

M2 informatique

IMC – Vérification logicielle

Durée : 90 minutes – sans document ni moyen de communication

Répondre dans les cadres prévus à cet effet.

Nom :

Signature :

Prénom(s) :

Exercice 1 (10 ●) Pour chaque technique, pour chaque item, entourez la réponse adéquate.

Interprétation abstraite

Domaine d'application : Ada, C, C#, Java, matériel/concurrence, tout

Niveau d'expertise : faible/moyen/élevé

Niveau d'automatisme : faible/moyen/élevé

Expressivité : faible/moyen/élevée

Un outil notable basé sur cette technique : Astrée, B, Coq, MS Code Contracts, Spin

Programmation par contrat

Domaine d'application : Ada, C, C#, Java, matériel/concurrence, tout

Niveau d'expertise : faible/moyen/élevé

Niveau d'automatisme : faible/moyen/élevé

Expressivité : faible/moyen/élevée

Un outil notable basé sur cette technique : Astrée, B, Coq, MS Code Contracts, Spin

Programmation par raffinement

Domaine d'application : Ada, C, C#, Java, matériel/concurrence, tout

Niveau d'expertise : faible/moyen/élevé

Niveau d'automatisme : faible/moyen/élevé

Expressivité : faible/moyen/élevée

Un outil notable basé sur cette technique : Astrée, B, Coq, MS Code Contracts, Spin

Model checking

Domaine d'application : Ada, C, C#, Java, matériel/concurrence, tout

Niveau d'expertise : faible/moyen/élevé

Niveau d'automatisme : faible/moyen/élevé

Expressivité : faible/moyen/élevée

Un outil notable basé sur cette technique : Astrée, B, Coq, MS Code Contracts, Spin

Programmation certifiée

Domaine d'application : Ada, C, C#, Java, matériel/concurrence, tout

Niveau d'expertise : faible/moyen/élevé

Niveau d'automatisme : faible/moyen/élevé

Expressivité : faible/moyen/élevée

Un outil notable basé sur cette technique : Astrée, B, Coq, MS Code Contracts, Spin

Nom :

Signature :

Prénom(s) :

Exercice 2 (10 •) On se donne une définition des expressions arithmétiques :

$$E ::= n \mid (* E E) \mid (+ E E)$$

où n parcourt Z , l'ensemble des entiers relatifs.

Par exemple, $(+ 1 (* 2 3))$ est une expression arithmétique correspondant à cette grammaire.

Donnez le pseudo-code d'un interpréteur I calculant la valeur d'une expression E quelconque.

Par exemple, $I[(+ 1 (* 2 3))] = 7$.

On considère l'assembleur A suivant, opérant sur une pile S d'entiers relatifs :

LOAD n – charge la constante n au sommet de la pile S ,

ADD – dépile le sommet de pile et l'élément suivant, calcule la somme puis l'empile sur S ,

MUL – dépile le sommet de pile et l'élément suivant, calcule le produit puis l'empile sur S .

Donnez le code assembleur correspondant à l'expression $(+ 1 (* 2 3))$.

Cet assembleur est exécuté par une machine à pile. La pile est gérée par les fonctions suivantes (on suppose qu'il n'y a pas de problème lié à cette gestion) :

empty() renvoie la pile vide,

push(n,S) renvoie la pile S où n est empilé,

pop(S) renvoie le couple (n,S') où n est le sommet de S et S' la pile S privée de son sommet.

Donnez le pseudo-code d'un interpréteur J de l'assembleur A .

Donnez le pseudo-code d'un compilateur C d'une expression arithmétique E quelconque.

Exprimez formellement que le compilateur C est *correct* relativement à l'interpréteur I et donnez l'idée générale de la preuve de cette propriété.