Introduction au Traitement Automatique du Langage

Mise en perspective historique : IA/informatique/linguistique

Frédéric Béchet, équipe TALEP, LIS UMR 7020

Qu’est ce que le « TAL » ?

Un domaine particulier de l’Intelligence Artificielle !

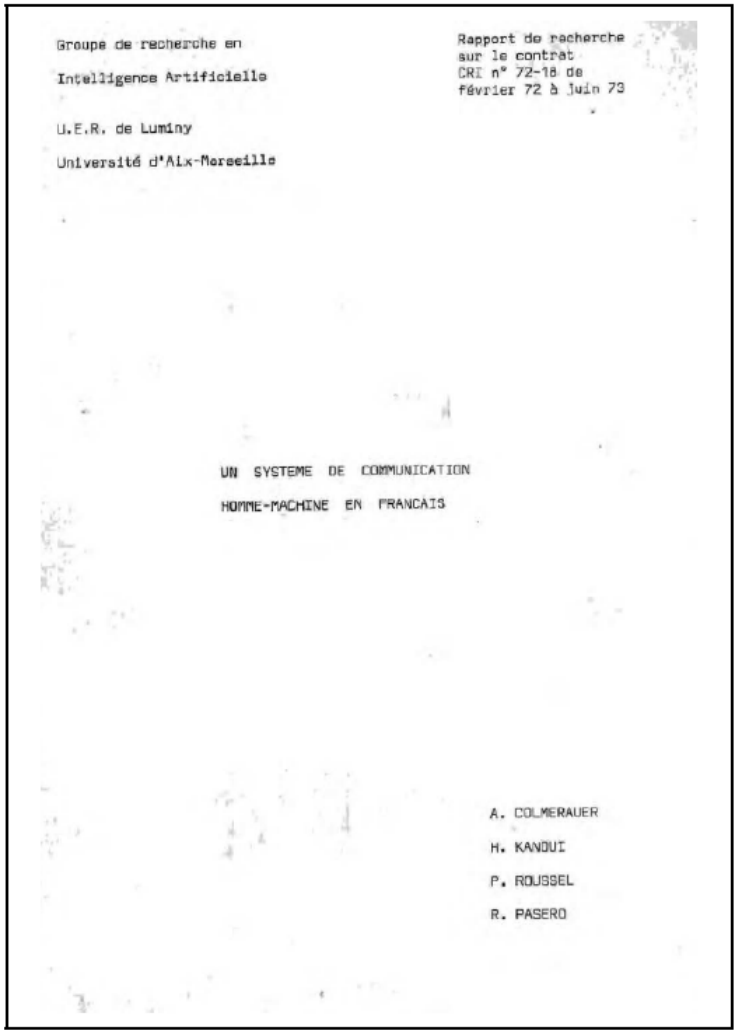
Intelligence Artificielle ?

Que dit la loi ?

- **intelligence artificielle**
  - *Abréviation : IA.*
  - *Domaine : Informatique.*
  - *Définition*
    - ⋆ *Champ interdisciplinaire théorique et pratique qui a pour objet la compréhension de mécanismes de la cognition et de la réflexion, et leur imitation par un dispositif matériel et logiciel, à des fins d'assistance ou de substitution à des activités humaines.*
  - *Voir aussi*
    - ⋆ *apprentissage automatique, apprentissage non supervisé, apprentissage par renforcement, apprentissage profond, apprentissage supervisé, dialogueur, réseau de neurones artificiels.*
  - *Équivalent étranger : artificial intelligence (AI).*
  - *Attention : Cette publication annule et remplace celle du Journal officiel du 22 septembre 2000.*

sources : JORF n0285 du 9 décembre 2018 texte n 58 : Vocabulaire de l'intelligence artificielle (liste de termes, expressions et définitions adoptés)

IA à Marseille : une histoire ancienne



Groupe de recherche en
Intelligence Artificielle

U.E.R. de Luminy
Université d'Aix-Marseille

Rapport de recherche
sur le contrat
CRI n° 72-18 de
février 72 à juin 73

UN SYSTEME DE COMMUNICATION
HOMME-MACHINE EN FRANCAIS

A. COLMERAUER
H. KANOUI
P. ROUSSEL
R. PASERO

AVANT - PROPOS

En février 1972, le Groupe d'Intelligence Artificielle de Luminy recevait une subvention de 180 000,00 francs dans le cadre du contrat CRI 72-18 intitulé "Communication homme-machine en langue naturelle avec déduction automatique". Ce contrat se termina en juin 1973.

L'objet du contrat était de mettre au point un système expérimental, mais très général de traitement de phrases en français, entrées à partir d'une console d'ordinateur, en vue de dialoguer avec un utilisateur.

L'approche de ce problème difficile fut menée sur trois fronts:

1) Du point de vue linguistique par une étude sous un angle particulier de la syntaxe et la sémantique du français. Signalons à ce sujet la thèse de 3è cycle de R. PASERO: "Représentation du français en logique du 1er ordre en vue de dialoguer avec l'ordinateur".

2) Du point de vue démonstration automatique, plusieurs méthodes ont été étudiées et essayées sur ordinateur. A ce sujet voir la thèse de 3è cycle de P. ROUSSEL: "Définition et traitement de l'égalité formelle en démonstration automatique".

3) Du point de vue purement informatique un langage de programmation très particulier, PROLOG, a été développé. Ce langage à base de démonstration automatique a servi à programmer tout le système décrit dans cette brochure.

Les retombées de ce travail, autres que le système décrit ici, sont multiples. Signalons entre autre que PROLOG a permis de démarer beaucoup d'autres recherches.

Il peut être intéressant de savoir comment l'argent du contrat a été dépensé:

- à payer beaucoup d'heures machine

- à inviter R. KOWALSKI d'Edimbourg et J. TRUDEL de Montréal.

- à prendre contact avec les principaux laboratoires d'intelligence artificielle des Etats-Unis et d'Angleterre.

- à payer une secrétaire et des frais de secrétariat.

Première partie

IA / Traitement Automatique des langues Informatique

Présentation générale par un survol du domaine, à travers son contexte historique et scientifique, avec de nombreux emprunts à :

- Logique et Sémantique des langues naturelles (D. Bonnay)
- Modèles de Langage et Analyse Syntaxique (Antoine Rozenknop)
- Sémantique distributionnelle, embeddings (Philippe Langlais)
- Méthodes Statistiques en Traitement des Langues : État des lieux et perspectives (François Yvon)
- Introduction à la sémantique formelle (Alain Lecomte)
- Et de nombreux emprunts à mes collègues Alexis Nasr, Benoît Favre, Carlos Ramisch

Le Traitement Automatique des Langues (Gazdar, 1996)

- théorie du calcul linguistique

  (...) is the study of the computational, mathematical and statistical properties of natural languages and systems for processing natural languages.

- pyscho-linguistique computationnelle

  Computational psycholinguistics involves the construction of psycho-logically motivated computational models of aspects of human NLP.

- outils de traitement pour des applications

  Applied NLP involves the construction of intelligent computational artefacts that process natural languages in ways that are useful to people other than computational linguists.

Le Traitement Automatique des Langues (Gazdar, 1996)

- théorie du calcul linguistique

  (...) is the study of the computational, mathematical and statistical properties of natural languages and systems for processing natural languages.

- pyscho-linguistique computationnelle

  Computational psycholinguistics involves the construction of psycho-logically motivated computational models of aspects of human NLP.

- outils de traitement pour des applications

  Applied NLP involves the construction of intelligent computational artefacts that process natural languages in ways that are useful to people other than computational linguists.

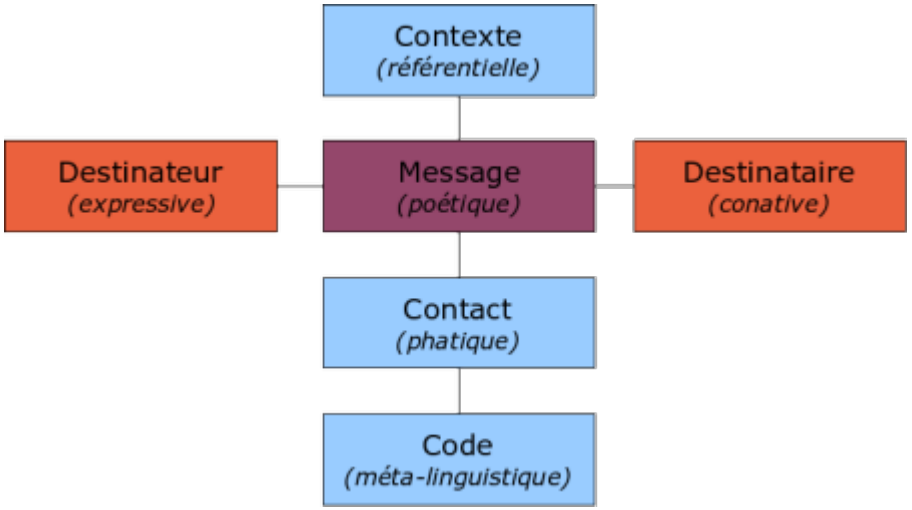
Histoire croisée : linguistique/logique/informatique/IA

(petite) histoire croisée de la linguistique et de l'informatique

- avant le XVIIIème siècle :
  - auteurs de grammaires descriptives
  - mathématiciens décrivant des méthodes générales de calcul et les inventeurs de "machines à calculer" mécaniques comme Pascal et Leibniz
- 1660
  - publication de la "Grammaire générale et raisonnée" (connue sous le titre "Grammaire de Port-Royal") d'Arnaud et Lancelot. Son ambition était de décrire les règles du langage en termes de principes rationnels universels
- aux XVIII et XIXème siècles
  - linguistique comparative et historique
  - logique "booléenne" (ou "propositionnelle") par Boole (1815-1864)
  - "logique des prédicats du 1er ordre" par Frege (1848-1925)
  - premiers projets de calculateurs mécaniques - Babbage (1791-1871)
- 1916
  - publication du "Cours de linguistique générale" du linguiste suisse Ferdinand de Saussure (1857-1913)
  - Introductions de concepts importants
    - langage = construction sociale d'un système de signes
      - signe = association arbitraire entre un signifiant et un signifié
    - Langage = faculté générale de s'exprimer au moyen de signes
    - Parole = utilisation concrète de signes linguistiques particuliers
    - deux axes d'analyse d'un discours, en tant que suite de signes :
      - axe syntagmatique
      - axe paradigmatique

(petite) histoire croisée de la linguistique et de l'informatique

- années 30-40 :
  - "cercle de Prague" développe la "linguistique structurale" - Roman Jakobson et Nicolas Troubetzkoy
  - phonologie : étude des sons élémentaires (les phonèmes)
  - Identification de six fonctions permises par le langage dans un contexte de communication :
    - fonction expressive permet au locuteur d'exprimer ses sentiments ;
    - fonction conative permet d'agir sur le destinataire (donner un ordre...)
    - fonction référentielle permet d'informer sur le monde extérieur
    - la fonction phatique permet de s'assurer du bon fonctionnement de la communication
    - la fonction poétique met l'accent sur la forme du message
    - la fonction métalinguistique permet de parler du langage grâce au langage

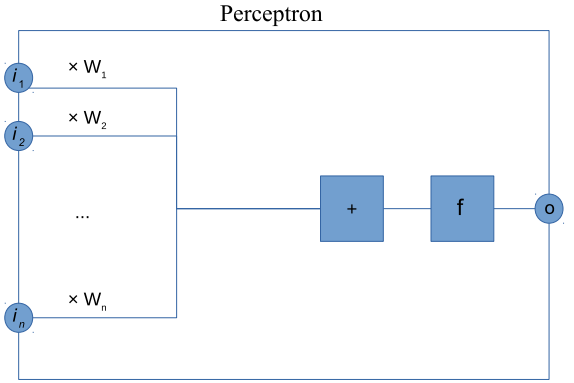


(petite) histoire croisée de la linguistique et de l'informatique

- 1936 : La Machine de Turing
  - Alan Turing (1912-1954) Mathématicien anglais qui propose en 1936 le dispositif plus tard appelé "machine de Turing" = date de naissance de l'informatique
- 1945 : Von Neumann (1903-1957)
  - Mathématicien et physicien américain propose un plan de construction des ordinateurs
- 1950 : « le test de Turing »
  - Dans "Machines de calcul et intelligence : « Je propose de réfléchir à la question : les machines peuvent-elles penser ? »
- 1951 : Marvin Minsky
  - SNARC (Stochastic Neural Analog Reinforcement Calculator), le premier simulateur de réseau neuronal, qui simule le comportement d'un rat apprenant à se déplacer dans un labyrinthe.
- 1952 : premier séminaire sur la Traduction Automatique au MIT
  - Organisé par Yehoshua Bar-Hillel (1915-1975)
- 1956 : séminaire de Darmouth (Marvin Minsky et John McCarthy)
  - Naissance du terme « Intelligence Artificielle »
- 1960-64 : Rapports Bar-Illel et ALPAC (Automatic Language Processing Advisory Committee)
  - Limites de la Traduction Automatique => fin des financements
- 1966 : dialoguer avec une machine : l'IA devient une réalité ?
  - "Eliza" de Weizenbaum = simulation d'un dialogue avec un psychothérapeute
- 1972 : Alain Colmerauer
  - Création du langage Prolog (PROgrammation en LOGique) à Marseille

(petite) histoire croisée de la linguistique et de l'informatique

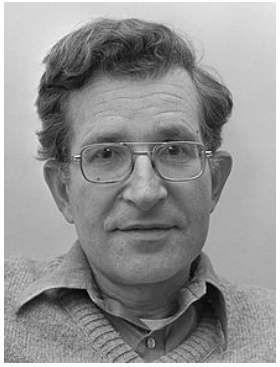
- 1890 W. James
  - Concept de mémoire associative - Loi de fonctionnement pour l'apprentissage
- 1943 W. McCulloch et W. Pitts
  - Modélisation du neurone biologique en neurone formel
- 1949 D. Hebb
  - Règle de Hebb
- 1957 F. Rosenblatt
  - Modèle du Perceptron
- 1969 M. Minsky et S. Papert
  - Mise en avant des limites du Perceptron - Abandon des recherches
- 1967-1982 S. Grossberg, T. Kohonen, etc.
  - Poursuite « déguisée » des recherches sur les réseaux de neurones
- 1982 J. J. Hopfield
  - Modèle de Hopfield - Théorie du fonctionnement et des possibilités des réseaux de neurones
- 1983 Machine de Boltzmann
  - Dépassement des limites du Perceptron, mais manque d'efficacité
- 1985 Algorithme de Rétropropagation du gradient, réseaux multicouches
- 2010 La « révolution du Deep Learning »



$$o = f\left(\sum_{k=1}^{n} i_k \cdot W_k\right)$$

(petite) histoire croisée de la linguistique et de l'informatique

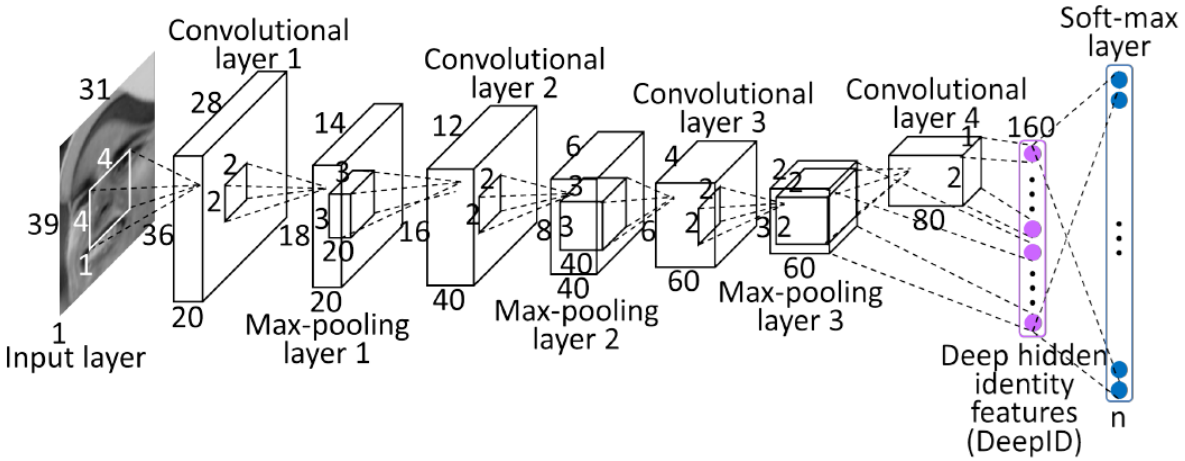
- Noam Chomsky (né en 1928) :
  - linguiste et activiste politique, professeur de linguistique au MIT depuis 1961
  - primauté de la syntaxe sur tous les autres niveaux d'analyse du langage
  - les hommes disposent à la naissance d'un "organe du langage" de nature mentale = notion de "grammaire universelle" innée
  - Distinction entre :
    - ***compétence linguistique (connaissance des règles de fonctionnement d'une langue)***
    - performance linguistique (mise en oeuvre effective de ces règles en compréhension ou en production)
  - but de toute théorie linguistique = explication des jugements de grammaticalité
  - la structure de surface (syntaxe) d'un énoncé détermine sa structure profonde (les relations sémantique).
  - 1957 : publication de "Syntactic Structures" - théorie des "grammaires génératives et transformationnelles"
  - années 80 : l'approche "Principle and Parameters" (Gouvernment and Binding)
  - En informatique, hiérarchie de Chomsky pour caractériser des familles de langages de complexité croissante
    - fondement de la théorie des langages formels, branche dans laquelle sont étudiées les propriétés des langages artificiels comme les langages de programmation informatiques.



It must be recognized that the notion “probability of a sentence" is an entirely useless one, under any known interpretation of this term. Noam Chomsky, 1969

(petite) histoire croisée de la linguistique et de l'informatique

- années 70-80 :
  - sémantique formelle pour représenter des connaissances et formaliser des raisonnements
  - nouvelles logiques (logique floue, logiques modales, etc.)
  - "scripts" (Roger Schank), "frames" (Marvin Minsky), "réseaux sémantiques", "graphes conceptuels" (John Sowa)
  - Les "systèmes experts", basés sur des modèles symboliques du même genre, constituent alors la vitrine de l'IA
  - Grammaires exprimée en PROLOG
- depuis les années 90 :
  - augmentation considérable de la capacité de stockage et de calcul des ordinateurs
  - développement exponentiel d'Internet
  - émergence d'une linguistique de corpus fondée sur l'exploitation de textes au format numérique
  - linguistique plus empirique, fondée sur les données plutôt que sur des modèles formels abstraits
  - progrès de "l'apprentissage automatique", branche de l'IA permettant à un programme d' « apprendre » à partir d'exemples
- Actuellement
  - Le règne de la nouvelle IA : le DEEP LEARNING !!!!



Méthodes formelles

- Méthodes basées sur la notion de ***compétence***
- Linguistique formelle
  - « réduire » la langue naturelle à une langue « formelle », « artificielle » sur laquelle pourront s'effectuer des calculs

TAL = Compilation ?

TAL – Méthodes formelles

- La syntaxe formelle est une théorie de la manière dont sont construites les phrases d'une langue L
- La sémantique formelle est une théorie de la manière dont ces phrases peuvent être interprétées du point de vue du sens

En quoi ces théories sont -elles dites formelles?

- nous ne prenons en compte que la forme des énoncés
  - on pourrait prendre en compte des contenus intuitifs, spiritualistes, qui n'ont pas de manifestation concrète
  - la forme d'un énoncé est sa seule caractéristique tangible, admise par tous les locuteurs d'une langue, on peut donc en faire une théorie
- nous essayons de représenter la signification des énoncés au moyen d'un langage formel
  - si ce n'était pas le cas : risque de régression infinie
  - un langage formel est non ambigu

Qu'est ce qu'un langage formel ?

Un langage formel est un système abstrait, doté d'un ensemble dénombrable de symboles et d'un nombre fini de règles.
En général on demandera qu'un tel langage soit:

- non ambigu (chaque expression signifiante est dotée d'une seule signification)
- récursif ou, au moins, "récursivement énumérable"

Deuxième point:
soit une expression quelconque P

- Cas idéal: on peut toujours en un temps fini déterminer si oui ou non elle appartient au langage: on dit que le langage est récursif ou décidable
- Cas plus faible: étant donné une expression correcte, on peut toujours arriver au bout d'un temps fini à dire qu'elle est correcte, mais si elle est incorrecte.... On dit en ce cas que le langage est seulement récursivement énumérable ou semi-décidable

Un ensemble de symboles : $p, q, r, s, ..., \wedge, \vee, \neg, \implies, (,)$
Un ensemble de règles :

- $p, q, r, s, ...$ sont des expressions correctes
- si A est une expression correcte, $(\neg A)$ aussi en est une
- si A, B sont des expressions correctes, $(A \wedge B)$, $(A \vee B)$, $(A \implies B)$ sont aussi des expressions correctes

Syntaxe

- Comment décrire complètement un langage ?
  - Liste exhaustive des phrases acceptables
- Syntaxe = lexèmes + structure syntaxique
  - partie visible du langage
  - Les règles de syntaxe définissent la forme des phrase admises dans le langage
- Lexèmes: unités syntaxiques élémentaires
  - mots-clés, identificateurs, opérateurs, séparateurs, …
- Structure syntaxique: spécification des séquence admissibles de lexèmes = notion commune de *grammaire*
  - Règles de formation des phrases acceptables
  - Arbres syntaxiques

Grammaire Formelle

- Usuellement, la spécification de la grammaire repose sur la définition d'un certain nombre de "catégories syntaxiques" et des relations existant entre elles.
- Au travers des catégories syntaxiques on peut reconnaître une "structure" dans une *phrase* du langage. Par exemple une *phrase* d'une langue naturelle peut avoir la structure suivante:

```
1. PHRASE  → SUJET  PREDICAT
2. SUJET  → PHRASE-NOMINALE
3. PHRASE-NOMINALE  → NOM-PROPRE
4. PHRASE-NOMINALE  → ARTICLE  NOM-COMMUN
5. PREDICAT  → VERBE
6. PREDICAT  → VERBE  PHRASE-NOMINALE
```

From: J Hulaas, Univ Genève

Grammaire Formelle

- Deux classes de symboles ont été introduites ici:

–les catégories syntaxiques (ici en majuscules):

•les ***symboles non terminaux***

–les mots constitutifs de la phrase lorsque le processus de génération se termine:

•les ***symboles terminaux***

- Le symbole non terminal utilisé pour débuter le processus de génération (dans notre exemple: *PHRASE*) est appelé :

–***symbole initial*** (ou *de départ* ou encore *axiome*).

- Le processus de génération consiste dans l'application, à chaque pas, d'une règle de réécriture appelée *production*, jusqu'à ce qu'aucune règle ne puisse être appliquée ou que l'on ait éliminé tous les symboles non terminaux.

From: J Hulaas, Univ Genève

Définition

Une ***grammaire formelle*** est un quadruplet (T, N, R, a) où:

- T est un ensemble fini non vide de symboles dit *alphabet terminal*, dont les éléments sont appelés *symboles terminaux* et sont ici par convention en lettres minuscules.
- N est un ensemble fini non vide de symboles dit *alphabet non terminal*. Les alphabets T et N sont disjoints, leur union définit l'alphabet global: $V = N \cup T$.
- R est l'ensemble fini et non vide des règles grammaticales, ou productions; chaque production est de la forme $\alpha \rightarrow \beta$ où $\alpha \in V^*$ est appelé *tête* ou *membre gauche*, et $\beta \in V^*$ est appelé *corps* ou *membre droit*. V^* est l'ensemble de toutes les séquences formées de symboles appartenant au vocabulaire V, y compris la chaîne vide dénotée ε. La tête α contient au moins un symbole non terminal.
- a constitue *l'axiome*, soit un élément particulier de N, ou *symbole de départ*.

From: J Hulaas, Univ Genève

Classification de Chomsky

- La définition des grammaires génératives donnée ci-dessus n'impose aucune contrainte sur les productions.

- En introduisant des limitations sur la forme de ces productions, Noam Chomsky a introduit en 1956 une classification hiérarchique des grammaires et des langages qui est très généralement acceptée.

- Chomsky s'intéresse avant tout aux langues naturelles, mais il n'en constitue pas moins un pionnier de l'informatique !

From: J Hulaas, Univ Genève

Classification hiérarchique de Chomsky

Les quatre types hiérarchiques de Chomsky sont:

- Le type 0: il n'y a aucune restriction sur les productions
  - Pas d'algorithme d'analyse efficace correspondant
- Le type 1: dite *dépendante du contexte*; toutes les productions ont la forme:

  $$pgq \rightarrow pdq \quad \text{où} \quad g \in N,\ p,q \in V^*,\ d \in V^* - \varepsilon$$

  et ε est la chaîne vide
  - En d'autres termes, *p* et *q* représentent le 'contexte'. On a toujours $|\ \alpha\ | \leq |\ \beta\ |$ (c'est-à-dire que longueur(α) $\leq$ longueur (β))
  - Toute grammaire dépendante du contexte possède un algorithme d'analyse syntaxique

From: J Hulaas, Univ Genève

Classification hiérarchique de Chomsky

- Le type 2: dite *indépendante du contexte*; toutes les productions ont la forme

$$A \rightarrow \beta \quad \text{où} \quad A \in N \text{ et } \beta \in V^*$$

c'est-à-dire que le symbole non terminal A peut être remplacé par β indépendamment du contexte dans lequel il se trouve.

- La grande majorité des langages de programmation sont décrits par une grammaire indépendante du contexte
- Les grammaires indépendantes du contexte ne sont pas un strict sous-ensemble des grammaires dépendantes du contexte (qui excluent la chaîne vide)

From: J Hulaas, Univ Genève

Exemple

La grammaire $G = (\{ E, T, F \}, \{ i, +, *, /, -, (,) \}, R, E)$
avec les règles de production R

- `E = T`
- `E = T + T;` `E = T - T`
- `T = F`
- `T = F * F;` `T = F / F`
- `F = i`
- `F = (E)`

génère le langage indépendant du contexte (noté *L(G)*) caractérisé comme contenant toutes les expressions arithmétiques de la variable *i*

From: J Hulaas, Univ Genève

Classification de Chomsky

- Le type 3: dite *régulière*; une telle grammaire régulière peut prendre deux formes:
  - soit $A \rightarrow tB$ ou $A \rightarrow t$, t étant une chaîne terminale ($t \in T^*$) et A et B des symboles non terminaux. Forme appelée *grammaire linéaire à droite*;
  - soit $A \rightarrow Bt$ ou $A \rightarrow t$, t étant une chaîne terminale ($t \in T^*$) et A et B des symboles non terminaux. Forme appelée *grammaire linéaire gauche*.
- Les grammaires régulières sont sont un sous-ensemble des grammaires indépendantes du contexte
- Les expressions régulières désignent la forme des règles des grammaires régulières

From: J Hulaas, Univ Genève

De la syntaxe vers la sémantique formelle

Richard Montague :

Il n'y a selon moi aucune différence théorique importante entre les langues naturelles et les langages artificiels des logiciens.

Syntaxe / sémantique formelle

Richard Montague :

Il n'y a selon moi aucune différence théorique importante entre les langues naturelles et les langages artificiels des logiciens.

Objectifs de la linguistique formelle :

syntaxe formelle caractériser de manière systématique les énoncés grammaticaux d'une langue donnée.

sémantique formelle sur la base de notre compréhension de la syntaxe, rendre compte de manière compositionnelle de la signification des phrases.

pragmatique formelle sur la base de notre compréhension de la syntaxe et de la sémantique, expliquer comment sont utilisés les énoncés.

Sémantique : calcul d'un sens et vérité ?

- Ceci suppose que les expressions aient une signification
- La solution la plus simple consiste à donner comme signification une valeur choisie dans un ensemble fini
- En général, on choisit la notion de valeur de vérité

Dans l'exemple précédent, certaines expressions ont une valeur de vérité égale à 1, d'autres une valeur de vérité égale à 0

On en vient à la **thèse de G. Frege**:

La signification d'une phrase réside dans ses conditions de vérité

Comprendre la signification de P c'est savoir à quelles conditions P est vraie

Calul d'un sens et vérité ?

G. Frege reprend et reformule l'antique théorie d'Aristote concernant la division entre sujet et prédicat:
que l'on ait:
les Grecs ont battu les Perses à la bataille de Platée
ou
les Perses furent battus par les Grecs à la bataille de Platée
on infère les mêmes conséquences, donc elles ont "même signification" (ou, dit Frege: "*même contenu conceptuel*")
or, elles ont des représentations sujet/prédicat différentes.
Frege remplace les notions de sujet et de prédicat par celles d'objet et de fonction
les Grecs ont battu les Perses = $\phi(\Gamma, \Pi)$

Logique des prédicats

Cela conduit à la logique des prédicats

Problème rencontré par Frege: la circularité (paradoxe de Russell, 1901)

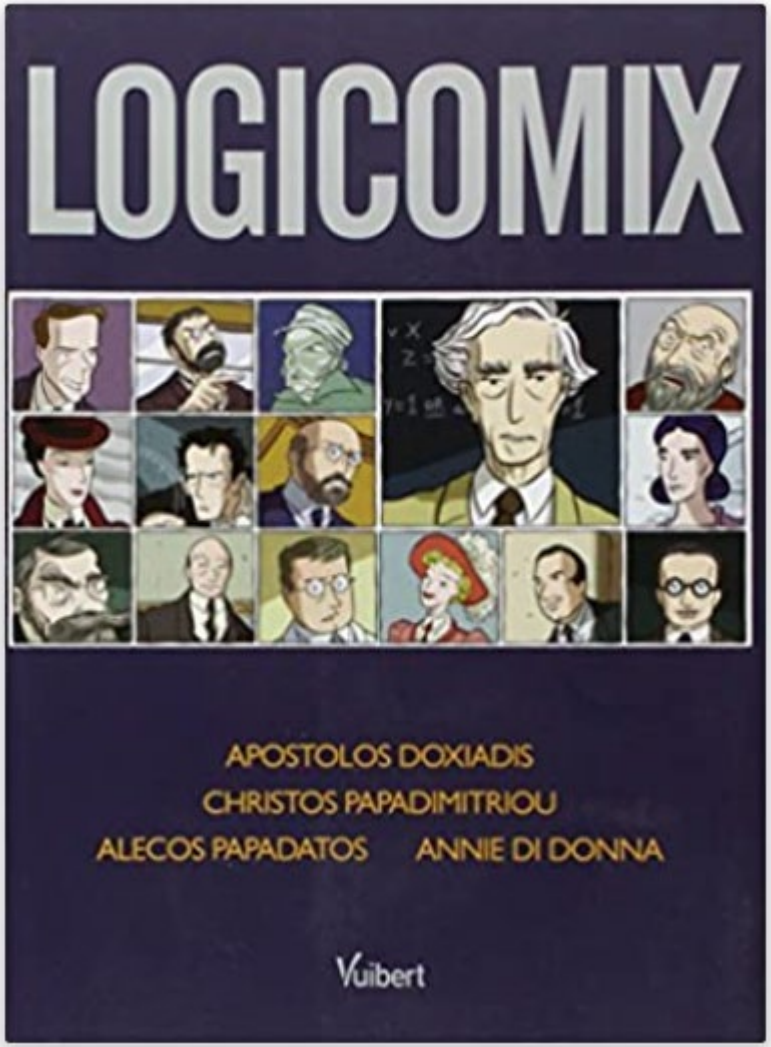
A priori, rien n'empêche qu'une fonction soit appliquée à elle même (cf. "être un prédicat est un prédicat" : $\phi(\phi)$ est vrai). Mais alors $\neg\phi(\phi)$ a aussi un sens, celui de "ne pas s'appliquer à soi-même". Soit donc Φ la fonction "ne pas s'appliquer à soi-même". Est-ce que $\Phi(\Phi)$ ou bien est-ce que $\neg\Phi(\Phi)$?

- si $\Phi(\Phi)$, alors "ne pas s'appliquer à soi-même" s'applique à soi-même, ce qui veut dire que "ne pas s'appliquer à soi-même" s'applique à "ne pas s'appliquer à soi-même", donc "ne pas s'appliquer à soi-même" ne s'applique pas à soi-même, c'est-à-dire: $\neg\Phi(\Phi)$
- si $\neg\Phi(\Phi)$, alors "ne pas s'appliquer à soi-même" ne s'applique pas à soi-même, ce qui veut dire que "ne pas s'appliquer à soi-même" ne s'applique pas à "ne pas s'appliquer à soi-même", donc "ne pas s'appliquer à soi-même" s'applique à soi-même, c'est-à-dire: $\Phi(\Phi)$

Logique d'ordre différents

Alors, il faut **hiérarchiser** les entités.
Il y a des entités d'ordre 0 (les individus), des entités d'ordre 1 (les propriétés d'individus, ou ensembles d'individus), des entités d'ordre 2 (les propriétés des propriétés qu'ont les individus, ou ensembles d'ensembles), et ainsi de suite...
On convient qu'une fonction d'ordre n ne peut s'appliquer qu'à un objet d'ordre $n - 1$
On appelle logique du premier ordre la logique qui ne comporte que des entités d'ordre 0 ou 1
Les autres sont des logiques dites d'ordre supérieur
parmi elles on distinguera la logique du second ordre

Idée lecture ludique sur ces thèmes !!



Et donc ….

- Le TAL est :
  - Une sorte de « compilation » des langages « humains »
  - Transformer le langage humain en langage « formel »
    - Similaire aux langages des mathématiciens/logiciens/informaticiens
- Faire du TAL nécessite donc :
  - de définir formellement ce qu'est un langage
  - de décrire son pouvoir d'expression
  - de développer des algorithmes permettant de
    - tester la « grammaticalité » d'une phrase (par rapport à un langage formel)
    - calculer le « sens » d'une phrase, c'est-à-dire sa véracité

Mais ….

Ce cours est-il donc un autre cours de Théorie des langages / compilation ?

- Et non !!
  - La vision « IA/logique/langage formel » du TAL n'est pas la seule vision possible !!
    - Le langage c'est « juste » de la « data »
    - Le langage c'est « juste » un signal
    - Le langage c'est « juste » une interface pour des applications
    - ….

Alors ?

Quelle « vision du TAL » allons nous étudier ?

- Un mélange !!
- Principalement des méthodes empiriques
  - Basées sur l'analyse de données
  - Et l'apprentissage automatique
- Mais ..
  - Les méthodes et représentations formelles sont toujours là !!
  - Comme inspiration/justification/explication/..

Finalement

- « Apprendre » le langage humain à une machine peut ressembler au processus d'apprentissage du langage chez l'enfant
  - Apprentissage par l'exemple
    - interaction avec parents / écoute / lecture
  - Apprentissage par l'enseignement
    - école => représentation formelle
- Nouvelle direction des recherches en sciences cognitives
  - Parallèle entre apprentissage machine et apprentissage humain