

LF – Théorie des langages formels

Sylvain Brandel

2022 – 2023

sylvain.brandel@univ-lyon1.fr



CM 4

AUTOMATES À ÉTATS FINIS

Automates à états finis

- Automate ou Machine
 - *Ang. Automaton / Machine*
- Automates ou Machines
 - *Ang. Automata / Machines ...*
- Automates (à états) finis déterministes
 - *Ang. Deterministic Finite Automata (DFA)*
- Automates (à états) finis non déterministes
 - *Ang. Nondeterministic Finite Automata (NFA)*
- Automates à pile
 - *Ang. Pushdown Automata (PDA)*
- Machines de Turing
 - *Ang. Turing Machines*

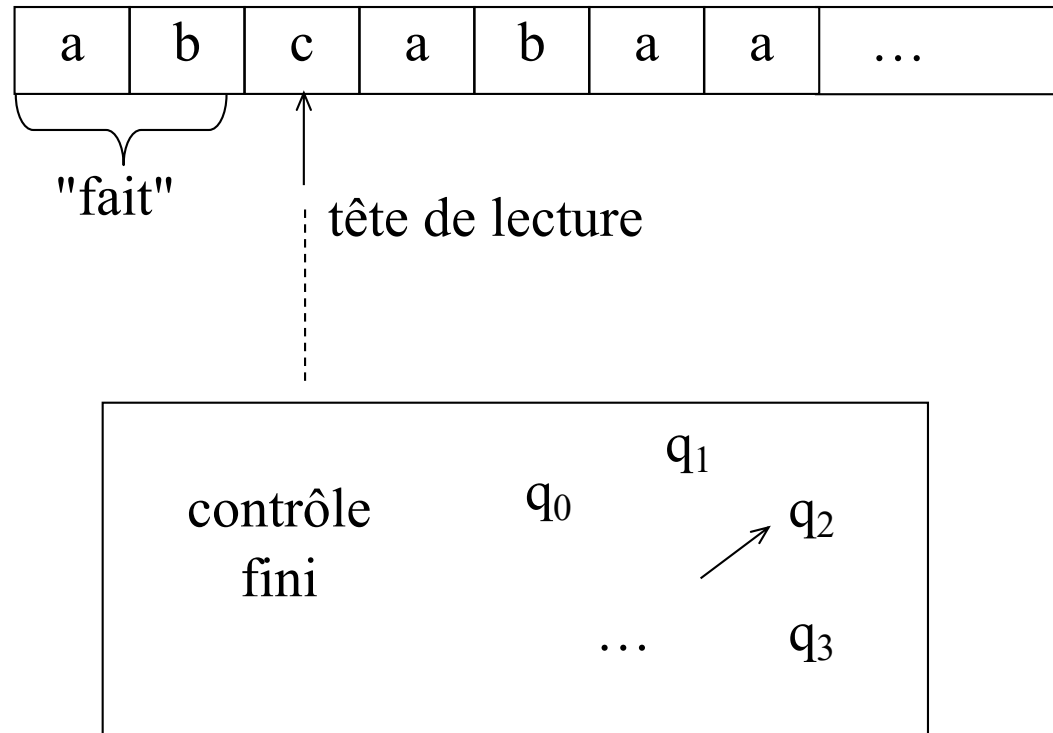
Automates à états finis

- Exemple concret
 - Machine à café dans le hall du déambu (pas sans contact)
 - Barrière de péage du périph' (pas liber-t ...)
 - Laverie (pas sans contact)
 - ...

Automates finis déterministes

- Simulation d'une machine très simple :
 - Mémorisation d'un **état**
 - Programme sous forme de graphe étiqueté indiquant les **transitions** possibles
 - Cette machine lit un **mot** en entrée
 - Ce mot décrit une suite d'actions et progresse d'état en état
 - Si le dernier état est un **état acceptant** et que le mot a été entièrement lu, on dit que le mot est accepté
- ⇒ Un automate permet de **reconnaître** un langage

Automates finis déterministes



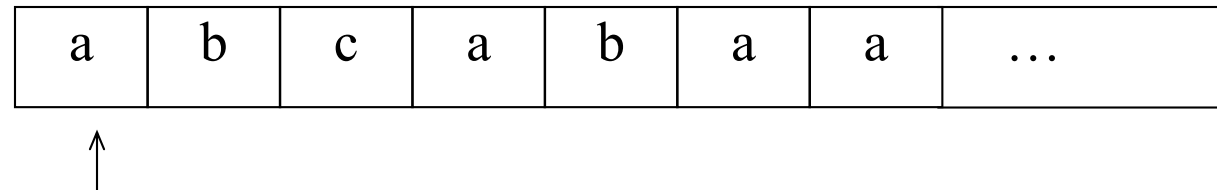
- Un état dépend uniquement
 - De l'état précédent
 - Du symbole lu

Automates finis déterministes

- Un automate **déterministe** fini est le quintuplet $M = (K, \Sigma, \delta, s, F)$ où :
 - K : ensemble fini (non vide) d'états
 - Σ : alphabet (ensemble non vide de symboles)
 - δ : **fonction** de transition : $K \times \Sigma \rightarrow K$
$$\delta(q, \sigma) = q' \quad (q' : \text{état de l'automate après avoir lu la lettre } \sigma \text{ dans l'état } q)$$
 - s : état initial : $s \in K$
 - F : ensemble des états finaux (**acceptants**) : $F \subset K$
- Si δ est une **application**, alors l'automate est complet

Automates finis déterministes

- Exécution



La machine

- lit a (qui est ensuite oublié),
 - passe dans l'état $\delta(s, a)$ et avance la tête de lecture,
 - répète cette étape jusqu'à ce que tout le mot soit lu ou plus de transition applicable
- La partie déjà lue du mot ne peut pas influencer le comportement à venir de l'automate
 - d'où la notion de **configuration**

Automates finis déterministes

- Configuration
 - état dans lequel est l'automate
 - mot qui lui reste à lire (partie droite du mot initial)
- Formellement
 - une configuration est un élément quelconque de $K \times \Sigma^*$
- Exemple
 - sur l'exemple précédent, la configuration est $(q_2, cabaa)$

Automates finis déterministes

- Le fonctionnement d'un automate est décrit par le passage d'une configuration C_1 à une configuration C_2
- C_2 est déterminée
 - en lisant un caractère
 - et en appliquant la fonction de transition
- Exemple
 - $(q_2, cbaaa) \rightarrow (q_3, abaa)$ si $\delta(q_2, c) = q_3$

Automates finis déterministes

- Un automate M détermine une relation binaire \vdash_M entre configurations définie par :
 - $\vdash_M \subset (K \times \Sigma^*)^2$
 - $(q, w) \vdash_M (q', w')$ ssi $\exists \sigma \in \Sigma$ tel que $w = \sigma w'$
et $\delta(q, \sigma) = q'$
- On dit alors qu'on passe de (q, w) à (q', w') **en une étape**

Automates finis déterministes

- On note \vdash_M^* la fermeture transitive réflexive de \vdash_M :

$(q, w) \vdash_M^* (q', w')$ signifie qu'on passe de (q, w) à (q', w') en zéro, une ou plusieurs étapes

- Un mot w est **accepté** par M ssi $(s, w) \vdash_M^* (q, \varepsilon)$, avec $q \in F$

- Le langage accepté par M est l'ensemble de tous les mots acceptés par M

Ce langage est noté $L(M)$

Automates finis déterministes

- Exemple $M = (K, \Sigma, \delta, s, F)$

- $K = \{q_0, q_1\}$

- $\Sigma = \{a, b\}$

- $s = q_0$

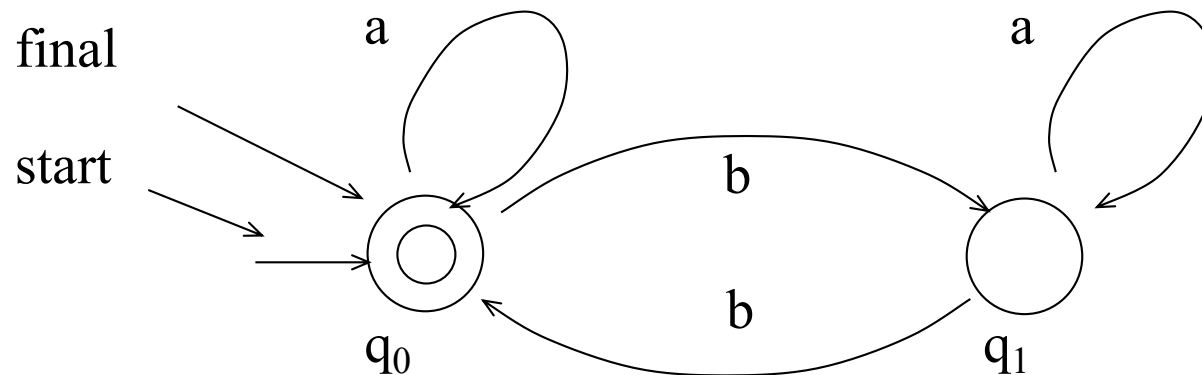
- $F = \{q_0\}$

$$\delta :$$

q	σ	$\delta(q, \sigma)$
q_0	a	q_0
q_0	b	q_1
q_1	a	q_1
q_1	b	q_0

\Leftrightarrow

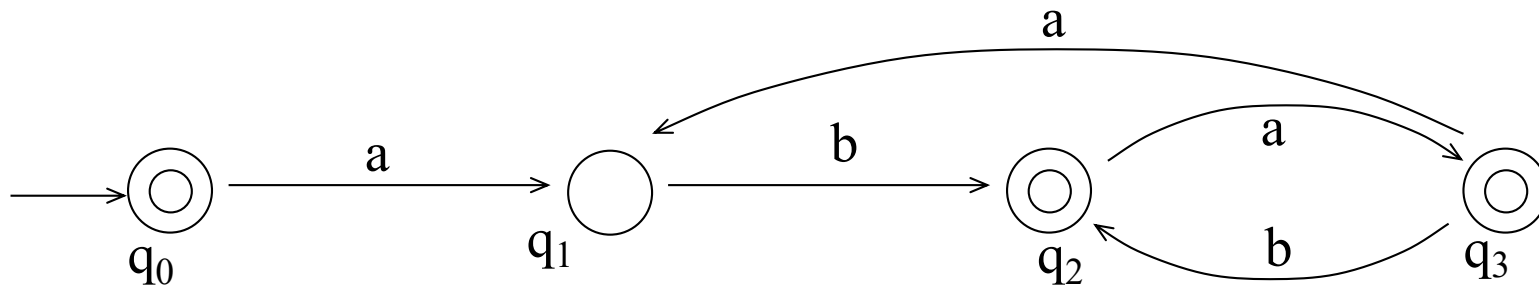
	a	b
q_0	q_0	q_1
q_1	q_1	q_0



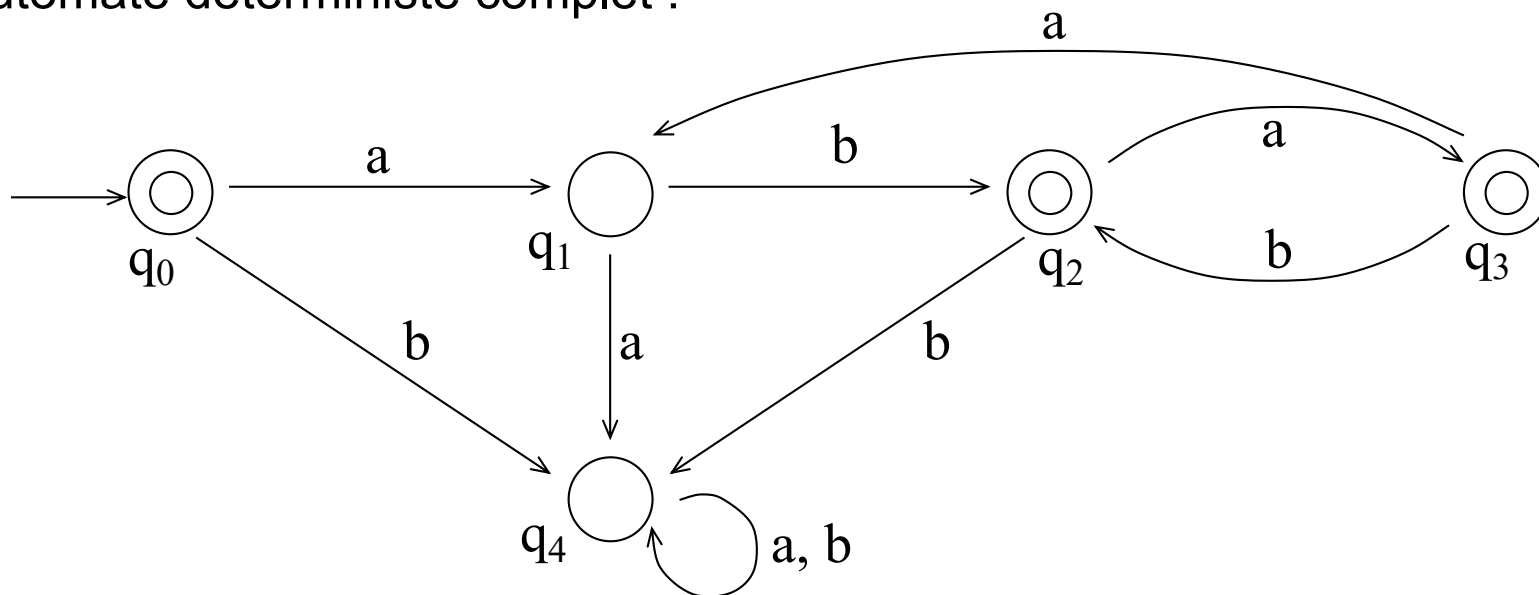
Automates finis déterministes

- $L = (ab \cup aba)^*$

Automate déterministe non complet :



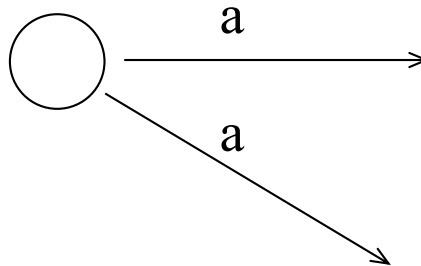
Automate déterministe complet :



Automates finis non déterministes

- Remplacer la **fonction** \vdash_M (ou δ) par une **relation**
- Une relation, c'est beaucoup plus général qu'une fonction
→ on a ainsi une classe plus large d'automates

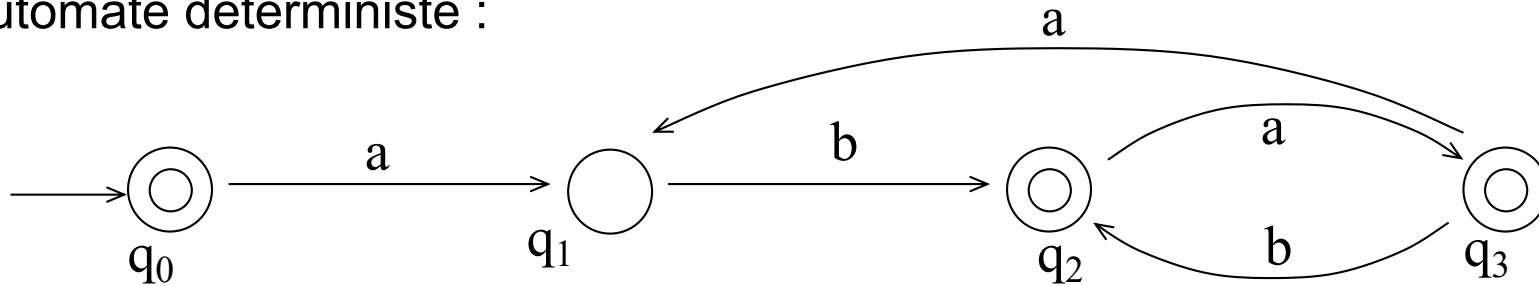
⇒ Dans un état donné, on pourra avoir :



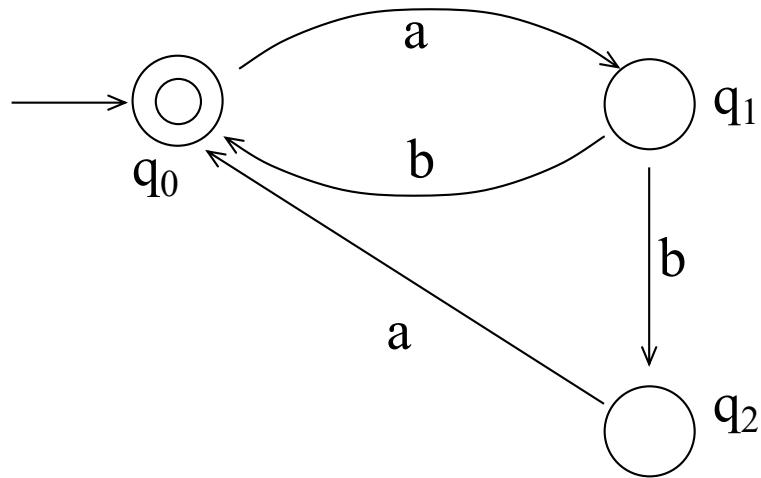
Automates finis non déterministes

- $L = (ab \cup aba)^*$

Automate déterministe :



Automate non déterministe :



Automates finis non déterministes

- Un automate **non déterministe** fini est le quintuplet $M = (K, \Sigma, \Delta, s, F)$ où
 - K : ensemble fini (non vide) d'états
 - Σ : alphabet (ensemble non vide de symboles)
 - Δ : **relation** de transition : $K \times \Sigma \times K$
 $(q, \sigma, q') \in \Delta$: σ –transition ($\sigma \in \Sigma$)
 - s : état initial : $s \in K$
 - F : ensemble des états finaux (**acceptants**) : $F \subset K$
- hormis Δ , le reste est identique à la formulation déterministe
- Δ peut-être définie comme une **application** $K \times \Sigma \rightarrow P(K)$

Automates finis non déterministes

- Une configuration est un élément de $K \times \Sigma^*$.
- Un automate M détermine une relation binaire \vdash_M entre configurations définie par :
 - $\vdash_M \subset (K \times \Sigma^*)^2$
 - $(q, w) \vdash_M (q', w')$ ssi $\exists \sigma \in \Sigma$ tel que $w = \sigma w'$
et $(q, \sigma, q') \in \Delta$

Automates finis non déterministes

- \vdash_M est une **relation** et non plus une fonction (automates déterministes)

Pour une configuration (q, w) , il peut y avoir plusieurs configurations (q', w') (ou aucune) tel que $(q, w) \vdash_M (q', w')$

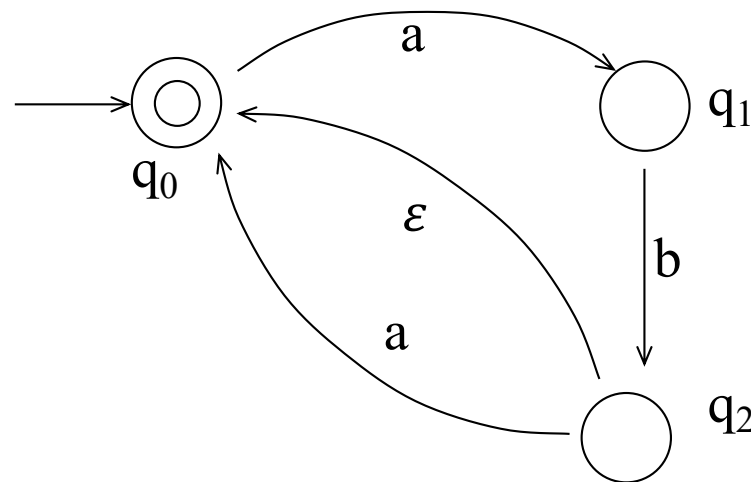
- On note comme avant \vdash_M^* la fermeture transitive réflexive de \vdash_M

- Un mot w est **accepté** par M ssi $(s, w) \vdash_M^* (q, \varepsilon)$, avec $q \in F$

- $L(M)$ est le langage de tous les mots acceptés par M

Automates finis non déterministes avec transitions spontanées

- Ajout de transitions vides
 - Il est possible d'étiqueter une flèche par le symbole ε .
- Autre formulation encore plus intuitive (?) de l'automate précédent :



Automates finis non déterministes avec transitions spontanées

- Un automate **non déterministe** fini **avec transitions spontanées** est le quintuplet $M = (K, \Sigma, \Delta, s, F)$ où :
 - K : ensemble fini (non vide) d'états
 - Σ : alphabet (ensemble non vide de symboles)
 - Δ : **relation** de transition : $K \times (\Sigma \cup \{\varepsilon\}) \times K$
 $(q, \sigma, q') \in \Delta$: σ –transition ($\sigma \in \Sigma$)
 - s : état initial : $s \in K$
 - F : ensemble des états finaux (**acceptants**) : $F \subset K$
- hormis la relation, la définition est identique à la formulation déterministe et à la formulation non déterministe

Automates finis non déterministes avec transitions spontanées

- Si $(q, \varepsilon, q') \in \Delta$: on a une ε – transition (transition spontanée)
→ On passe de q à q' sans lire de symbole dans le mot courant.
- Une configuration est un élément de $K \times \Sigma^*$
- M décrit une relation binaire entre configurations qu'on note \vdash_M :
 $(q, w) \vdash_M (q', w')$
ssi il existe un mot d'au plus une lettre $u \in \Sigma \cup \{\varepsilon\}$
tel que $w = uw'$
et $(q, u, q') \in \Delta$

Automates finis non déterministes avec transitions spontanées

- \vdash_M est une **relation** et non plus une fonction (automates déterministes)
 - (q, ε) peut être en relation avec une autre configuration (après une ε – transition)
 - pour une configuration (q, w) , il peut y avoir plusieurs configurations \vdash_M (ou aucune) telles que $(q, w) \vdash_M (q', w')$
- On note comme avant \vdash_M^* la fermeture transitive réflexive de \vdash_M
- Un mot w est **accepté** par M ssi $(s, w) \vdash_M^* (q, \varepsilon)$, avec $q \in F$
- $L(M)$ est le langage de tous les mots acceptés par M

Retour sur les grammaires algébriques

- Théorème

Un langage est rationnel si et seulement si il est engendré par une grammaire régulière

- Ce théorème exprime que tout langage rationnel est algébrique (et non l'inverse)
- On peut utiliser la preuve de ce théorème pour :
 - Passer d'un automate (déterministe ou non) à une grammaire
 - Passer d'une grammaire à un automate

Langages rationnels et grammaires algébriques

- Exemple ($G \rightarrow M$)

$G = (V, \Sigma, R, S)$ une grammaire régulière avec :

– $V = \{S, X, Y\}$

– $\Sigma = \{a, b\}$

– $R = \{S \rightarrow aX \mid bY, X \rightarrow aS \mid a, Y \rightarrow bS \mid b\}$

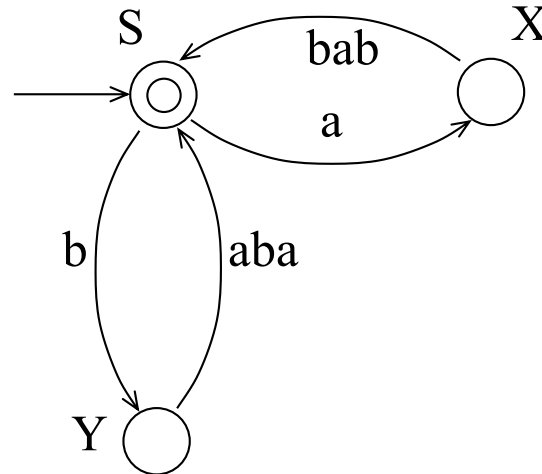
- Soit M tel que $L(M) = L(G)$. Pour chaque non-terminal de G on crée un état dans M de la façon suivante :

– Si $A \rightarrow wB \in R$ alors on crée dans M une transition de l'état A vers l'état B étiquetée par w

– Si $A \rightarrow w \in R$ alors on crée dans M une transition de l'état A vers l'état F , où F est le seul état introduit dans M qui ne correspond à aucun non-terminal de G

Langages rationnels et grammaires algébriques

- Exemple ($M \rightarrow G$)
Soit l'automate :



- Soit G une grammaire régulière telle que $L(G) = L(M)$. Pour chaque transition de M on crée une règle dans G de la façon suivante :
 - Pour toute transition de l'état p vers l'état q étiquetée par w , on crée la règle correspondante dans G : $P \rightarrow wQ$
 - Pour tout état final f de M , on crée dans G une règle d'effacement : $F \rightarrow \varepsilon$

Et maintenant ?

- Déterministe ou non déterministe ?
 - Même classe de langages reconnus
 - Donc équivalents
- Langages rationnels et langages reconnus par les automates finis ?
 - Même classe de langages
- Un langage est-il rationnel ?
 - Preuve de rationalité
- Un automate est-il minimal ?
 - Automate standard