Université de Nice-Sophia Antipolis École Supérieure en Sciences Informatiques

2005–2006

Examen Mathématiques Discrètes
du 30 Novembre 2005

Durée : 1 heure

1	
2	
3	
4	
5	
6	

Aucun document autorisé.

Toutes vos réponses doivent être justifiées par une démonstration ou un contre-exemple. Si vous pensez que le texte d'une question est ambigu (voire erroné) faites une hypothèse raisonnable et écrivez la sur votre copie.

On supposera que si m est un mot, on dispose de la méthode $m.vide()$ qui retourne vrai si et seulement si le mot est vide, et que si m est un mot non vide , on dispose des méthodes $m.premiereLettre()$, $m.derniereLettre()$, $m.debutMot()$, $m.finMot()$, $m.longueur()$ qui retournent respectivement la premiere lettre de m, la dernière lettre de m, le mot m privé de sa dernière lettre, le mot m privé de sa première lettre, la longueur de m . Si vous le souhaitez vous pouvez racourcir l'écriture des noms de ces méthodes en $m.pL()$, $m.dL()$, $m.fM()$, $m.dM()$, $m.l()$.On dispose aussi de la méthode $m.concat(m')$ qui retourne le mot mm' et la methode l.toMot() qui retourne le mot composé de la seule lettre l.

1 Définitions inductives [4 points]

Soit P le sous ensemble des mots binaires, composés des mots admettant un nombre pair de "1" et de "0". Donnez et prouvez un schéma inductif pour P.

Soit E l'ensemble défini inductivement par

Base : $\epsilon \in E$

Règles :

$m \in E \rightarrow 00m \in E$
$m \in E \rightarrow m00 \in E$
$m \in E \rightarrow m11 \in E$
$m \in E \rightarrow 11m \in E$
$m \in E \rightarrow 1m1 \in E$
$m \in E \rightarrow 0m0 \in E$
$m \in E \rightarrow 01m01 \in E$
$m \in E \rightarrow 10m10 \in E$

Montrez que P est inclus dans E.

- Montrons par récurrence sur k que tout mot de P de longueur inférieure ou égale à k appartient à E

  Pour $k = 0$ c'est vrai. Supposons l'hypothèse vraie au rang k et soit m un mot de P de longueur $k+1$. Si m commence par 0 alors
  - s'il termine par 0, il s'écrit $0m'0$ et m' appartient à P. Par hypothèse de récurrence m' est donc dans E et par définition de E, on a donc m dans E.

- sinon il termine par 1 et s'écrit $0m'1$. m' ne peut pas être vide, sa première lettre est soit 0 soit 1.
  - si m' commence par 0, alors $m = 00m"1$ et $m"1$ appartient à P. Par hypothèse de récurrence $m"1$ est donc dans E et donc par définition de E on a m dans E.
  - sinon $m = 01m"1$, m" ne peut pas être vide, sa dernière lettre est soit 0 soit 1.
    - si m" termine par 0, alors $m = 01p01$ et p appartient à P. Par hypothèse de récurrence p est donc dans E et donc par définition de E on a m dans E.
    - sinon $m = 01p11$, et $01P$appartient à P. Par hypothèse de récurrence p est donc dans E et donc par définition de E on a m dans E.

Le cas où m commence par 1 se traite de manière analogue

.....

2 Désaccentuation [3 points]

Soit A l'alphabet usuel a 26 lettres, soit $A_{acc} = A \cup \{é,è,ê\}$. Donner une définition inductive de la fonction de A^*_{acc} dans A^* qui remplace toutes les occurences d'un e accentué par un e.

Donner aussi un code récursif.

- $enleveAccent(\epsilon) = \epsilon$
- $enleveAccent(a.m) = a.enleveAccent(m)$, *si* $a \notin$ *é,è,ê*
- $enleveAccent(a.m) = e.enleveAccent(m)$, *sinon*

```
public mot enleveAccent (mot m){
 if (m.vide()) {return m ;}
 if m.pL()== 'é' ||m.pL()='è' ||m.pL='ê' {return m.concat('e'.toMot())}
 else return m.pL().toMot().concat(m.fM().enleveAccent()) ;
}
```

.....

3 Palindrome [3 points]

Soit A un alphabet, définir inductivement la fonction

$fairePalindrome : A^* \rightarrow A^*$

telle que $fairePalindrome(m)$ est le mot palindrome dont la moitié gauche est m. Par exemple, $fairePalindrome(abc) = abccba$

Donner aussi un code récursif.

- $fairePalindrome(\epsilon) = \epsilon$
- $fairePalindrome(am) = amfairePalindrome(m)a$

```
public mot fairePalindrome (mot m){
 if (m.vide()) {return m ;}
 return m.pL().toMot().concat(fairePalindrome(m.fM())).concat(m.pL().toMot()) ;
}
```

.....

4 Prefixe [3 points]

Soit A un alphabet, définir inductivement la fonction
$Prefixe : A^*xA^* \rightarrow \{faux,vrai\}$
telle que $Prefixe(m,m')$ est vraie si et seulement si m est un prefixe de m'.
Donner aussi un code récursif.

- $Prefixe(\epsilon,m') = vrai$
- *Si* $m \neq \epsilon$, $Prefixe(m,\epsilon) = faux$
- $Prefixe(am,bm') = ((a = b)\ et\ Prefixe(m,m'))$

```
public boolean prefixe (mot m, mot m'){
if (m.vide()) {return true ;}
if (m'.vide()) {return false ;}
return (m.pL()==m'.pL())&& prefixe(m.fM(),m'fM());
}
```

.....

5 Facteur [4 points]

Soit A un alphabet, définir inductivement la fonction
$Facteur : A^*xA^* \rightarrow \{faux,vrai\}$
telle que $Facteur(m,m')$ est vraie si et seulement si m est un facteur de m', c'est à dire s'il existe un prefixe p de m' et un suffixe s de m' tels que $m' = pms$. Vous pouvez utiliser toutes les fonctions définies aux questions précédentes.
Donner aussi un code récursif.

- $Facteur(\epsilon,m') = vrai$
- *Si* $m \neq \epsilon$, $Facteur(m,\epsilon) = faux$
- $Facteur(am,bm') = ((a = b)\ et\ Prefixe(m,m'))\ ou\ Facteur(am,m')$

```
public boolean facteur (mot m, mot m'){
if (m.vide()) {return true ;}
if (m'.vide()) {return false ;}
return (m.pL()==m'.pL())&& prefixe(m.fM(),m'fM())||facteur(m,m'.fM());
}
```

.....

6 SousMot [3 points]

Soit A un alphabet, définir inductivement la fonction
$SousMot : A^*xA^* \rightarrow \{faux,vrai\}$
telle que $SousMot(m,m')$ est vraie si et seulement si m peut être obtenu à partir de m' en supprimant certaines lettres de m' pas necessairement consécutives. Ainsi,SousMot(exemple, elxeerrrmddprsleee)=vrai alors que SousMot (exemple, ekxksmple)= faux
Donner aussi un code récursif.

- $sousMot(\epsilon,m') = vrai$
- *Si* $m \neq \epsilon$, $sousMot(m,\epsilon) = faux$
- $sousMot(am,bm') = ((a = b)\ et\ sousMot(m,m'))\ ou\ sousMot(am,m')$

```
public boolean sousMot (mot m, mot m'){
if (m.vide()) {return true ;}
if (m'.vide()) {return false ;}
return (m.pL()==m'.pL())&& sousMot(m.fM(),m'fM())||sousMot(m,m'.fM());
}
```

.....