

S.A.004 LANGAGES FORMELS ET COMPILATEURS**1. Données de l'unité de cours/module**

Faculté	Ordinateurs, Informatique et Microélectronique				
Catégorie/département	Ingénierie Logicielle et Automatique				
Cycle des études	Licence, 1er cycle				
Programme des études	0612.3 Science des données				
Anne d'étude	Semestre	Type d'évaluation	Catégorie formative	Catégorie facultative	Credits ECTS
II (étude à plein temps);	4	E	D – Discipline du domaine professionnel	O - unité de cours obligatoire	5
III (étude à fréquence réduite);	5				

2. Durée totale estimée

Nombre total d'heures dans le programme	De quelles				
	Heures auditives		Travail individuel		
	Cours	Laboratoire/seminaire	Projet de l'année	L'étude du matériel théorique	L'étude du matériel théorique
jour: 150	30	15/30	-	30	45
fréquence réduite: 150	12	6/12	-	90	30

3. Prérequis pour accéder à l'unité de cours/module

Selon le programme	Mathématiques supérieures ; Mathématiques discrètes; Structures de données et algorithmes ; Analyse et conception d'algorithmes, programmation C.
Selon les compétences	Abstraction de situations réelles pour formuler des problèmes et des modèles mathématiques et connaissance d'un langage de programmation structuré

4. Conditions de réalisation du processus éducatif pour

Cours	Pour la présentation de la matière théorique en classe, un tableau noir ou électronique, un projecteur et un écran sont nécessaires. Les retards des étudiants, ainsi que les conversations téléphoniques pendant le cours, ne seront pas tolérés.
Laboratoire/seminaire	Les étudiants compléteront des rapports selon les conditions imposées par les consignes méthodiques. Le délai de remise des travaux de laboratoire est de deux semaines après leur achèvement. Pour la soumission tardive du travail, ceci est déduit de 1 pct./semaine de retard.

5. Compétences spécifiques accumulées

Compétences professionnelles	CI. Concernant les fondements scientifiques et techniques des technologies de l'information <ul style="list-style-type: none"> — Identification et définition de concepts, théories, méthodes des sciences fondamentales et appliquées comme support des langages formels ; — Expliquer les solutions d'ingénierie en utilisant des techniques, des concepts et des principes de la théorie des automates finis. — Résoudre des problèmes dans les domaines de la conception de compilateurs en appliquant notamment des techniques et des méthodes au sein des langages formels. — Le choix de critères et de méthodes pour analyser les avantages et les inconvénients des méthodes et procédures appliquées à la résolution de problèmes au sein des langages formels. — Modélisation de problèmes typiques des sciences appliquées en utilisant l'appareil mathématique au sein d'automates finis, automates à mémoire de pile.
Compétences transversales	CT1. Appliquer les principes, normes et valeurs de l'éthique professionnelle. CT2. Identifier, décrire et réaliser les activités organisées en équipe avec le développement des

	capacités de communication et de collaboration, mais aussi avec la prise en charge de différents rôles (d'exécution et de gestion). CT3. Faire preuve d'un esprit d'initiative et d'action pour mettre à jour ses connaissances en matière de culture professionnelle, économique et organisationnelle.
--	---

6. Objectifs de l'unité de cours/module

L'objectif general	Former les compétences nécessaires pour utiliser des langages formels pour résoudre des problèmes dans différents domaines
Objectifs spécifiques	Acquérir des compétences en analyse formelle du langage Utiliser des méthodes de résolution de problèmes dans le domaine des langages formels

7. Contenu de l'unité de cours/du module

Le thème des activités didactiques	Nombre d'heures	
	à plein temps	à fréquence réduite
Le sujet des conférences		
T1. Notions préliminaires : vocabulaire, chaîne, langue.	1	1
T2. Méthodes de description des langues. La méthode générative. Exemples.	1	
T3. Grammaires Chomsky et langages classifiables formels.	1	1
T4. Grammaires régulières et automates finis. Définitions, exemples.	1	1
T5. Automates finis déterministes (AFD) et non déterministes (AFND). Algorithme de réduction de l'AFND en AFD équivalent.	2	1
T6. Equivalence des grammaires régulières et des automates finis. Théorème GAF.	1	
T7. Le lemme du pompage et ses applications.	1	
T8. Expressions régulières.	1	
T9. Equivalence des automates finis et des grammaires régulières. Théorème AFG.	1	1
T10. Arbres de dérivation, théorème de branchement. Transformations équivalentes sur les grammaires indépendantes du contexte.	1	
T11. Suppression des symboles inutiles. Suppression des symboles improductifs.	1	
T12. Éliminer les ϵ -productions. Suppression des noms.	1	
T13. Forme normale de Chomsky.	2	2
T14. Éliminer la récursion gauche. Forme normale de Greibach. Le théorème UVWXY et ses applications.	3	2
T15. Automates à mémoire de pile : définitions, exemples.	3	1
T16. La matrice de priorité simple.	3	1
T17. Grammaire LL1.	3	1
T18. Algorithme d'analyse syntaxique universelle CYK.	3	
Total des conférences :	30	12
Le thème des activités didactiques	Nombre d'heures	
	à plein temps	à fréquence réduite
Le thème des séminaires		
S1. Exemples de problèmes de vocabulaire résolu, de création de chaînes et de description du langage	2	1
S2. Décrire et créer une grammaire formelle basée sur des vocabulaires. Exemples. Méthodes de représentation de AF. Conversion de G en AF.	2	1
S3. Transformation AFND en AFD en appliquant des algorithmes de transformation.	2	1
S4. Théorème GAF. Le lemme du pompage. Expressions régulières. Exemples.	2	1
S5. Théorème AFG. Le lemme de branchement.	2	
S6. Suppression des symboles inutiles et inaccessibles.	2	1
S7. GIC dans FNC.	2	1
S8. GIC avec récursion gauche. Conversion de GIC en FNG.	2	1
S10. Applications du théorème UVWXY. Exemples.	1	1
S11. Automates avec mémoire de pile.	3	1
S12. Application de matrices de priorité simples dans les langages de programmation.	4	1
S14. Construction de la table d'analyse. Pile d'analyse.	3	1

S15. Analyse syntaxique CYK.	3	1
Total des conférences :		
Total des travaux/séminaires en laboratoire :	30	12
Le thème des activités didactiques	Nombre d'heures	
	à plein temps	à fréquence réduite
Le thème des travaux de laboratoire		
L1. Grammaires formelles, classification Chomsky. Arbres de dérivation, génération de langage. Théorème GAF. Méthodes de représentation de AF.	4	2
L2. Equivalence des grammaires régulières, des expressions régulières et des automates finis. L'algorithme pour réduire l'AFND à un AFD équivalent. Le lemme du pompage.	4	2
L3. Forme normale Chomsky. Forme normale de Greibach.	4	1
L4. MPS. LL1.	3	1
Total des travaux de laboratoire/séminaires:	15	6

8. Références bibliographiques

Principales	<ol style="list-style-type: none"> 1. Luca-Dan Șerbănați. Limbaje de programare și compilatoare. Editura Academiei, București, 1987. 2. Toader Jucan. Limbaje formale și automate. MATRIX ROM, București, 1999, 163 pp. 3. Creangă, C. Reischer, D. Simovici : Introducere algebrică în informatică, vol II, Limbaje formale, Ed. Junimea, 1974 4. Gheorghe Grigoraș. Limbaje formale și tehnici de compilare. Universitatea "Al. I. Cuza", Iași, 1985 5. Adrian Atanasiu. Bazele informaticii.- Universitatea București, 1987 6. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. т.1,2. Москва, Мир, 1978. 7. Касьянов В.Н., Поттосин И.В. Методы построения трансляторов. – Новосибирск, Наука, 1986. 8. Хантер Р. Проектирование и конструирование компиляторов. Москва, Финансы и статистика, 1984 9. Льюис Ф., Розенкранц Д., Стирнз Р. Теоретические основы проектирования компиляторов. Москва, Мир, 1979. 10. Пратт Т. Языки программирования. Разработка и реализация. Москва, Мир, 1979. 11. Рейфорд-Смит В.Дж. Теория формальных языков. Вводный курс. Москва, Радио и связь, 1988
Suplimentaires	<ol style="list-style-type: none"> 1. Linz Peter, An introduction to formal languages and automata. Jones&Bartlett Learning, fifth edition, 2012. 2. Hopcroft John E., Motwani Rajeev, Ullman Jeffrey D. Introduction to Automata Theory, Languages and Computation. Addison – Wesley, second edition, 2001. 3. A.V.Aho, R.Sethi, J.D.Ullman. Compilers: Principles, Techniques, and Tools, Addison-Wesley, 1986. 4. Jean-Paul Tremblay, Paul G. Sorenson. The theory and practice of compiler writing. McGraw-Hill Book Company, 1985. 5. S. Marcus : Gramaticisi automate finite, Editura Academiei, Bucuresti, 1964. 6. J. E. Hopcroft, R. Motwani, J.D. Ullman. Introduction to Automata Theory, Languages, and Computation. Second Edition. Addison Wesley, 2001, 521 p. 7. J. Hopcroft, J. Ulman: Formal Languages and their Relations to Automata, Addison Wesley Publ. Comp., 1969. 8. L. Livovschi, N. Tandareanu, s.a. Bazele informaticii, Ed. did. ped. 1981. 9. Arto Salomaa. Formal Languages, Academic Press, 1973.

9. Évaluation

Périodique		Actuelle	Étude individuelle	Projet/thèse	Examen
EP 1	EP 2				
Étude à plein temps					
15%	15%	15%	15%		40%
Étude à fréquence réduite					
25%			25%		50%
Norme de performance minimale					

10. Critères d'évaluation

Activité	Composantes d'évaluation	Méthode d'évaluation, Critères d'évaluation	Poids dans la note finale de l'activité	Le poids dans l'évaluation de la discipline
Étude à plein temps				
Évaluation périodique I	Contenu théorique, thèmes 1 à 5	Test sur MOODLE	100%	15%
Évaluation périodique II	Contenu théorique, thèmes 6-14	Test sur MOODLE	100%	15%
Évaluation actuelle	Activité pratique	Discussions en séminaires	50%	15%
		Rapports	50%	
Étude individuelle	Résoudre des tâches selon les thèmes	Rapports	100%	15%
Evaluation finale	Contenu théorique et pratique	Épreuve écrite. Marquage selon barème	100%	40%
Étude à fréquence réduite				
Evaluation périodique I	Contenu théorique, thèmes 1 à 5	Test sur MOODLE	40%	25%
Evaluation périodique II	Contenu théorique, thèmes 6-14	Test sur MOODLE	40%	
Évaluation actuelle	Activité pratique	Rapports	20%	
Étude individuel	Résoudre des tâches selon les thèmes	Rapports	100%	25%
Evaluation finale	Contenu théorique et pratique	Épreuve écrite. Marquage selon barème	100%	50%