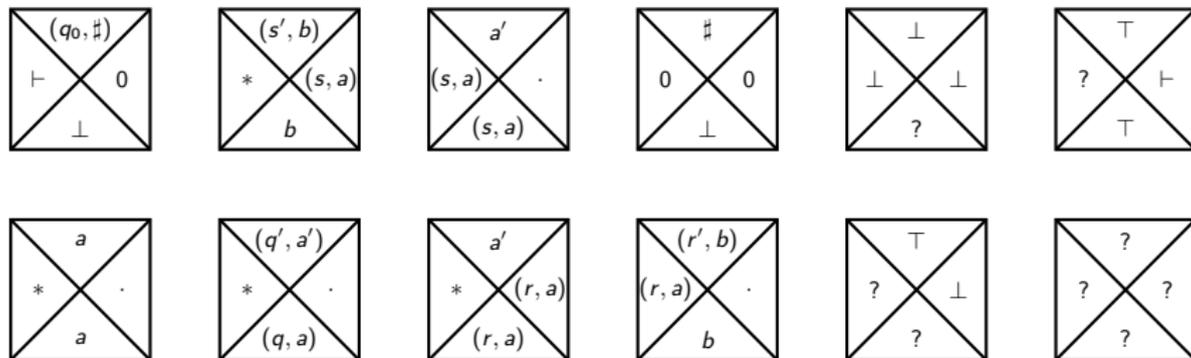
Pour montrer qu'un problème P est indécidable, une stratégie de démonstration est :

- 1 On suppose que l'on sait résoudre le problème P à l'aide d'un programme \mathcal{P} .
- 2 On transforme une entrée du problème de l'arrêt en entrée du problème P . La transformation est telle que les réponses **OUI/NON** restent inchangées.
- 3 Pour résoudre le problème de l'arrêt, il suffirait donc de transformer l'entrée et d'appliquer \mathcal{P} .
- 4 On conclut : comme le programme \mathcal{P} ne peut pas exister, alors P est indécidable.

Machines de Turing et tuiles de Wang

On peut coder des calculs de machine de Turing dans des tuiles de Wang :

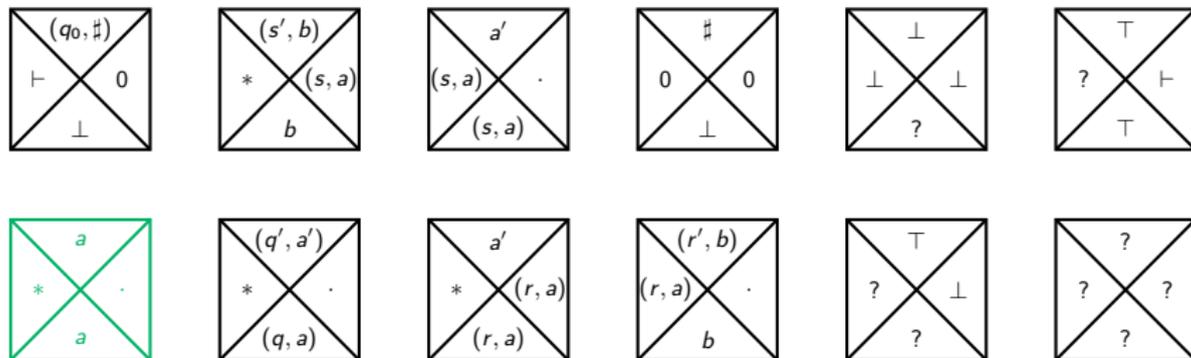
- ▶ pas de tête de calcul
- ▶ configuration initiale $(\infty \# \infty, q_0)$
- ▶ $\delta(q, a) = (q', a', \cdot)$
- ▶ $\delta(r, a) = (r', a', \rightarrow)$
- ▶ $\delta(s, a) = (s', a', \leftarrow)$



Machines de Turing et tuiles de Wang

On peut coder des calculs de machine de Turing dans des tuiles de Wang :

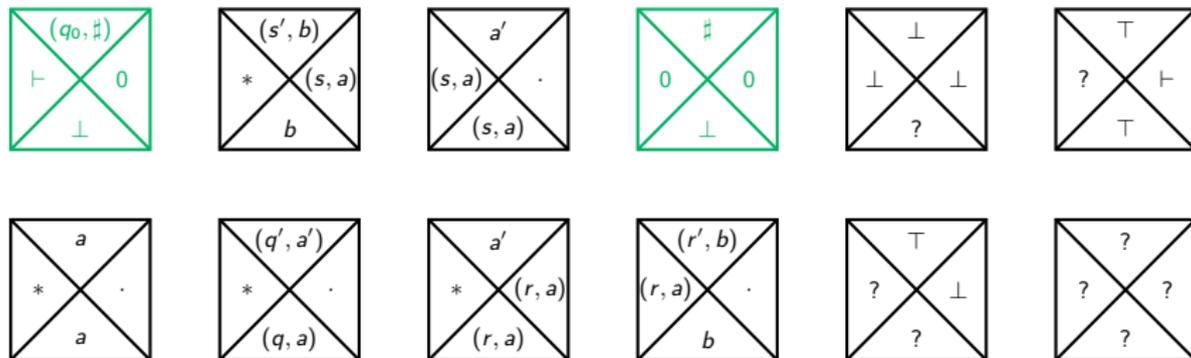
- ▶ pas de tête de calcul
- ▶ configuration initiale $(\infty \# \infty, q_0)$
- ▶ $\delta(q, a) = (q', a', \cdot)$
- ▶ $\delta(r, a) = (r', a', \rightarrow)$
- ▶ $\delta(s, a) = (s', a', \leftarrow)$



Machines de Turing et tuiles de Wang

On peut coder des calculs de machine de Turing dans des tuiles de Wang :

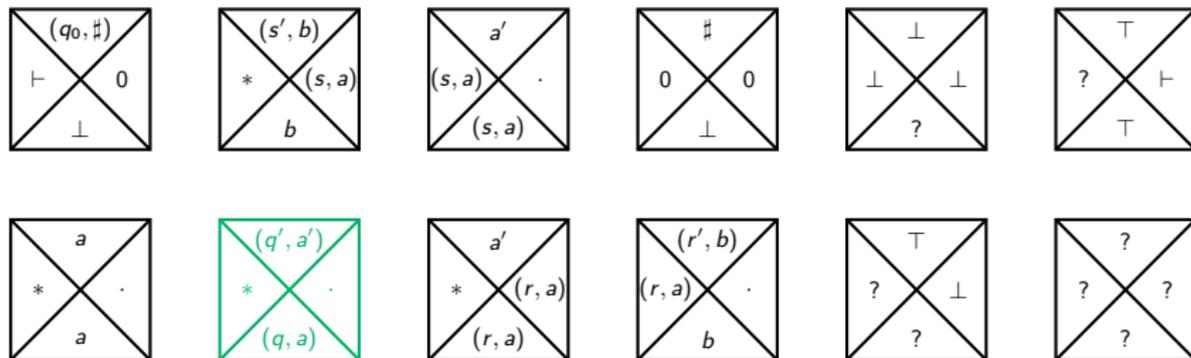
- ▶ pas de tête de calcul
- ▶ configuration initiale (${}^{\infty}\#\infty, q_0$)
- ▶ $\delta(q, a) = (q', a', \cdot)$
- ▶ $\delta(r, a) = (r', a', \rightarrow)$
- ▶ $\delta(s, a) = (s', a', \leftarrow)$



Machines de Turing et tuiles de Wang

On peut coder des calculs de machine de Turing dans des tuiles de Wang :

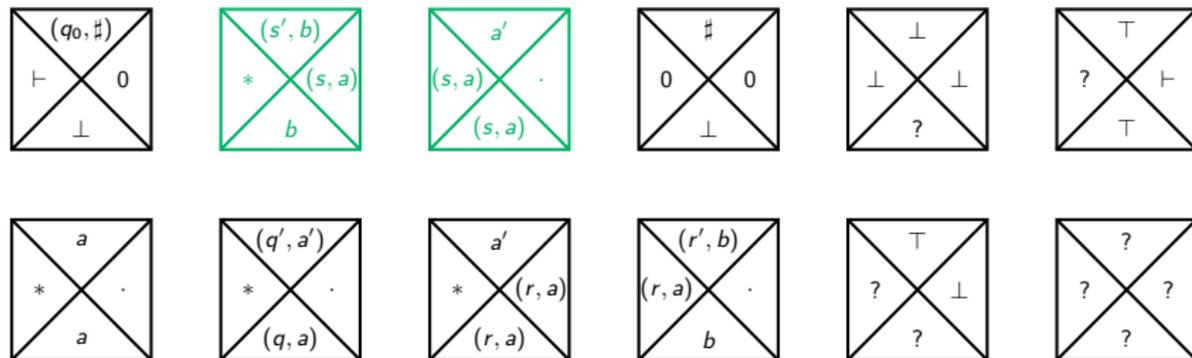
- ▶ pas de tête de calcul
- ▶ configuration initiale $(\infty \# \infty, q_0)$
- ▶ $\delta(q, a) = (q', a', \cdot)$
- ▶ $\delta(r, a) = (r', a', \rightarrow)$
- ▶ $\delta(s, a) = (s', a', \leftarrow)$



Machines de Turing et tuiles de Wang

On peut coder des calculs de machine de Turing dans des tuiles de Wang :

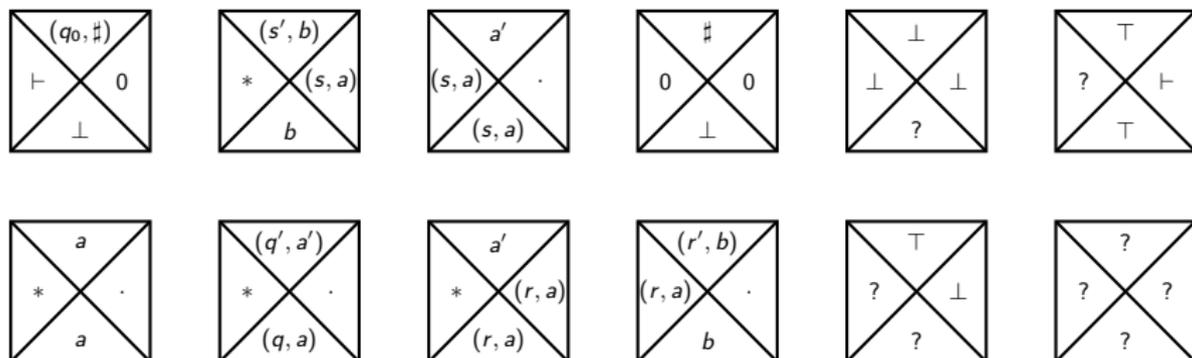
- ▶ pas de tête de calcul
- ▶ configuration initiale $(\infty \# \infty, q_0)$
- ▶ $\delta(q, a) = (q', a', \cdot)$
- ▶ $\delta(r, a) = (r', a', \rightarrow)$
- ▶ $\delta(s, a) = (s', a', \leftarrow)$



Machines de Turing et tuiles de Wang

On peut coder des calculs de machine de Turing dans des tuiles de Wang :

- ▶ pas de tête de calcul
- ▶ configuration initiale $(\infty \# \infty, q_0)$
- ▶ $\delta(q, a) = (q', a', \cdot)$
- ▶ $\delta(r, a) = (r', a', \rightarrow)$
- ▶ $\delta(s, a) = (s', a', \leftarrow)$



Ce qu'on veut : τ admet un pavage ssi \mathcal{M} ne s'arrête pas sur l'entrée vide.

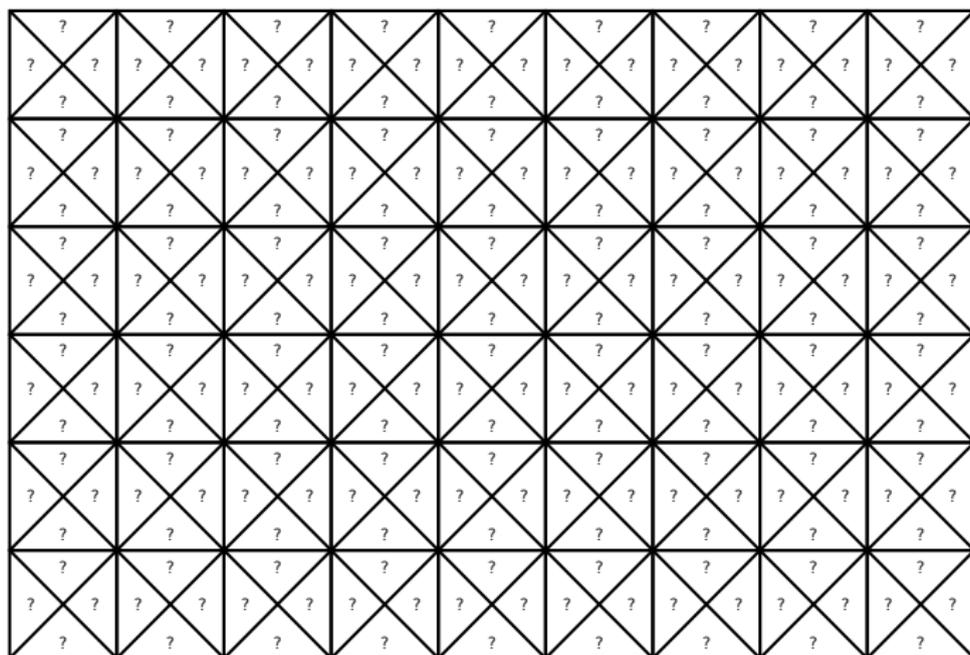
Pavages possibles ?

On **supprime** les tuiles sur lesquelles apparaît un état d'acceptation q_a .

Pavages possibles ?

On **supprime** les tuiles sur lesquelles apparaît un état d'acceptation q_a .

Si \mathcal{M} ne s'arrête pas sur l'entrée vide, il existe un pavage. Mais...



Le problème du Domino à origine fixée

Ce que l'on n'a pas montré :

Pas encore un théorème

Le problème du Domino est indécidable.

Le problème du Domino à origine fixée

Ce que l'on n'a pas montré :

Pas encore un théorème

Le problème du Domino est indécidable.

Ce que l'on a montré :

Théorème (Kahr, Moore & Wang 1962, Büchi 1962)

Le problème du Domino à origine fixée est indécidable.

avec

Problème du Domino à origine fixée

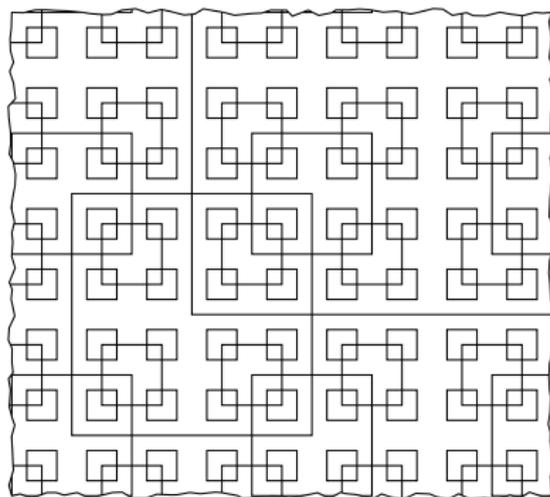
Input : Un jeu fini de tuiles de Wang τ , une tuile $t \in \tau$

Output : **Oui** s'il existe un pavage valide par τ avec la tuile t à l'origine, **Non** sinon.

Et le problème du Domino ?

Solution

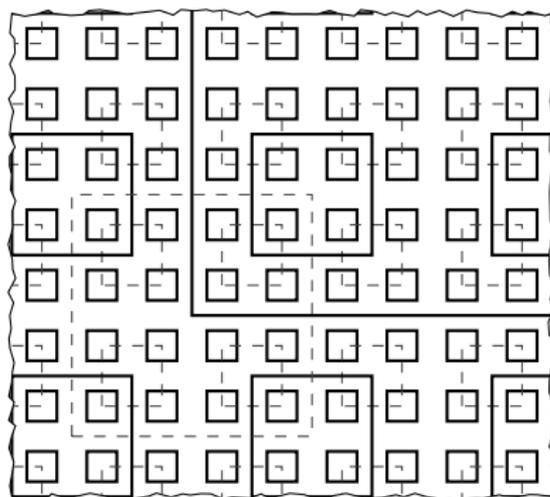
Un peu de machines de Turing, un peu de pavage de Robinson...



Et le problème du Domino ?

Solution

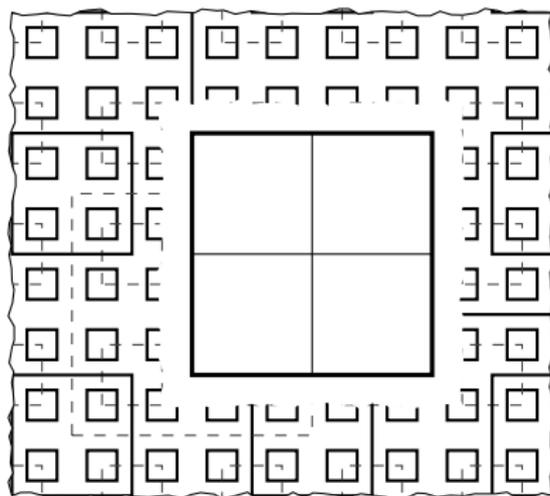
Un peu de machines de Turing, un peu de pavage de Robinson...



Et le problème du Domino ?

Solution

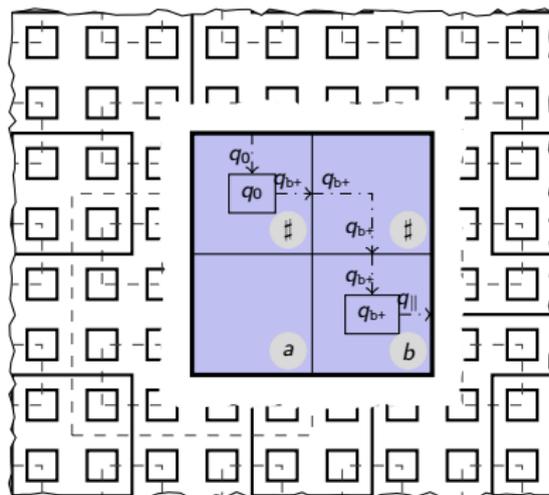
Un peu de machines de Turing, un peu de pavage de Robinson...



Et le problème du Domino ?

Solution

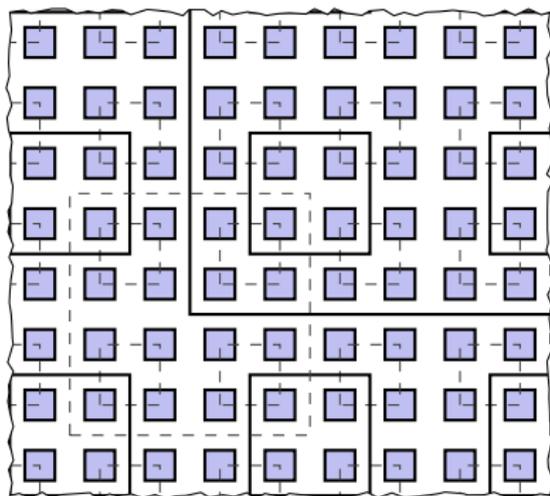
Un peu de machines de Turing, un peu de pavage de Robinson...



Et le problème du Domino ?

Solution

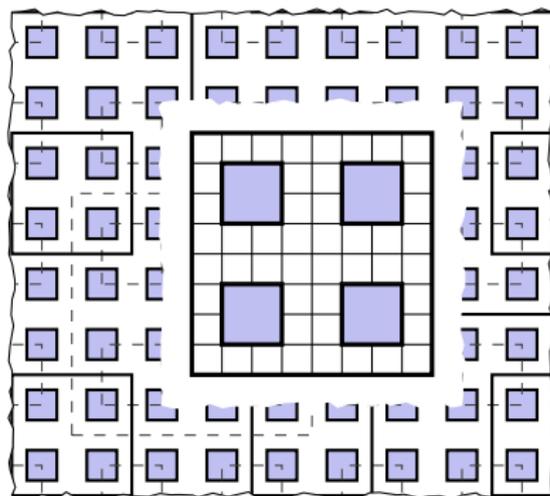
Un peu de machines de Turing, un peu de pavage de Robinson...



Et le problème du Domino ?

Solution

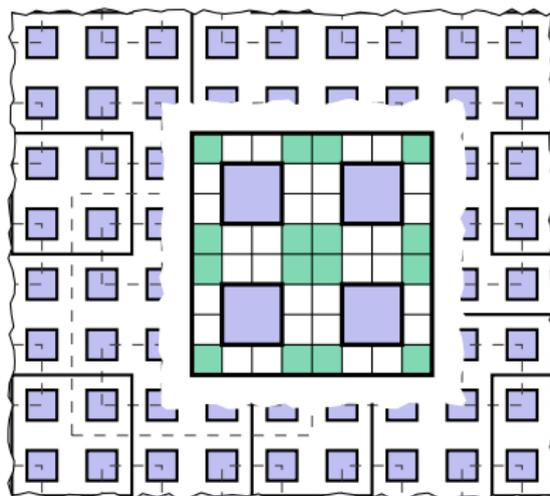
Un peu de machines de Turing, un peu de pavage de Robinson...



Et le problème du Domino ?

Solution

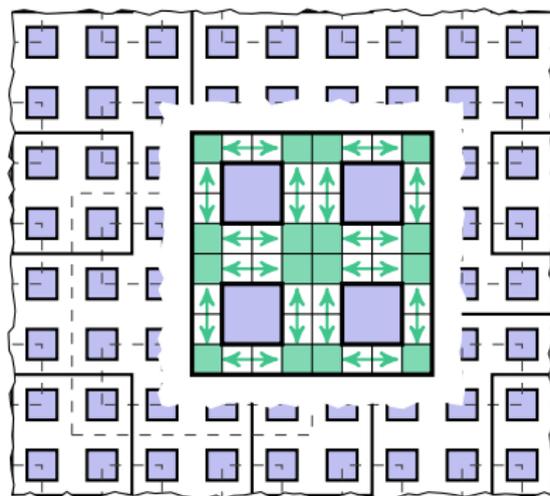
Un peu de machines de Turing, un peu de pavage de Robinson...



Et le problème du Domino ?

Solution

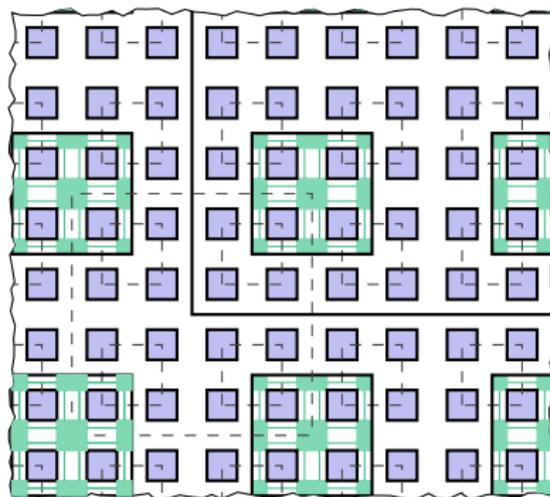
Un peu de machines de Turing, un peu de pavage de Robinson...



Et le problème du Domino ?

Solution

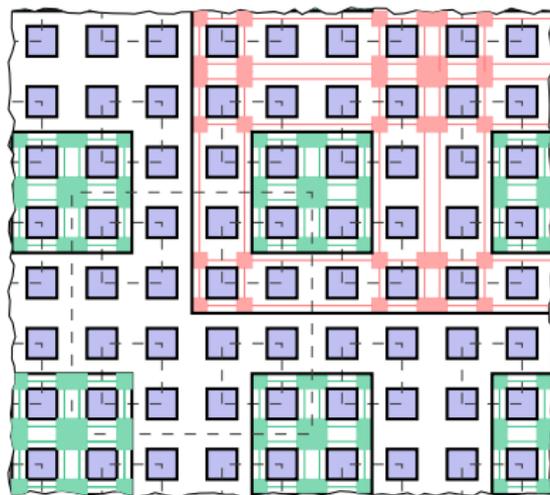
Un peu de machines de Turing, un peu de pavage de Robinson...



Et le problème du Domino ?

Solution

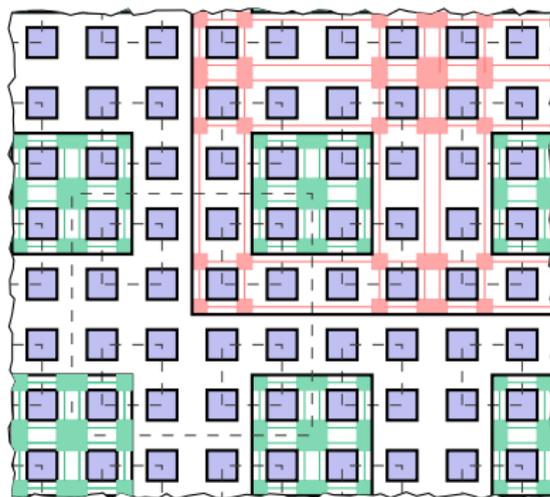
Un peu de machines de Turing, un peu de pavage de Robinson...



Et le problème du Domino ?

Solution

Un peu de machines de Turing, un peu de pavage de Robinson...



Théorème (Berger 1966, Robinson 1971)

Le problème du Domino est indécidable.

Merci !!



DU TROP FORT AU CARRÉMENT FAIBLE :

LES AXIOMES DES MATHÉMATIQUES ET LA MATHÉMATIQUE DES AXIOMES

Itai Ben Yaacov

Lundi 13/02/2017 à 20h

à l'Université Lyon 1, La Doua
Bât. des mathématiques



SOIRÉES MATHÉMATIQUES DE LYON

TRAILERS ET VIDÉOS
SUR
WWW.MMI-LYON.FR

Public : Étudiants post-bac
Entrée libre et gratuite



T 04 72 43 11 80
contact@mmi-lyon.fr

MMI portée par



Tutelles



Partenaires SML

