

Programmation

Jean Goubault-Larrecq

LSV/CNRS UMR 8643 & INRIA Futurs projet SECSI & ENS Cachan

61 avenue du président-Wilson, F-94235 Cachan Cedex

`goubault@lsv.ens-cachan.fr`

Phone: +33-1 47 40 75 68 Fax: +33-1 47 40 75 21

21 novembre 2003

Résumé

Ce document sert de notes de cours pour le premier quart du cours de programmation du magistère STIC, ENS Cachan, édition 2003–2004. Il s’agit de la version 1, qui date du 02 octobre 2003 (leçon 1), du 16 octobre 2003 (leçon 2), du 20 octobre 2003 (leçon 3), du 23 octobre 2003 (leçon 4).

Table des matières

1	Leçon 1	3
1.1	Une brève introduction aux langages de programmation	3
1.2	Quelques bases de théorie de l’ordre	10
1.2.1	Points fixes et boucles	10
1.2.2	Treillis complets et théorème de Tarski	13
1.2.3	Cpos, fonctions Scott-continues	15
2	Leçon 2	19
2.1	Langages impératifs, le langage C	19
2.1.1	Affectations	19
2.1.2	Tableaux, structures	21
2.1.3	Structures de contrôle	24
2.2	Langages fonctionnels, le cas de mini-Caml	27
3	Leçon 3	32
3.1	Architecture et assembleur	32
3.1.1	Mémoires	32
3.1.2	Le processeur	35

3.1.3	Modes d'adressage et formats d'instructions	39
3.1.4	Formalisation	42
3.2	La sémantique dénotationnelle (de mini-Caml)	46
3.2.1	Domaines, le modèle \mathbb{P}_ω de Plotkin	47
4	Leçon 4	52
4.1	Sémantique dénotationnelle de mini-Caml	52
4.1.1	Quel genre de cpo nous faut-il ?	52
4.1.2	Sémantique dénotationnelle des expressions	55
4.1.3	Sémantique dénotationnelle des programmes	58
5	Leçon 5	60
5.1	Sémantique opérationnelle grands pas de mini-Caml	60
5.1.1	Clôtures	60
5.1.2	Appel gauche-droite, par valeur, par nécessité, par référence	63
5.1.3	Autres règles	64
5.1.4	Sémantique opérationnelle des programmes	65
5.1.5	Arbres de dérivations, récurrence sur les dérivations	66
5.1.6	Règles dérivées	69
5.1.7	Correction de la sémantique concrète par rapport à l'abstraite	72
A	Guide de référence rapide de l'assembleur Pentium	76