Théorie des Langages
Expressions rationnelles, Lemme d'Arden et Quotients gauches

Yannick Le Nir Gaspard Férey

Contributions de :
Taisa Guidini Goncalves

CY TECH

yannick.lenir@eisti.fr gaspard.ferey@inria.fr

Rappels

Langages, grammaires et automates

- Langages *rationnels* ou *réguliers* ou *de type 3*
  - grammaire de type 3
  - expressions régulières
  - automates états finis

Théorème de Kleene

Un langage est régulier ssi il est reconnu par un automate d'états finis.

Langages et automates

Objectifs

- Une nouvelle méthode de définition d'un langage : les expressions régulières.
- Génération d'une expression régulière à partir d'un automate
  - mise en équation
  - lemme d'Arden
- Construction de l'automate minimal à partir d'une expression régulière
  - méthode des quotients gauches

Expressions rationnelles ou régulières

Construction

Soit A un alphabet.

- $\forall x \in A$, x est une expression régulière (ER)
- ε (mot vide) et $\emptyset$ (ensemble vide) sont des ER
- si E_1 et E_2 sont des ER, $E_1 \; + \; E_2$ est une ER
- si E_1 et E_2 sont des ER, $E_1.E_2$ est une ER
  Note : le point peut être omis.
- si E est un ER, E^* est une ER

Exemples

$(0 \; + \; 1)^*$	$110^* + \varepsilon$	$A^+ = A(A^*)$
$(0(10)^*)1$	$0(1 + \emptyset)$	$A \cup B = A + B$
$(10)^*(01)^*$	$(0^* + 1^*)^*$	$\{a, b, c\}^* = (a + b + c)^*$
		$A^n B^n$

Remarques

Remarques

- On s'autorise l'usage de parenthèses pour clarifier l'ordre des opérations.
- La concaténation est associative donc on note $A(BC) = (AB)C = ABC$.
- L'étoile $(*)$ est prioritaire sur la concaténation qui est prioritaire sur l'union $(+)$:
  - $AB^* = A(B^*)$
  - $A + BC = A + (BC)$
  - $AB + C = (AB) + C$
  - $A + B^* = A + (B^*)$

Langages rationnels (réguliers)

Définition

Soit E une ER. Le langage $\mu(E)$ est défini par

$$\forall x \in A, \mu(x) = \{x\} \qquad \mu(\emptyset) = \emptyset$$
$$\mu(E_1 + E_2) = \mu(E_1) \cup \mu(E_2) \qquad \mu(\varepsilon) = \{\varepsilon\}$$
$$\mu(E_1 \cdot E_2) = \mu(E_1) \cap \mu(E_2) \qquad \mu(E^*) = \mu(E)^*$$

Propriété fondamentale

Un langages L est rationnel (régulier) si et seulement si il existe une expression rationnelle (régulière) E telle que $L = \mu(E)$.

Remarque

On confondra souvent par abus de langage, une expression et le langage qu'elle représente.

Langages et expressions

Equivalence

- Un langage peut être produit par des grammaires différentes (cf TD 1).
  S'il est rationnel, certaines d'entre elles sont de type 3.
- Un langage rationnel peut être reconnu par des automates différents (cf TD 2).
- Un langage rationnel peut également être représenté par différentes expressions (dites équivalentes).

Exemples

- Les expressions $(a^*b^*)^*$ et $(a + b)^*$ sont équivalentes et représentent toutes les deux le langage $\{a, b\}^*$
- Par contre $(aa + b)^*$ et $(aa + aab + bb)^*$ ne sont pas équivalentes, car par exemple b appartient au langage associé à la première mais pas à celui de la seconde.

Automates et langages

De l'automate au langage

- Déterminer le système d'équations à partir de l'automate.
- Déterminer sa solution : le langage reconnu par l'automate.
- Résolution via lemme d'Arden.

Mise en équations : algorithme

Construction d'une ER à partir d'un AFD

Soit $< Et, F, Ev, \Psi, q_0 >$ un AFD.

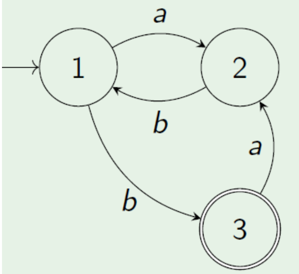
- Les évènements forment l'alphabet : $A := Ev$
- Pour chaque état $q_i \in Et$ on définit une inconnue Y_i représentant le langage reconnu par l'AFD si q_i était l'état initial.
- Pour chaque état $q_i \in Et$, toutes les transitions (q_i, e, q_f) produisent l'équation :

$$Y_i = \sum_{(q_i, e, q_f) \in \Psi} e \cdot Y_f \quad (+ \varepsilon \text{ si } q_i \in F)$$

- La solution (langage reconnu par l'AFD) est la valeur de l'inconnue Y_0 correspondant à l'état initial q_0.

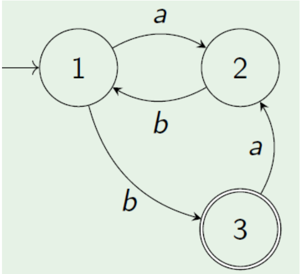
Mise en équations : exemple

Exemple



Mise en équations : exemple

Exemple



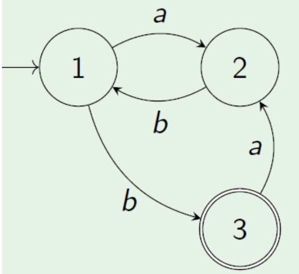
Y_1

Y_2

Y_3

Mise en équations : exemple

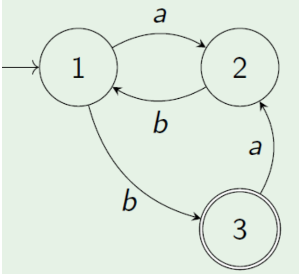
Exemple



$$
\begin{aligned}
Y_1 &= aY_2 + bY_3 \\
Y_2 & \\
Y_3 &
\end{aligned}
$$

Mise en équations : exemple

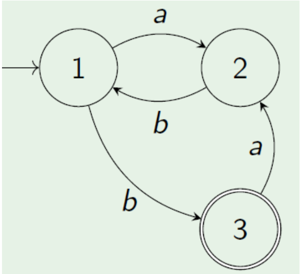
Exemple



$$
\begin{aligned}
Y_1 &= aY_2 + bY_3 \\
Y_2 &= bY_1 \\
Y_3 &
\end{aligned}
$$

Mise en équations : exemple

Exemple



$$
\begin{aligned}
Y_1 &= aY_2 + bY_3 \\
Y_2 &= bY_1 \\
Y_3 &= aY_2 + \varepsilon
\end{aligned}
$$

Lemme d'Arden

Lemme

Soient A et D des langages.

Soit (E) une équation de la forme $X = A.X + D$.

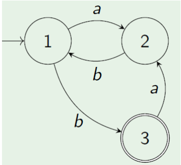
1. Le langage $X = A^*D$ est une solution de (E)
2. Toute solution de (E) contient A^*D (solution minimale).
3. Si $\varepsilon \notin A$ alors A^*D est l'unique solution de (E).

Résolution de l'équation

Principe

- Générer le système d'équations
- Appliquer plusieurs fois le lemme d'Arden ($X = A.X + D \Rightarrow X = A^*D$) pour résoudre le système d'équations
- La solution est le langage reconnu par l'AFD

Exemple



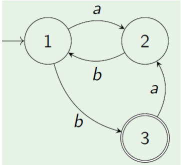
$$
\begin{aligned}
Y_1 &= aY_2 + bY_3 \\
Y_2 &= bY_1 \\
Y_3 &= aY_2 + \varepsilon
\end{aligned}
$$

Résolution de l'équation

Principe

- Générer le système d'équations
- Appliquer plusieurs fois le lemme d'Arden ($X = A.X + D \Rightarrow X = A^*D$) pour résoudre le système d'équations
- La solution est le langage reconnu par l'AFD

Exemple



$$Y_1 = aY_2 + b(aY_2 + \varepsilon)$$

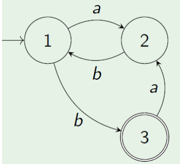
$$\begin{aligned} Y_1 &= aY_2 + bY_3 \\ Y_2 &= bY_1 \\ Y_3 &= aY_2 + \varepsilon \end{aligned}$$

Résolution de l'équation

Principe

- Générer le système d'équations
- Appliquer plusieurs fois le lemme d'Arden ($X = A.X + D \Rightarrow X = A^*D$) pour résoudre le système d'équations
- La solution est le langage reconnu par l'AFD

Exemple



$$
\begin{aligned}
Y_1 &= aY_2 + bY_3 \\
Y_2 &= bY_1 \\
Y_3 &= aY_2 + \varepsilon
\end{aligned}
$$

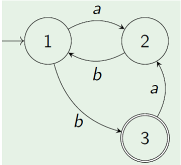
$$
\begin{aligned}
Y_1 &= aY_2 + b(aY_2 + \varepsilon) \\
&= (a + ba)Y_2 + b
\end{aligned}
$$

Résolution de l'équation

Principe

- Générer le système d'équations
- Appliquer plusieurs fois le lemme d'Arden ($X = A.X + D \Rightarrow X = A^*D$) pour résoudre le système d'équations
- La solution est le langage reconnu par l'AFD

Exemple



$$\begin{aligned} Y_1 &= aY_2 + bY_3 \\ Y_2 &= bY_1 \\ Y_3 &= aY_2 + \varepsilon \end{aligned}$$

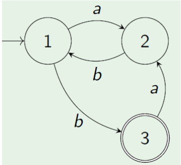
$$\begin{aligned} Y_1 &= aY_2 + b(aY_2 + \varepsilon) \\ &= (a + ba)Y_2 + b \\ &= (a + ba)bY_1 + b \end{aligned}$$

Résolution de l'équation

Principe

- Générer le système d'équations
- Appliquer plusieurs fois le lemme d'Arden ($X = A.X + D \Rightarrow X = A^*D$) pour résoudre le système d'équations
- La solution est le langage reconnu par l'AFD

Exemple



$$\begin{aligned} Y_1 &= aY_2 + bY_3 \\ Y_2 &= bY_1 \\ Y_3 &= aY_2 + \varepsilon \end{aligned}$$

$$\begin{aligned} Y_1 &= aY_2 + b(aY_2 + \varepsilon) \\ &= (a + ba)Y_2 + b \\ &= (a + ba)bY_1 + b \\ &=_{\text{Arden}} ((a + ba)b)^*b \end{aligned}$$

Langages et automates

Du langage à l'automate minimal

- Déterminer les états à partir du langage
- Déterminer les transitions
- $\rightarrow$ construction via quotients gauches

Quotients gauches

Définition

Soit L un langage sur un alphabet A.
Les quotients gauches de L sont notés $a^{-1}L$ pour $a \in A$ et sont définis par :

$$a^{-1}L = \{w \in A^* / aw \in L\}$$

C'est l'ensemble des mots de L commencant par le symbole a privé de ce premier symbole a.

Quotients gauches

Propriété

Soient L_1 et L_2 deux langages sur un alphabet A.
Soient $a, b \in A$ tels que $a \neq b$.

$$
\begin{aligned}
a^{-1}(a) &= \varepsilon \\
a^{-1}(b) &= \emptyset \\
a^{-1}(L_1L_2) &= \begin{cases} (a^{-1}L_1)L_2 + a^{-1}L_2 & \text{si } \varepsilon \in L_1 \\ (a^{-1}L_1)L_2 & \text{sinon} \end{cases} \\
a^{-1}(L_1 + L_2) &= a^{-1}L_1 + a^{-1}L_2
\end{aligned}
$$

Quotients gauches

Théorème

L'ensemble des quotients gauches de L, $Q(L)$, est fini

Proposition

Soit A un AFD et $L = L(A)$. On a $Q(L) = \{L_q / q \in L\}$ fini. On peut construire un automate minimal qui reconnait L et dont chaque état correspond à un élément de $Q(L)$.

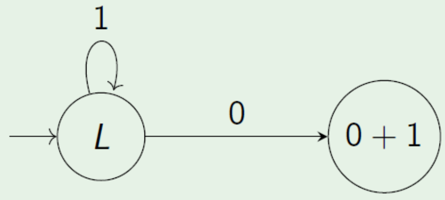
Construction de l'automate

Principe

- Chaque quotient est un état de l'automate
- Chaque transition permet de passer d'un état à l'autre
- L'état final est le quotient chaîne vide ε
- L'état poubelle est le quotient vide

Exemple

Soit l'E.R. $L = 1^*0(0+1)$
$0^{-1}L = 0 + 1$ et $1^{-1}L = L$



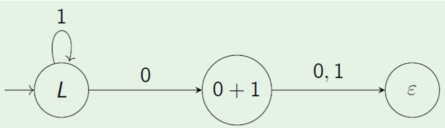
Construction de l'automate

Principe

- Chaque quotient est un état de l'automate
- Chaque transition permet de passer d'un état à l'autre
- L'état final est le quotient chaîne vide ε
- L'état poubelle est le quotient vide

Exemple

Soit l'E.R. $L = 1^*0(0+1)$
$0^{-1}(0+1) = \varepsilon$ et $1^{-1}(0+1) = \varepsilon$



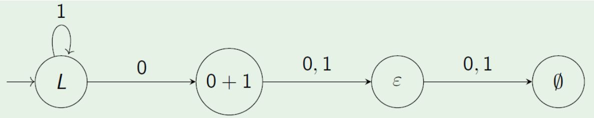
Construction de l'automate

Principe

- Chaque quotient est un état de l'automate
- Chaque transition permet de passer d'un état à l'autre
- L'état final est le quotient chaîne vide ε
- L'état poubelle est le quotient vide

Exemple

Soit l'E.R. $L = 1^*0(0+1)$
$0^{-1}\varepsilon = \emptyset$ et $1^{-1}\varepsilon = \emptyset$



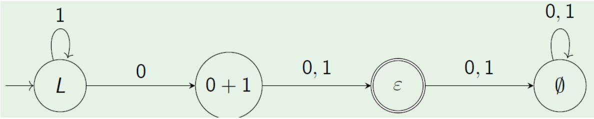
Construction de l'automate

Principe

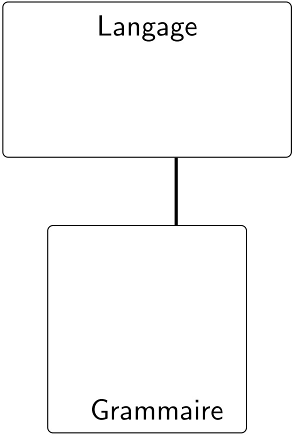
- Chaque quotient est un état de l'automate
- Chaque transition permet de passer d'un état à l'autre
- L'état final est le quotient chaîne vide ε
- L'état poubelle est le quotient vide

Exemple

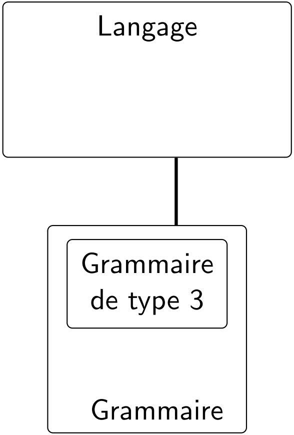
Soit l'E.R. $L = 1^*0(0+1)$
$0^{-1}\emptyset = \emptyset$ et $1^{-1}\emptyset = \emptyset$



Récapitulons



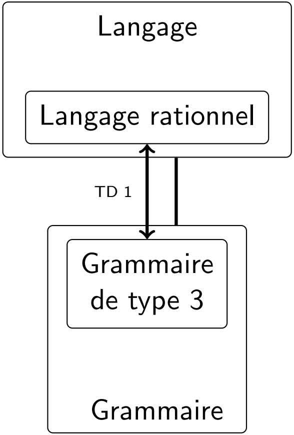
Récapitulons



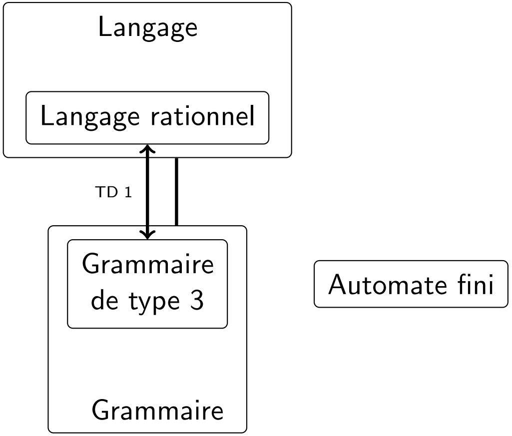
Yannick Le Nir, Gaspard Férey

Contributions de : Taisa Guidini Goncalves

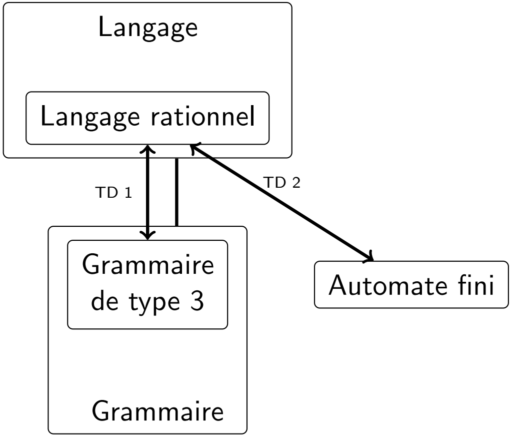
Récapitulons



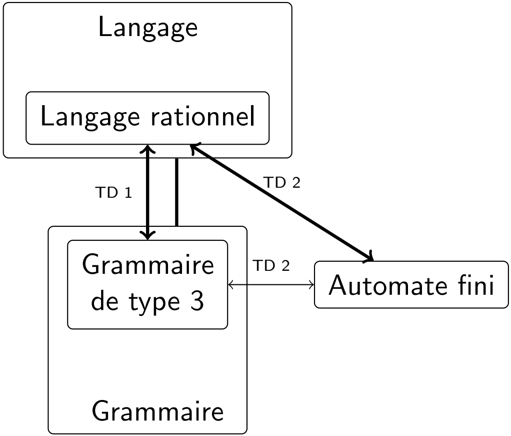
Récapitulons



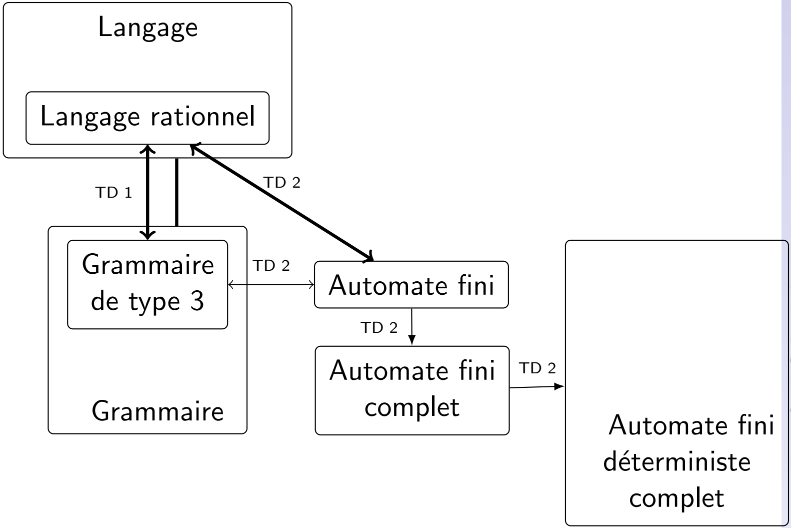
Récapitulons



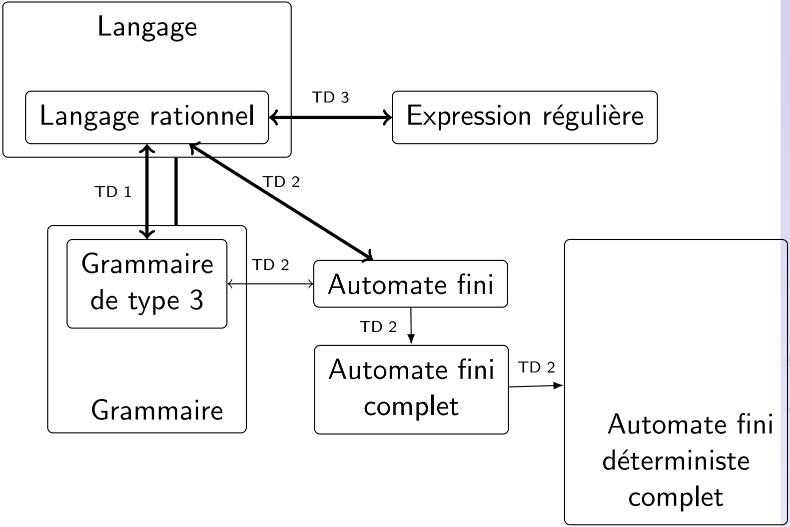
Récapitulons



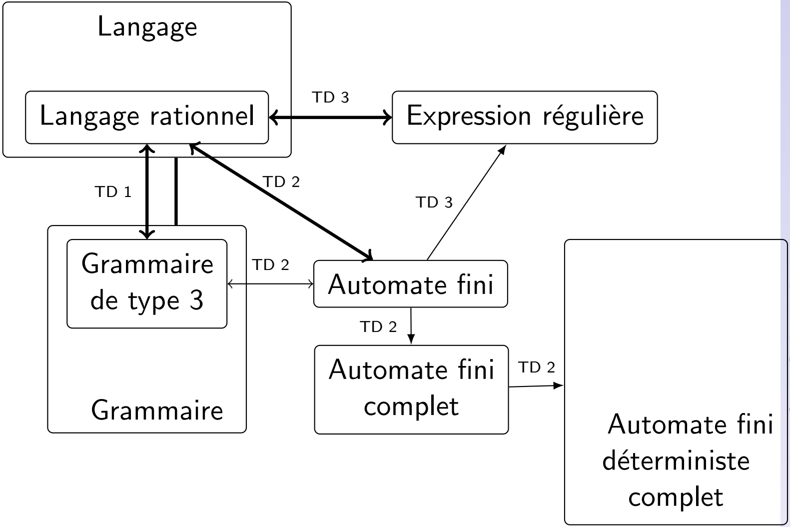
Récapitulons



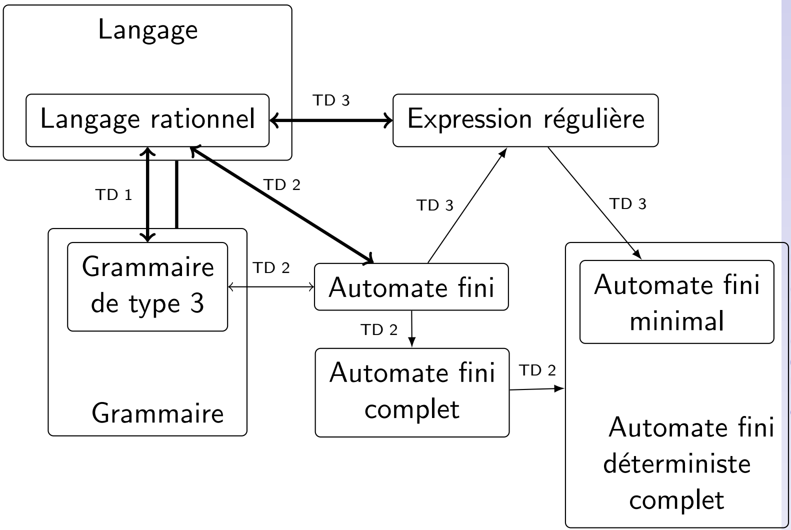
Récapitulons



Récapitulons



Récapitulons



Algorithme de déterminisation

Principe

- Construire une expression régulière à partir de l'automate fini non déterministe (mise en équation et lemme d'Arden).
- Appliquer les quotients gauches pour construire l'automate fini minimal déterministe équivalent à l'automate fini non déterministe.

Preuve du lemme d'Arden

Lemme

(E) $\qquad X = A.X + D$

1. $X = A^*D$ est solution de (E).
2. A^*D est la solution minimale.
3. Si $\varepsilon \notin A$ alors A^*D est l'unique solution de (E).

Preuve de 1.

$$\begin{aligned} A^*D &= (A^+ + \varepsilon)D \\ &= (AA^* + \varepsilon)D \\ &= (AA^*)D + \varepsilon D \\ &= A(A^*D) + D \end{aligned}$$

Preuve du lemme d'Arden

Lemme

$(E) \qquad X = A.X + D$

1. $X = A^*D$ est solution de (E).
2. A^*D est la solution minimale.
3. Si $\varepsilon \notin A$ alors A^*D est l'unique solution de (E).

Preuve de 2.

Soit X solution de (E).
Montrons par récurrence sur n que $A^nD \subset X$:

- $n = 0$: $D \subset X$
- $n > 0$: $A^nD = A(A^{n-1}D)$.
  Or par hypothèse de recurrence, $A^{n-1}D \subset X$
  Donc $A^nD = A(A^{n-1}D) \subset AX \subset X$.

Donc $\forall n, A^nD \subset X$ et on en conclut $A^*D \subset X$.

Preuve du lemme d'Arden

Lemme

$(E) \qquad X = A.X + D$

1. $X = A^*D$ est solution de (E).
2. A^*D est la solution minimale.
3. Si $\varepsilon \notin A$ alors A^*D est l'unique solution de (E).

Preuve de 3.

Supposons $\varepsilon \notin A$. Soit X solution de (E) et $w \in X$.
Montrons $w \in A^*D$ par récurrence forte sur $|w|$.

- $|w| = 0$: $w = \varepsilon$ possible ssi $w \in D \subset A^*D$.
- $|w| > 0$: si $w \notin D$, alors nécessairement $w \in AX$.
  Donc $w = aw'$ avec $a \in A$ et $w' \in X$.
  Comme $\varepsilon \notin A$, $|a| > 0$ et $|w'| < |w|$.
  Par hypothèse de récurrence, $w' \in A^*D$ et donc
  $w \in AA^*D \subset A^*D$.