

## Le séquenceur pour faire de la simulation

### 1) Introduction : effectuer des simulations

#### 1.A) Présenter succinctement notre démarche

Dans le quotidien, faire de la simulation à propos du monde réel consiste à imaginer comment son état évolue au fil du temps, sous l'action des agents qui agissent sur lui.

Nous venons d'étudier les séquenceurs, qui sont des Systèmes Formels (SF) très simples, et tout à l'heure, nous les utiliserons pour faire de la simulation, car ils s'avèrent un bon outil pour cela.

Cependant, au préalable, posons quelques jalons, afin d'étudier les bases qui, ensuite, nous seront nécessaires dans ce chapitre.

#### a) Partir des mathématiques

Dans ce cours d'IA, nous sommes partis du domaine mathématique des SF, car ils constituent l'outil formel de base pour étayer une démonstration rigoureuse.

#### b) Les séquenceur : des SF très simples

Ensuite, nous avons introduit les séquenceurs en les présentant comme une restriction des SF. Ainsi nous obtenons un outil très simple, mais déjà formel ; et rapidement nous l'utiliserons pour simuler, car il est adapté à cette tâche.

#### c) Implémentation électronique des séquenceurs

Maintenant, notre démarche consiste à faire le lien entre les séquenceurs et le quotidien (psychologie, linguistique, sociologie...). En effet, nous travaillons à rendre nos outils plus concrets, i.e. à montrer comment le cycle d'un séquenceur peut s'implémenter concrètement.

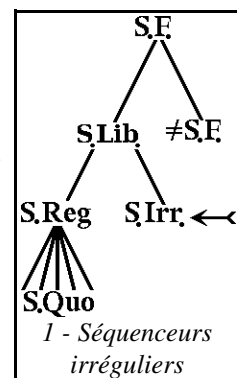
À la fin du chapitre précédent, nous décrivons la première étape de cette implémentation. Elle consiste à montrer comment ce travail peut s'effectuer au moyen d'un circuit logique électronique. Nous montrons que les séquenceurs pouvaient être implémentés selon deux architectures : les irréguliers en extension, et les réguliers en compréhension. Ce faisant, nous ouvrons la voie à des implémentations encore plus concrètes des séquenceurs, sous forme de neurones formels.

#### d) Évacuer les séquenceurs irréguliers

Dans la famille des séquenceurs, apparaissent ceux qui sont irréguliers, par exemple, la donnée d'un itinéraire arbitraire le long d'un chemin. Ils fonctionnent selon un domaine biscornu, sinueux, que nous devons définir en extension. Mais ce mode de définition est peu pratique, car il nous oblige à lister exhaustivement tous les états du système. Finalement, nous travaillerons rarement avec eux, et préférons utiliser les S.Réguliers.

#### e) Travailler avec les séquenceurs réguliers (S.Réguliers)

Les S.Réguliers possèdent un comportement souvent très simple car il est définissable en compréhension : leur domaine peut être défini par itération. Leurs transitions, (changements d'états) peuvent être définis par le simple moyen de règles. Par exemple, c'est le cas de  $\mathbb{N}$ , l'ensemble des entiers naturels [Pea 89] [Pea 67]).



Pour ces raisons, nous choisissons de travailler avec eux. C'est pourquoi, dans le chapitre précédent, nous en avons décrits précisément six.

### f) Jusqu'aux Séquenceurs Quotidiens (S.Quotidiens)

Notre but final est d'utiliser les *S.Quotidiens* pour modéliser quelques scènes simples de la vie de tous les jours. Exemple :

lieu 4,	Le lieu de cette entité est 4.
taille 3,	La taille de cette entité est 3.
vitesse 12,	La vitesse de cette entité est 12.

...

Dans les exemples de *S.Quotidiens* que nous explorerons, le domaine est tellement régulier que chaque état du système se réduit à une valeur typée. Elle se décrit au moyen de 2 attributs, i.e. de 2 facettes : la facette *valeur* et la facette *type*.

#### D'un côté, la facette *valeur*

Les états de ce domaine sont très réguliers alors ils peuvent être définis itérativement. De cette propriété, il ressort qu'ainsi, l'état du système est souvent une valeur, et plus particulièrement un numérique.

#### De l'autre côté, la facette *type*

Chaque facette *type* est constante. Elle indique un trait invariant de l'état du système, et ainsi, introduit une restriction de type : le système prend ses états au sein d'un domaine de variation où tous ces éléments possèdent un trait en commun (au sein d'une classe, d'une famille). Par exemple, tous sont des lieux ; ou tous sont des tailles ; tous, des longueurs...

## 1.B) Présentation des S.Réguliers

Tout d'abord, les *S.Réguliers* servent à généraliser la description de la vie courante. Ainsi, ils ne fournissent qu'un unique attribut : la *valeur* de l'état du système.

Ensuite, cette facette possède la nature d'une *valeur* : elle sert à une démarche de quantification.

### a) La facette valeur est instanciée par un scalaire

#### Démarche de quantification

Cette facette valeur est instanciée par un scalaire, i.e. par un nombre qui sert à quantifier. L'apparition de scalaires utilisés pour décrire l'état du système tient au fait que nous travaillons sur un domaine très régulier. Il est construit itérativement, il doit être ordonné, complet, invariant par translation pour les opérations de quantification.

Par exemple, si le système est dans l'état 4, et si chaque transition augmente la valeur de son état d'un intervalle élémentaire valant 1, après 2 transitions, il doit se trouver dans l'état 6 ( $4+2=6$ ).

Et de façon plus générale, si le système est dans l'état quantifié  $N$ , et si chaque transition augmente la valeur de son état de  $n$  intervalles élémentaires unitaires, alors, après une transition, il doit se trouver dans l'état  $N+n$ .

#### Utilisation de scalaires

Seuls les espaces scalaires peuvent satisfaire à ces contraintes : en physique nous tombons sur des nombres réels, puis complexes.

Cependant, dans le cadre des séquenceurs, nous utilisons plus simplement des nombres entiers : binaires ou numériques. En effet, le domaine peut être construit selon la méthode de Giuseppe Peano [Pea 89] [Pea 67].

## b) Démarche, de généralisation au prix d'une d'abstraction

Au passage, nous devons signaler ici que le traitement de simulation s'effectue au prix d'une généralisation, obtenue par abstraction.

En effet, partant de l'état du monde représenté par une valeur typée, la démarche de généralisation consiste à délaissier le type, puisqu'il demeure constant, invariant au fil des transitions d'état du système : nous ne traitons que la valeur.

Il s'agit donc ici d'une démarche d'abstraction<sup>1</sup> car, dans la description de l'état du système, nous délaissions la facette concrète *type*, pour focaliser sur la facette *valeur* qui elle est abstraite et générale. En effet, elle s'applique universellement à beaucoup de types (longueur, largeur, poids, nombre, date, distance...).

## c) Passer du qualitatif de l'anticipation, au quantitatif de la simulation

Cette généralisation permet de quantifier l'état du système.

### Agents Croyances pour anticiper

Chez les agents *Croyances*<sup>2</sup>, qui mémorisent le résultat de leurs actions passées, apparaît la démarche d'anticipation unaire, i.e. à profondeur 1. Elle sert à modéliser les transitions et à anticiper le résultat de la prochaine action : si celui-ci est négatif, elle l'inhibe.

Pour ce faire, l'être a seulement besoin de savoir si, le résultat de l'expérience passée, réalisée dans les mêmes conditions, était positive ou négative. Donc l'anticipation peut se contenter d'une mémorisation grossière (par exemple binaire) de la valeur obtenue au terme de la précédente action.

Évidemment, un être doté d'une telle faculté en tire un avantage sélectif certain. Il survit facilement à ses congénères qui ne la possèdent pas.

### Agents CDI pour planifier

Chez les agents supérieurs (CDI)<sup>3</sup>, apparaît la démarche de simulation, capable de modéliser, au fil des transitions, l'évolution de la valeur de l'état du système. Alors apparaît la démarche d'anticipation à profondeur N. Elle est basée sur la simulation qui permet de modéliser les transitions d'états. Ainsi l'agent peut anticiper le résultat de chacune des prochaines séquences d'actions envisageables : après avoir envisagé, simulé puis évalué toutes les possibilités, l'agent choisira la plus avantageuse. C'est ainsi qu'il planifie<sup>4</sup>.

Mais pour ce faire, l'être a besoin de connaître l'évaluation de chaque état du système obtenu après chaque transition/action. Donc, cette démarche de planification d'une séquence d'action est basée sur la simulation de N transitions qui requiert une mémorisation numérique du résultat de chaque action/transition.

Ici, l'avantage sélectif, procuré par cette aptitude, est encore plus important.

## 1.C) Présentation des S.Quotidiens

### a) Fabriquer des S.Quotidiens à partir de S.Réguliers

Nous disposons des S.Réguliers, qui prennent leur valeur dans un domaine numérique et notre but final est de les prolonger pour modéliser quelques scènes simples de la vie de tous les jours. Par exemple, pour décrire une entité que nous observons, nous écrirons :

lieu 4,	Elle est dans le lieu 4.
taille 3,	Sa taille est 3.

1 Étymologie de *Abstraire* - ab-trahere : tirer vers (le haut).

2 Dans les chapitres d'introduction, nous avons entrevus les agents CDI, nous les étudierons plus tard en détail.

3 Nous les avons entrevus dans les chapitres d'introduction, nous les étudierons plus tard en détail.

4 Nous avons présenté rapidement la planification, elle se détaillée dans le cours sur les agents.

vitesse 12, | Sa vitesse est 12...

### **b) Démarche de spécialisation**

Pour construire un S.Quotidien à partir d'un S.Régulier nous le typons en l'incarnant, i.e. en le plongeant dans la matière.

Concrètement, il agit dans le même domaine, en faisant varier la même grandeur physique. Donc, pour le décrire et le modéliser, nous lui rajoutons un type (lieu, taille, vitesse...).

Ainsi, un S.Quotidien est caractérisé par la donnée de 2 facettes : *valeur* et *type*.

### **c) Le type : une marque d'appartenance à une classe**

Ce type, que nous rajoutons au S.Quotidien, marque l'appartenance de tous les états de son domaine, à une famille, une classe : à l'intérieur de cet ensemble d'états, les éléments possèdent une propriété commune : par exemple, ce sont essentiellement des lieux, ou des tailles, ou des vitesses...

## 2) Taxinomie des séquenceurs

Nous venons de décrire différentes familles de séquenceurs ; maintenant, rangeons-les au sein d'un arbre d'héritage (illustration ci-dessous, à droite).

### 2.A) Les SF

Tout au sommet de cet arbre, apparaissent les systèmes formels.

### 2.B) Séquenceurs libres (SL) ou *séquenceurs à états libres* : sous-classe des SF

Dans le haut de cet arbre, dessous le nœud des SF, se trouvent les séquenceurs à états libres. Ils sont définis et construits comme des séquenceurs dont le domaine est constitué de variables libres<sup>5</sup> et dont le graphe découle de la donnée des règles de réécriture du SF.

#### a) Les SL généralisent une grande variété de séquenceurs

De par leur nature, les séquenceurs sont abstraits. Grâce à cette construction, ils généralisent l'ensemble de tous les séquenceurs imaginables (de toutes les formes de graphe, et de tous les procès<sup>6</sup> possibles).

#### b) Décrire le traitement effectué par les SL

Maintenant, voyons pourquoi le procès qu'ils effectuent est indéterminable. Cependant, nous constaterons qu'ils effectuent un traitement caractérisé par la forme de leur graphe.

#### c) Séquenceurs dont le domaine est constitué de variables libres

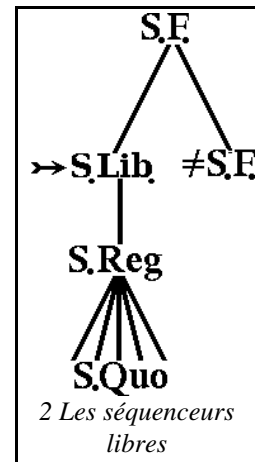
D'un côté, la donnée des règles de réécriture détermine la topologie du graphe du séquenceur, mais regardons plutôt le contenu des état de ce graphe. Nous voyons que les règles des SL font référence à des variables libres, donc au sein d'un séquenceur les procès (les transitions d'état produites par l'application des règles de réécriture) demeurent très variables et donc difficiles à caractériser.

Les SL sont construits au moyen de symboles non connotés, i.e. les états du graphe sont étiquetés par des variables libres. À chaque transition d'état, ce remplissage non contraint laisse le procès effectué par les SL totalement indéterminé. Ceci les rend tellement abstraits que nous n'envisageons pas de les charger de sens. La verbalisation du procès de réécriture se fait au moyen de verbes abstraits. Elle se limite pratiquement à constater que *le système change d'état*.

Ainsi il est impossible d'utiliser ces systèmes pour faire de la représentation, pour modéliser un micro-monde, et encore moins pour simuler des transformations en son sein.

#### d) Séquenceurs dont le graphe est déterminé par la donnée des règles

Par contre, pour un séquenceur, la donnée d'un jeu de règles détermine la topologie de son graphe. Au début de ce cours, pour passer des SF aux séquenceurs, nous avons spécifié des restrictions sur les règles afin qu'elles induisent une régularité dans les graphes. Ainsi émergent des structures caractéristiques : notamment des parcours linéaires, bouclés ou confluents. En

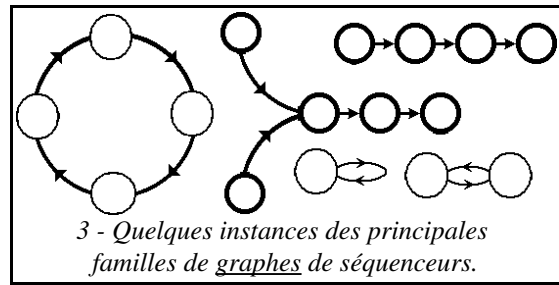


5 Variables qui peuvent être instanciées sans restriction de type.

6 Procès : transition d'état. Ici il s'agit de la transition, du changement de l'état du système formel.

conclusion, seule la topologie de ce type de séquenceur caractérise le traitement qu'il produit à l'occasion du parcours de son graphe.

Contrairement au traitement décrit par le procès de réécriture, celui effectué au niveau du graphe est plus facilement caractérisable. À chaque topologie, il est plus aisé d'attacher une verbalisation. Quand le séquenceur parcourt une période très courte, nous pouvons dire qu'il oscille ; quand elle s'allonge, nous disons qu'il boucle ou qu'il cycle. Enfin, quand la période devient très grande, le fait que le graphe se reboucle, constitue une information, un événement. Alors, nous commentons : à partir d'ici il recommence son cycle.



La figure ci-contre, présente une instance de chaque principale famille de graphe de séquenceur : à gauche, une boucle ; en haut à droite, une séquence linéaire ; au milieu, un confluent ; au milieu en bas, un état permanent ; et en bas à droite, un bistable.

### e) Note sur une convention de notation graphique

Dans un SL, chaque état est étiqueté par une variable libre différente car il s'avère impossible de dégager une régularité au sein de son domaine. Ainsi, dans le dessin ci-contre, aucun des états n'est étiqueté par un symbole, car nous avons choisi cette autre convention de notation, plus intuitive : les états sont représentés vides pour exprimer qu'il ne sont pas contraints. C'est pourquoi appelle cette famille les séquenceurs à états libres : une fois la topologie du graphe définie, le contenu de chaque état peut être instancié par n'importe quel symbole.

## 2.C) S.Réguliers, un domaine numérique défini en compréhension

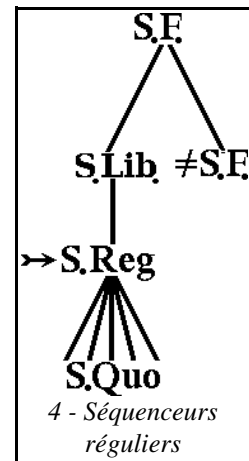
### a) Introduction

Au milieu de la taxinomie (arborescence), se trouvent les séquenceurs réguliers (S.Réguliers).

Par construction, ils héritent de leurs parents les séquenceurs libres, des graphes réguliers (linéaires, bouclés ou confluents).

### b) Le domaine des S.Réguliers est défini en compréhension<sup>7</sup>

Depuis l'état de départ  $e_0$ , il est construit par récurrence, en utilisant une fonction f binaire analytique :  $e_1=f(e_0)$ ,  $e_2=f(e_1)$  ...  $e_{n+1}=f(e_n)$ . Donc ils prennent leurs valeurs dans des sous-ensembles de  $\mathbb{Z}$  (entiers positifs ou négatifs).



### c) Les états du domaine des S.Réguliers sont définis en compréhension

Par définition, les états des S.Réguliers sont calculés en itérant une formule de récurrence : il sont définis en compréhension, en fonction de l'état précédent :

$$e_{n+1} = f(e_n) \quad // \text{ où } f \text{ est une fonction}$$

À la réécriture suivante, nous obtenons de même :

$$e_{n+2} = f(e_{n+1}) \quad // \text{ où } f \text{ est la même fonction car elle est définie en compréhension}$$

En remplaçant, dans la ligne ci-dessus,  $e_{n+1}$  par sa valeur  $f(e_n)$  nous calculons :

$$e_{n+2} = f(f(e_n))$$

À la réécriture suivante, nous procédons de même :

$$e_{n+3} = f(f(f(e_n)))$$

<sup>7</sup> En mathématiques, la définition d'un ensemble est dite *en compréhension*, si nous fournissons une méthode qui permet de construire systématiquement la liste de ses éléments.

Finalement nous déduisons la forme générale :

$$e_{n+k} = f^k(e_n)$$

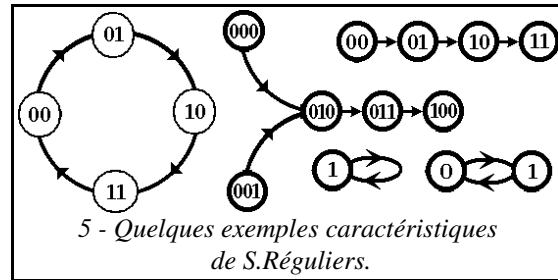
Nous concluons que le  $k^{\text{ième}}$  terme est obtenu par itération, en composant  $k$  fois la fonction  $f$ .

#### d) Par construction, leur graphe est celui d'un séquenceur

Dans la famille des séquenceurs libres nous avons déjà remarqué et décrit des topologies caractéristiques de graphes : *linéaires*, *bouclés* et *confluents*. Puisque les SL sont parents des S.Réguliers, ces derniers, dès leur création, héritent aussi de ces structures.

Remarque : dans la suite de notre étude, nous délaisserons un peu les *confluents* pour focaliser de préférence sur les *bouclés* et les *linéaires*.

Pour illustrer les S.Réguliers, dans la figure ci-dessus, nous en présentons cinq exemples : à gauche, une boucle ; en haut à droite, une séquence linéaire ; au milieu, un confluent ; au milieu en bas, un état permanent ; et en bas à droite, un bistable.

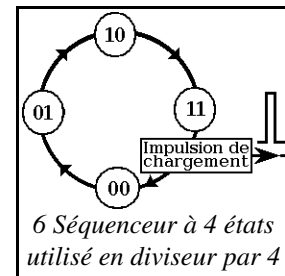


#### e) Aspect effectif des S.Réguliers

De par leur construction, ces séquenceurs peuvent modéliser les phénomènes de la vie quotidienne. Puisqu'ils sont généraux, et construits par abstraction, leurs symboles binaires ne sont pas chargés de contenu, mais leurs graphes sont chargés de propriétés structurantes, et leurs états sont déterminés itérativement par leurs procès.

Par exemple, regardons d'abord ce séquenceur en boucle caractérisé par la topologie de son graphe. Son domaine est {00, 01, 10, 11} et ses règles de transition sont : 00→01, 01→10, 10→11, 11→00.

D'un côté, cette structure de graphe, qui s'apparente au décodeur/démultiplexeur de la logique, est utilisée dans la technologie *token ring* des réseaux pour distribuer des jetons alternativement aux nœuds de l'anneau : soit le cas d'un sous-système  $n$ , connecté à l'état  $N$  ; à chaque instant élémentaire, le séquenceur passe à l'état suivant, quand il arrive à l'état  $N$ , alors  $n$  reçoit un jeton et est validé.



Et de l'autre côté, nous pouvons encore l'utiliser pour faire un diviseur modulo 4 : à chaque instant élémentaire, le séquenceur reçoit une impulsion et passe à l'état suivant. À chaque fois qu'il transite de l'état 11 à l'état 00, il envoie une impulsion à un autre sous-système. Ainsi, ce dernier reçoit 4 fois moins d'impulsions que le séquenceur.

Finalement, nous constatons ici que *la topologie du séquenceur fixe le facteur de la division modulo*, et que ses procès déterminent le lieu des impulsions/validations. En conclusion, la prestation fournie par le système dépend de sa structure : elle relève à la fois de sa topologie et de ses procès.

## 2.D) En conclusion :

Dans notre construction, depuis l'ensemble immense des SF, chaque étape a élagué drastiquement. Mais l'ensemble de départ est si grand qu'à l'arrivée, il reste encore de nombreux séquenceurs construits sur des séries entières définies par itération/récurrence. Par exemple, nous pouvons citer les six instances de séquenceurs traitées au début du poly :

Type de séquenceur	Son domaine de définition	Sa formule de récurrence
cellule permanente	{0}	$e_{n+1} = e_n$
alternateur	{0, 1,}	$e_{n+1} = \gamma e_n$ ; (avec $\gamma 1=0$ ; et $\gamma 0=1$ ).
séquenceur linéaire	{000, 001, 010, 011, 100}	$e_{n+1} = 1 + (e_n)$
boucle	{000, 001, 010, 011}	si $(e_n=11)$ $e_{n+1} = 0$ sinon $e_{n+1} = 1 + (e_n)$
décroissance linéaire	{100, 11, 10, 1, 0}	$e_{n+1} = e_n - 1$
série géométrique de base $r = 1,5$	{100, 110, 1001, 1101, 10011}	$e_{n+1} = 1,5 * e_n$

## 2.E) Les Séquenceurs Quotidiens (S.Quotidiens)

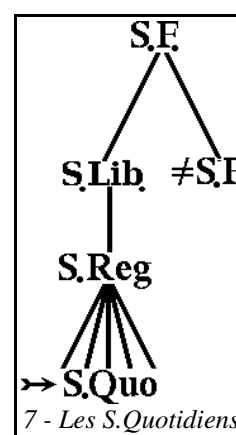
Selon une définition informatique, ce sont des instances typées de S.Réguliers. Mais, de façon plus imagée, nous pouvons encore dire que ce sont des S.Réguliers, plongés dans des domaines particuliers, par lesquels le cogniticien<sup>8</sup> s'emploie à modéliser la vie de tous les jours.

### a) Positionnement

Ensuite, juste dessous les S.Réguliers, nous plaçons les S.Quotidiens.

#### Quelles sont les raisons de ce placement ?

Les S.Quotidiens sont placés dessous leurs parents car ils héritent de leur structure (graphe et procès) et parce qu'ils sont plus spécifiques : ils possèdent en plus une facette *type*, qui caractérise la spécificité du domaine où ils sont plongés.



### b) Le travail du cogniticien est de modéliser le réel, la vie quotidienne

La vie quotidienne est structurée par de multiples règles : les lois physico-chimiques de la nature, celles des échanges socio-économiques au sein groupe, celles de la psychologie.

Le travail du cogniticien est de modéliser ces structures et ces comportements. Il doit trouver un S.Régulier dont la sémantique, i.e. le fonctionnement (graphe + procès) soit proche de celui du monde à modéliser.

Il prend une instance de S.Régulier. Il lui ajoute un type, i.e. un marqueur spécifique au domaine où se déroule l'action modélisée. Ainsi le cogniticien obtient un micro-monde, un séquenceur qui se comporte, évolue, change d'état. Mais ce dernier possède un domaine dont les états sont typés.

Ainsi les S.Quotidiens sont obtenus en plongeant les S.Réguliers dans un domaine spécifique. Le cogniticien instancie les S.Réguliers, et leur ajoute un type provenant du domaine où ils sont plongés. Ce faisant, les états numériques des S.Réguliers deviennent des états typés à deux facettes, de la forme (type, nombre)<sup>9</sup>.

### c) Plonger un S.Régulier dans un domaine lui fait hériter le type correspondant

#### Exemple de la base de temps :

Par exemple, plonger un S.Régulier dans le temps (la dimension temporelle), le transforme en horloge, en base de temps. À la lumière de ce nouvel éclairage, *passer d'un état à un autre se*

8 En IA, le cogniticien est un spécialiste de la représentation des connaissances. Il observe le monde, analyse son état et les règles qui le gouvernent, et travaille à décrire l'état du micro-monde de départ de la simulation.

9 Nous pouvons encore trouver d'autre formes équivalentes : (nombre, type), typenombre, type<sub>nombre</sub>, [type nombre]. Ex : nom1, nom<sub>1</sub>, [nom 1].



verbalise en : *passer de l'heure n à l'heure n+1*. Ainsi, l'état 0 du S.Régulier, se concrétise en *le temps 0* que nous pouvons noter indifféremment :  $t_0$ ,  $temps0$ , ou encore (*temps 0*).

#### Exemple de la base spatiale :

Par exemple, plonger un S.Régulier dans l'espace (la dimension spatiale), le transforme en chemin, en trajet. À la lumière de ce nouvel éclairage, *passer d'un état à un autre* se verbalise en : *passer du lieu n au lieu n+1*. Ainsi, le lieu 0 du S.Régulier, se concrétise en *le lieu 0* que nous pouvons noter indifféremment :  $lieu_0$ ,  $lieu0$ , ou encore (*lieu 0*).

#### d) Plonger un séquenceur conserve graphe et procès mais change ses symboles

Au passage, nous retrouvons ici ce qui est annoncé en introduction. Plonger un séquenceur dans un domaine spécifique, d'un côté conserve son graphe et ses procès, mais de l'autre particularise, type les symboles de son domaine. Nous lui attribuons des symboles connotés, i.e. que dans chaque cas, nous transformons les nombres binaires du S.Régulier dans des symboles typés, appartenant à la même famille.

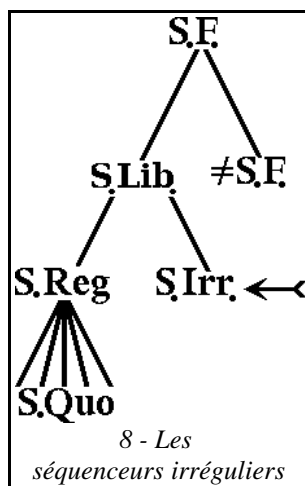
Ainsi, dans l'exemple du *plongeon temporel*, les symboles  $t_i$  du S.Quotidien obtenus appartiennent à une classe, une famille, à un type du *temps*. De même, dans celui du *plongeon spatial*, les symboles qui sont fabriqués appartiennent à une classe, une famille, à un type de l'*espace* : quand nous verbalisons, nous les nommons des *lieux*.

#### e) Conclusion :

Appliquer un S.Régulier à un domaine particulier, lui attribue un type. Les transformations qu'il subit cessent d'être des procès abstraits. En s'incarnant dans un domaine particulier, le S.Régulier prend chair, se concrétise, et devient un S.Quotidien, qui permet de simuler des actions sur ce domaine.

Dans la suite de ce polycopié, nous nous attacherons à montrer comment les séquenceurs, s'avèrent d'excellents outils pour faire de la simulation.

## 2.F) Les séquenceurs irréguliers (SI), au domaine défini en extension



Enfin, au sein de la taxinomie où nous classons les séquenceurs, plaçons les séquenceurs irréguliers (SI).

Parmi les exemples de séquenceurs que nous traitons, les (SI) sont seulement définis en extension. Plusieurs raisons peuvent expliquer cette absence de définition en compréhension :

- Ils possèdent un domaine très irrégulier.
- Ils possèdent un domaine trop difficilement descriptible en compréhension.
- Leur domaine a été nommé par des conventions arbitraires que nous ignorons, traduits dans un langage dont nous ne possédons pas le dictionnaire. Ex : les séquenceurs spatiaux, basés sur les lieux : le trajet *Chartres*→*Nantes*, ou encore le trajet *LieuA*→*LieuZ*→*LieuE*→*LieuR*→*LieuT*.

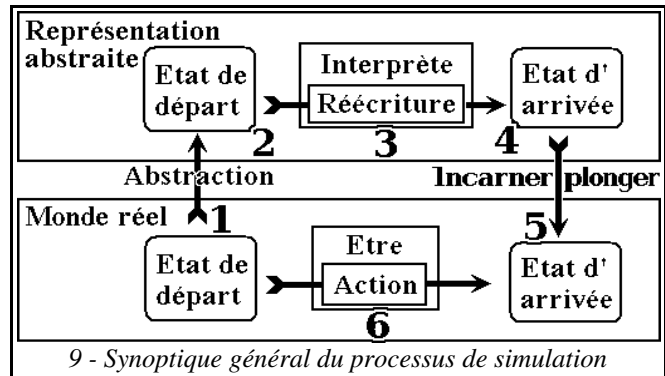
Bref, nous ne disposons pas de définition en compréhension de leur domaine : il se représente seulement en extension. Par ailleurs, leur fonctionnement est représenté exhaustivement par la donnée d'une liste ou d'un tableau de règles.

Pour ces raisons, quand nous ferons de la simulation, nous traiterons d'un côté les S.Réguliers, et nous simulerons autrement avec les SI.

### 3) Principe général et synoptique de la simulation

Après avoir classé les différentes familles de séquenceurs au sein d'une taxinomie, étudions maintenant le principe général et le synoptique de la simulation en IA.

Elle implique six étapes différentes que, successivement, nous allons analyser et détailler.



#### 3.A) Étape 1 : départ depuis le micro-monde réel

Au départ, nous nous donnons un micro-monde, avant qu'il ne subisse une action.

#### 3.B) Étape 2 : une représentation du micro-monde

Une étape de description, abstraction décrit en termes formels l'état du micro-monde. Il constitue son état de départ.

#### 3.C) Étape 3 : la simulation proprement dite

L'étape 3 consiste en la simulation proprement dite : étape après étape, un interprète applique des règles pour réécrire la représentation abstraite de l'état du micro-monde.

#### 3.D) Étape 4 : le résultat de ce traitement simulé

À l'issue de la démarche de simulation de l'étape 3, nous obtenons le résultat de ce traitement : il décrit l'état d'arrivée abstrait calculé du micro-monde.

#### 3.E) Étape 5 : état d'arrivée simulé du micro-monde

Finalement, la dernière étape de cette simulation plonge le résultat 4, l'état formel calculé du micro-monde, dans la réalité quotidienne en l'incarnant dans le monde réel. Ainsi il fournit l'état d'arrivée simulé.

#### 3.F) Étape 6 : évolution réelle du monde, au fil du temps

Sur le synoptique général du processus de simulation, nous représentons aussi l'étape 6 qui correspond à l'évolution réelle du monde, au fil du temps ; il s'agit bien sûr de l'étape que nous simulons au travers des cinq étapes précédentes.

Remarque : si la simulation est de bonne qualité, l'état d'arrivée simulé du micro-monde correspond à l'état réel du monde. Sinon, ce n'est pas le cas...

## 4) Observer le monde réel et modéliser sa structure avec un séquenceur

Avant de pouvoir expliquer la démarche de modélisation du monde réel (de son état et de ses règles), au moyen d'un séquenceur, nous devons introduire une définition :

### 4.A) Introduction de la notion de *S.Quotidiens équivalents*

#### a) Définition de deux *S.Quotidiens équivalents*

Deux *S.Quotidiens* sont équivalents s'ils présentent la même forme profonde, i.e. s'ils possèdent le même graphe, et si, à chaque étape de ce graphe, ils effectuent le même procès. En fait, ils ne diffèrent que par leur typage.

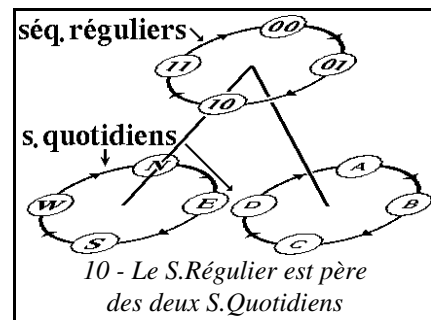
Parfois des *S.Quotidiens* sont équivalents car ils dérivent du même parent : ils possèdent une même forme profonde (un même graphe et un même procès), mais présentent deux formes de surface typées différemment, car ils sont plongés dans des mondes concrets dissemblables.

#### b) Classe d'équivalence de *S.Quotidiens*

Quand deux *S.Quotidiens* présentent la même structure profonde, i.e. le même graphe et les mêmes procès, ils sont équivalents. Nous les rangeons dans la même classe d'équivalence, et ils sont représentés par le même *S.Régulier* parent.

#### c) Exemple - le séquenceur qui boucle de 00 à 11

Dans le dessin ci-contre, regardons en bas à gauche, le *S.Quotidien* en boucle, dont le domaine est constitué des lettres *N, E, S, W* (pour nord, est, sud, ouest). Selon les conventions de la navigation aérienne, ces caps correspondent aux valeurs  $0^\circ, 90^\circ, 180^\circ, 270^\circ$ . Pour abstraire ce séquenceur, nous ôtons son typage en degré d'angle, nous divisons ces valeurs par 90, et nous le généralisons par le séquenceur *S*, dont le domaine  $D = \{00, 01, 10, 11\}$  permet de compter les objets d'un ensemble de quatre éléments.



Dans le dessin ci-contre, regardons en bas à droite, le *S.Quotidien* en boucle, dont le domaine est constitué des lettres *A, B, C* et *D*. Selon le code ASCII, ces lettres correspondent aux valeurs hexadécimales *65, 66, 67* et *68*. Pour abstraire ce séquenceur, nous ôtons son typage en soustrayant à ces valeurs *65*, et nous le généralisons avec le même séquenceur *S*.

Finalement, sur les deux exemples précédents, nous déduisons, qu'au prix de ces deux aménagements de type (division par 90 et soustraction de 65), les deux *S.Quotidiens* sont équivalents, car ils sont représentés par le même parent : le *S.Régulier S*.

### 4.B) Plongeon *S.Réguliers*→*S.Quotidiens* : de l'abstrait au concret

Matérialiser un *S.Régulier* en le plongeant dans le monde quotidien lui fournit un type. Pour incarner, matérialiser deux *S.Réguliers* équivalents, nous typons chacun d'eux, en le plongeant dans un monde réel et quotidien différent. Ainsi, au niveau de sa forme de surface, chaque *S.Quotidien* reçoit un type particulier et est rangé dans une classe spécifique.

#### a) Quelques précisions sur le verbe plonger

Le terme *plonger* est un jargon rapide et facile d'épistémologue. Maintenant nous allons détailler et formaliser cette opération au moyen de la famille des *S.Réguliers*.

Il s'agit de plonger un S.Régulier dans différentes situations de la vie quotidienne. Le même séquenceur se trouve donc utilisé plusieurs fois. Il conserve la même structure (caractérisée par son graphe et ses procès), mais il est mis en œuvre dans différents contextes, i.e. dans différents domaines : des symboles issues de différentes familles étiquettent les nœuds de ses graphes.

### b) Exemple 1 :

Soit le séquenceur à croissance linéaire  $S$ , dont le domaine numérique binaire est :  $\{001, 010, 011, 100, 101\}$ . Nous le plongeons dans différents domaines : alphabétique, décimal, spatial et temporel. Ainsi nous obtenons les quatre instances suivantes de séquenceurs :

$S_1$ dont le domaine est : $\{a, b, c, d, e\}$ .	// Domaine alphabétique
$S_2$ dont le domaine est : $\{1, 2, 3, 4, 5\}$ .	// Domaine décimal
$S_3$ dont le domaine est : $\{lieu_1, lieu_2, lieu_3, lieu_4, lieu_5\}$ .	// Domaine spatial
$S_4$ dont le domaine est : $\{t_1, t_2, t_3, t_4, t_5\}$ .	// Domaine temporel

### c) Exemple 2 :

Soit le séquenceur irrégulier, dont le domaine défini en extension est :  $\{A, Z, E, R, T\}$ . Nous le plongeons dans les domaines alphabétique et spatial. Ainsi nous obtenons les deux instances suivantes de séquenceurs :

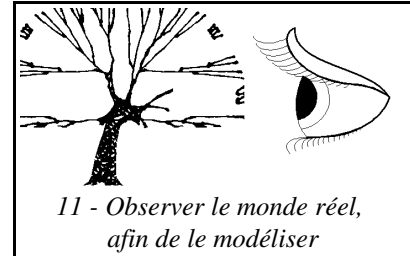
$S_1$ dont le domaine est : $\{LettreA, LettreZ, LettreE, LettreR, LettreT\}$ .	// Domaine alphabétique
$S_2$ dont le domaine est : $\{Paris, Chartres, Le\_Mans, Angers, Nantes\}$ .	// Domaine spatial

Dans ce cas, les ensembles sont moins réguliers et la démarche ne fonctionne pas parfaitement : elle est à peine inversible.

## 4.C) Observer, pour modéliser le monde réel avec un S.Régulier

### a) Étape 1 d'analyse

Quand nous observons le monde réel afin de le modéliser au moyen d'un S.Régulier, nous passons d'abord par une étape d'analyse. Le but est d'identifier le S.Quotidien correspondant au monde analysé, au niveau de son graphe et aussi de ses procès.



### b) Étape 2 d'abstraction

Puis vient une étape d'abstraction (ab - trahere : tirer de quelque chose). Nous généralisons, nous transformons le quotidien vers le régulier, en abandonnant le typage. Abstraire un séquenceur qui est plongé dans un domaine, c'est abandonner son type et ne garder que son graphe et ses procès numériques.

### c) Exemple - des séquenceurs qui bouclent de 1 à 5

Par exemple, reprenons les quatre séquenceurs à croissance linéaire  $S_i$  de l'exemple précédent. Nous pouvons réécrire leur domaine ainsi :

$S_1$ dont le domaine est : $\{lettre1^{i\grave{e}re}, lettre2^{i\grave{e}me}, lettre3^{i\grave{e}me}, lettre4^{i\grave{e}me}, lettre5^{i\grave{e}me}\}$	//type alphabétique
$S_2$ dont le domaine est : $\{1, 2, 3, 4, 5\}$ .	// type décimal
$S_3$ dont le domaine est : $\{lieu_1, lieu_2, lieu_3, lieu_4, lieu_5\}$ .	// type spatial
$S_4$ dont le domaine est : $\{t_1, t_2, t_3, t_4, t_5\}$ .	// type temporel

Ainsi, en laissant tomber leur type les séquenceurs  $S_1$ ,  $S_3$  et  $S_4$  s'abstraient d'abord en  $S_2$ , puis ce dernier s'abstrait en  $S$  dont le domaine numérique binaire est :  $\{001, 010, 011, 100, 101\}$ .

Finalement, pour chacune de ces classes, nous dégagons un représentant. Ce dernier apparaît bien comme une abstraction du séquenceur, car il ne possède plus de type, mais conserve le graphe et les procès.

**d) Exemple - des séquenceurs qui bouclent de 1 à 7**

Regardons par exemple le conte de Blanche-Neige, et particulièrement le monde de ses sept nains. Il se décrit par la liste : *Atchoum, Prof, Grincheux, Simplet, Dormeur, Timide et Joyeux* ; qui permet de construire le S.Quotidien :  $S_n = \{Nain1, Nain2, Nain3, Nain4, Nain5, Nain6, Nain7\}$ . En abandonnant le typage nous abstrayons  $S_n$  en  $S$ , S.Régulier dont le domaine est constitué des entiers de 001 à 111 :  $D_S = \{001 \dots 111\}$ . Défini en compréhension, il permet de compter les objets d'un ensemble de sept éléments.

Prenons un autre exemple, la liste des jours de la semaine : *lundi, mardi, mercredi, jeudi, vendredi, samedi et dimanche*. Elle permet de construire le S.Quotidien :  $S_j = \{Jour1, Jour2, Jour3, Jour4, Jour5, Jour6, Jour7\}$ . En abandonnant le typage nous abstrayons  $S_j$  vers  $S$ , le S.Régulier précédent.

Les deux S.Quotidiens  $S_j$  et  $S_n$  sont équivalents. Le premier  $S_j$  peut ainsi compter les jours de la semaine, et le second  $S_n$  les nains de Blanche-neige ; mais quand ils sont abstraits, dans la classe des SF réguliers, les symboles du domaine  $\{001 \dots 111\}$  de leur père  $S$  ne sont plus typés, s'avèrent moins chargés de contenu ; et sont semblables.

**4.D) Bilan**

À ce stade nous avons montré comment nous observons le monde réel afin de le modéliser au moyen d'un S.Régulier. Maintenant détaillons les étapes de simulations effectuées au moyen de séquenceurs.

## 5) Le séquenceur utilisé en simulation

Le séquenceur, est un outil très simple, cependant, il est déjà capable de servir en IA pour faire de la simulation.

### 5.A) La simulation : imaginer l'évolution de notre monde réel

Commençons par préciser la **définition de la simulation classique en IA**. D'abord, Pour simuler l'évolution d'un monde réel, un être doit posséder un modèle suffisamment précis du support et des agents qui y évoluent. Ensuite, il doit connaître l'état de départ de ce monde, la position du mobilier et des agents sur le support, et encore leurs buts, plans et projets. Enfin, il doit réfléchir et imaginer l'évolution de ce monde sous l'action des agents.

Pour l'instant, munis des séquenceurs, nous n'allons pas pouvoir effectuer une simulation aussi fine ; cependant, voici ce que nous allons réussir à faire.

### 5.B) Les étapes d'une simulation effectuée au moyen d'un séquenceur irrégulier

#### a) Introduction

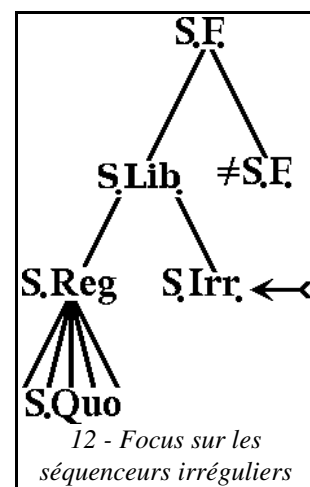
Tout d'abord, nous commençons à faire de la simulation au moyen des séquenceurs irréguliers, i.e. dont le domaine et les règles ne peuvent être définis qu'en extension.

#### b) Étape 1 et 2 - Démarche de modélisation : écrire le séquenceur

##### Démarche de description au moyen d'un séquenceur

À ce stade du cours nous travaillons seulement avec les séquenceurs. Ce sont les seuls outils de travail dont nous disposons. Nous devons utiliser un séquenceur pour décrire le monde analysé, et pour cela nous savons que ce dernier dérive d'un SF. Il est seulement un peu transformé : nous l'avons légèrement contraint par quelques spécifications portant sur la formation des règles de réécriture.

Donc, comme pour un SF, dans une démarche de modélisation, nous devons au moins décrire l'état de départ du micro-monde et les règles de réécriture. Par contre, nous n'avons pas à décrire l'état d'arrivée. En effet, ici, nous fonctionnons en simulation, et donc, notre but est de le deviner.

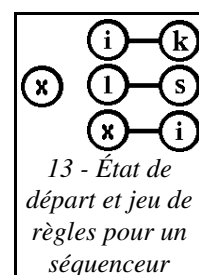


##### La description de l'état de départ constitue une démarche d'abstraction :

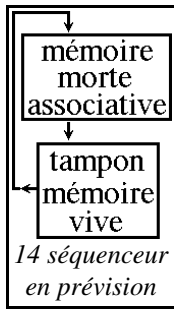
Pour décrire l'état de départ du système, l'informaticien observe le monde réel, puis il analyse sa perception. Dans une démarche d'abstraction, il décrit l'état du monde analysé. Pour ce faire, dans le cadre de cette modélisation au moyen d'un séquenceur, il doit décrire l'état de départ du système en fournissant un unique symbole. Par exemple, dans la figure ci-dessous, à droite, il fournit la lettre *x*.

##### Démarche d'abstraction : Décrire la base de règles

Ensuite, l'informaticien observe les transformations que subit le monde réel. Dans une démarche d'analyse et d'abstraction, il décrit, en extension, les règles correspondantes, de transformation (de réécriture) sous la forme :



ÉtatDuMondeAvantLaTransformation → ÉtatDuMondeAprèsLaTransformation.



Notez que la figure ci-dessus à droite présente un exemple de base de règles pour un séquenceur irrégulier. Puisqu'elle se réduit à la forme  $Symbole1 \rightarrow Symbole2$ , elle peut aussi se mémoriser sous la forme d'une mémoire morte associative  $Clé \rightarrow Valeur$  (figure de gauche).

### La démarche de catégorisation n'est pas forcément exacte

Quand un informaticien décrit, modélise une situation au moyen d'un séquenceur, il fait aussi une démarche de catégorisation, i.e. une démarche qui n'est pas forcément rigoureuse. Il connaît (ou croit connaître) les différents comportements, i.e. les multiples trajets du système, et il essaie de les regrouper dans une catégorie. Mais rien ne garantit qu'il y parvienne. Certains traits particuliers du processus modélisé peuvent lui avoir échappé.

Nous retombons ici sur le problème du lien *signifiant*  $\rightarrow$  *signifié*, qui correspond encore au problème que nous avons déjà évoqué, i.e. à celui du contenu du symbole que nous utilisons pour étiqueter une catégorie.

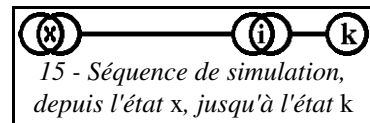
En conclusion, dans ce cours nous avons clairement dit que le processus formel d'application des règles dans les SF est rigoureux, par contre, nous constatons ici que l'étape de description/modélisation du monde réel peut pêcher.

### c) Étape 3 - Transition d'état, par application de règle de réécriture

Ensuite, le système simule l'évolution de l'état du micro-monde en déroulant une séquence : tant que c'est possible, il applique la règle de réécriture permettant une transition, et fournit un nouvel état.

Tant qu'il trouve une règle dont la tête correspond à l'état du micro-monde, il l'applique, il efface l'état du micro-monde correspondant à cette tête de règle pour le remplacer par la queue de règle, et donc il obtient un nouvel état du système. Ainsi, à chaque procès, le symbole qui décrit l'état du micro-monde est remplacé.

Dans l'exemple ci-contre, partant de l'état  $x$ , il applique d'abord la règle  $x \rightarrow i$ , et obtient l'état  $i$  ; ensuite, en appliquant la règle  $i \rightarrow k$ , il obtient l'état  $k$ .



### d) Étape 4 - L'arrêt du système donne l'état d'arrivée du micro-monde

Si, le système arrive dans un état tel qu'aucune règle ne peut se déclencher, la situation obtenue correspond à une condition d'arrêt. La séquence de transitions s'arrête : le système stoppe. Nous obtenons l'état d'arrivée calculé du micro-monde : il est réduit à un symbole.

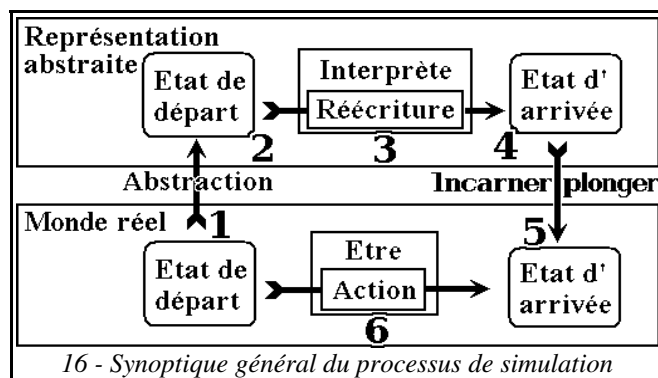
Dans l'exemple ci-dessus, plus aucune règle ne se déclenche, alors nous sommes arrivés à l'état final  $k$ .

### e) Étape 5 - Exploitation du résultat obtenu

Ainsi le traitement effectué fournit un symbole qui décrit l'état d'arrivée calculé du micro-monde.

### f) Étape 6 - Vérification du calage du modèle

En comparant cet état du micro-monde obtenu par simulation avec celui que nous pouvons observer dans la réalité, nous pouvons nous faire une idée de la qualité de la modélisation et, si nécessaire, recalibrer la simulation, i.e. réécrire les règles.



Remarque : ce recalage du modèle, n'est pas anodin ; il est très complexe et constitue une démarche nouvelle traitée plus longuement à la fin du cours, dans la partie à propos du traitement cognitif.

### g) Appliquer à un exemple de gestion de l'espace : vérifier un itinéraire séquentiel

Maintenant nous décrivons une application de cette méthode, nous simulons, au moyen d'un séquenceur, la vérification d'un itinéraire. L'agent se pose la question suivante : *en commençant depuis ce point de départ, et en suivant le chemin, vais-je arriver au but que je me suis fixé ?*

Avant de commencer la vérification d'un itinéraire, l'agent doit d'abord avoir enregistré, dans une mémoire morte associative, la liste des étapes *départ*→*arrivée* qu'il connaît.

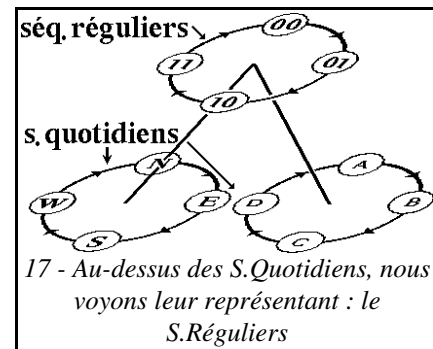
Ensuite, il simule : partant d'un point initial, il vérifie si, avec la connaissance qu'il possède au sujet du monde réel, il trouve l'itinéraire, i.e. il teste si, au bout de plusieurs transitions/dérivations, il peut atteindre le lieu but qu'il s'est donné.

## 5.C) Simulation au moyen de S.Réguliers ou quotidiens

Après avoir décrit comment effectuer une simulation au moyen des séquenceurs irréguliers, passons maintenant aux S.Réguliers.

Dans ce cours, en classant les séquenceurs au sein d'une arborescence, nous avons travaillé sur la relation *S.Régulier*↔*S.Quotidiens*, et avons montré comment un S.Régulier peut représenter, être le père de plusieurs S.Quotidiens.

Maintenant nous continuons de présenter le mécanisme de la simulation, mais nous travaillons dans le cadre plus organisé des S.Réguliers plongés dans le monde quotidien.



### a) Intro. : quantifier l'évolution du monde, deviner les valeurs de ses états futurs

De façon générale, une simulation modélise l'évolution du monde réel sous l'action des agents qui le peuplent. Or le travail de description de l'état de départ du monde demande d'y reconnaître des modèles déjà connus. Ainsi, au travers de cette recherche, il introduit déjà de la standardisation, i.e. de la régularité.

De plus, ici nous choisissons de travailler avec des S.Réguliers, alors ce choix introduit encore plus d'ordre. Ainsi nous travaillons au niveau ultime de représentation abstraite pour décrire le monde : nous cessons d'utiliser des symboles classiques pour traiter seulement des symboles numériques binaires. Nous en sommes arrivés à une démarche de quantification qui nous permettra de décrire le monde au moyen de nombres entiers.

Nous utilisons un S.Régulier numérique : partant du résultat de l'évaluation de l'état de départ du système, en appliquant la formule récurrente qui caractérise la fonction de ce S.Régulier, nous représentons un histogramme, i.e. nous traçons l'évolution séquentielle de la courbe des quantifications effectuées sur le micro-monde.

Voyons maintenant cela en détail.

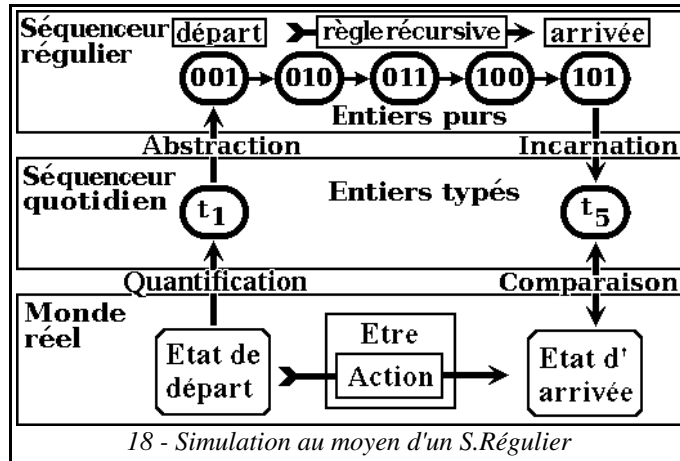
### b) Le travail de modélisation consiste en une triple démarche d'abstraction

Dans une simulation effectuée au moyen des S.Réguliers et quotidiens, la modélisation induit une triple démarche d'abstraction.

#### 1) D'abord une démarche de modélisation de l'état de départ



Correspondant à la démarche classique de simulation, nous trouvons d'abord un travail de modélisation : le cogniticien observe la vie réelle et décrit l'état d'un micro-monde quotidien. Puisque il travaille avec un séquenceur, cette description se limite à la fourniture d'un état typé à deux facettes : (type, nombre). Le symbole numérique quantifie son état. Le type provient de la vie de tous les jours (temps, prix, difficulté, taille, surface, poids, production...). Dans l'exemple ci-contre, nous pouvons voir l'entier typé  $t_1$ . Il exprime la valeur au temps 1).



## 2) Ensuite : démarche d'abstraction

Puis, dans une démarche d'abstraction, le cogniticien délaisse le type  $t$  du domaine et, ensuite, travaille seulement sur la valeur non typée.

## 3) Puis, analyse : identifier le S.Régulier correspondant, trouver la fonction récurrente

Le cogniticien observe les transformations que subit le monde réel. Dans une démarche d'analyse, il recherche le S.Régulier et surtout la fonction récurrente correspondante, qui, à chaque cycle, quantifie l'évolution du monde.

Au terme de cette analyse, il identifie son représentant, le séquenceur père ; ainsi il dispose du S.Régulier qui modélise le processus<sup>10</sup> au moyen d'une fonction calculée sur des symboles très abstraits ; car évidemment, ici il obtient une fonction numérique sur des entiers binaires.

Cette simulation, sert à quantifier le procès. Donc ici, la modélisation atteint ici un degré ultime de régularité, elle est effectuée au moyen de la donnée d'un unique entier.

## 4) Transmission des résultats au S.Régulier

Puis, à l'issue de ces deux démarches, cette étape transmet deux résultats. D'abord, au moyen d'une abstraction, elle délaisse le type  $t$  du domaine et transmet au S.Régulier, seulement la valeur, i.e. l'entier non typé  $I$ . Ensuite, à l'issue de cette démarche d'analyse, elle lui transmet la fonction numérique à évaluer itérativement.

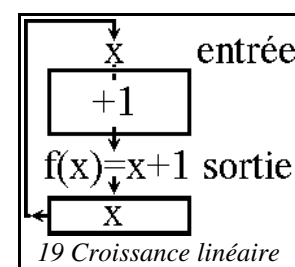
## c) Transition d'état par application de règle de réécriture effectuée sur un S.Régulier

Le système simulateur dispose maintenant des deux éléments/données nécessaires pour quantifier chaque étape de l'évolution de son domaine. Au moyen du S.Régulier, sachant la formule récurrente qui le caractérise et l'évaluation numérique de l'état de départ, il effectue la simulation proprement dite : la transition d'états.

Ainsi, partant de cet entier  $I$ , valeur de l'état initial du micro-monde, l'informaticien simule son évolution en déroulant une séquence : tant que c'est possible, il applique itérativement au registre mémoire, la fonction numérique récurrente qui effectue le calcul, et fournit un nouvel état quantifié du micro-monde, i.e. une nouvelle valeur binaire.

Ici, puisque nous sommes dans le cadre d'un traitement numérique, le processus se ramène à calculer les valeurs du système qui évoluent à chaque impulsion de chargement du registre mémoire.

Dans l'exemple traité, nous obtenons la suite : 10, 11, 100, 101 ; qui constitue un histogramme des valeurs prises par le système.



<sup>10</sup> Ici *processus* correspond à la traduction du terme anglais *process*. Il correspond à un *traitement de l'information*. Ne pas le confondre avec le terme français *procès* qui correspond à un changement de l'état du monde.

#### d) L'arrêt du système

Puisque le système applique itérativement une fonction numérique récurrente, une fois lancé, il continue : il est très autonome et les conditions d'arrêt sont moins nombreuses. Alors, il faut aussi prévoir un regard<sup>11</sup> extérieur qui à chaque cycle teste une condition d'arrêt pour stopper le processus. Dans l'exemple traité ci-dessus, le système s'arrête quand son état numérique est 101.

#### e) Nous obtenons l'état d'arrivée du micro-monde

Quand la séquence de transition s'arrête, le système stoppe. Nous obtenons l'état d'arrivée quantifié du micro-monde : il est réduit à un symbole numérique : ici le nombre binaire 101.

#### f) Retour au réel du quotidien

Après avoir quantifié le comportement abstrait du modèle, nous retournons ce résultat final à ses origines, i.e. nous le plongeons dans le monde concret, en lui rajoutant le type que nous avons enlevé à l'étape de généralisation (temps, prix, difficulté, taille, surface, poids, production...).

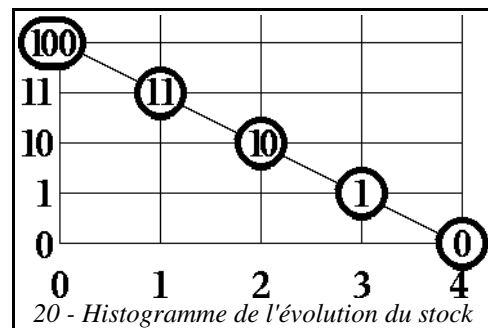
#### g) Remarque : le test et le recalage du modèle relèvent de l'apprentissage

À ce stade, nous avons terminé le cycle de simulation proprement dit. Mais, ensuite, au vu du résultat calculé par la simulation, il se peut que nous passions à d'autres démarches : parfois, nous testons à nouveau le modèle, pour tirer des conclusions, afin de le recalibrer. Ces ajustements effectués sur lui, ne sont pas anodins. Ils constituent un fonctionnement nouveau en apprentissage, qui sera traité plus longuement à la fin du cours, dans la partie relative au traitement cognitif.

#### h) Exemple de traitement quantitatif : gestion de stock

Nous avons déjà décrit ce fonctionnement : dans un circuit logique combinatoire, nous câblons une fonction numérique. Le registre mémoire est initialisé à la valeur  $x_0$  ; à chaque impulsion de l'horloge, il passe de l'état  $x$  à l'état suivant :  $x - k$ . Nous obtenons une ligne décroissante qui modélise la diminution du stock géré au fil du temps.

L'historique ci-contre montre la courbe obtenue pour les valeurs  $x_0=100$  et  $k=1$ .



À l'issue de cette simulation, l'expérimentateur déduit qu'à l'étape 4, le stock est épuisé.

### 5.D) Application : le séquenceur, bien que simple, capable de prévoir le futur

Après avoir présenté le séquenceur utilisé en simulation, ici nous le présentons dans une autre application : la prévision à long terme.

#### a) Le séquenceur est déjà capable de prévoir le futur

Quand le modèle servant à simuler est bien au point, nous pouvons l'utiliser au sein d'agents capables d'anticiper. Dans un tel fonctionnement, quand la simulation dans le micro-monde s'arrête, le système fournit le futur état d'arrivée calculé du monde.

En le comparant avec le but visé, l'agent peut tirer des enseignements. Si la prévision est prometteuse, alors il décide de continuer l'action dans le monde réel, mais s'il considère qu'elle induit un problème, il peut décider de l'inhiber.

<sup>11</sup> Nous retrouvons ici la notion de *méta-regard*.

Par le biais de cette prévision à long terme, l'être vivant obtient une certaine emprise sur son environnement ; nous ne pouvons pas forcément conclure qu'il le maîtrise, mais nous pouvons déjà dire qu'il le subit moins.

**b) Note : quel peut être l'intérêt d'une telle simulation ?**

Le lecteur critique, qui connaîtrait le fonctionnement de la simulation classique en IA, pourrait formuler des réserves. En effet, le séquenceur ainsi utilisé en simulation fournit une information bien tenue à propos du monde simulé : elle est réduite à un unique symbole. Alors quel est son intérêt ?

Tout d'abord, le premier intérêt de ce séquenceur est qu'il est très simple. Il est déjà capable de prévoir le futur, pourtant, l'implémentation électronique ou informatique de ce processus si précieux demeure extrêmement légère. Elle se réduit à un registre temporaire rebouclé sur lui-même au moyen d'une mémoire morte associative ou d'une fonction numérique récurrente. Cependant il faut reconnaître que nous n'avons pas précisé comment implémenter une structure de contrôle. En effet, il nous reste à expliquer le départ et l'arrêt du traitement, i.e. comment sont implémentés l'initialisation et l'arrêt de notre montage.

Nous pouvons encore répondre à cette critique, en plaidant qu'en IA nous utilisons avec profit, une structure qui pourtant est encore moins performante. En effet, il existe une planification fournissant elle aussi un unique symbole, mais limitée à une profondeur unaire : elle prévoit seulement le futur proche. C'est l'anticipation immédiate, qui cependant présente un intérêt. Dans les SMA (système multi agents) elle évite à l'agent des dangers imminents et mortels : la chute dans un précipice, la menace d'un prédateur à l'affût... Alors si le séquenceur ne fournit lui aussi qu'une information réduite à un symbole, par contre il prévoit à long terme, i.e. à profondeur N, à échéance de plusieurs étapes. Alors, il est donc encore plus intéressant que l'anticipation immédiate.

Enfin, il faut aussi mettre en avant le contexte. Certes cette planification au moyen d'un séquenceur est piteuse comparée à la planification classique de l'IA symbolique, mais elle trouve sa place dans un contexte de vie artificielle, quand des agents de deux espèces différentes sont en concurrence. En effet, le mutant qui le premier implémente cette nouvelle aptitude se trouve avantagé : il augmente ses chances de survie et de reproduction. Ainsi, il serait privilégié par le mécanisme de sélection naturelle.

De plus, il faut ajouter que la planification classique symbolique, à chaque itération de son traitement, utilise une étape de simulation pour calculer l'état d'arrivée obtenu en appliquant l'action à tester.

**c) Conclusion : efficace et bon marché**

En IA, et en particulier en Vie Artificielle, il existe des contextes où le séquenceur en simulation présente déjà bien des intérêts pour l'agent qui l'implémente. Un tel agent augmente ses chances de résister au combat pour la vie.

## 6) Conclusion

À l'issue de ce cours sur les séquenceurs, mine de rien, nous avons touché à plusieurs démarches de l'IA.

D'abord nous avons rapidement évoqués les démarches d'anticipation, de planification, de méta-regard.

Mais surtout nous avons formalisé les démarches de généralisation, abstraction ; typage, incarnation, spécification, spécialisation ; et surtout de simulation.

## 7) Bibliographie

[Pea 89] Guisepe Peano  
*The principles of arithmetic, presented by a new method.*  
Jean van Heijenoort, 1889.

[Pea 67] Guisepe Peano  
*A Source Book in Mathematical Logic, 1879-1931.*  
Harvard Univ. Press : 83-97, 1967.

## Sommaire

Le séquenceur pour faire de la simulation.....	1
1) Introduction : effectuer des simulations.....	1
1.A) Présenter succinctement notre démarche.....	1
a) Partir des mathématiques.....	1
b) Les séquenceur : des SF très simples.....	1
c) Implémentation électronique des séquenceurs.....	1
d) Évacuer les séquenceurs irréguliers.....	1
e) Travailler avec les séquenceurs réguliers (S.Réguliers).....	1
f) Jusqu'aux Séquenceurs Quotidiens (S.Quotidiens).....	2
1.B) Présentation des S.Réguliers.....	2
a) La facette valeur est instanciée par un scalaire.....	2
b) Démarche, de généralisation au prix d'une d'abstraction.....	3
c) Passer du qualitatif de l'anticipation, au quantitatif de la simulation.....	3
1.C) Présentation des S.Quotidiens.....	3
a) Fabriquer des S.Quotidiens à partir de S.Réguliers.....	3
b) Démarche de spécialisation.....	4
c) Le type : une marque d'appartenance à une classe.....	4
2) Taxinomie des séquenceurs.....	5
2.A) Les SF .....	5
2.B) Séquenceurs libres (SL) ou séquenceurs à états libres : sous-classe des SF.....	5
a) Les SL généralisent une grande variété de séquenceurs.....	5
b) Décrire le traitement effectué par les SL.....	5
c) Séquenceurs dont le domaine est constitué de variables libres.....	5
d) Séquenceurs dont le graphe est déterminé par la donnée des règles.....	5
e) Note sur une convention de notation graphique.....	6
2.C) S.Réguliers, un domaine numérique défini en compréhension.....	6
a) Introduction.....	6
b) Le domaine des S.Réguliers est défini en compréhension.....	6
c) Les états du domaine des S.Réguliers sont définis en compréhension.....	6
d) Par construction, leur graphe est celui d'un séquenceur.....	7
e) Aspect effectif des S.Réguliers.....	7
2.D) En conclusion : .....	7
2.E) Les Séquenceurs Quotidiens (S.Quotidiens).....	8
a) Positionnement.....	8
b) Le travail du cogniticien est de modéliser le réel, la vie quotidienne.....	8
c) Plonger un S.Régulier dans un domaine lui fait hériter le type correspondant.....	8
d) Plonger un séquenceur conserve graphe et procès mais change ses symboles.....	9
e) Conclusion : .....	9
2.F) Les séquenceurs irréguliers (SI), au domaine défini en extension.....	9
3) Principe général et synoptique de la simulation.....	10
3.A) Étape 1 : départ depuis le micro-monde réel.....	10

3.B) Étape 2 : une représentation du micro-monde.....	10
3.C) Étape 3 : la simulation proprement dite.....	10
3.D) Étape 4 : le résultat de ce traitement simulé.....	10
3.E) Étape 5 : état d'arrivée simulé du micro-monde.....	10
3.F) Étape 6 : évolution réelle du monde, au fil du temps.....	10
4) Observer le monde réel et modéliser sa structure avec un séquenceur.....	11
4.A) Introduction de la notion de S.Quotidiens équivalents.....	11
a) Définition de deux S.Quotidiens équivalents.....	11
b) Classe d'équivalence de S.Quotidiens.....	11
c) Exemple - le séquenceur qui boucle de 00 à 11.....	11
4.B) Plongeon S.Réguliers→S.Quotidiens : de l'abstrait au concret.....	11
a) Quelques précisions sur le verbe plonger.....	11
b) Exemple 1 :.....	12
c) Exemple 2 :.....	12
4.C) Observer, pour modéliser le monde réel avec un S.Régulier.....	12
a) Étape 1 d'analyse.....	12
b) Étape 2 d'abstraction.....	12
c) Exemple - des séquenceurs qui bouclent de 1 à 5.....	12
d) Exemple - des séquenceurs qui bouclent de 1 à 7.....	13
4.D) Bilan.....	13
5) Le séquenceur utilisé en simulation.....	14
5.A) La simulation : imaginer l'évolution de notre monde réel.....	14
5.B) Les étapes d'une simulation effectuée au moyen d'un séquenceur irrégulier.....	14
a) Introduction.....	14
b) Étape 1 et 2 - Démarche de modélisation : écrire le séquenceur.....	14
c) Étape 3 - Transition d'état, par application de règle de réécriture.....	15
d) Étape 4 - L'arrêt du système donne l'état d'arrivée du micro-monde.....	15
e) Étape 5 - Exploitation du résultat obtenu.....	15
f) Étape 6 - Vérification du calage du modèle.....	15
g) Appliquer à un exemple de gestion de l'espace : vérifier un itinéraire séquentiel.....	16
5.C) Simulation au moyen de S.Réguliers ou quotidiens.....	16
a) Intro. : quantifier l'évolution du monde, deviner les valeurs de ses états futurs.....	16
b) Le travail de modélisation consiste en une triple démarche d'abstraction.....	16
c) Transition d'état par application de règle de réécriture effectuée sur un S.Régulier.....	17
d) L'arrêt du système.....	18
e) Nous obtenons l'état d'arrivée du micro-monde.....	18
f) Retour au réel du quotidien.....	18
g) Remarque : le test et le recalage du modèle relèvent de l'apprentissage.....	18
h) Exemple de traitement quantitatif : gestion de stock.....	18
5.D) Application : le séquenceur, bien que simple, capable de prévoir le futur.....	18
a) Le séquenceur est déjà capable de prévoir le futur.....	18
b) Note : quel peut être l'intérêt d'une telle simulation ?.....	19
c) Conclusion : efficace et bon marché.....	19
6) Conclusion.....	20

---

7) Bibliographie.....20