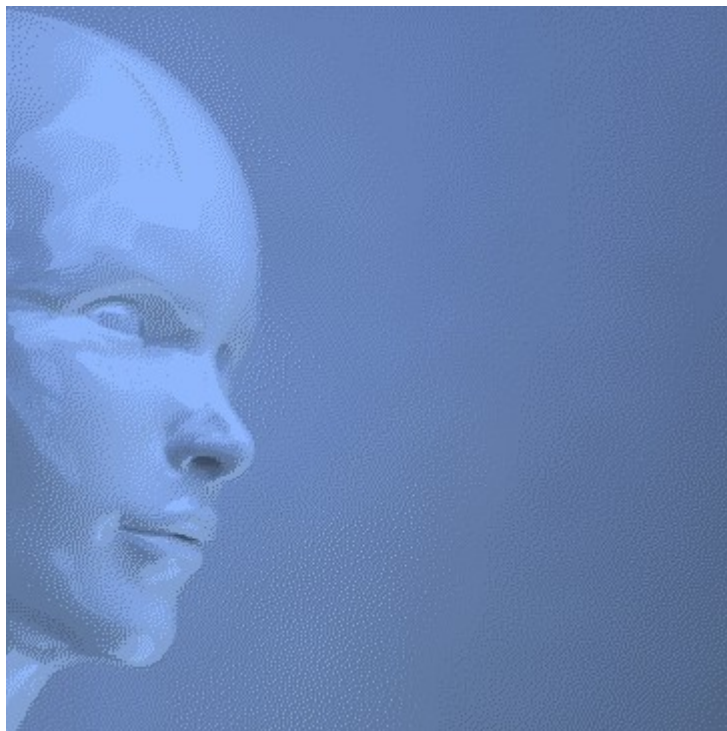


# Est ce que les ordinateurs pourrons penser?

## Découvrez-le vous-même



Cette théorie n'a pas été mis à jour pour 2010.

Voulez-vous aider à traduire?

S'il vous plaît contactez-moi via <https://www.mafait.org/contact>.

## Table des matières

1. Introduction.....	3
1.1. Le concept « Thinknowlogy ».....	3
2. L'algèbre et la logique dans le langage naturel.....	4
2.1. Une définition généralité/spécialité.....	4
2.1.1. Une relation-spécificité.....	5
2.1.2. Un groupement de généralités.....	5
2.1.2.1. Un groupement de spécificités.....	5
2.1.3. Une affectation.....	6
2.1.3.1. Une affectation-spécificité.....	6
2.1.3.2. Une affectation-généralité.....	6
2.2. Une sélection.....	7
2.5. Poser des questions par soi-même.....	8
2.6. Chercher les relations causales.....	8
2.7. Semantic ambiguity.....	9
2.7.1. Autonomous semantic disambiguation.....	9
3. La construction de la structure de connaissance.....	10
3.1. La création d'une généralité/spécificité.....	10
3.1.1. La création de nouveaux mots.....	11
3.1.2. La création d'un lien spécificité.....	11
3.1.3. La création d'un groupement-spécificité.....	12
3.1.4. La création d'une affectation.....	13
3.1.4.1. La création d'une affectation-spécificité.....	13
3.1.4.2. La création d'une affectation-généralité.....	14
3.2. La création d'une sélection.....	15
3.2.1. L'exécution des sélections.....	15
4. Dialogues.....	16
4.1. La grammaire.....	16
4.1.1. La lecture d'une phrase.....	16
4.1.1.1. Seules les structures connues de phrase sont acceptées.....	17
4.1.2. Répondre par une phrase.....	17
4.1.2.1. Poser des questions de lui-même.....	17
5. Applications.....	18

## 1. Introduction

- Pouvons-nous réaliser plus de choses avec un logiciel informatique que seulement des tâches de calculs, de rentrées de données et l'automatisation de processus de travail?
- Pourrons-nous un jour poser des questions à un moteur de recherche sur Internet au lieu de chercher des mots, des combinaisons de mots sur une page d'Internet?
- Est ce que les ordinateurs pourrons comprendre un jour le langage naturel?
- Les ordinateurs peuvent traiter des données. Mais est ce que les ordinateurs pourront un jour traiter des informations? Autrement dit, pourrons-nous un jour programmer l'information?

Il y a eu beaucoup de tentatives de faites pour faire « penser » les ordinateurs, mais sans concept qui soutient ces tentatives, de telles tentatives restent vaines. En fait on ne peut réussir qu'avec un concept bien élaboré.

### 1.1. Le concept « Thinknowlogy »

Le concept « Thinknowlogy » est basé sur les principes suivants:

1. Le langage de programmation est basé sur l'algèbre et la logique.
2. Tout individu a un sens inné pour l'algèbre et la logique, même si toute fois ce sens inné est plus développé chez certains individus que chez d'autres. Cela se remarque très bien dans le langage naturel.
3. En combinant l'algèbre et la logique dans le langage naturel avec l'algèbre et la logique dans le langage de programmation, il est possible de programmer dans le langage naturel.
4. Le but final est de programmer l'information.

Premièrement nous verrons comment l'algèbre et la logique du langage naturel peut-être combinées avec l'algèbre et la logique du langage de programmation, ceci nous permettant de programmer en langage naturel.

Dans un second temps nous verrons comment l'information peut-être programmée.

## 2. L'algèbre et la logique dans le langage naturel

Ci-dessous nous citerons quelques aspects de l'algèbre et de la logique dans le langage naturel.

### 2.1. Une définition généralité/spécialité

L'homme organise ses pensées en les groupant, en les mettant dans des casiers, en généralisant. Généraliser signifie en fait: séparer les faits principaux, des faits secondaires. Les faits principaux sont appelés généralités et les faits secondaires sont appelés spécificités.

Le processus de séparation des généralités, des spécificités se trouve caché au niveau le plus bas du langage. Considérons la phrase: "*Jean est un homme.*".

"*Jean*" est le sujet et "*homme*" est ce qui caractérise "*Jean*". Autrement dit: "*Jean*" est la généralité de la spécificité "*homme*", parce que "*homme*" dit quelque chose de spécifique sur le sujet "*Jean*".

Plus de spécificités sur Jean:

- "*Jean est un père.*"
- "*Jean est un boulanger.*"

Dans les phrases citées ci-dessus nous avons utilisé un article indéfini: "*Jean est un homme.*", "*Jean est un père.*" et "*Jean est un boulanger.*". Cette structure dans le concept Thinknowlogy est nommée une **définition généralité/spécialité**.

### 2.1.1. Une relation-spécificité

Considérons la phrase: “*Jean est le père de Pierre.*”.

A travers cet exemple nous pouvons également déduire que la généralité “*Jean*” a une relation de “*père*” avec “*Pierre*”. “*Père*” est ici appelé une **relation-spécificité** et “*Pierre*” est appelé la relation.

Dans cet exemple la relation-spécificité est également une affectation parce qu’un article défini est utilisé: “*Jean est le père de Pierre.*”.

De même les phrases “*Bush est le président des États-Unis.*” et “*Obama est le président des États-Unis.*” ont des relation-spécificités. “*Bush*” et “*Obama*” ont en effet une relation-“*président*” avec des “*États-Unis*”.

### 2.1.2. Un groupement de généralités

Dans l’exemple précédent, les généralités étaient à chaque fois les mêmes et les spécificités étaient différentes.

Considérons maintenant les phrases suivantes:

- “*Bush est le président des États-Unis.*”
- “*Obama est le président des États-Unis.*”

Dans ce cas les spécificités sont semblables, mais les généralités sont différentes. Parce que “*Bush*” et “*Obama*” ont dans ce contexte apparemment affiliés, ces généralités doivent être rassemblées.

Nous appelons cela un **groupement de généralités**.

#### 2.1.2.1. Un groupement de spécificités

Dans une définition généralité/spécialité, la généralité peut avoir plusieurs spécificités qui sont liées les unes aux autres, comme par exemple: “*Un verre est plein ou vide.*”. Ou de façon plus spécifique: “*Un verre est plein, à moitié plein, ou vide.*”.

Un tel groupe de spécificités est nommé un **groupement de spécificités**.

### 2.1.3. Une affectation

Un certain article donne l'état actuel d'une structure généralité/spécialité. Nous appelons cela une **affectation**.

#### 2.1.3.1. Une affectation-spécificité

Exemple: "*Le verre est à moitié plein.*".

Nous avons ici une structure généralité/spécialité avec un groupement de spécificités dans lequel un article défini est utilisé. Nous appelons cela une **affectation-spécificité**.

Une affectation-spécificité est en même temps une structure généralité/spécialité, car ceci est également valable: "*Un verre est à moitié plein.*".

Étant donné qu'une indication donne l'état actuel des choses, l'état peut donc changer: Nous aurons tout d'abord la situation suivante "*Le verre est plein.*". Puis quand le verre est vidé, nous aurons: "*Le verre est vide.*".

#### 2.1.3.2. Une affectation-généralité

Prenons la phrase:

- "*Bush est le président des États-Unis.*"
- "*Obama est le président des États-Unis.*"

Ce sont des affectations car un article défini est utilisé. Il est vrai que les généralités sont différentes et qu'elles forment un groupement de généralités. C'est pourquoi ces affectations sont appelées des **affectations-généralité**.

## 2.2. Une sélection

Le langage naturel connaît ce qu'on appelle des sélections, comme dans la phrase suivante: *“Si le feu est orange ou rouge, alors tu dois t'arrêter.”*. Donc dans le cas où *“le feu est orange ou rouge”* s'en suit l'action *“Tu dois t'arrêter.”*.

Souvent, d'autres alternatives sont possibles: sinon si le feu est vert, l'action est *“Tu dois filer.”*. En résumé: *“si le feu est rouge ou orange, tu dois t'arrêter, sinon tu dois filer.”*.

Une sélection se compose donc de 2 ou 3 parties:

- une condition (*“si le feu est orange ou rouge”*),
- une action (*“Tu dois t'arrêter.”*)
- et éventuellement une action alternative (*“Tu dois filer.”*).

Ici nous avons à faire aux structures généralité/spécialité:

- La condition: *“Un feu est orange ou rouge.”*;
- L'action: *“Tu dois t'arrêter ou filer.”*.

Aussi bien les conditions et les actions sont des affectations:

- Les affectations d'une condition sont utilisées pour définir si la condition donnée est vraie. La condition dans l'exemple est donc vraie, si une des deux affectations est réalisée: *“Le feu est orange.”* ou *“Le feu est rouge.”*;
- En fonction de ce résultat, l'affectation de l'action ou de l'action alternative sera réalisée: *“Tu dois t'arrêter.”* ou *“Tu dois filer.”*.

Cette théorie n'a pas été mis à jour pour 2010.

Voulez-vous aider à traduire?

S'il vous plaît contactez-moi via <http://mafait.org>.

## ***2.5. Poser des questions par soi-même***

Si le système n'a pas toutes les informations nécessaires, est-il concevable que le système demande à l'utilisateur ce qu'il lui manque? Même des informations contradictoires peuvent être révélées et par le biais de questions posées, corrigées. Nous élaborerons cela par la suite.

## ***2.6. Chercher les relations causales***

Dans un texte, il est possible de chercher les relations causales et donc d'amener un système à tirer des conclusions sur la base des informations qui sont déjà connues. Nous élaborerons cela par la suite.



## 2.7. Semantic ambiguity

Two types of ambiguity can be distinguished: **static** ambiguity and **dynamic** (e.g. time-related) ambiguity.

An example of static ambiguity:

- “*Boston is a city in both the United States and the United Kingdom*”.

An example of dynamic ambiguity:

- “*Bush is inaugurated as (the) president of the United States.*”, because George H. W. Bush was inaugurated in 1989, and his son George W. Bush was inaugurated in 2001, and re-inaugurated in 2005.

### 2.7.1. Autonomous semantic disambiguation

When a sentence is entered and its context (semantics) is not clear, the system can either:

- use deduction to determine which context is meant by the user;
- make an assumption, when the meant context cannot be determined, but when it is quite obvious;
- or ask a question, when the system has no clue about the context.

### 3. La construction de la structure de connaissance

Quelqu'un qui a déjà programmé a sans doute reconnu dans le chapitre précédent les principes de base du langage de programmation: Un **assignment** sous la forme d'affectation, une structure **if-then-else** dans le sens d'une sélection et peut-être une **déclaration d'une variable** sous la forme d'une définition généralité/spécialité: on donne la structure de la variable, mais on n'attribue pas encore de valeur à la variable.

Grâce à ces points communs entre le langage naturel et le langage de programmation, on peut construire une structure de connaissance. Ainsi se produit un lien entre le langage naturel et le langage de programmation qui rend la programmation dans le langage naturel possible, et pose les bases pour un ordinateur « pensant ».

Ci-dessous nous allons expliquer en plusieurs étapes comment une structure de connaissance peut-être construite à partir de phrases en langage naturel.

#### **3.1. La création d'une généralité/spécificité**

Un élément important de la structure de connaissance est la généralité/spécialité. Nous allons décrire ci-dessous les différents aspects de cette structure et comment à partir de cela la structure de connaissance peut être construite.

### 3.1.1. La création de nouveaux mots

Considérons la phrase: “*Jean est un père.*”.

Le système doit tout d’abord créer les mots “*Jean*” et “*père*”, avant de pouvoir créer la structure de connaissance. Donc si ces mots n’existent pas encore dans le système ou s’ils ne sont pas grammaticalement les mêmes, nous devons tout d’abord les créer.

#### Jean

Le mot “*Jean*” s’écrit avec une lettre capitale et est le premier mot de la phrase. Il peut donc appartenir au groupe grammatical des noms propres ou à un groupe grammatical inconnu, si le mot ne commençait pas par une lettre capitale.

Dans ce cas deux mots sont créés:

- “*Jean*” du genre grammatical *noms propres*;
- “*jean*” ((sans lettre capitale) d’un genre grammatical inconnu.

Lors de la construction de la structure de connaissance, un des deux mots seulement sera utilisé. Le mot non utilisé sera ensuite supprimé afin que le système ne soit pas encombré par des données non utilisées. Nous reviendrons sur ce point plus tard.

#### père

Le mot “*père*” est précédé d’un article, de ce fait le genre grammatical du mot est clair: c’est un substantif.

Dans la phrase “*Un verre est plein, à moitié plein ou vide.*”, il n’est pas clair à quel genre grammatical les mots “*plein*”, “*à moitié plein*” et “*vide*” appartiennent. Mais nous partons du principe qu’ils appartiennent tous les trois au même genre grammatical.

### 3.1.2. La création d’un lien spécificité

A partir du moment où le système connaît les mots, nous pouvons procéder à leur assemblage. Plus justement dit: le mot-généralité peut être assemblé au mot-spécificité.

Exemple: “*Jean est un père.*” et “*Jean est un boulanger.*”.

Dans la première phrase un lien de spécificité est établi entre le mot-généralité “*Jean*” et le mot-spécificité “*père*” et dans la deuxième phrase le lien est établi entre “*Jean*” et “*boulanger*”.

Dans la structure de connaissance il est donc clair que, Jean est à la fois un père et un boulanger.

### 3.1.3. La création d'un groupement-spécificité

Considérons la phrase: “*Un verre est plein, à moitié plein ou vide.*”.

Ici la généralité “*verre*” à plusieurs spécificités qui dépendent les unes des autres, à l’inverse du premier exemple où à la généralité sont attachées deux spécificités qui sont indépendantes l’une de l’autre.

Ces spécificités qui sont dépendantes les unes des autres sont ‘regroupées’. Pour réaliser cela nous devons utiliser des liens de groupement pour les relier les unes aux autres:

- “*plein*” est lié à “*à moitié plein*” par un lien de groupement montant;
- “*à moitié plein*” est lié à “*vide*” par un lien de groupement montant;
- “*vide*” est lié à “*à moitié plein*” par un lien de groupement descendant;
- “*à moitié plein*” est lié à “*plein*” par un lien de groupement descendant.

Le lien de groupement ‘montant’ ou ‘descendant’ donne l’ordre de l’énumération.

Considérons les phrases suivantes:

- “*Bush est le président des États-Unis.*”
- “*Obama est le président des États-Unis.*”

Les généralités “*Bush*” et “*Obama*” sont liées par un lien de groupement qui les relie l’une l’autre:

- “*Bush*” est lié à “*Obama*” par un lien de groupement montant;
- “*Obama*” est lié à “*Bush*” par un lien de groupement descendant.

### 3.1.4. La création d'une affectation

Une affectation est une structure généralité/spécialité avec une valeur attribuée. C'est pourquoi il faut d'abord construire une structure généralité/spécificité pour ensuite construire la structure d'affectation.

Ainsi dans l'exemple "*Le verre est à moitié plein.*", le lien-spécificité entre le "*verre*" et "*à moitié plein*" doit être premièrement créé – si celui-ci n'existe pas encore – et ensuite un lien d'affectation entre "*verre*" et "*à moitié plein*" devra être créé de la même façon.

Une structure généralité/spécialité est en principe statique. Une structure d'affectation est dynamique car la situation peut changer.

#### 3.1.4.1. La création d'une affectation-spécificité

Considérons la phrase: "*Un verre est plein, à moitié plein ou vide.*".

Une attribution ne peut, à l'intérieur d'un regroupement-spécificité, que rendre une des trois situations: "*plein*", "*à moitié plein*" ou "*vide*".

Considérons la situation (affectation) "*Le verre est à moitié plein.*". Si le contenu du verre est bu, la situation est devenue: "*Le verre est vide.*". Et donc le lien d'affectation de "*verre*" à "*à moitié plein*" disparaît, car cette situation appartient au même regroupement-spécificité. Ensuite il y a un lien d'affectation qui est créé de "*verre*" à "*vide*", afin de signifier que la situation du verre est maintenant vide.

#### Plusieurs affectations par généralité

Si une structure généralité/spécialité a plusieurs groupements-spécificité, une structure peut aussi avoir plusieurs affectations.

Par exemple: si tu dis tout d'abord: "*Jean est le père de Pierre.*", et un peu plus tard "*Jean est le boulanger de Marie.*", "*Jean*" a dans ce cas deux affectations, étant donné que "*père*" et "*boulanger*" ne forment pas un regroupement-spécificité.

Afin d'ajouter à la relation-spécificité que "*Jean est le père de Pierre.*", le mot-généralité "*Jean*" doit être relié à la spécificité "*père*" et à la relation "*Pierre*" par un lien spécificité.

On utilise donc un lien-spécificité contenant une seconde indication. Ce lien particulier est aussi appelé **lien-relation-spécificité** ou **lien-relation**.

Ce lien-relation doit être lu de la façon suivante: “Jean” a une relation-“père” avec “Pierre”. De même “Bush” et “Obama” ont une relation-“président” avec “les États-Unis”.

### **3.1.4.2. La création d'une affectation-généralité**

Considérons les phrases:

- “Bush est le président des États-Unis.”
- “Obama est le président des États-Unis.”

“Bush est le président des États-Unis.” vient en premier lieu, puis après les élections ce sera “Obama est le président des États-Unis.”, ainsi l’affectation de “Bush” à “le président des États-Unis” doit être supprimée et l’affectation de “Obama” à “le président des États-Unis” devra être créée.

## 3.2. La création d'une sélection

Tout comme la généralité/spécialité, la sélection est un élément de base pour la structure de connaissance.

Pour construire une structure de connaissance à partir d'une sélection, il faut tout d'abord construire une structure généralité/spécialité à partir de laquelle la sélection sera construite. Ensuite la condition sera enregistrée sur une [liste-condition](#) opgeslagen, l'action sur une [liste-action](#) et l'action alternative sur une [liste-action-alternative](#).

Considérons la phrase: *“Si le feu est orange ou rouge, tu dois t'arrêter, et sinon tu dois filer.”*.

Tout d'abord nous aurons la généralité/spécialité *“Le feu est orange ou rouge.”*, *“Tu dois t'arrêter.”* et *“Tu dois filer.”*. Ensuite nous aurons sur la liste des conditions *“Le feu est orange ou rouge.”* avec une référence à l'action associée et à l'action alternative. *“Tu dois t'arrêter.”* (sur la liste-action) et *“Tu dois filer.”* (sur la liste-action-alternative).

### 3.2.1. L'exécution des sélections

Après que le système ait lu une phrase et l'ait traitée, le système lit toutes les conditions dans la structure de connaissance et exécute éventuellement l'action associée ou l'action alternative. Ce processus sera répété jusqu'à ce qu'il n'y ait plus de modification possible dans le système. Le système est alors « en repos ». Ensuite le système pourra lire la phrase suivante et la traiter.

C'est un peu comme si le système « pensait », mais en fait c'est tout simplement un programme écrit en langage naturel qui est exécuté.

## 4. Dialogues

Dans les chapitres précédents nous avons transformé des phrases du langage naturel en une structure de connaissances. Mais pour parvenir à dialoguer avec l'utilisateur, il faut que le système puisse répondre par des phrases lisibles. Pour cela, il faut que la structure de connaissances soit de nouveau transformée en langage naturel.

### 4.1. La grammaire

Afin de pouvoir lire et répondre dans la langue choisie, il est nécessaire que le système connaisse la grammaire de cette langue. Ainsi, il faudra définir auparavant la grammaire de chaque langue sélectionnée.

Pour que le système demeure flexible, la grammaire ne sera pas programmée « hard-coded », dans le système, mais sera enregistrée dans un fichier informatique sous forme de texte. Pendant le démarrage du système, il sera procédé à la lecture de 1 ou plusieurs fichiers informatiques de grammaire.

Il est aussi possible de dialoguer dans plusieurs langues et finalement il doit être possible que le système traduise automatiquement, mais en ce qui concerne cette dernière possibilité je préfère y revenir ultérieurement.

#### 4.1.1. La lecture d'une phrase

La grammaire est utilisée lors de la lecture d'une phrase:

- Pour contrôler si la phrase est conforme à la grammaire définie;
- Pour différencier les différentes structures, telles que les généralité/spécialités, les affectation et les sélections;
- Pour définir à l'avance certains mots (à traiter ultérieurement).



#### **4.1.1.1. Seules les structures connues de phrase sont acceptées**

Nous avons vu précédemment que pour transformer des phrases en structure de connaissances, il est nécessaire d'avoir un concept pour chaque structure de phrase (généralité/spécialités, affectation et sélections), afin de pouvoir traduire l'information donnée par ces phrases, dans une structure de connaissances. Nous avons vu également que ces structures de phrases étaient enregistrées dans un fichier informatique de grammaire.

Autrement dit: le système accepte seulement des structures de phrases qui sont enregistrées dans un fichier informatique de grammaire, parce que pour ces structures seulement il existe un concept qui permet de transformer l'information contenue dans ces phrases en structure de connaissances.

**Le but final de ce projet est de penser un concept pour un nombre le plus grand possible de structures de phrases. Un concept qui traduit l'information contenue dans ces structures de phrases en une structure de connaissances.**

Cela prendra donc encore beaucoup de temps et coûtera beaucoup d'efforts pour atteindre ce but, mais il sera alors possible de traiter l'information automatiquement; l'ordinateur 'pensant' est alors en vue.

#### **4.1.2. Répondre par une phrase**

On utilise la grammaire dans une réponse phrasée pour:

- Sélectionner la structure de phrase voulue;
- Pour que les mots que le système veut présenter à l'utilisateur, forment une phrase grammaticalement correcte.

##### **4.1.2.1. Poser des questions de lui-même**

Le système peut reconnaître les lacunes et les contradictions dans une structure de connaissances (à traiter ultérieurement).

En plus de cela, le système peut poser des questions, de lui-même, à l'utilisateur sur les lacunes et les contradictions trouvées, pouvant ainsi compléter, corriger, éclaircir les informations enregistrées. Cela peut paraître comme un comportement intelligent et donc renforcer le sentiment que les ordinateurs peuvent penser.

## 5. Applications

Cette théorie n'a pas été mis à jour pour 2010.

Voulez-vous aider à traduire?

S'il vous plaît contactez-moi via <http://mafait.org>.