Khawarizm

D.E ZEGOUR

École Supérieure d'Informatique

ESI

Khawarizm

Contenu de la présentation

Généralités

Utilisation

Langage Z

Documentation

Khawarizm

G E N E R A L I T E S

KHAWARIZM est un environnement de programmation

But : Apprendre et approfondir les principales structures de données et de fichiers.

KHAWARIZM

offre la possibilité d'écrire des algorithmes dans un langage algorithmique (langage Z), de les indenter, de les dérouler ou les simuler et de les convertir automatiquement vers les langages de programmation PASCAL et C.

Khawarizm

G E N E R A L I T E S

Outils :

- Un éditeur

- Un indenteur

- Un interpréteur

- Un simulateur

- Un traducteur vers Pascal ou C.

Khawarizm

G E N E R A L I T E S

Utilisation du langage Z

Doté de machines abstraites simulant les principales structures de données et de fichiers.

Objectifs :

Expérimentation sur les vecteurs, les structures, les listes linéaires chaînées, les listes bilatérales, les files d'attente, les piles, les arbres de recherche binaire, les arbres de recherche m-aire, les fichiers.

Manipulation de structures de données complexes telles que liste de files d'attente, liste de piles, arbre de listes, liste de piles de vecteurs, etc...

Khawarizm

UTILISATION

1.{Familiarisation avec un langage algorithmique arbitraire : langage Z}

Apprendre le langage algorithmique utilisé.
Utiliser l'aide fourni

2. {Edition de l'algorithme}
Ecrire un algorithme ou corriger un algorithme existant.

3. {Vérification syntaxique}

Lancer le module arranger.
Répéter tant qu'il ya des erreurs : Corriger les erreurs et Relancer le module Indenter

A ce stade, votre algorithme est bien écrit et il a été indenté pour vous.

Khawarizm

U T I L I S A T I O N

4. {Exécution}

Lancer l'exécution de votre algorithme

Les fenêtres montrent alors
- les données lues par votre algorithme (Bouton Données)
- les écritures émises par votre algorithme (Bouton Résultats)

Ou bien votre algorithme donne les résultats attendus ou pas. Dans ce dernier cas, lancer la simulation pour essayer de déterminer les erreurs de logique.

Khawarizm

UTILISATION

5. {Simulation}

Lancer la simulation de votre algorithme (exécution avec trace)

Les fenêtres montrent alors les données lues par votre algorithme (Bouton Données), les écritures émises par votre algorithme (Bouton Résultats) et tous les changements effectués sur les objets utilisés (Bouton Simulation)

Vous avez ainsi la trace complète de votre algorithme que vous pouvez imprimer et l'analyser pour détecter les erreurs.

Si vous désirez dérouler pas à pas votre algorithme, demander une trace.

Khawarizm

UTILISATION

6. {Trace}

Redemander la simulation avec trace
Vous pouvez alors suivre pas à pas l'évolution de votre algorithme, sortir de la boucle courante ou même du module courant.

Afin d'éviter d'avoir une trace complète qui peut être longue il est possible de limiter la longueur des boucles utilisées dans votre algorithme.

Vous pouvez changer les modes de simulation (voir "Options" du menu).

Khawarizm

U T I L I S A T I O N

7. {Passage vers un PASCAL ou C}

Une fois que votre algorithme "tourne", il est possible de le traduire automatiquement en PASCAL ou en C.
Deux fenêtres montrent votre algorithme et l'autre le résultat de la traduction.

Vous pouvez consulter l'aide concernant le passage vers PASCAL ou C.
Dans cette aide, vous trouverez

- les équivalents Z vers PASCAL et Z vers C.
- toutes les implémentations des machines Z.

La tâche de Khawarizm s'arrête à ce niveau là .

Khawarizm

U T I L I S A T I O N

8. {Programmation PASCAL ou C}

Utiliser le compilateur PASCAL ou C, pour finaliser définitivement votre programme.

Vous pouvez rajouter tous les modules de saisie des données et de restitution des résultats.

Khawarizm

LANGAGE Z

Généralités

> Un Z-algorithme est un ensemble de modules parallèles dont le premier est principal et les autres sont soient des actions composées (ACTION) soient des fonctions de type quelconque (FONCTION).

> Le langage Z admet les modules récursifs.

> Les objets globaux sont déclarés dans le module principal.

> La communication entre les modules se fait via les paramètres et les variables globales.

Khawarizm

LANGAGE Z

Généralités

> Le langage permet
- tout type de paramètres : scalaires, structures,liste , file, vecteurs, pile, arbres et même les type complexes.
- l'allocation dynamique de tableaux et de structures
- l'affectation globale de tout type

> Quatre types standard sont autorisés : ENTIER, BOOLEAN, CAR, CHAINE.

> Certaines fonctions usuelles sont prédéfinies : MOD, MAX et MIN.

> Le langage est l'ensemble des algorithmes abstraits, écrits à base de modèles (machinesabstraites).

Khawarizm

LANGAGE Z

Généralités

> On définit ainsi des machines abstraites sur les objets composés (structures), les vecteurs de dimension quelconque, les piles, les files d'attentes,les arbres de recherche binaire, les arbres de recherche m-aire, les listes mono-directionnelles, les listes bidirectionnelles.

> Les vecteurs et les structures peuvent être statiques ou dynamiques.

> Le langage permet les types composés du genre PILE DE FILES DE LISTES DE ....dont la dernière citée est de type scalaires ou structure.

> Le langage est doté des opérations de haut niveau permettant de construire des listes, des arbres, des files d'attente, etc. à partir d'un ensemble de valeurs (expressions) ou de structures.

Khawarizm

LANGAGE Z

Structure d'un Z-algorithme

SOIENT
{ Objets locaux et globaux }
{ Annonce des modules }
DEBUT
{ Instructions }
FIN
Module 1
Module 2
....
Module n
Chaque module peut être soit une fonction soit une action.

Khawarizm

LANGAGE Z

Définition d'une action

```
ACTION Nom (P1, P2, ..., Pn)
        { Objets locaux et paramètres }
DEBUT
        { Instructions }
FIN
```

Définition d'une fonction

```
FONCTION Nom (P1, P2, ...,Pn) : Type
        { Objets locaux et paramètres }
DEBUT
        { Instructions }
FIN
```

Khawarizm

LANGAGE Z

Exemple d'un Z-algorithme

```
SOIENT
   L1, L2 DES LISTES;
   Rech, Tous : FONCTION(BOOLEEN);
DEBUT
   CREER_LISTE(L1, [2, 5, 9, 8, 3, 6 ]);
   CREER_LISTE(L2, [12, 5, 19, 8, 3, 6, 2,9]);
   ECRIRE( Tous(L1, L2) )
FIN
```

Recherche séquentielle dans une liste

```
FONCTION Rech ( L, Val ) : BOOLEEN
SOIENT
    L UNE LISTE; Val UN ENTIER;
DEBUT
 SI L = NIL : Rech := FAUX
 SINON
   SI VALEUR(L) = Val
      Rech := VRAI
   SINON
      Rech := Rech(SUIVANT(L), Val )
   FSI
  FSI
FIN
```

Est-ce que tous les éléments dans L1 sont dans L2 ?

```
FONCTION Tous ( L1, L2 ) : BOOLEEN
SOIENT
   L1, L2 DES LISTES;
DEBUT
 SI L1 = NIL : Tous := VRAI
 SINON
   SI NON Rech(L2, VALEUR(L1) )
      Tous := FAUX
   SINON
      Tous := Tous(SUIVANT(L1), L2)
   FSI
 FSI
FIN
```

Khawarizm

LANGAGE Z

Objets

Les objets peuvent être des scalaires : ENTIER, BOOLEEN, CAR, CHAINE.

Les objets peuvent être des machines abstraites :
FILE(Files d'attente) PILE,
STRUCTURE, VECTEUR,
LISTE, LISTEBI(Listes bidirectionnelles),
ARB(Arbres de recherche binaire), ARM (Arbres de recherche m-aire),
FICHIER.

Les objets peuvent être complexes, c'est-à-dire une combinaison de machines abstraites.

Khawarizm

LANGAGE Z

Objets

Exemples
Scalaires

A, B, C DES BOOLEENS ; Ch UNE CHAINE ;

Machines abstraites :

A UN ARB;
L1, L2 DES LISTES ;
A UNE STRUCTURE(CHAINE, ENTIER) ;
F UN FICHIER DE (ENTIER, VECTEUR(10)) ENTETE ENTIER BUFFER BUF1, BUF2

Objets complexes

V1 UN VECTEUR(10, 60) DE (CAR, ENTIER) ;
Y UNE LISTE DE PILES DE VECTEUR(10)

Khawarizm

LANGAGE Z

Expressions

Comme dans les tous langages de programmation.

Exemples :

(B+C) / F
NON Trouv
(X # 5) ET NON Trouv
F(x) <> 5
(A ET B) OU C

Khawarizm

LANGAGE Z

Instructions

V désigne une variable, E une expression et Idf un nom de module.
[ ] désigne une partie facultative, { } un ensemble.

Affectation :	V := E
Lecture :	LIRE(V1, V2, ...)
Ecriture :	ECRIRE(E1, E2, ...)
Appel :	APPEL Idf [(E1, E2, ...)]
Conditionnelle :	SI E [:] { Instructions } [SINON { Instructions }] FSI
Répétitive :	TQ E [:] (Forme 1) { Instructions } FTQ
Répétitive :	POUR V := E1, E2,E3 (Forme 2) E3 désigne le pas. { Instructions } FPOUR

Khawarizm

LANGAGE Z

Machines abstraites

Vecteurs : ELEMENT, AFF_ELEMENT, ALLOC_TAB, LIBER_TAB (Si tableau dynamique)

Structures : STRUCT, AFF_STRUCT, ALLOC_STRUCT, LIBER_STRUCT(si structure dynamique)

Listes : ALLOUER, LIBERER, VALEUR, SUIVANT, AFF_ADR, AFF_VAL

Listes bidirectionnelles : ALLOUER,LIBERER, VALEUR, SUIVANT, AFF_VAL,PRECEDENT, AFF_ADRD, AFF_ADRG

Piles : CREERPILE, EMPILER, DEPILER,PILEVIDE

Files : CREERFILE, ENFILER, DEFILER, FILEVIDE

Khawarizm

LANGAGE Z

Machines abstraites

Arbres de recherche binaire : CREERNOEUD,FG, FD, PERE, LIBERERNOEUD, AFF_FG, AFF_FD, AFF_PERE, INFO, AFF_INFO

Arbres de recherche m-aire : CREERNOEUD,FILS, LIBERERNOEUD, AFF_FILS, INFOR, AFF_INFO, DEGRE, AFF_DEGRE, PERE, AFF_PERE

Fichiers : OUVRIR, FERMER, ENTETE, AFF_ENTETE,LIRESEQ, LIREDIR, ECRIRESEQ, ECRIREDIR, RAJOUTER, ALLOC-BLOC,FINFICH

Khawarizm

LANGAGE Z

Macro-Opérations

CREER_ARB, CREER_LISTE,
CREER_LISTEBI,CREER_ARM,
CREER_PILE,CREER_FILE,
INIT_VECTEUR(ou INIT_TABLEAU)
INIT_STRUCT

Exemple
CREER-LISTE (L, [12, 23, 67, I, I+J])
Crée la liste linéaire chaînée L avec les valeurs entre crochets dans l'ordre indiqué.

Khawarizm

DOCUMENTATION

Téléchargement du Logiciel:

Khawarizm II+ (Win 32) : http://zegour.esi.dz/Ftp/Khawarism2_plus-32.rar

Khawarizm II+ (Win 64) : http://zegour.esi.dz/Ftp/Khawarism2_plus-64.zip

Documentation:

Documentation complète (PDF) : http://zegour.esi.dz/Cours/Khawarizm_plus.pdf

Dépliant sur le langage Z (PDF) : http://zegour.esi.dz/Developpement/Seriez/Depliant%20z2.pdf

Khawarizm

DOCUMENTATION

Documentation Intégrée au logiciel

Introduction (TXT)

Présentation (TXT)

Utilisation (TXT)

Exposé (HTML)

Documentation complète sur Khawarism (HTML)

Description du langage Z (HTML)