AFD Vs AFN

En résumé

- Les AFNs n'apportent aucun gain en expressivité : la prise en compte de transitions non déterministes est sans effet sur le langage reconnu.
- Les AFNs sont plus faciles à construire à partir d'autres représentations des langages et sont donc utiles pour certaines preuves ou algorithmes.
- Ils fournissent également des machines bien plus (dans certains cas exponentiellement plus) "compactes" que les AFD, ce qui n'est pas une propriété négligeable.

Théorème de déterminisation

Pour tout AFN A défini sur Σ, il existe un AFD A' équivalent à A. Si A a n états, alors A' a au plus 2^n états.

AFN --> AFD : Déterminisation

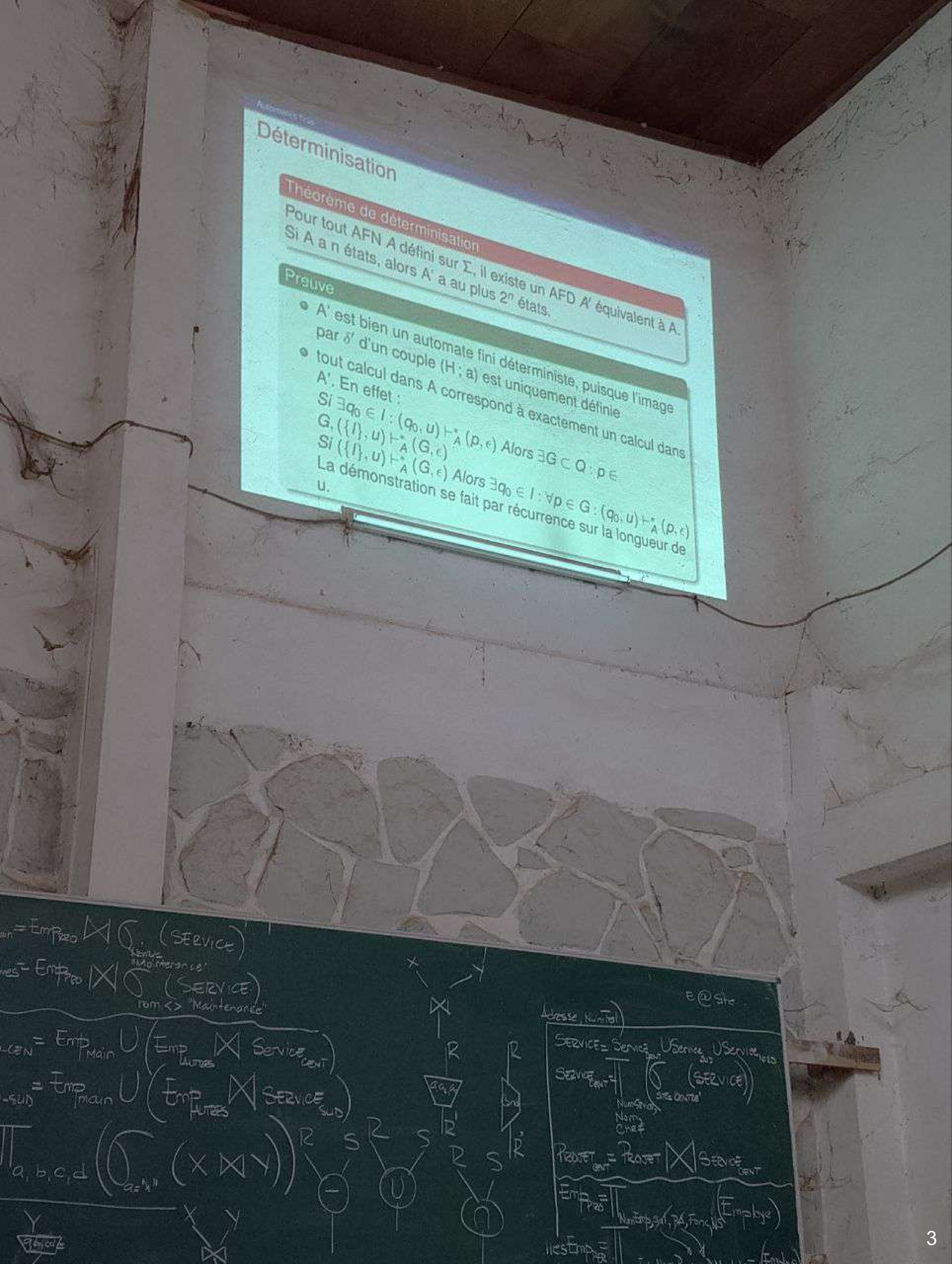
Preuve

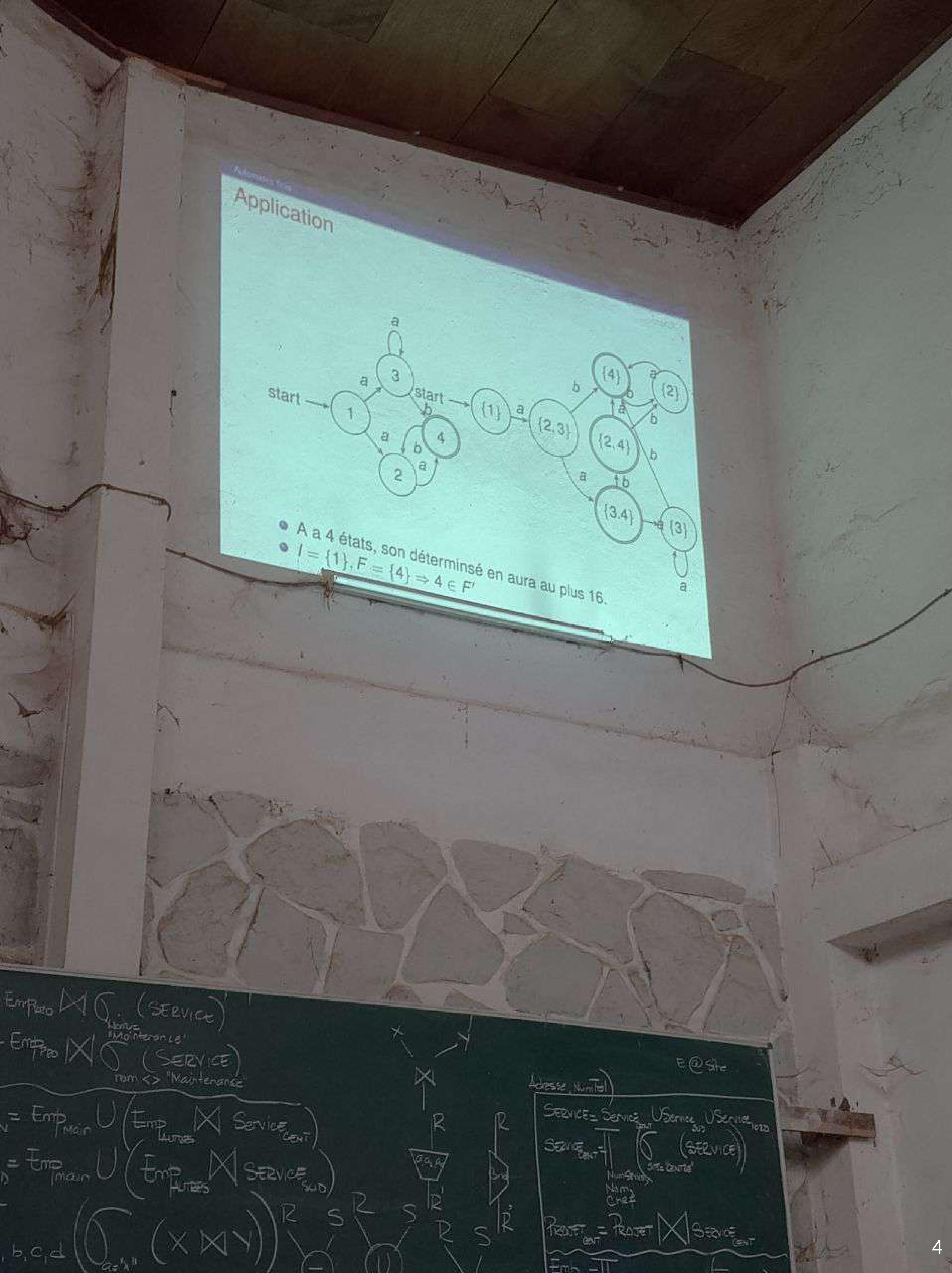
Soit $A = (\Sigma, Q, I, F, \delta)$. On considère $A' = (\Sigma, 2^Q, I, F', \delta')$ avec

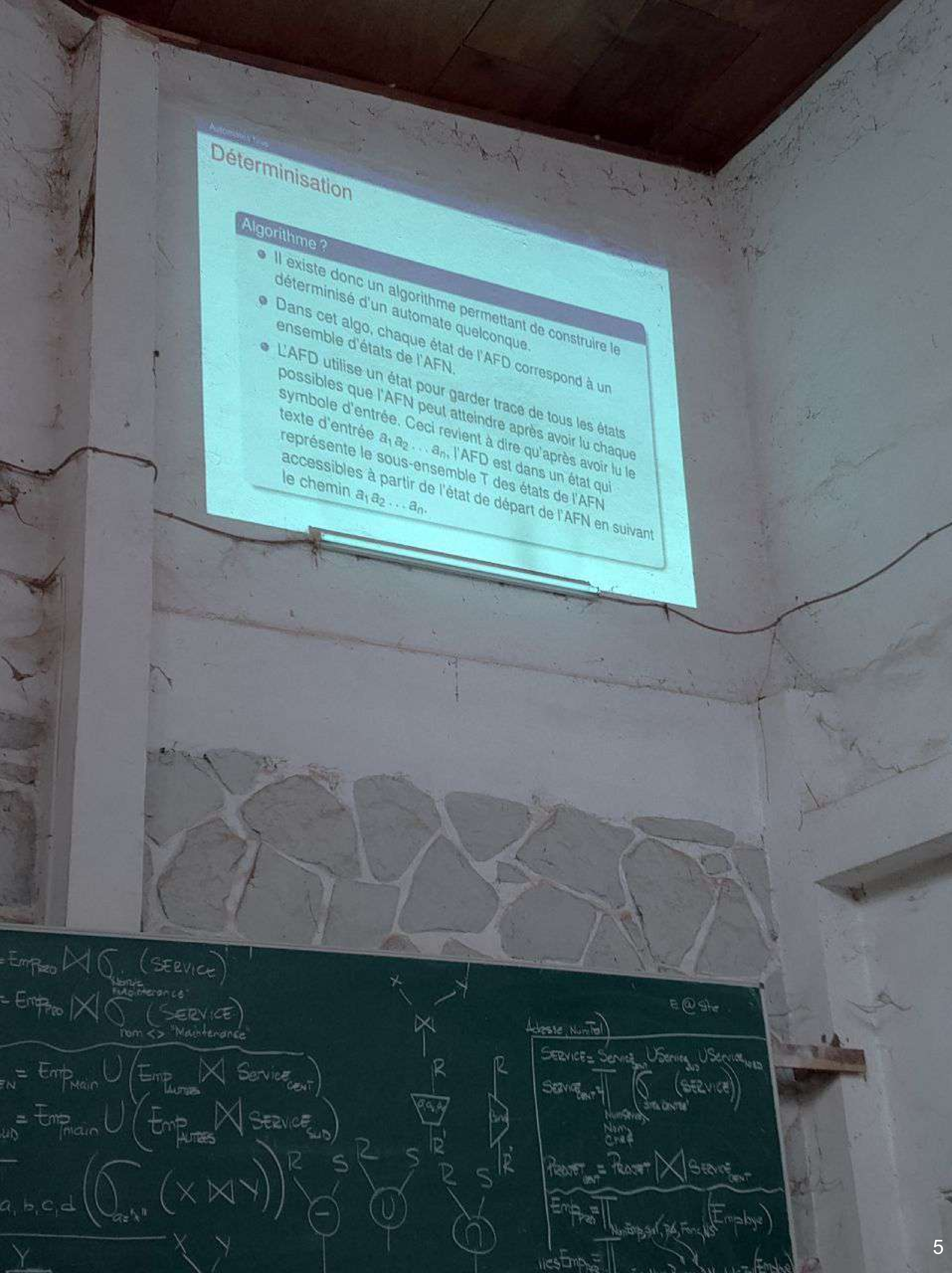
- $F' = \{G \subset Q \mid F \cap G \neq \emptyset\}$
- $\forall G \subset Q, \forall a \in \Sigma, \delta'(G, a) = \bigcup_{q \in G} \delta(q, a)$

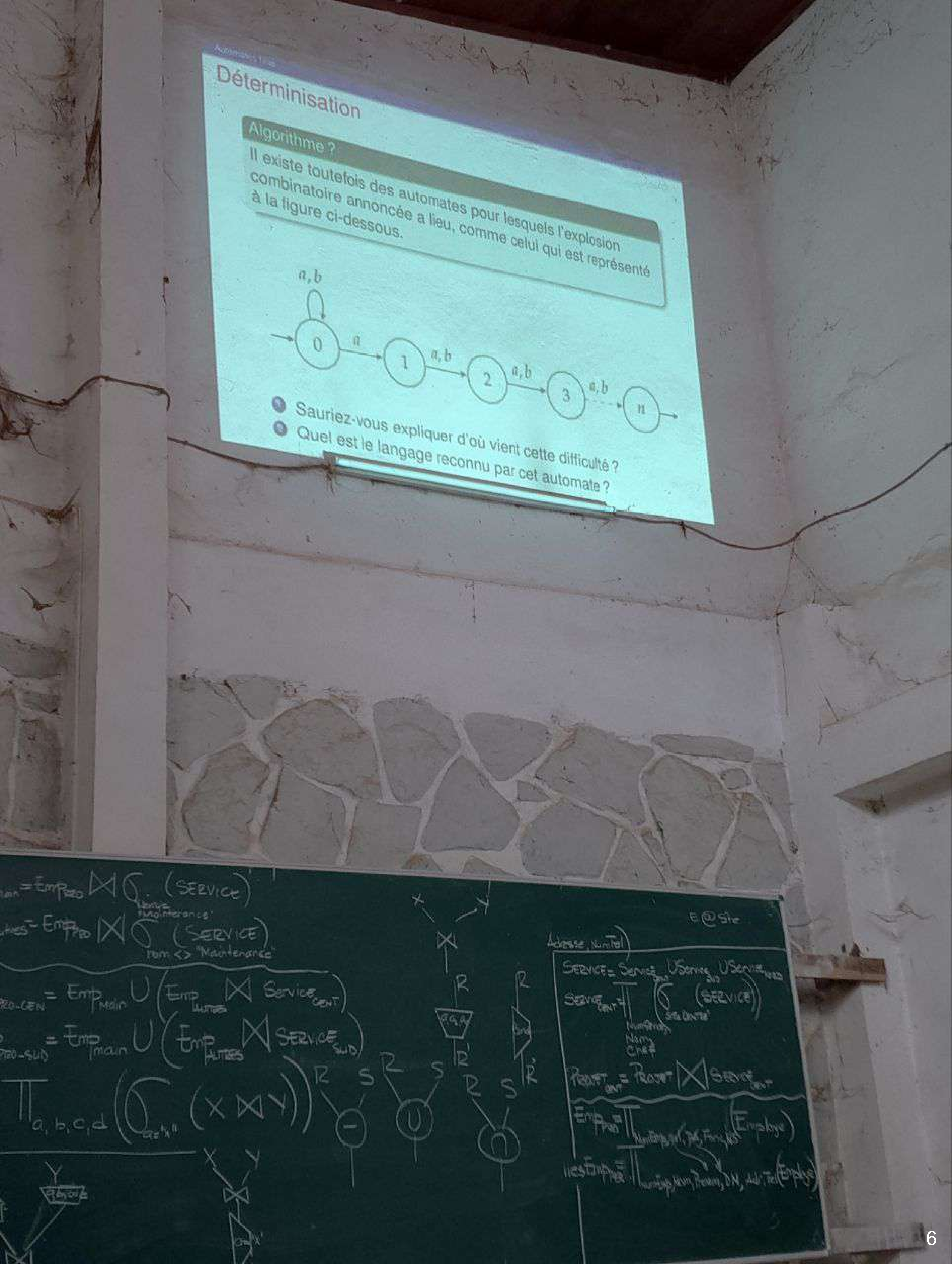
Ainsi A' est un automate déterministe équivalent à A. On l'appelle aussi le déterminisé (ou encore le standardisé) de A.

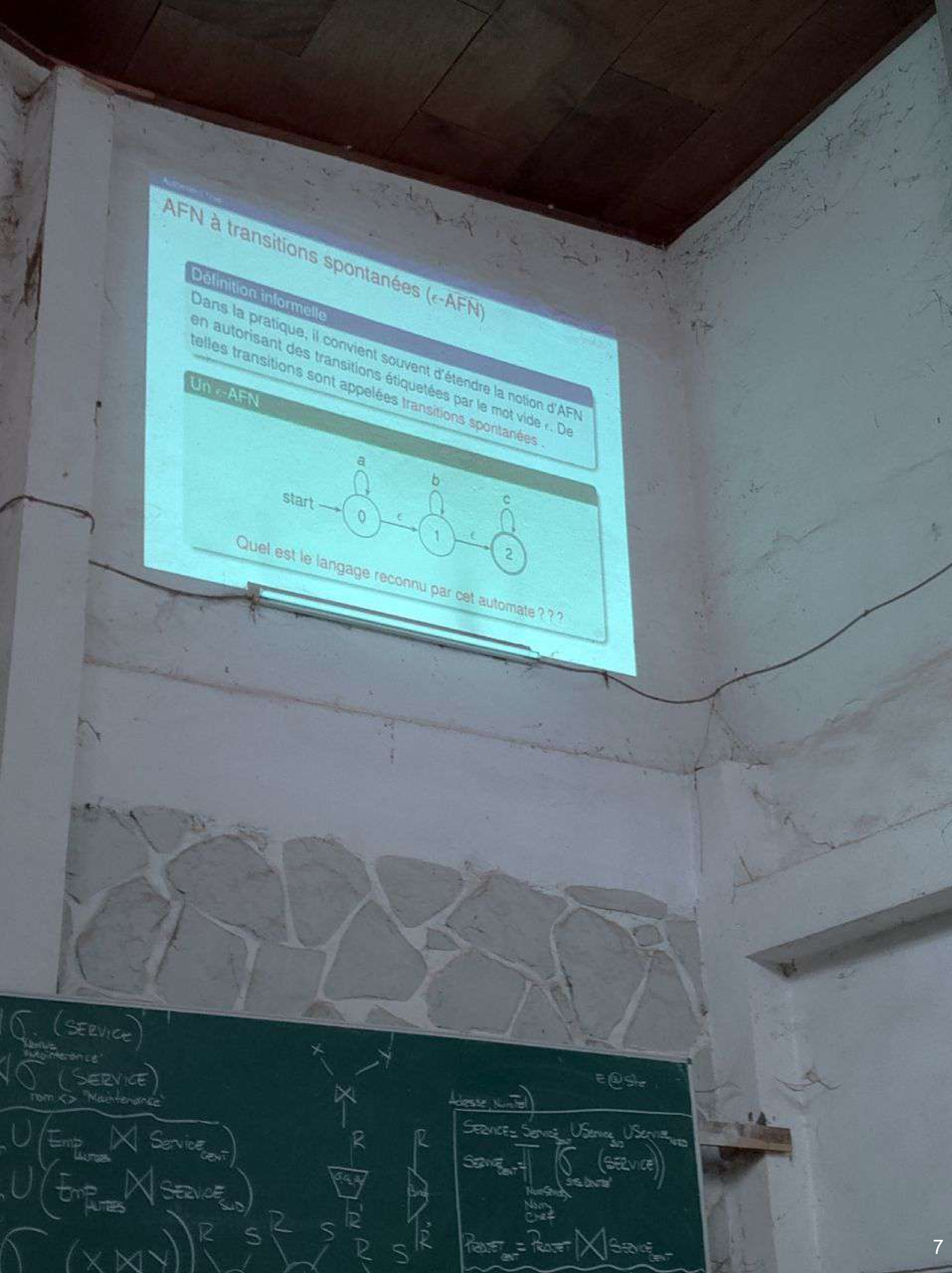
- Les états de A' sont donc associés de manière biunivoque à des sous-ensembles de Q (il y en a un nombre fini), l'état initial étant l'ensemble $\{I\}$;
- Chaque partie contenant un état final de A donne lieu à un état final de A' ;
- La transition sortante d'un sous-ensemble E, étiquetée par a, atteint l'ensemble de tous les états de Q atteignables depuis un état de E par une transition étiquetée par a.

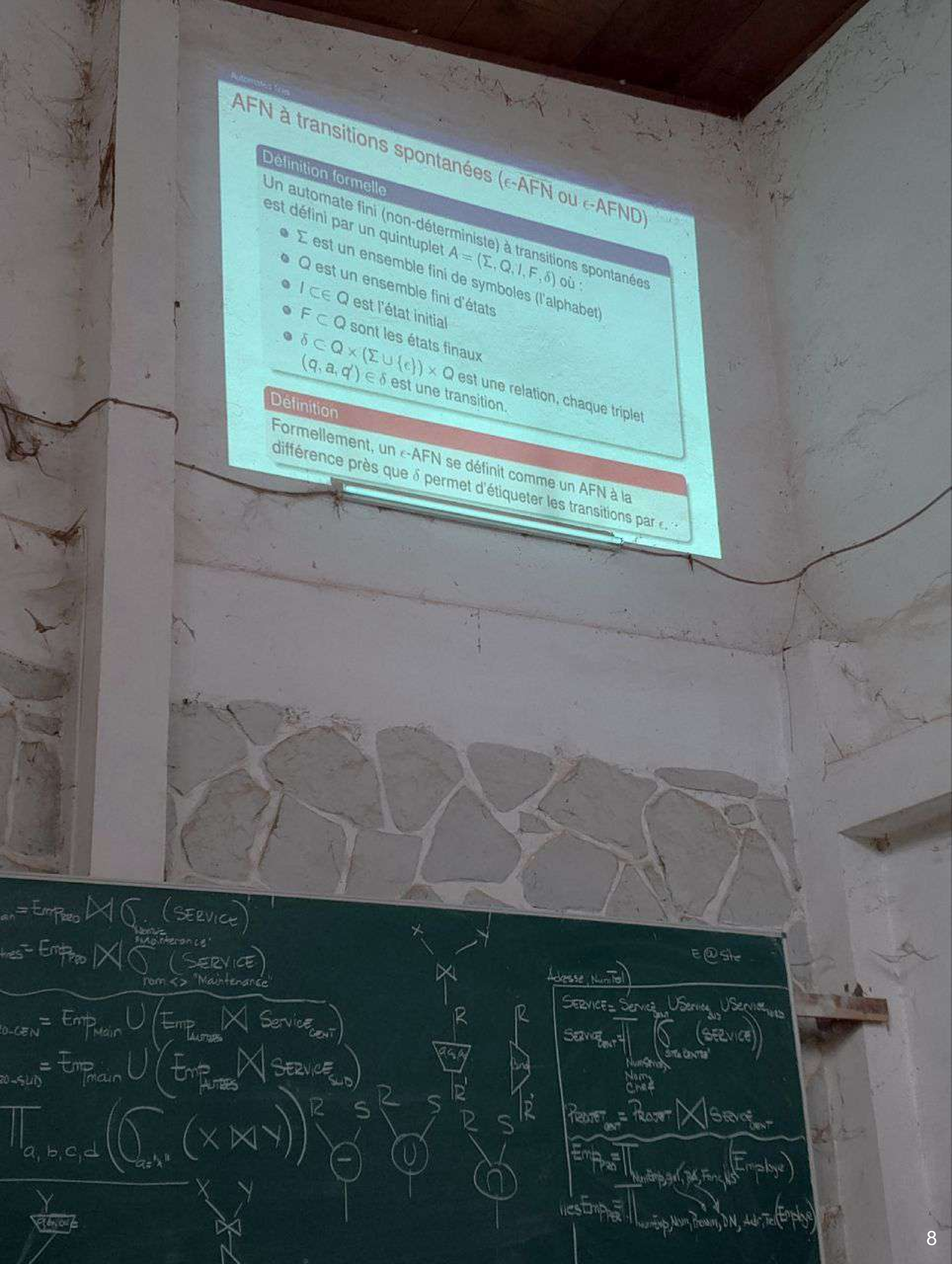

Déterminisation
Théorème de déterminisation
Pour tout AFN A défini sur Σ, il existe un AFD A' équivalent à A.
Si A a n états, alors A' a au plus 2^n états.
Preuve
A' est bien un automate fini déterministe, puisque l'image par δ' d'un couple (H ; a) est uniquement définie
tout calcul dans A correspond à exactement un calcul dans A'. En effet :
$Si\ \exists q_0 \in I : (q_0, u) \vdash^*_A (p, \epsilon)\ Alors\ \exists G \subset Q : p \in G, (\{I\}, u) \vdash^*_A (G, \epsilon)$
$Si\ (\{I\}, u) \vdash^*_A (G, \epsilon)\ Alors\ \exists q_0 \in I : \forall p \in G : (q_0, u) \vdash^*_A (p, \epsilon)$
La démonstration se fait par récurrence sur la longueur de u.
SERVICE = Service$_{CENT}$ U Service$_{SUD}$ U Service$_{NORD}$
Adresse, NumTel
E @ Site

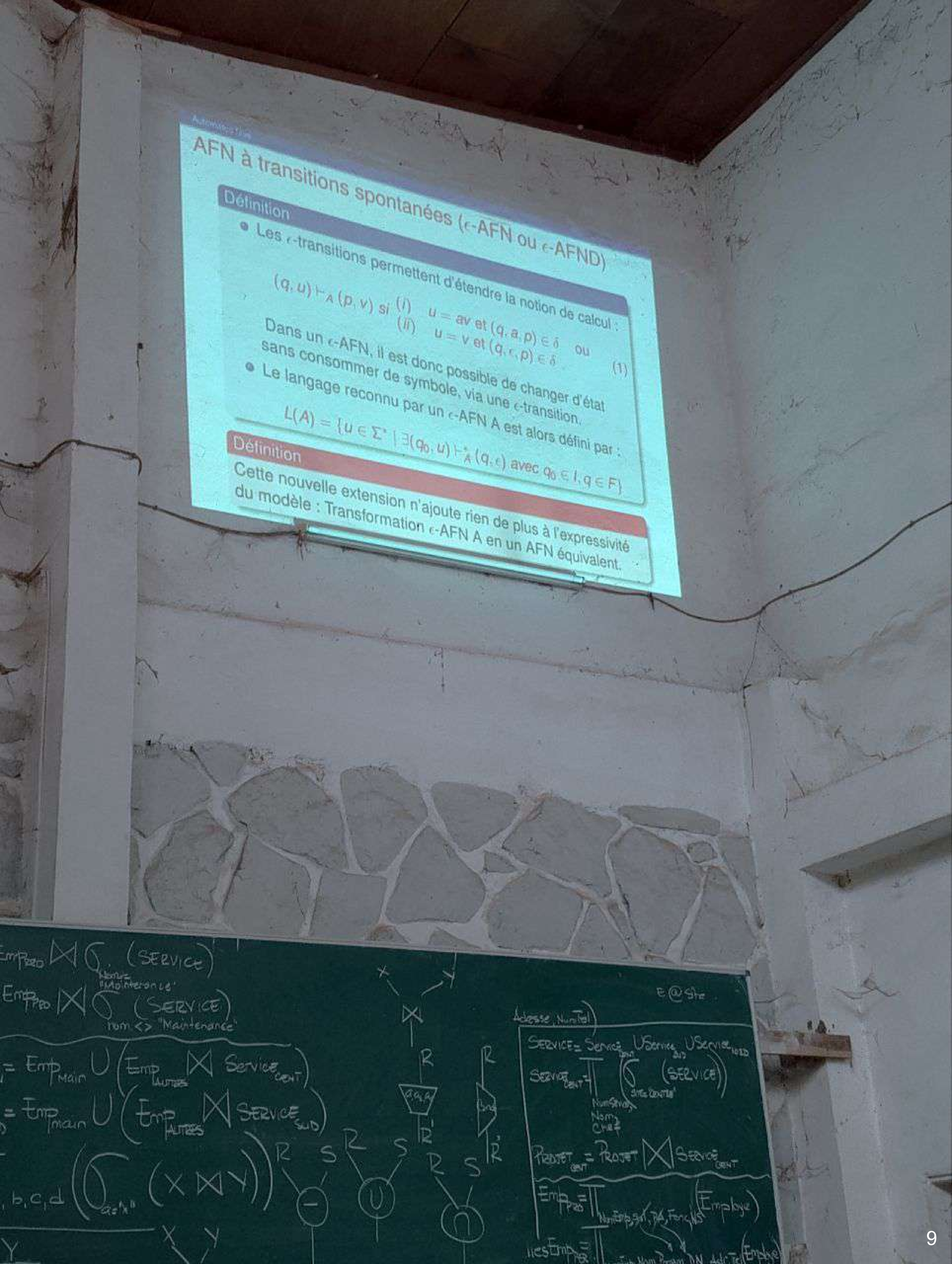

Application
a
3
start
1
a
b
4
b
a
2
a
start
{1}
a
{2,3}
b
{4}
a
b
{2}
a
b
{2,4}
b
a
b
{3,4}
{3}
a
A a 4 états, son déterminsé en aura au plus 16.
I = {1}, F = {4} ⇒ 4 ∈ F'

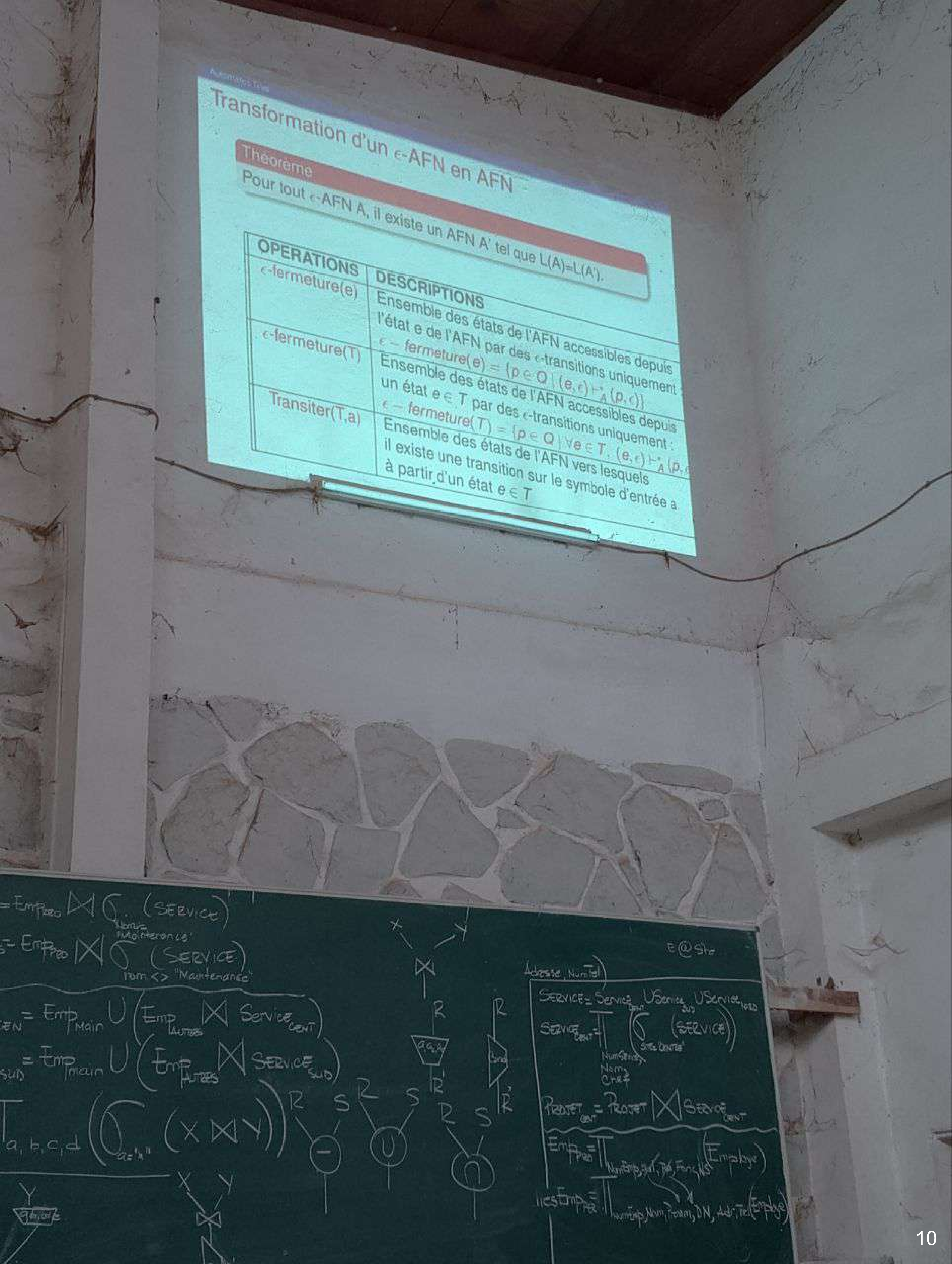

Déterminisation
Algorithme ?
Il existe donc un algorithme permettant de construire le déterminisé d'un automate quelconque.
Dans cet algo, chaque état de l'AFD correspond à un ensemble d'états de l'AFN.
L'AFD utilise un état pour garder trace de tous les états possibles que l'AFN peut atteindre après avoir lu chaque symbole d'entrée. Ceci revient à dire qu'après avoir lu le texte d'entrée $a_1 a_2 \ldots a_n$, l'AFD est dans un état qui représente le sous-ensemble T des états de l'AFN accessibles à partir de l'état de départ de l'AFN en suivant le chemin $a_1 a_2 \ldots a_n$.

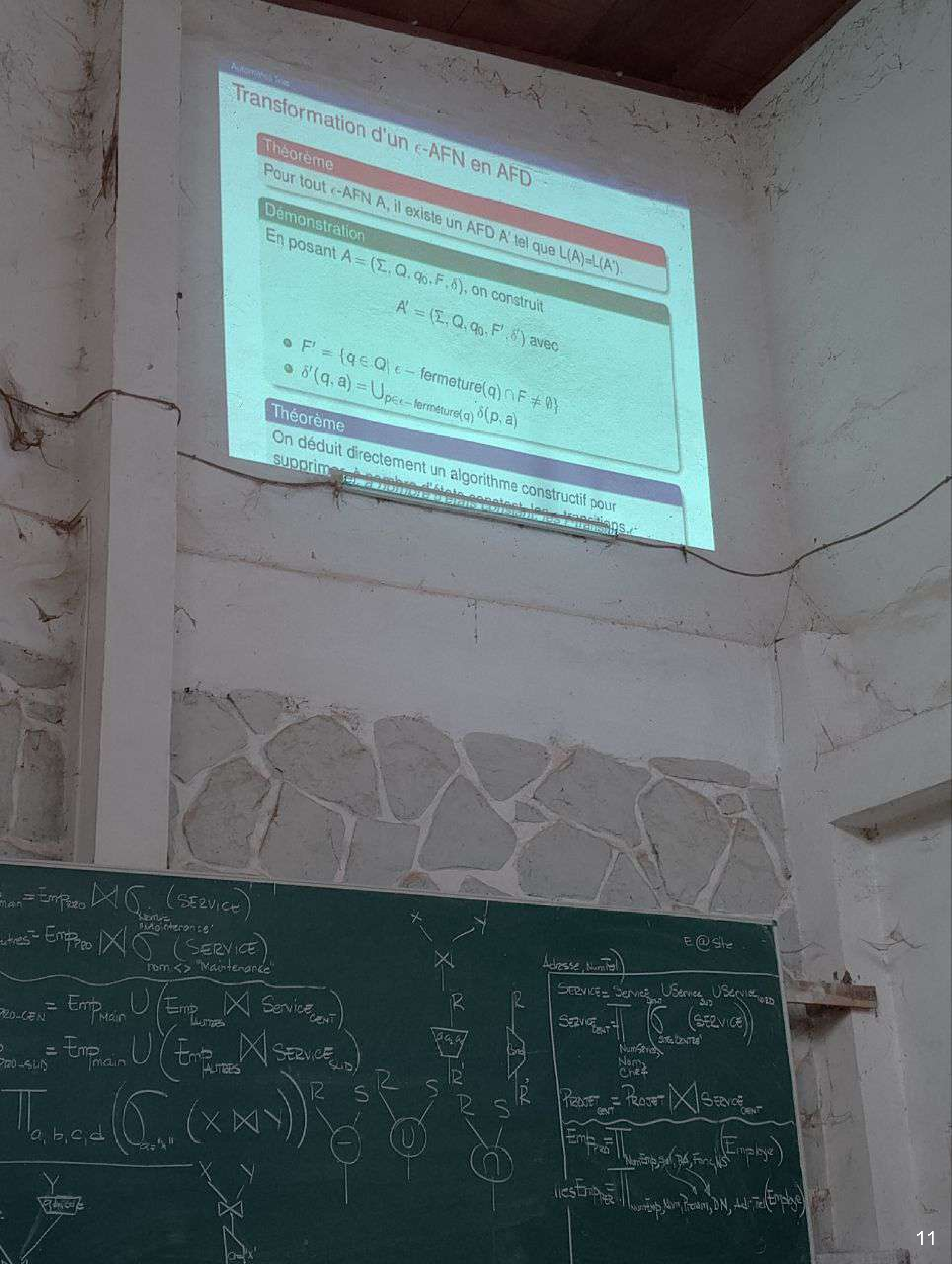

Déterminisation
Algorithme ?
Il existe toutefois des automates pour lesquels l'explosion combinatoire annoncée a lieu, comme celui qui est représenté à la figure ci-dessous.
a, b
0
a
1
a, b
2
a, b
3
a, b
n
Sauriez-vous expliquer d'où vient cette difficulté ?
Quel est le langage reconnu par cet automate ?

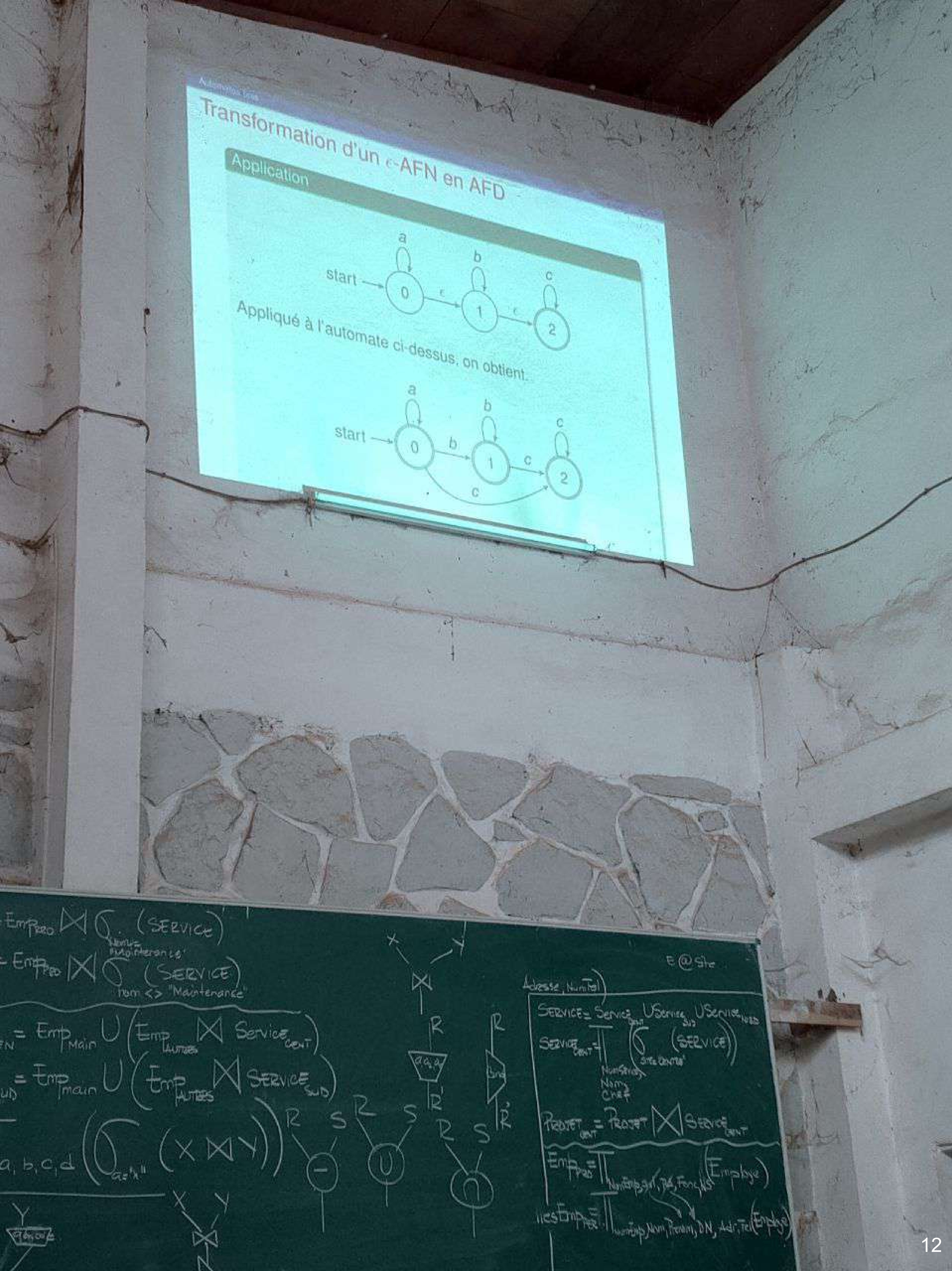

AFN à transitions spontanées (ε-AFN)
Définition informelle
Dans la pratique, il convient souvent d'étendre la notion d'AFN en autorisant des transitions étiquetées par le mot vide ε. De telles transitions sont appelées transitions spontanées.
Un ε-AFN
start
a
b
c
0
1
2
ε
ε
Quel est le langage reconnu par cet automate ???

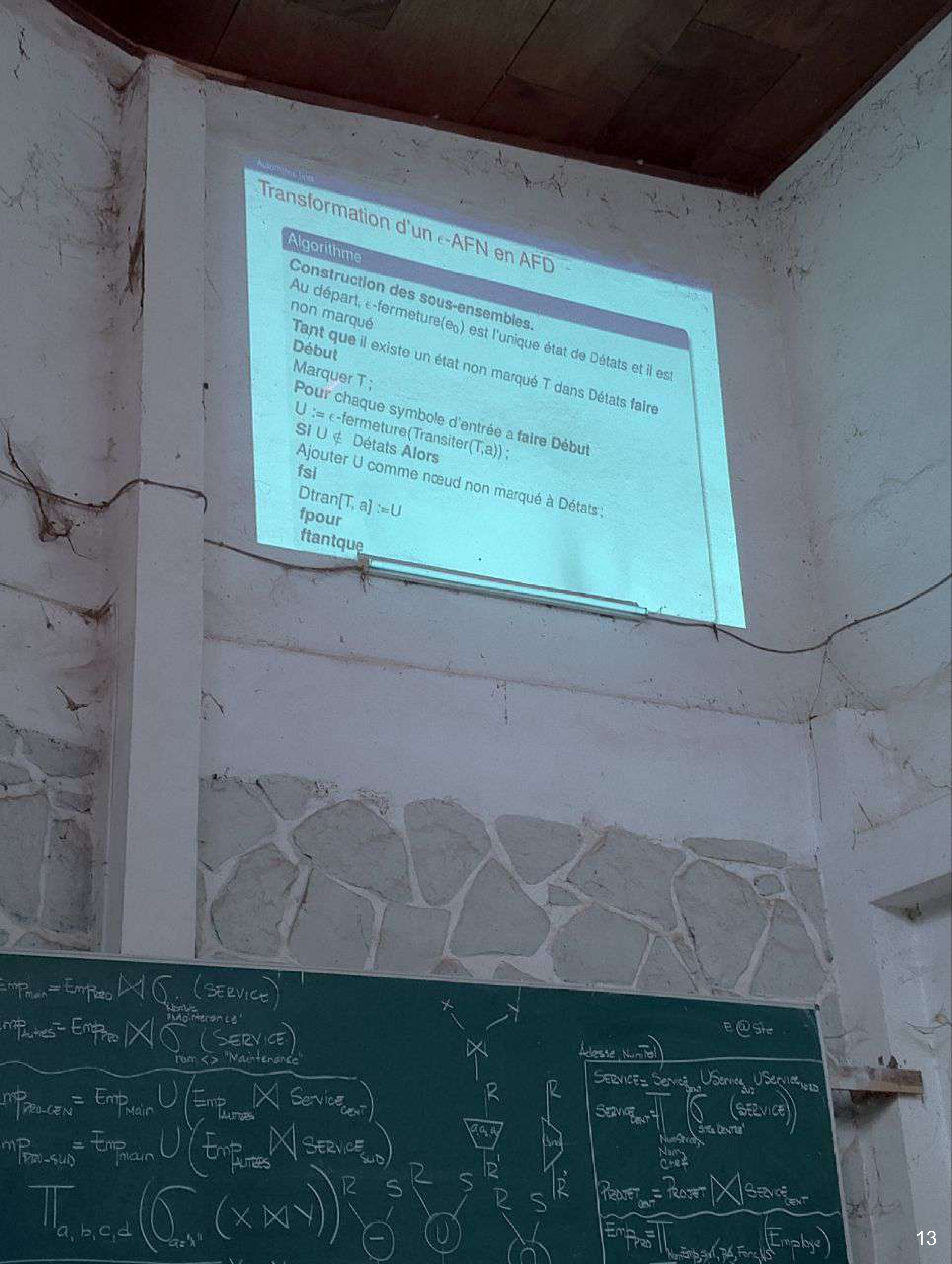

AFN à transitions spontanées (ε-AFN ou ε-AFND)
Définition formelle
Un automate fini (non-déterministe) à transitions spontanées est défini par un quintuplet $A = (\Sigma, Q, I, F, \delta)$ où :
Σ est un ensemble fini de symboles (l'alphabet)
Q est un ensemble fini d'états
$I \subset\in Q$ est l'état initial
$F \subset Q$ sont les états finaux
$\delta \subset Q \times (\Sigma \cup \{\epsilon\}) \times Q$ est une relation, chaque triplet $(q, a, q') \in \delta$ est une transition.
Définition
Formellement, un ε-AFN se définit comme un AFN à la différence près que δ permet d'étiqueter les transitions par ϵ.
E @ Site
SERVICE = Service_CENT U Service_SUD U Service_NORD
PROJET_CENT = PROJET ⋈ SERVICE_CENT

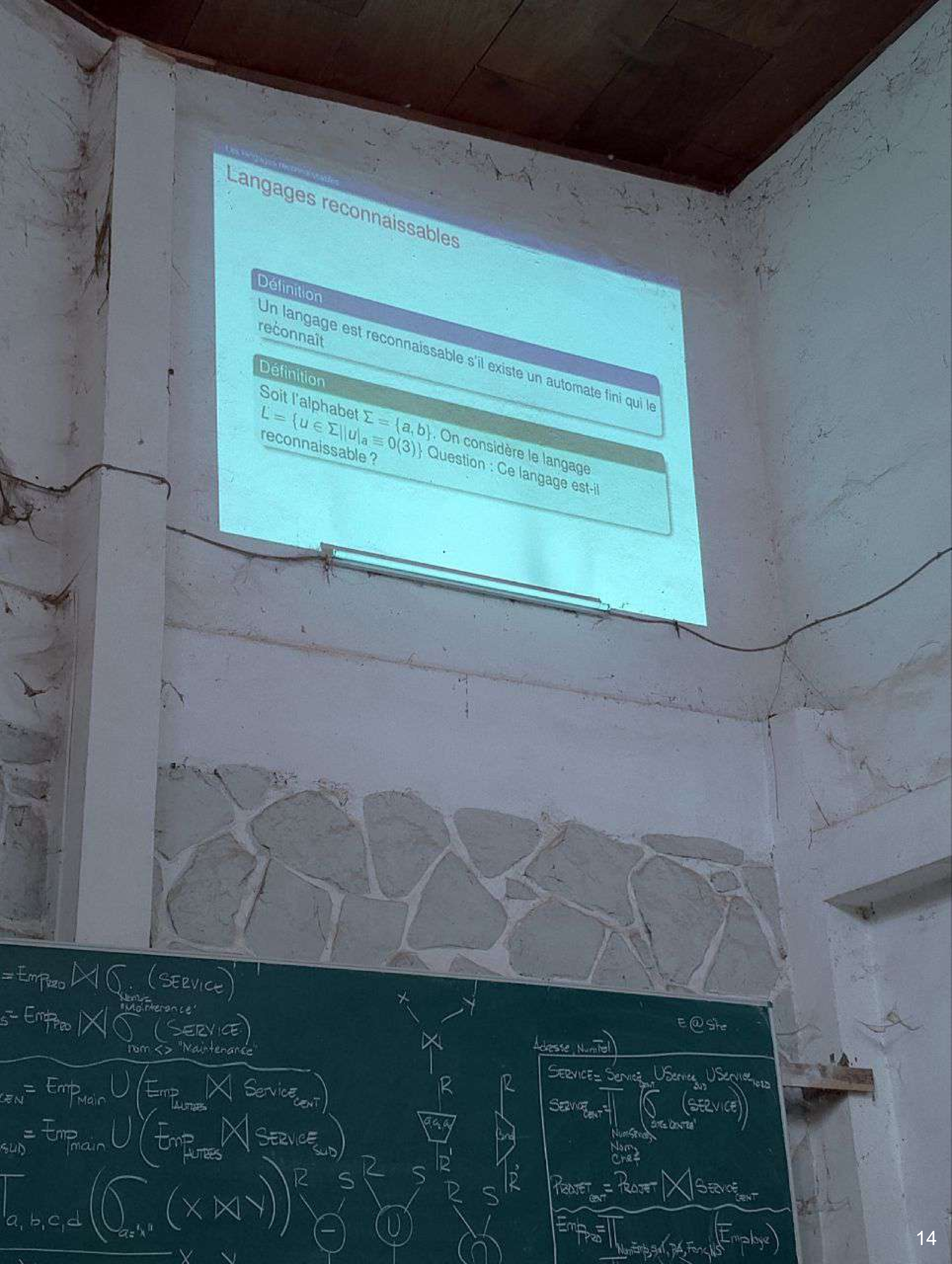

AFN à transitions spontanées (ϵ-AFN ou ϵ-AFND)
Définition
Les ϵ-transitions permettent d'étendre la notion de calcul :
$(q,u) \vdash_A (p,v)$ si (i) $u = av$ et $(q,a,p) \in \delta$ ou
(ii) $u = v$ et $(q,\epsilon,p) \in \delta$
(1)
Dans un ϵ-AFN, il est donc possible de changer d'état sans consommer de symbole, via une ϵ-transition.
Le langage reconnu par un ϵ-AFN A est alors défini par :
$L(A) = \{u \in \Sigma^* \mid \exists (q_0,u) \vdash_A^* (q,\epsilon)$ avec $q_0 \in I, q \in F\}$
Définition
Cette nouvelle extension n'ajoute rien de plus à l'expressivité du modèle : Transformation ϵ-AFN A en un AFN équivalent.


Transformation d'un ϵ-AFN en AFN
Théorème
Pour tout ϵ-AFN A, il existe un AFN A' tel que L(A)=L(A').
OPERATIONS
DESCRIPTIONS
ϵ-fermeture(e)
Ensemble des états de l'AFN accessibles depuis l'état e de l'AFN par des ϵ-transitions uniquement
ϵ-fermeture(T)
Ensemble des états de l'AFN accessibles depuis un état e ∈ T par des ϵ-transitions uniquement :
Transiter(T,a)
Ensemble des états de l'AFN vers lesquels il existe une transition sur le symbole d'entrée a à partir d'un état e ∈ T


Transformation d'un ϵ-AFN en AFD
Théorème
Pour tout ϵ-AFN A, il existe un AFD A' tel que L(A)=L(A').
Démonstration
En posant $A = (\Sigma, Q, q_0, F, \delta)$, on construit
$A' = (\Sigma, Q, q_0, F', \delta')$ avec
$F' = \{q \in Q \mid \epsilon - \text{fermeture}(q) \cap F \neq \emptyset\}$
$\delta'(q, a) = \bigcup_{p \in \epsilon-\text{fermeture}(q)} \delta(p, a)$
Théorème
On déduit directement un algorithme constructif pour
E@Site
Adresse, NumTel


Transformation d'un ε-AFN en AFD
Application
start
a
b
c
0
1
2
ε
Appliqué à l'automate ci-dessus, on obtient.
start
a
b
c
0
1
2
SERVICE
"Maintenance"
Service
SERVICE
E@Site
Adresse, NumTel
Employe


Transformation d'un ϵ-AFN en AFD
Algorithme
Construction des sous-ensembles.
Au départ, ϵ-fermeture(e0) est l'unique état de Détats et il est non marqué
Tant que il existe un état non marqué T dans Détats faire
Début
Marquer T ;
Pour chaque symbole d'entrée a faire Début
U := ϵ-fermeture(Transiter(T,a)) ;
Si U ∉ Détats Alors
Ajouter U comme nœud non marqué à Détats ;
fsi
Dtran[T, a] :=U
fpour
ftantque


Langages reconnaissables
Définition
Un langage est reconnaissable s'il existe un automate fini qui le reconnaît
Définition
Soit l'alphabet $\Sigma = \{a, b\}$. On considère le langage $L = \{u \in \Sigma \mid |u|_a \equiv 0(3)\}$ Question : Ce langage est-il reconnaissable ?