Limites des langages rationnels

Est ce que tout langage est rationnel?
Existe-t-il des langages non rationnels?

Hypothèse: tout est rationnel

- Les langages rationnels sont reconnaissables : ils sont en bijection avec les automates.
- Un automate peut être décrit par un texte
- Ce texte correspond (codage ASCII) à un nombre binaire
- D'où que les langages reconnaissables sont en bijection avec $\mathbb{N}$.

Hypothèse: tout est rationnel

- On énumère les automates finis sur un alphabet à une lettre et on les ordonne dans une liste :
  - A_0 le premier automate
  - A_1 le second automate
  - …
- On énumère les langages rationnels sur un alphabet à un seul lettre :
  - L_0 reconnu par A_0
  - L_1 reconnu par A_1
  - …

Tableau mots/langages

	L_0	L_1	L_2	L_3
w_0	O	O	N	N
w_1	N	N	O	N
w_2	O	N	O	N
w_3	N	O	O	N

$T[i,j]$=Oui si $w_i \in L_j$
Non sinon

$w_i \in D \Leftrightarrow w_i \notin L_i$
D n'est pas dans T

D n'est pas dans T

- Si D était dans le tableau, il existerait j tel que $D=L_j$.

- Puisque $D=L_j$, si
  - $w_j \in L_j$ alors, par définition de D, $w_j \notin D \Rightarrow$ contradiction
  - $w_j \notin L_j$ alors, par définition de D, $w_j \in D \Rightarrow$ contradiction

- l'ensemble des langages est infini, mais non dénombrable

D n'est pas dans T

- Il existe des langages qui ne sont pas rationnels

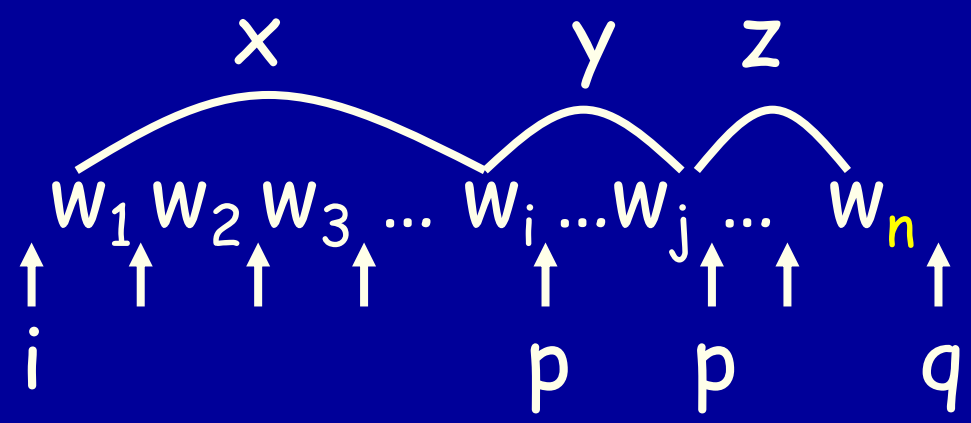
Preuve par technique de diagonalisation due à Cantor.

Très utile pour montrer qu'un ensemble infini n'est pas dénombrable.

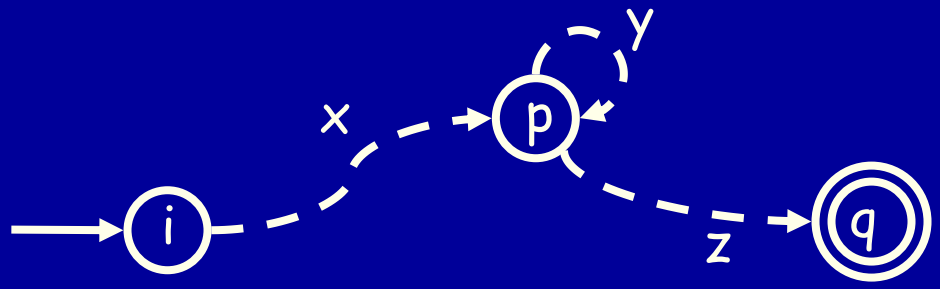
Un langage qui n'est pas rationnel

Le langage $L = \{0^k 1^k \mid k \geq 0\}$

- Supposons L rationnel; il existe A un AFD à n états qui le reconnaît. Choisissons w un mot de L de longueur ≥ n (par exemple $w=0^n1^n$). Que se passe-t'il lors de la lecture de w?
- En lisant les n premiers 0, un état p de A est visité plusieurs fois (on passe par n+1 états pour lire n symboles).



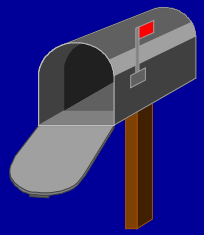
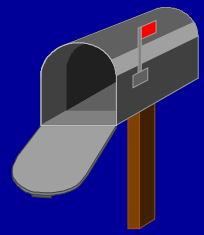
Le langage $L = \{0^k 1^k \mid k \geq 0\}$



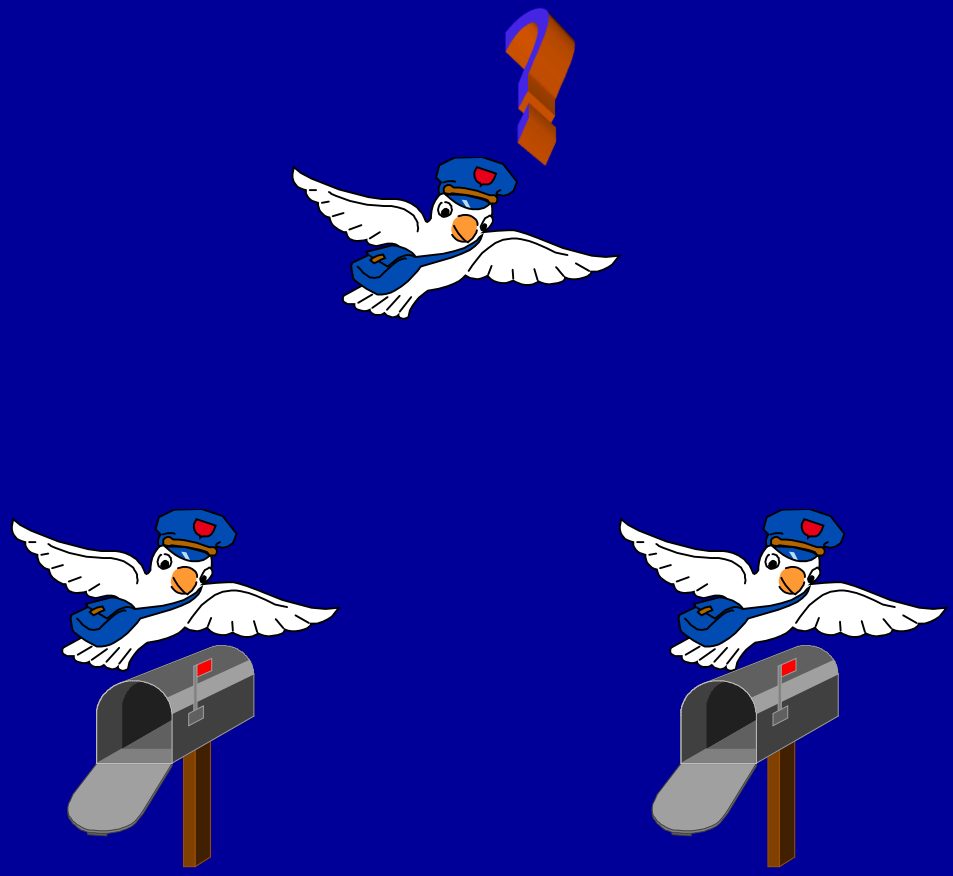
- Puisque $w=xyz \in L$, $xz \in L$ ainsi que $xyyz$ et, $\forall i \geq 0$, xy^iz. Mais chacun de ces mots possède plus ou moins de 0 que de 1, une contradiction.
- application simple
  - du principe des pigeons pour les anglo-saxons
  - des tiroirs de Dirichlet chez nous
  - ou aussi tiroirs et chaussettes …

Principe des pigeons



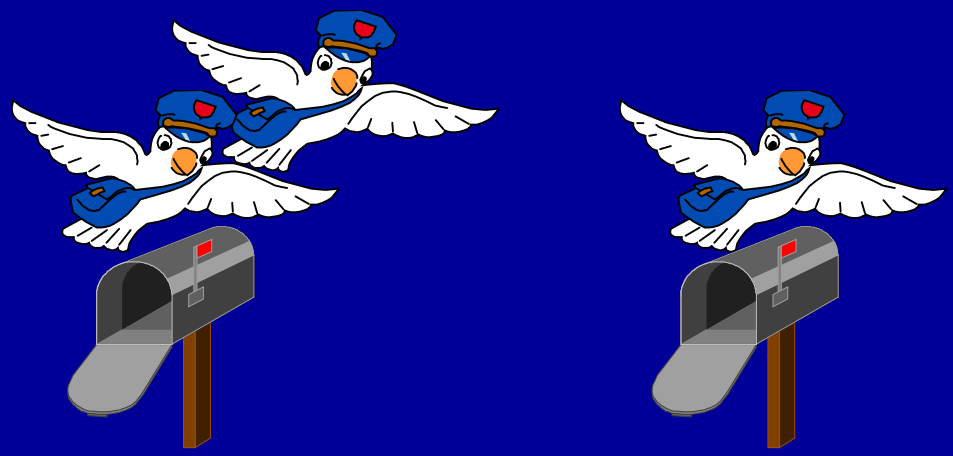



Principe des pigeons



Principe des pigeons

Si on cherche à mettre n pigeons dans m cages (n>m), une cage contiendra plus d'un pigeon



Ce qu'on vient de faire

- On a montré qu'il existe au moins un langage non rationnel, le langage

$$L = \{0^k 1^k : k \geq 0\}$$

- **But** : trouver une technique pour montrer la non rationalité d'un langage, i.e. pour décider le problème :
  - Donnée : L un langage
  - Question : L est-il non rationnel?

Le lemme de la pompe

Technique de démonstration

- On utilise un résultat sur les langages rationnels: le lemme de la pompe.
- Il exprime une propriété particulière des rationnels.
- Si un langage ne possède pas cette propriété, il n'est pas rationnel.

- **Propriété:** *Tout mot (suffisamment long) d'un langage rationnel contient un facteur qui peut être itéré autant que l'on veut de telle sorte que le mot résultant est toujours dans la langage.*

Le lemme de la pompe

- Si L est rationnel, alors il existe un nombre n tel que pour tout mot w de L, $|w| \geq n$, w peut être factorisé en w=xyz de telle sorte que
  1. Pour tout $i \geq 0$, $xy^iz \in L$
  2. $|y| > 0$
  3. $|xy| \leq n$
- Quand w est factorisé en xyz, soit x soit z peut être ε mais la condition 2 assure que $y \neq \varepsilon$.
- La condition 3 assure que le préfixe xy est de longueur au plus n. Cette condition est utile pour certains langages.

Exemple pour $L = \{0^k 1^k \mid k \geq 0\}$

- Supposons L rationnel. Alors par le lemme, il existe n tel que pour tout mot $w=xyz$, $y \neq \varepsilon$, $|xy| \leq n$ et $\forall i$, $xy^i z \in L$.
- En particulier pour $w=0^n1^n$. Comme $|xy| \leq n$, y ne contient que des zéros. Alors pour i=0, le mot $xz \notin L$. Une contradiction
- L n'est pas rationnel

Remarques

- Observons que le lemme dit

  L rationnel $\Rightarrow$ L satisfait le lemme

- Mais on ne sait rien pour la réciproque:

- Si L satisfait le lemme, on ne sait pas si L est rationnel

Utilisation du lemme

1. On suppose que L est rationnel
2. Le lemme ⇒ tout mot de longueur ≥n du langage peut être « gonflé »
3. Trouver w (|w| ≥ n) qui ne peut pas être gonflé, **quelle que soit** sa factorisation.
4. Une contradiction pour chaque factorisation
5. L n'est donc pas rationnel

Point délicat

- Le point 3. est le plus délicat

3. Trouver w ($|w| \geq n$) qui ne peut pas être gonflé, **quelle que soit** sa factorisation.

Il faut :

trouver un mot qui, pour toute factorisation, permet de trouver une valeur de i (la valeur de répétition) qui nous mène à un mot qui n'est pas de L. On contredit ainsi le lemme

Pour toute factorisation

Pourquoi faut-il trouver un mot qui, pour toute factorisation, permet de trouver une valeur de *i* (la valeur de répétition) qui nous mène à un mot qui n'est pas dans L ?

Pour toute factorisation

- On fait un raisonnement par l'absurde :
  - On utilise le fait

    L rationnelle $\Rightarrow$ "L vérifie le lemme"
  - Équivalent à

    P $\equiv$ "L non rationnelle" $\vee$ "L vérifie le lemme"
  - Par l 'absurde : il faut nier P

$\neg$P $\equiv$ "L rationnelle" $\wedge$ "L ne satisfait pas le lemme"

- Que veut dire que L ne satisfait pas le lemme ?

Exemple $L=\{w\in\{0,1\}^*:|w|_0=|w|_1\}$

- Supposons L rationnel et soit n la valeur fixée par le lemme.
- On choisit $w=0^n1^n$. On peut alors factoriser w en accord avec le lemme

$$w=xyz \text{ avec } |xy|\leq n \text{ et } |y|>0$$

- y ne contient que des 0 et xy^2z n'est plus dans le langage; une contradiction

Une présentation comme jeu

- L'utilisation du lemme peut être présenté comme un jeu entre deux joueurs (vous et un adversaire) :
  - Votre but est de prouver que L n'est pas rationnel.
  - Les correspondances des quantificateurs :

    vous ~ $\forall$ **et** adversaire ~ $\exists$

  1. Vous choisissez L.
  2. L'adversaire choisit n.
  3. Vous choisissez $w \in L$, $|w| \geq n$.
  4. L'adversaire choisit x,y,z. $w=xyz$, $|xy| \leq n$, $|y| \geq 1$.
  5. Vous choisissez i tel que $xy^iz \notin L$.
- Chaque choix peut dépendre des précédents.

Une présentation comme jeu - exemple

1. $L=\{w\in(a+b)^* : |w|_a \leq |w|_b\}$
2. n
3. $w=a^n b^n$
4. $w=xyz,\ x=a^j,\ y=a^k,\ z=a^{n-j-k}b^n,\ j\geq 0,\ k>0,\ j+k\leq n$

   $(|xy|=j+k\leq n,\ |y|=k>0)$
5. $i=2$:

$$xy^2z=a^j a^k a^k a^{n-j-k}b^n=a^{n+k}b^n\notin L$$

Conclusion : L n'est pas rationnel !

Prouver la non rationalité

- Pour montrer que L n'est pas rationnel :
  on fait un raisonnement par l'absurde.
  - On utilise le raisonnement avec des AFD
  - On utilise le lemme de la pompe
- Autre méthode :
  on utilise les propriétés de clôture

Union	Intersection	Etoile	Concatenation	Substitution
Oui	Oui	Oui	Oui	Oui

Exemple $L=\{w\in\{0,1\}^*:|w|_0=|w|_1\}$

Autre méthode :

- Supposons $L=\{w\in\{0,1\}^*:|w|_0=|w|_1\}$ rationnel
- Par les propriétés de clôture, $L\cap 0^*1^*$ doit être rationnel (0^*1^* est rationnel)
- $L\cap 0^*1^*= \{0^n1^n : n\geq 0\}$
- Comme $\{0^n1^n : n\geq 0\}$ n'est pas rationnel, L ne peut être rationnel.

La génération des langages

Origine des grammaires

- Tentatives de formalisation du langage naturel
- But : décrire précisément les règles permettant de construire des phrases syntaxiquement correctes d 'une langue

- Échec de la linguistique mais réussite pour des langues plus simples = langages informatiques

Exemple pour la linguistique

- Phrase :Sujet Verbe
- Sujet : Pronom
- Pronom : il | elle
- Verbe : dort | écoute

Règles

Symboles terminaux

- Avec ces 4 règles, on peut alors construire les phrases:
  - il écoute
  - il dort
  - elle écoute
  - elle dort

En informatique

- DecimalNumeral:
  - 0
  - NonZeroDigit $Digits_{opt}$
- Digits:
  - Digit
  - Digits Digit
- Digit:
  - 0
  - NonZeroDigit

- NonZeroDigit: one of
  1 2 3 4 5 6 7 8 9

définition d'un décimal java

Grammaire informatique

- Ensemble de règles de la forme
  - Digit:
    0
    NonZeroDigit

- Décrit la manière de construire le langage
- Inversement, un automate nous permet de reconnaître les mots du langage

Forme de Backus-Naur BNF

- Description analytique d'une grammaire informatique
- Utile à l 'analyse syntaxique (1ere étape de compilation)
- Catégories syntaxiques : suite de mots commençant par une majuscule sans espace
  - **OpérateurAdditif, NonZeroDigit, Digit**
- Alternatives : Une barre verticale sépare les alternatives
  - **Digit: 0|NonZeroDigit**
- Mots clés : en gras
  - **class, float, switch, boolean**
- Éléments optionnels : Les crochets encadrent les éléments optionnels
  - **DecimalNumeral: 0| NonZeroDigit [Digits]**
- Éléments répétés : encadrés par des accolades
  - **Identificateur : Lettre {Lettre | Chiffre}**

Flottants JAVA BNF

- FloatingPointLiteral:
  Digits . [Digits] [ExponentPart] [FloatTypeSuffix] |
  . Digits [ExponentPart] [FloatTypeSuffix] |
  Digits ExponentPart [FloatTypeSuffix] |
  Digits [ExponentPart] FloatTypeSuffix
- ExponentPart: ExponentIndicator SignedInteger
- ExponentIndicator: e | E
- SignedInteger: [Sign] Digits
- Sign: +| -
- FloatTypeSuffix: f | F | d | D

Les grammaires formelles

- Principe de base : ensemble de règles qui engendrent les mots d'un langage
- sortes de règles de réécriture
  - Une suite de symboles peut être remplacée par une nouvelle suite de symboles
  - Les mots engendrés sont ceux obtenus en appliquant les règles à partir d'un symbole de départ

Définition

- Une grammaire $G=(N,T,R,S)$
  - N : ensemble des symboles non terminaux
  - T : ensemble des symboles terminaux
  - $R \subseteq (N \times (N \cup T)^*)$: ensemble fini de règles de réécriture, les productions
  - $S \in N$: symbole de départ également appelé axiome
- Les mots engendrés sont ceux obtenus en appliquant les règles à partir du symbole de départ et qui ne contiennent plus que des symboles terminaux
- **Exemple** :

$G=(N=\{S,A,B\},T=\{0,1\},R=\{S \rightarrow ASB;\ S \rightarrow \varepsilon;\ A \rightarrow 0;\ B \rightarrow 1\},S)$

Conventions d'écriture

- Les non terminaux sont représentés par des majuscules
- Les terminaux sont représentés par des minuscules
- Les productions $(X, \alpha) \in R$ sont notées $X \rightarrow \alpha$
- L'axiome est le plus souvent noté S (start)
- Les symboles de $N \cup T$ sont appelés les symboles grammaticaux.
- Il sont représentés par les lettres minuscules grecques : $\alpha, \beta, \gamma, \ldots$
- Si $X \rightarrow \alpha_1$, $X \rightarrow \alpha_2, \ldots, X \rightarrow \alpha_k \in R$ avec X comme partie gauche, on peut écrire

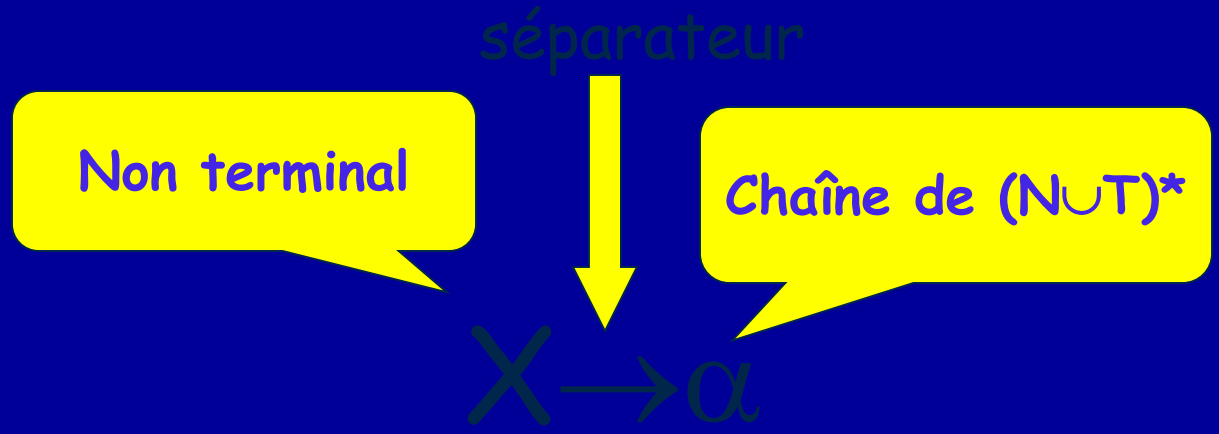
$$X \rightarrow \alpha_1 | \alpha_2 | \ldots | \alpha_k$$

Observations

- Les terminaux sont les symboles de base à partir desquels les mots sont formés; on les appelle des unités lexicales
- Les non terminaux sont des variables syntaxiques qui dénotent un ensemble de chaînes qui aident à la spécification du langage
- Exemple :
  - Lettre $\rightarrow$ A|B|C|…|Z|a|b|…|z
  - Chiffre $\rightarrow$ 0|1|…|9
  - Identificateur $\rightarrow$ Lettre {Lettre | Chiffre}
- Les terminaux sont les lettres et les chiffres
- Les non terminaux sont {Lettre, Chiffre, Identificateur} qui aident à la compréhension du langage

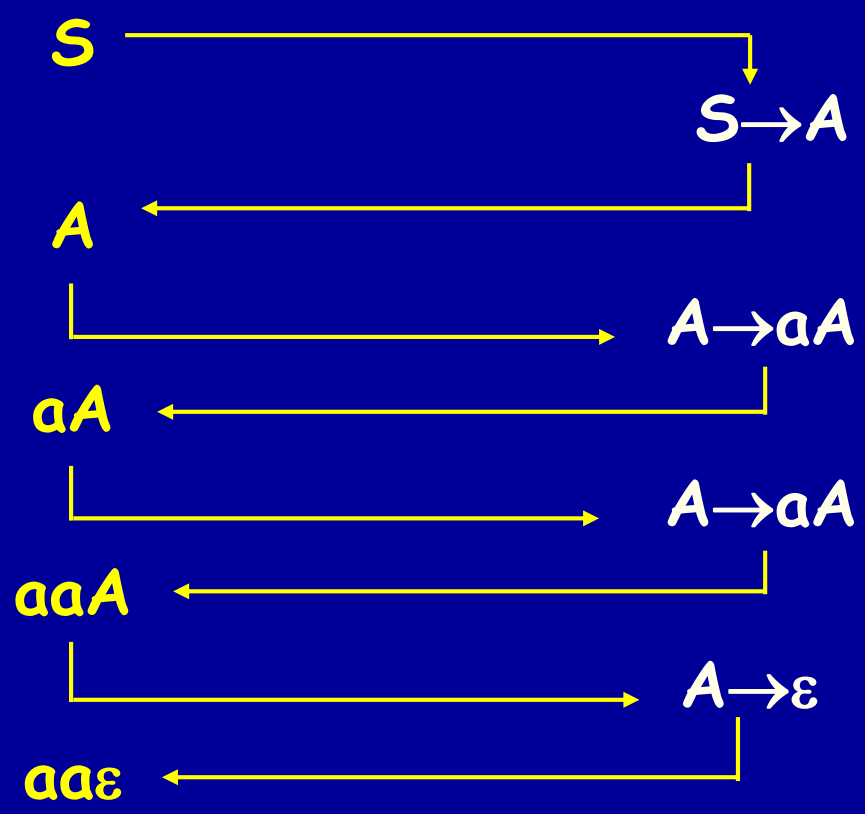
Observations

- Les productions spécifient la manière dont les terminaux et les non terminaux peuvent être combinés pour former des chaînes.
- Chaque production $X \rightarrow \alpha$ consiste en



Exemple

- G=(N,T,R,S)
  - N={S,A,B}
  - T={a,b}
  - R={S→A|B, A→aA|ε ,
    B→bB|ε}
- Définit une grammaire



Partant de S on a pu engendrer le mot aa

Une « vieille » connaissance

- $G=(N,T,R,S)$
  - $N=\{S\}$
  - $T=\{a,b\}$
  - $R=\{S\rightarrow \varepsilon,\ S\rightarrow aSb\}$

- Définit une grammaire pour $\{a^n b^n : n\geq 0\}$ non rationnel

$$
\begin{array}{ccccccccc}
S & \rightarrow aSb & \rightarrow aaSbb & \rightarrow aaaSbbb & \rightarrow aaaaSbbbb & \rightarrow aaaaaSbbbbb \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
\varepsilon & ab & aabb & aaabbb & aaaabbbb &
\end{array}
$$