

Compresseur de Huffman

Contents

1 Analyse

1.1 Types de données abstraits

Nom: ArbreDeHuffman

Paramètre: Element

Utilise: Naturel, Booleen

Opérations: arbreDeHuffman: Element \times **Naturel** \rightarrow ArbreDeHuffman

combiner: ArbreDeHuffman \times ArbreDeHuffman \rightarrow ArbreDeHuffman

obtenirPondération: ArbreDeHuffman \rightarrow **Naturel**

obtenirElement: ArbreDeHuffman \rightarrow Element

estUneFeuille: ArbreDeHuffman \rightarrow **Booleen**

obtenirArbreGauche: ArbreDeHuffman \rightarrow ArbreDeHuffman

obtenirArbreDroit: ArbreDeHuffman \rightarrow ArbreDeHuffman

Axiomes:

- obtenirElement (arbreDeHuffman(élément,pondération)) = élément
- obtenirPondération (arbreDeHuffman(élément,pondération)) = pondération
- obtenirPondération (combiner(arbreGauche,arbreDroit)) = obtenirPondération(arbreGauche)+obtenirPondération(arbreDroit)
- estUneFeuille (arbreDeHuffman(élément,pondération)) = Vrai
- estUneFeuille (combiner(arbreGauche,arbreDroit)) = Faux
- obtenirArbreGauche (combiner(arbreGauche,arbreDroit)) = arbreGauche

- obtenirArbreDroit (combiner(arbreGauche,arbreDroit)) = arbreDroit

Préconditions: obtenirElement(arbre)estUneFeuille(arbre)

non estUneFeuille(arbre)

obtenirArbreGauche(arbre):obtenirArbreDroit(arbre): non estUneFeuille(arbre)

Nom: Octet

Utilise: **Booleen**, Bit, \natural

Opérations: octet: $\text{Bit} \times \text{Bit} \times \text{Bit} \times \text{Bit} \times \text{Bit} \times \text{Bit} \times \text{Bit} \times \text{Bit} \rightarrow \text{Octet}$

estEgal: $\text{Octet} \times \text{Octet} \rightarrow \mathbf{Booleen}$

obtenirBit: $\text{Octet} \times 1..8 \rightarrow \text{Bit}$

Sémantiques: octet: Crée un octet avec les 8 bits spécifiés.

estEgal: Vérifie si les deux octets sont égaux.

obtenirBit: Obtient le bit à la position spécifiée dans l'octet (position de 0 à 7).

Axiomes:

- obtenirBit (00000001, 1) = 1
- estEgal (octet1,octet2) = (obtenirBit (octet1, 1..8) = obtenirBit (octet2, 1..8))