

Sémantique et Traduction des Langages

Majeure Sciences et Ingénierie du Logiciel

Marc Pantel

2020 – 2021

Organisation

- ▶ Cours : 10 séances – Marc Pantel
- ▶ TD : 8 séances – Marc Pantel/Neeraj Singh
- ▶ TP : 9 séances – Marc Pantel/Neeraj Singh
- ▶ Mini Projets (20%) en binôme : 2 séances de suivi + 1 test
Finalisation travaux de TP sur langage fonctionnel MINIML
Finalisation travaux de TP sur langage impératif MINIC
- ▶ Projet (40%) en quadrinôme : 5 séances de suivi + 1 test
Extension du langage MINIC avec technologies objets
- ▶ Examen (40%) : 1h30 avec documents
- ▶ Urgent : Constituer les quadrinômes et binômes associés
- ▶ Alternative 1 : Pas de confinement
 - ▶ Travaux Dirigés en présenciel
 - ▶ Travaux Pratiques à décider
- ▶ Alternative 2 : Confinement
 - ▶ Combinaison TD/TP à distance

Organisation

- ▶ Cours : 10 séances – Marc Pantel
- ▶ TD : 8 séances – Marc Pantel/Neeraj Singh
- ▶ TP : 9 séances – Marc Pantel/Neeraj Singh
- ▶ Mini Projets (20%) en binôme : 2 séances de suivi + 1 test
Finalisation travaux de TP sur langage fonctionnel MINIML
Finalisation travaux de TP sur langage impératif MINIC
- ▶ Projet (40%) en quadrinôme : 5 séances de suivi + 1 test
Extension du langage MINIC avec technologies objets
- ▶ Examen (40%) : 1h30 avec documents
- ▶ Urgent : Constituer les quadrinômes et binômes associés
- ▶ Alternative 1 : Pas de confinement
 - ▶ Travaux Dirigés en présenciel
 - ▶ Travaux Pratiques à décider
- ▶ Alternative 2 : Confinement
 - ▶ Combinaison TD/TP à distance

Organisation

- ▶ Cours : 10 séances – Marc Pantel
- ▶ TD : 8 séances – Marc Pantel/Neeraj Singh
- ▶ TP : 9 séances – Marc Pantel/Neeraj Singh
- ▶ Mini Projets (20%) en binôme : 2 séances de suivi + 1 test
Finalisation travaux de TP sur langage fonctionnel MINIML
Finalisation travaux de TP sur langage impératif MINIC
- ▶ Projet (40%) en quadrinôme : 5 séances de suivi + 1 test
Extension du langage MINIC avec technologies objets
- ▶ Examen (40%) : 1h30 avec documents
- ▶ **Urgent :** Constituer les quadrinômes et binômes associés
- ▶ Alternative 1 : Pas de confinement
 - ▶ Travaux Dirigés en présenciel
 - ▶ Travaux Pratiques à décider
- ▶ Alternative 2 : Confinement
 - ▶ Combinaison TD/TP à distance

Organisation

- ▶ Cours : 10 séances – Marc Pantel
- ▶ TD : 8 séances – Marc Pantel/Neeraj Singh
- ▶ TP : 9 séances – Marc Pantel/Neeraj Singh
- ▶ Mini Projets (20%) en binôme : 2 séances de suivi + 1 test
Finalisation travaux de TP sur langage fonctionnel MINIML
Finalisation travaux de TP sur langage impératif MINIC
- ▶ Projet (40%) en quadrinôme : 5 séances de suivi + 1 test
Extension du langage MINIC avec technologies objets
- ▶ Examen (40%) : 1h30 avec documents
- ▶ Urgent : Constituer les quadrinômes et binômes associés
- ▶ Alternative 1 : Pas de confinement
 - ▶ Travaux Dirigés en présenciel
 - ▶ Travaux Pratiques à décider
- ▶ Alternative 2 : Confinement
 - ▶ Combinaison TD/TP à distance

Organisation

- ▶ Cours : 10 séances – Marc Pantel
- ▶ TD : 8 séances – Marc Pantel/Neeraj Singh
- ▶ TP : 9 séances – Marc Pantel/Neeraj Singh
- ▶ Mini Projets (20%) en binôme : 2 séances de suivi + 1 test
Finalisation travaux de TP sur langage fonctionnel MINIML
Finalisation travaux de TP sur langage impératif MINIC
- ▶ Projet (40%) en quadrinôme : 5 séances de suivi + 1 test
Extension du langage MINIC avec technologies objets
- ▶ Examen (40%) : 1h30 avec documents
- ▶ **Urgent** : Constituer les quadrinômes et binômes associés
- ▶ Alternative 1 : Pas de confinement
 - ▶ Travaux Dirigées en présenciel
 - ▶ Travaux Pratiques à décider
- ▶ **Alternative 2 : Confinement**
 - ▶ Combinaison TD/TP à distance

Plan du cours

- ▶ Introduction
 - ▶ Rappels : Modélisation, Automates et Graphes, GLS
 - ▶ Architecture générale
 - ▶ Formes de sémantique
- ▶ Interprétation
 - ▶ Sémantique opérationnelle
 - ▶ Sémantique axiomatique
- ▶ Compilation
 - ▶ Table des Symboles, Arbre abstrait
 - ▶ Typage
 - ▶ Modèle mémoire, Génération de code
 - ▶ Sémantique translationnelle, dénotationnelle
- ▶ Vérification de correction

Plan du cours

- ▶ Introduction
 - ▶ Rappels : Modélisation, Automates et Graphes, GLS
 - ▶ Architecture générale
 - ▶ Formes de sémantique
- ▶ **Interprétation**
 - ▶ **Sémantique opérationnelle**
 - ▶ **Sémantique axiomatique**
- ▶ Compilation
 - ▶ Table des Symboles, Arbre abstrait
 - ▶ Typage
 - ▶ Modèle mémoire, Génération de code
 - ▶ Sémantique translationnelle, dénotationnelle
- ▶ Vérification de correction

Plan du cours

- ▶ Introduction
 - ▶ Rappels : Modélisation, Automates et Graphes, GLS
 - ▶ Architecture générale
 - ▶ Formes de sémantique
- ▶ Interprétation
 - ▶ Sémantique opérationnelle
 - ▶ Sémantique axiomatique
- ▶ Compilation
 - ▶ Table des Symboles, Arbre abstrait
 - ▶ Typage
 - ▶ Modèle mémoire, Génération de code
 - ▶ Sémantique translationnelle, dénotationnelle
- ▶ Vérification de correction

Plan du cours

- ▶ Introduction
 - ▶ Rappels : Modélisation, Automates et Graphes, GLS
 - ▶ Architecture générale
 - ▶ Formes de sémantique
- ▶ Interprétation
 - ▶ Sémantique opérationnelle
 - ▶ Sémantique axiomatique
- ▶ Compilation
 - ▶ Table des Symboles, Arbre abstrait
 - ▶ Typage
 - ▶ Modèle mémoire, Génération de code
 - ▶ Sémantique translationnelle, dénotationnelle
- ▶ Vérification de correction

Rappels

- ▶ Modélisation :
 - ▶ Structure algébrique des langages
 - ▶ Spécification des langages :
 - ▶ Expressions régulières,
 - ▶ Grammaire (règles de production, EBNF, Conway)
- ▶ Automates et Théorie des Langages
 - ▶ Automates, Automates à piles, Analyseur descendant récursif
 - ▶ Générateurs d'analyseurs lexicaux et syntaxiques
- ▶ Ingénierie Dirigée par les Modèles
 - ▶ Métamodèles :
 - ▶ Représentation abstraite du langage (MOF),
 - ▶ Règles de bonne formation (OCL)
 - ▶ Syntaxe concrète texte : Xtext

Rappels

- ▶ Modélisation :
 - ▶ Structure algébrique des langages
 - ▶ Spécification des langages :
 - ▶ Expressions régulières,
 - ▶ Grammaire (règles de production, EBNF, Conway)
- ▶ Automates et Théorie des Langages
 - ▶ Automates, Automates à piles, Analyseur descendant récursif
 - ▶ Générateurs d'analyseurs lexicaux et syntaxiques
- ▶ Ingénierie Dirigée par les Modèles
 - ▶ Métamodèles :
 - ▶ Représentation abstraite du langage (MOF),
 - ▶ Règles de bonne formation (OCL)
 - ▶ Syntaxe concrète texte : Xtext

Rappels

- ▶ Modélisation :
 - ▶ Structure algébrique des langages
 - ▶ Spécification des langages :
 - ▶ Expressions régulières,
 - ▶ Grammaire (règles de production, EBNF, Conway)
- ▶ Automates et Théorie des Langages
 - ▶ Automates, Automates à piles, Analyseur descendant récursif
 - ▶ Générateurs d'analyseurs lexicaux et syntaxiques
- ▶ Ingénierie Dirigée par les Modèles
 - ▶ Métamodèles :
 - ▶ Représentation abstraite du langage (MOF),
 - ▶ Règles de bonne formation (OCL)
 - ▶ Syntaxe concrète texte : Xtext

Principes essentiels

Communication = Echange d'informations

- Besoins :
- ▶ Représenter les informations possibles
 - ▶ Reconnaître une information
 - ▶ Exploiter une information

Organisation stratifiée : information structurée

Informatique : Science du traitement de l'information

Computer science : Science de la « machine à calculer »

- Essentiel :
- ▶ Description et manipulation de l'information (langage),
 - ▶ Traitement d'une information quelconque,
 - ▶ Traitement d'une manipulation quelconque

- D'où :
- ▶ Description formelle du langage
 - ▶ Génération automatique des outils de manipulation

Références bibliographiques

- ▶ Hopcroft, Ullman, Introduction to automata theory, languages and computation, Addison-Wesley, 1979.
- ▶ Stern, Fondements mathématiques de l'informatique, McGraw-Hill, 1990.
- ▶ Carton, Langages formels, calculabilité et complexité, Vuibert, 2008.
- ▶ Aho, Sethi, Ullman, Compilateurs : Principes, Techniques et Outils, InterEditions, 1989.
- ▶ Fisher, Leblanc, Crafting a compiler in ADA/in C, Benjamin Cummings, 1991.
- ▶ Wilhem, Maurer, Les compilateurs : Théorie, construction, génération, Masson, 1994.
- ▶ Appel, Modern Compiler Implementation in Java/ML/C, Cambridge University Press, 1998.
- ▶ Winskel, The formal semantics of programming languages : An introduction, MIT Press, 1993.
- ▶ Lämmel, Software Languages : Syntax, Semantics and Metaprogramming, Springer (under review), 2017.

Exemple : fichier /etc/hosts

- Fichier tel qu'il est affiché :

```
#_Ceci_est_un_commentaire

127.0.0.1      ↪hal9000_localhost

#_En_voici_un_autre

147.127.18.144 ↪phoenix.enseeiht.fr
```

- Informations brutes : caractères

0000000	#	sp	C	e	c	i	sp	e	s	t	sp	u	n	sp	c	o
0000020	m	m	e	n	t	a	i	r	e	nl	nl	1	2	7	.	0
0000040	.	0	.	1	ht	h	a	l	9	0	0	0	sp	l	o	c
0000060	a	l	h	o	s	t	nl	nl	#	sp	E	n	sp	v	o	i
0000100	c	i	sp	u	n	sp	a	u	t	r	e	nl	nl	1	4	7
0000120	.	1	2	7	.	1	8	.	1	4	4	ht	p	h	o	e
0000140	n	i	x	.	e	n	s	e	e	i	h	t	.	f	r	nl
0000160																

Analyse lexicale

- Informations élémentaires : **commentaire**, **nombre**, **identificateur**, `.` (unités lexicales)

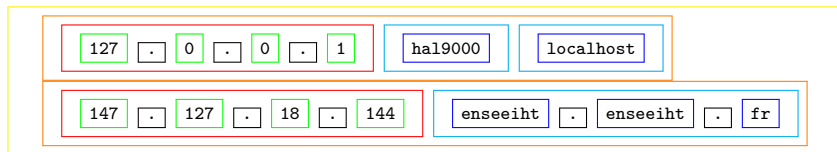
- Résultat de l'analyse lexicale :

```
# Ceci est un commentaire 127 . 0 . 0 . 1
hal9000 localhost # En voici un autre 147 .
127 . 18 . 144 enseiht . enseiht . fr
```

- Spécification des unités lexicales : Expressions régulières
 - Commentaire : $\#[\backslash n]^*\backslash n$
 - Nombre : $[0 - 9]^+$
 - Identificateur : $[a - bA - B][a - bA - B0 - 9]^*$

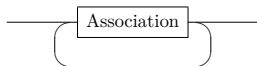
Analyse syntaxique

- Informations structurées (unités syntaxiques) :
 - Premier niveau : **adresse IP**, **nom qualifié**
 - Deuxième niveau : **association**
 - Troisième niveau : **document**

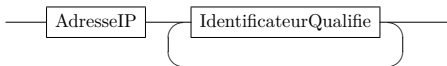


- Spécification des unités syntaxiques : Grammaires (notation de Conway)

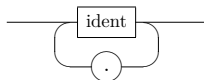
Document



Association



IdentificateurQualifie

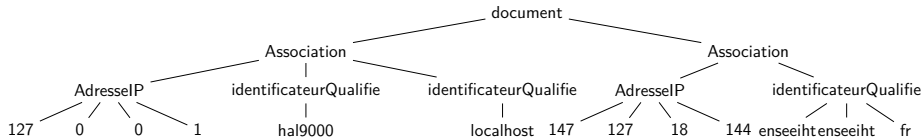


AdresseIP



Analyse sémantique

- Structure arborescente associée :



- Exploitation des informations : association nom qualifié/adresse IP (unités sémantiques)

hal9000

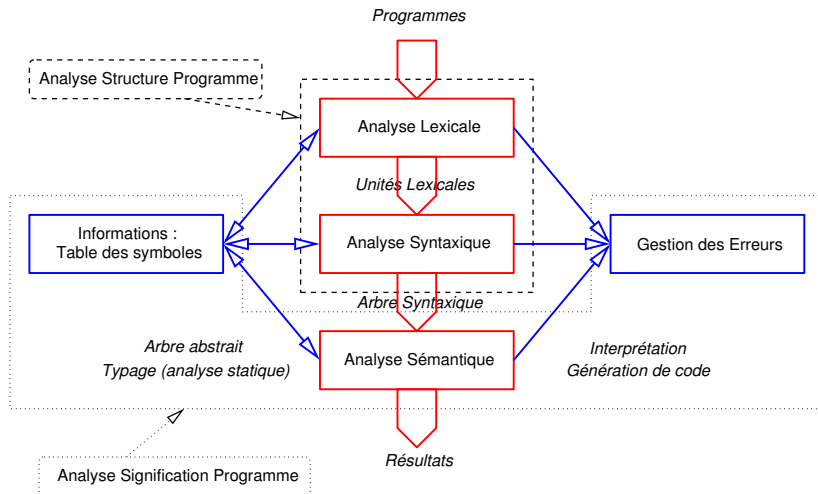
localhost

enseiht . enseiht . fr

127 . 0 . 0 . 1
127 . 0 . 0 . 1

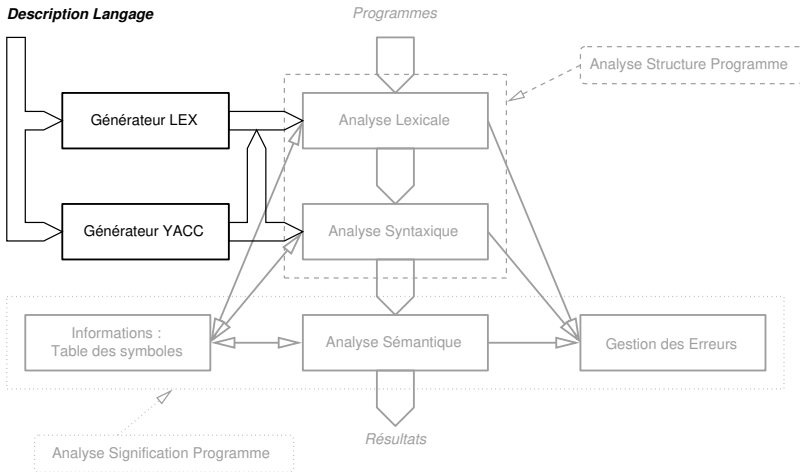
147 . 127 . 18 . 144

Structure d'un outil



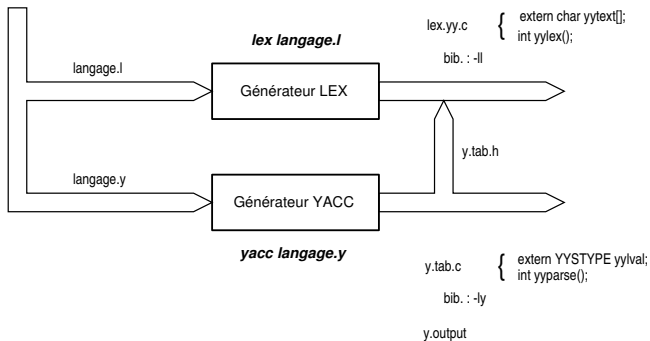
Exemple lex et yacc

Description Langage



Exemple lex et yacc

Description Langage



Définitions

- ▶ Caractère/Symbole : Unité élémentaire d'information
- ▶ Unité lexicale (lexème, mot) : Séquence de caractères
- ▶ Unité syntaxique (arbre syntaxique, syntème, phrase) : Arbre d'unités lexicales
- ▶ Unité sémantique : diverses (arbre abstrait, table des symboles, type, code généré, résultat évaluation, ...)

Comment organiser les informations ?

- ▶ Objectif : Exploitation des informations
- ▶ Règle : Choisir le bon niveau de précision
- ▶ Unité lexicale : Bloc élémentaire d'information pertinente
- ▶ Unité syntaxique : Élément structurant de l'information

Sémantique formelle des langages

- ▶ Objectif : Modélisation la sémantique avec des outils mathématique
- ▶ Atteindre la qualité de la modélisation de la syntaxe
- ▶ Etudier la cohérence et la complétude
- ▶ Prouver la correction des outils
- ▶ Générer automatiquement les outils
- ▶ Différentes formes :
 - ▶ Sémantique opérationnelle : Mécanisme d'exécution des programmes
 - ▶ Sémantique axiomatique : Mécanisme de vérification des programmes
 - ▶ Sémantique translationnelle : Traduction vers un autre langage équipé d'une sémantique formelle
 - ▶ Sémantique dénotationnelle : Traduction vers un formalisme mathématique
- ▶ Validation des sémantique par étude équivalence entre formes

- ▶ Expressions :
- ▶ Partie fonctionnelle :
- ▶ Partie Impérative :

miniML

- ▶ Expressions :
- ▶ **Partie fonctionnelle :**
- ▶ Partie Impérative :

miniML

- ▶ Expressions :
- ▶ Partie fonctionnelle :
- ▶ **Partie Impérative :**

miniML : Expressions

► Règles de production

$$\begin{array}{lcl} Expr & \rightarrow & Ident \\ & | & Const \\ & | & Expr \text{ Binaire } Expr \\ & | & Unaire Expr \\ & | & (Expr) \end{array}$$
$$Const \rightarrow entier \mid boolean$$
$$Unaire \rightarrow - \mid !$$
$$\begin{array}{lcl} Binaire & \rightarrow & + \mid - \mid * \mid / \mid \% \mid \& \mid \mid \\ & | & == \mid != \mid < \mid <= \mid > \mid >= \end{array}$$