

Autism

En résumé

- 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 détermination
Pour tout AFN A ...

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.

1

As a result of the above, the following is proposed:

Théorème de détermination
Pour tout AFN A défini
Si A a n états

Théorème de détermination
 Pour tout AFD 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 AFD
- par δ' d'

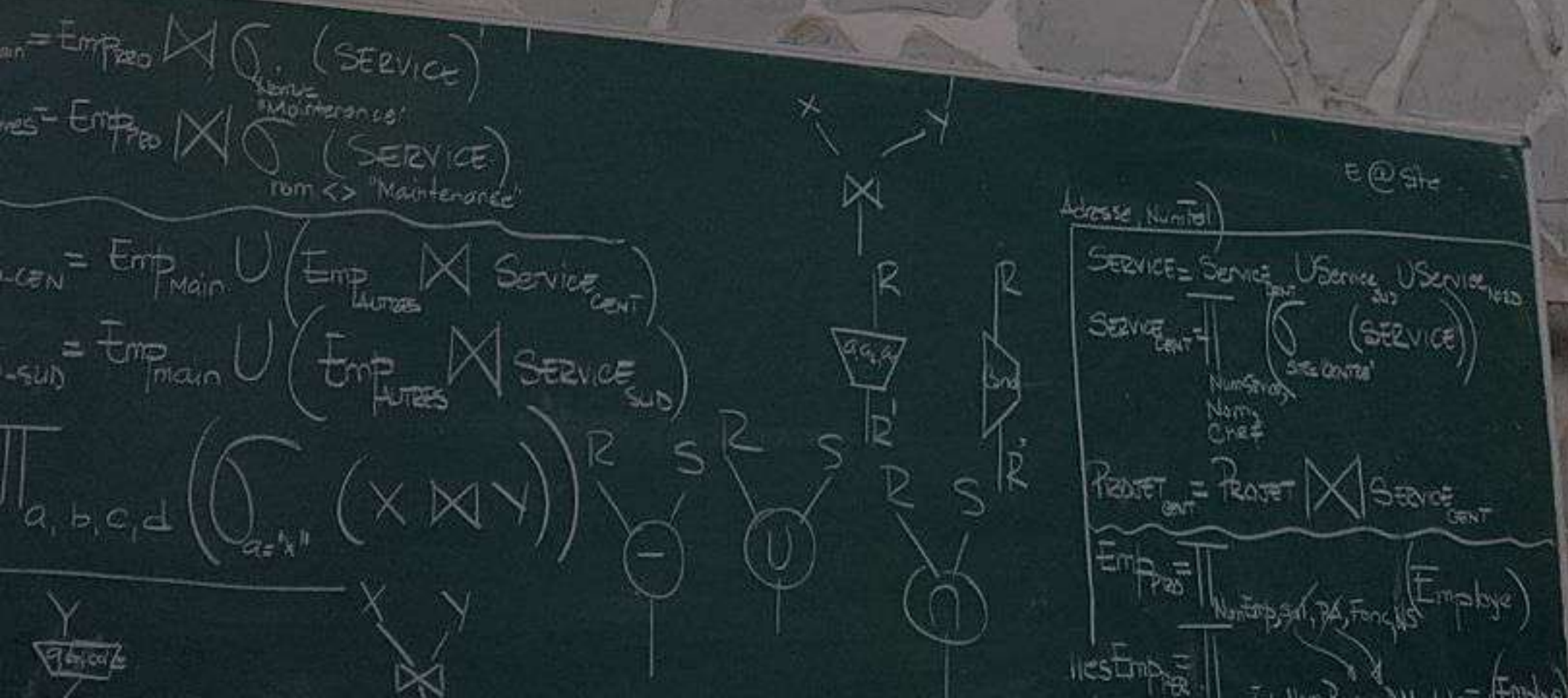
Prove

- 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 $\exists q \in Q$
 $G, (\{I\}, u) \vdash_A^* (G, \epsilon)$
 Si $(\{I\}, u) \vdash_A^* (G, \epsilon)$

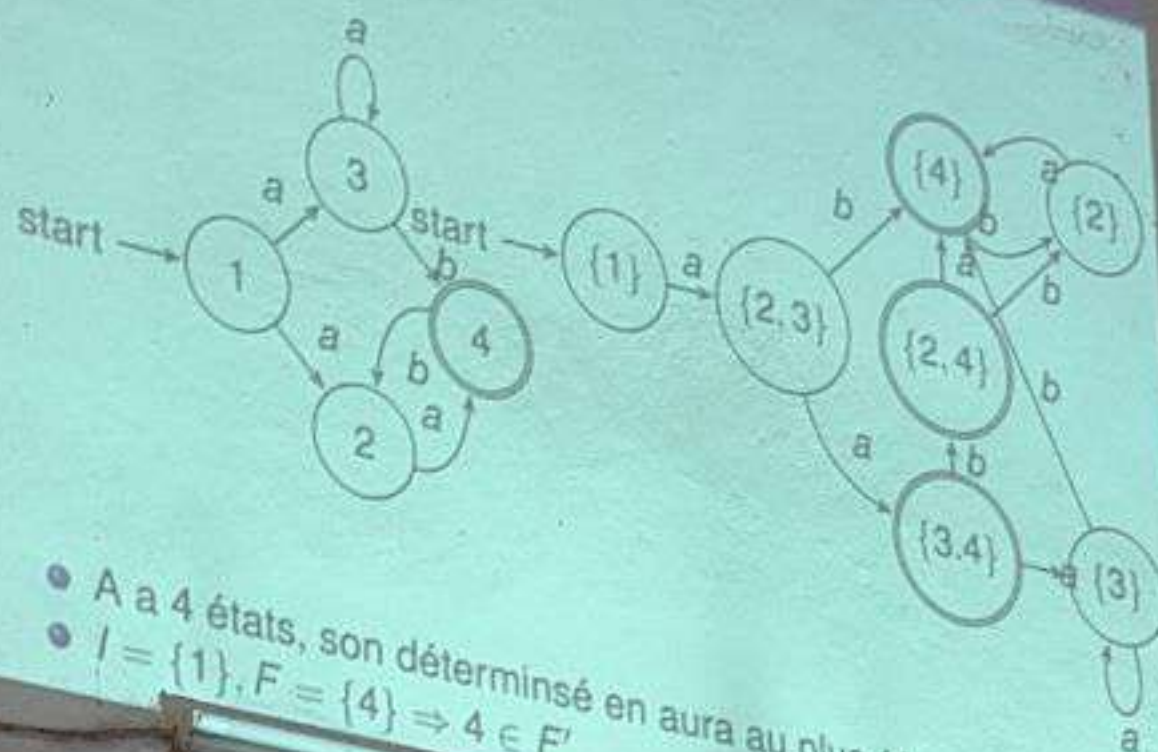
Si $\exists q_0 \in I : (q_0, u) \vdash_A^* (p, \epsilon)$ Alors $\exists G \subset Q : p \in G, (\{I\}, u) \vdash_A^* (G, \epsilon)$
 Si $(\{I\}, u) \vdash_A^* (G, \epsilon)$ Alors $\exists q_0 \in I : \forall p \in G : (q_0, u) \vdash_A^* (p, \epsilon)$
 La démonstration se fait par récurrence sur la longueur de u .

La démonstration

... $\exists q_0 \in I : \forall p \in G : (q_0, u) \vdash_A^* (p, \epsilon)$
 ... se fait par récurrence sur la longueur de



Automates FSA Application



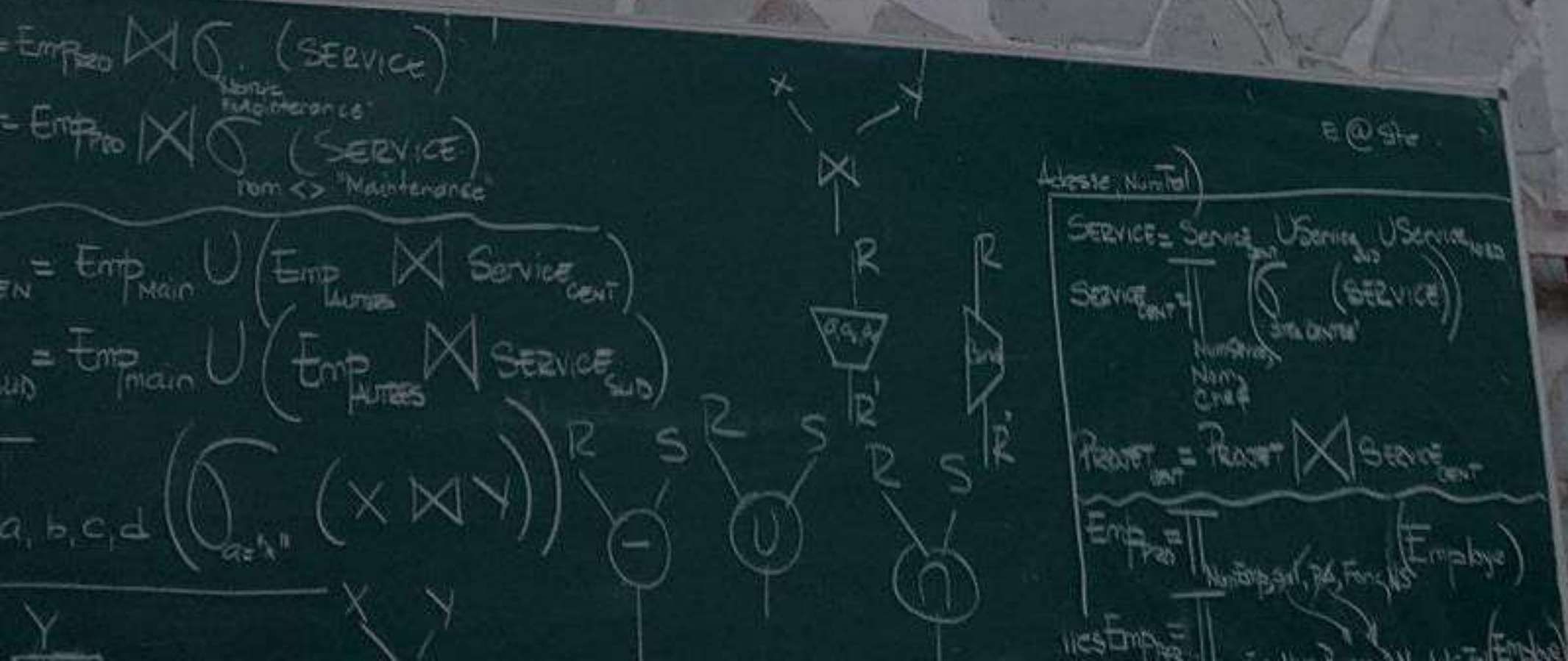
- A a 4 états, son déterminisé en aura au plus 16.
- $I = \{1\}, F = \{4\} \Rightarrow 4 \in F'$



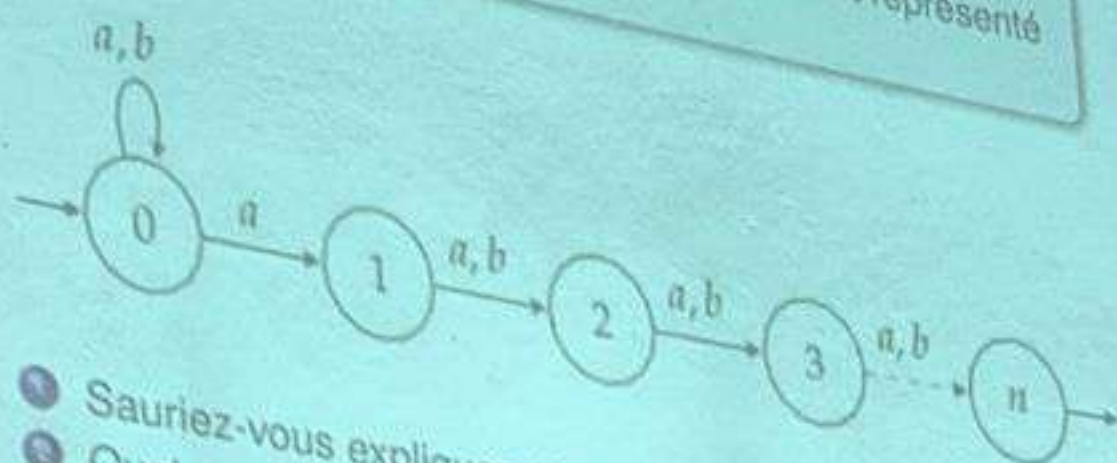
Déterminisation

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 $a_1 a_2 \dots a_n$, 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 $a_1 a_2 \dots a_n$.



Il existe toutefois des automates pour lesquels l'explosion combinatoire annoncée a lieu, comme celui qui est à la figure ci-dessous.



- Sauriez-vous expliquer d'où vient cette difficulté ?
- Quel est le langage reconnu par cet automate ?

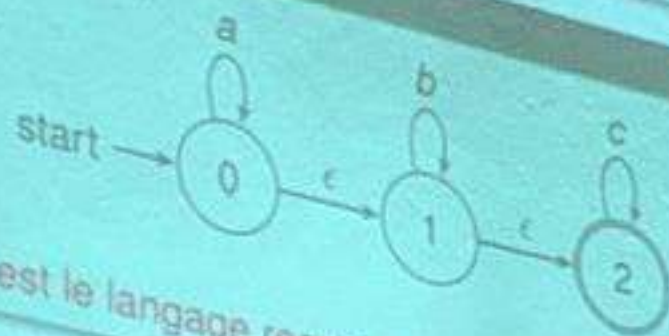
[illegible]

AFN à transitions spontanées (r-AFN)

Définition informelle
Dans la pratique
en c...

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 -AFN



Quel est le langage reconnu par cet automate ?? ?

(SERVICE)
 from "Maintenance"

$Service_{ent} = Service_{ent} \cup Service_{ent}$
 $Service_{ent} = \{ (Service) \}$
 $Project_{ent} = Project_{ent} \times Service_{ent}$

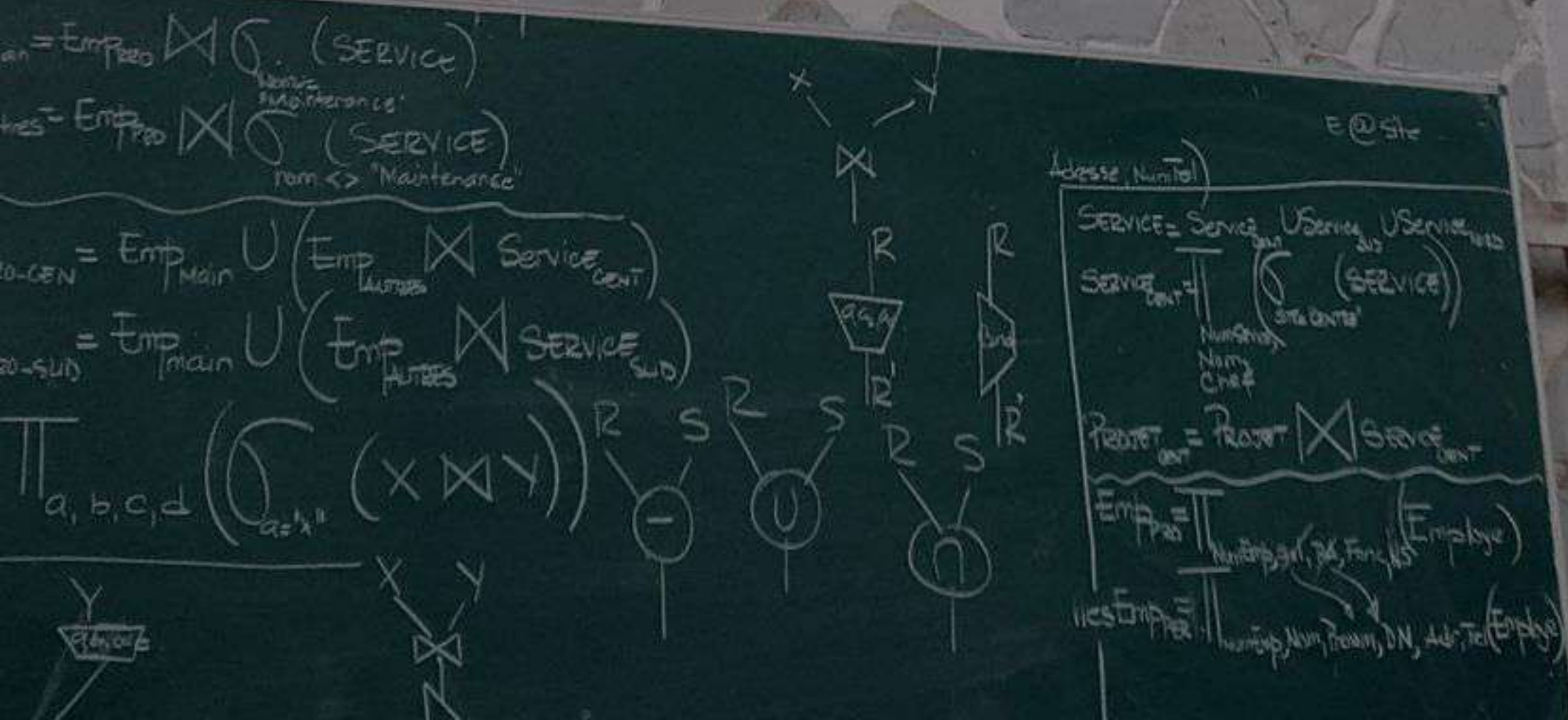
Un auteur

- Σ est un ensemble fini de symboles
- Q est un ensemble fini de symboles
- $I \subseteq Q$

- Q est l'ensemble fini d'états
 - $F \subset Q$ sont les états finaux
 - $\delta \subset Q \times (\Sigma \cup \{c\}) \times Q$ est une relation, chaque triplet $(q, a, q') \in \delta$ est une transition.
- Définition**

Definition

Formellement, un ϵ -AFN se définit comme un AFN à la différence près que δ permet d'étiqueter les transitions par ϵ .



Définition

- Les ϵ -transitions permettent d'étendre la notion de calcul :
 - (i) $u = \epsilon v$ et $(q, \epsilon) \rightarrow (p, v)$
 - (ii) $u = \epsilon v$ et $(q, \epsilon) \rightarrow (p, \epsilon)$

$(q, u) \vdash_A (p, v)$ si (i) $u = av$ et $(q, a, p) \in \delta$ ou
 (ii) $u = v$ et $(q, \epsilon, p) \in \delta$ (1)

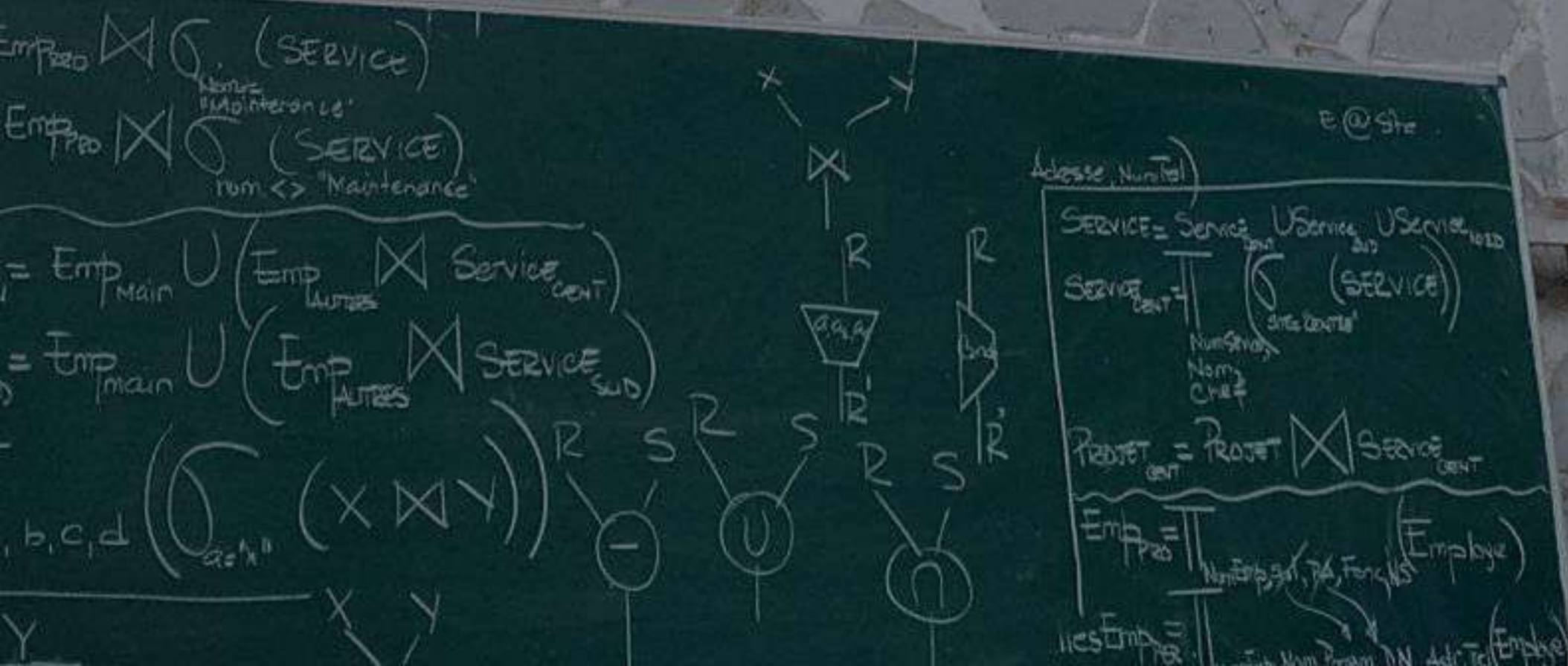
Dans un ϵ -AFN, il est donc possible de changer d'état sans consommer de symbole, via une ϵ -transition.

- Le langage reconnu par un c-AFN A est alors défini par : $L(A) = \{u \in \Sigma^* \mid \exists (q_0, u) \vdash^* (q, \epsilon) \text{ pour un } q \in F\}$

$L(A) = \{u \in \Sigma^* \mid \exists (q_0, u) \vdash_A^* (q, \epsilon) \text{ avec } q_0 \in I, q \in F\}$

Definition

Cette nouvelle extension n'ajoute rien de plus à l'expressivité du modèle : Transformation ϵ -AFN A en un AFN équivalent.

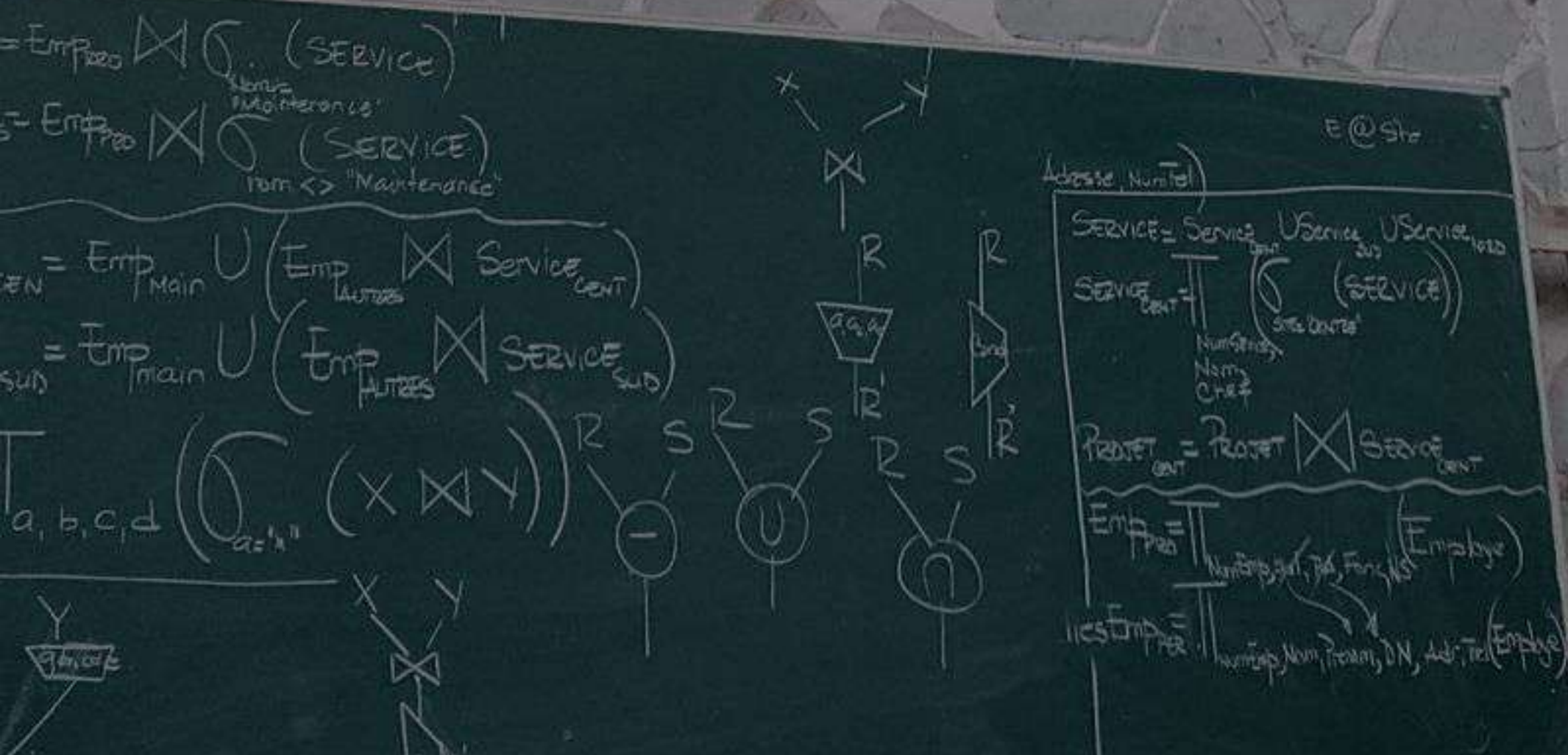


Transformation d'un ϵ -AFN en AFN

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 $\epsilon\text{-fermeture}(e) = \{p \in Q \mid (e, \epsilon) \vdash_A^* (p, \epsilon)\}$
ϵ -fermeture(T)	Ensemble des états de l'AFN accessibles depuis un état $e \in T$ par des ϵ -transitions uniquement : $\epsilon\text{-fermeture}(T) = \{p \in Q \mid \forall e \in T, (e, \epsilon) \vdash_A^* (p, \epsilon)\}$
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 $e \in T$



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

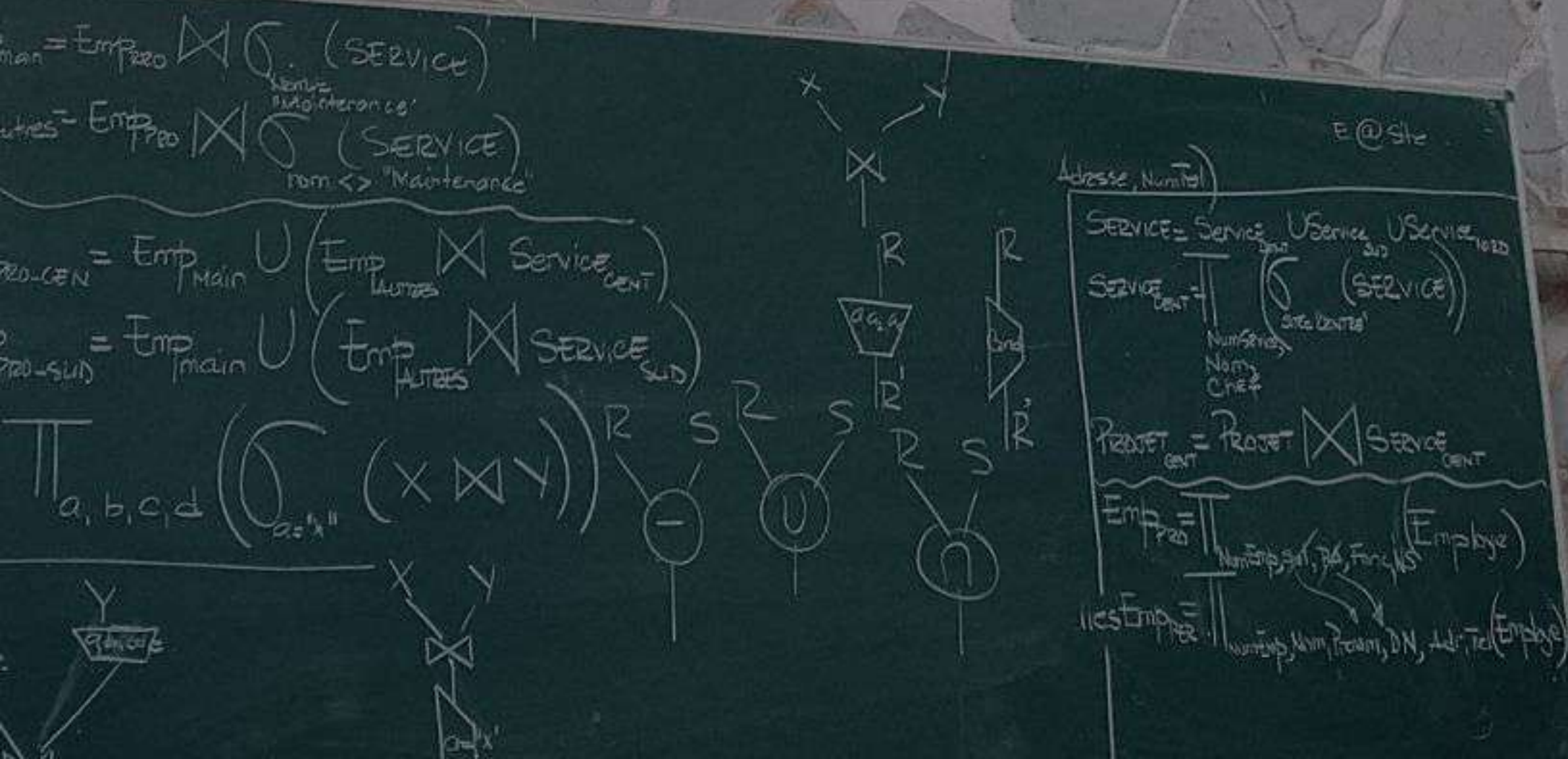
En posant $A = (\Sigma, Q, q_0, F, \delta)$, on construit

$$A' = (\Sigma, Q, q_0, F', \delta') \text{ avec}$$

- $F' = \{q \in Q \mid \epsilon\text{-fermeture}(q) \cap F \neq \emptyset\}$
- $\delta'(q, a) = \bigcup_{p \in \epsilon\text{-fermeture}(q)} \delta(p, a)$

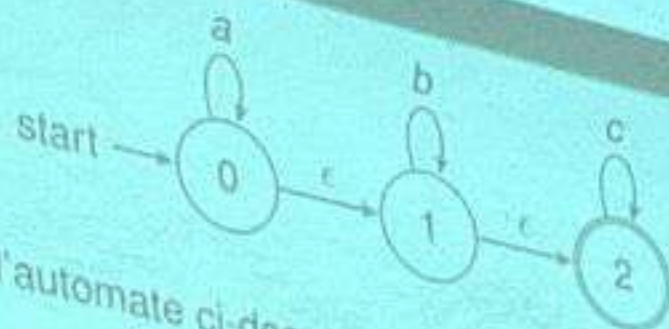
Théorème

On déduit directement un algorithme constructif pour supprimer le nombre d'états constant les transitions.

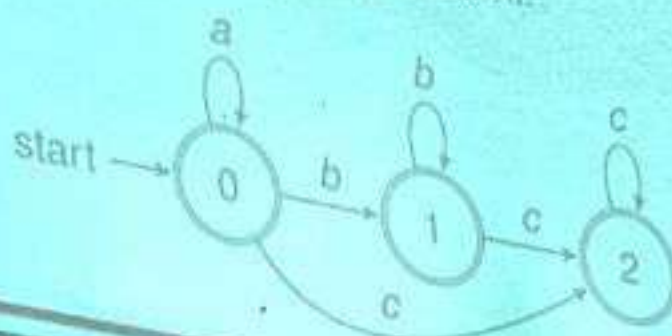


Transformation d'un ϵ -AFN en AFD

Application



Appliqué à l'automate ci-dessus, on obtient.



Emp_{pro} \times (SERVICE)
 Emp_{pro} \times (SERVICE)
 nom \leftrightarrow "Maintenance"

$\text{Emp}_{\text{main}} \cup (\text{Emp}_{\text{autres}} \times \text{Service}_{\text{cent}})$

$\text{Emp}_{\text{main}} \cup (\text{Emp}_{\text{autres}} \times \text{SERVICE}_{\text{SUD}})$

$a, b, c, d \left(\begin{matrix} \text{ } \\ \text{ } \end{matrix} \right) (x \times y) \begin{matrix} R \\ S \end{matrix}$

$a = 'x'$



Adresse, NumTel

SERVICE = Service_{cent} \cup Service_{sud} \cup Service_{autres}
 SERVICE_{cent} = $\left(\begin{matrix} \text{ } \\ \text{ } \end{matrix} \right) \text{ (SERVICE)}$
 NonServ, NonChef

PROJET_{cent} = PROJET \times SERVICE_{cent}

Emp_{pro} = $\left(\begin{matrix} \text{ } \\ \text{ } \end{matrix} \right) \text{ (Employe)}$

les Emp_{pro} = $\left(\begin{matrix} \text{ } \\ \text{ } \end{matrix} \right) \text{ (Employe)}$

Construction des sous-ensembles.
Au départ, ε -fermeture(e_0) est l'ensemble des états accessibles à partir de e_0 en ne franchissant que des transitions étiquetées ε .
Tant qu'il existe un état e appartenant à ε -fermeture(e_0) tel que $e \xrightarrow{a} e'$ pour un $a \in \Sigma$, on ajoute e' à ε -fermeture(e_0).

Début

Pour chaque symbole d'entrée à faire Début
 $U := \epsilon$ -fermeture(Transiter(T, a)) ;
 Si $U \neq \emptyset$ Alors
 Ajouter U

Ajouter Un

tsi

pour

stantque

Adresse, Nummer)
 SERVICE = Service, UserService, UserService, ...
 SERVICE_{cent} = (SERVICE)
 (Service, UserService, UserService, ...)
 NumStrich,
 Name,
 Chef
 PROJET = PROJET \times SERVICE_{cent}
 Emp_{pas} = (Employee)
 NumStrich, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 8

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 ?

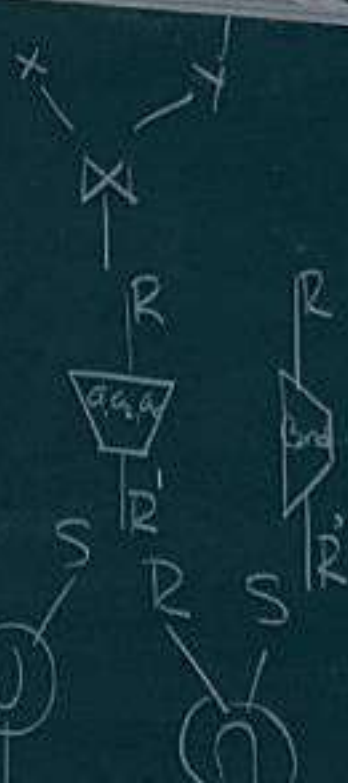
$= Emp_{P20} \times (SERVICE)$

$= Emp_{P20} \times (SERVICE)$
nom <> "Maintenance"

$= Emp_{main} \cup (Emp_{AUTRES} \times Service_{CENT})$

$= Emp_{main} \cup (Emp_{AUTRES} \times SERVICE_{SUD})$

$T_{a,b,c,d}((a=x)(x \times y))$
 $a=x$



Adresse, NumTel)

$SERVICE = Service_{CENT} \cup Service_{SUD}$
 $SERVICE_{CENT} = \{ (SERVICE) \}$
NumServ, Nom, Chà

$PROJET_{CENT} = PROJET \times SERVICE_{CENT}$

$Emp_{P20} = \{ (Employee) \}$
NumEmp, Sal, PA, Fonct