

Pràctica 1

Introducció

El llenguatge de programació escollit per a les pràctiques és Mathematica [Wolfram,91]. En les implementacions que es van a dur a terme, cal manejar objectes abstractes com són els autòmats i les gramàtiques, sobre els quals es realitzaran una sèrie de transformacions. Tant uns com les altres poden ser concebuts com a estructures més complexes realitzades sobre la base d'altres més senzilles. L'interès de les pràctiques no estriba a aconseguir implementacions eficients, sinó a utilitzar un llenguatge que facilite la construcció d'aquests objectes abstractes i el maneig dels mateixos. Tenint en compte tots aquests aspectes, vegem a continuació algunes de les característiques de Mathematica que ho fan adequat per al seu ús en aquestes pràctiques.

Mathematica és un paquet de desenvolupament per a aplicacions de tipus general en les quals els aspectes de desenvolupament matemàtic, algebraic, numèric, simbòlic i gràfic juguen un paper preponderant. Mathematica està dissenyat per a tractar tant càlculs matemàtics, així com aquells càlculs numèrics tradicionalment duts a terme en FORTRAN. La majoria de funcions especials de la física matemàtica i funcions matricials, ja estan construïdes en Mathematica. A més, té una extensa capacitat gràfica per a la visualització dels resultats dels càlculs.

L'aplicacions de Mathematica engloben pràcticament totes les àrees d'investigació i desenvolupament tant en investigacions científiques, d'enginyeria, econòmiques, arquitectura, etc. Dins d'aquestes mateixes àrees es poden utilitzar tant com a eina de treball com a eina docent per a aquelles matèries que comporten un alta càrrega de desenvolupament matemàtic.

Mathematica com a llenguatge de programació es diferencia del FORTRAN o C o PASCAL, d'una banda per la seua habilitat en el tractament d'expressions matemàtiques, nombres, expressions simbòliques, etc., i per una altra en què és un llenguatge interpretat. Com a conseqüència de ser interpretat, un càlcul tarda més temps a executar-se que en un llenguatge compilat, no obstant açò, l'escriure un programa en Mathematica requereix una fracció del temps necessari per a escriure el mateix programa en C, i el que és més important, permet concentrar els esforços en els detalls conceptuals i no en els de implementació.

El sistema Mathematica és interactiu, la qual cosa significa que no cal compilar programes. En lloc d'açò, els càlculs es fan típicament executant una línia cada vegada, i per tant, els resultats intermedis poden veure's immediatament. Mathematica és potent, pot usar-se per a realitzar grans càlculs interactius, evitant els típics errors algebraics, i els resultats es poden comprovar a l'instant. El llenguatge Mathematica és concís, càlculs complicats poden ser escrits en poques línies (encara que cal controlar les limitacions de memòria de l'ordinador usat). És també flexible, poden establir-se connexions amb altres programes ja existents, etc.. Mathematica està sent desenvolupat per Wolfram Research, Inc. Actualment està disponible en pràcticament totes les plataformes d'ordinadors. Els codis es poden transportar en fitxers ASCII o en fitxers Mathematica Notebook, si es treballa amb versions compatibles.

En aquesta primera pràctica es pretén introduir alguns conceptes elementals del llenguatge associat a l'aplicació Mathematica, així com el desenvolupament d'aplicacions relacionades amb paraules i llenguatges emprant aquesta eina.

Abans de començar

La finestra de Mathematica és totalment típica. No obstant, té algunes peculiaritats. Les més importants són:

- Els noms són distints si s'escriuen en majúscules o en minúscules.
- Per executar una instrucció cal teclejar **shift+intro**.
- Les instruccions es van numerant de manera automàtica i es disposa en **cel·les**. Cada instrucció du associada una cel·la d'entrada (In[.]) i una altra d'eixida (Out[.]). La seua es pot veure en línies verticals a la dreta.
- Mentre dura l'execució d'una instrucció la línia de la cel·la d'entrada es manté doble. Si s'entra en un bucle indefinit es pot interrompre l'execució teclejant **Alt+.** i si no funciona s'executa **Quit Kernel** del menú **Evaluation**. La primera de les opcions només interromp l'execució en curs, mentre la segona anul·la totes les instruccions de la sessió actual.
- En tot moment es pot demanar ajuda interactiva sobre la sintaxi d'un comando escrivint el signe **?** seguit del nom del comando (o part del mateix, acabat amb *****).
- Mathematica sagna el text de manera automàtica quan es realitzen els programes. Si el text està mal sagnat acostuma a ser per una sintaxi incorrecta.

Llistes en Mathematica

Una de les estructures de dades més importants en Mathematica (i, sense dubte la que nosaltres usarem més) és la llista. Es poden usar llistes tant per a les paraules dels llenguatges amb els que treballem, com per a les transicions dels autòmats que accepten aquests llenguatges.

Les llistes es representen entre claus i els seus elements estan separats per comes, per exemple: `list1 = {a, b, {c, d}}` assigna a la variable **list1** la llista amb primer element **a**, segon element **b** i tercer la llista `{c, d}`. L'element *i*-èssim d'una llista es referencia com `nom[[i]]`. Així, `list1[[3]]` és `{c, d}`. Observeu que els elements de la llista no cal que siguin homogenis.

Operacions amb llistes

Per estalviar temps començarem contruint llistes de manera automàtica. Amb la instrucció `l=Table[Random[Integer, {1, 9}], {i, 20}]` (**shift+intro**) assignem a la variable **l** una llista de 20 nombres enters aleatoris en el rang de 1 a 9 (naturalment es pot variar tant la quantitat com el rang).

Es demana construir dues llistes **l1** e **l2**, i provar els comandos següents:

Important: els comandos que es descriuen a continuació, excepte els que s'indica explícitament, no actualitzen les llistes. Per poder disposar del resultat cal assignar-lo a una variable.

- `Length[l1]`: Torna la longitud de la llista.
- `Join [l1, l2]`: Concatena dues llistes.
- `Union[l1, l2]`: Torna una llista amb els elements que es troben en l1 o l2 i els ordena.
- `Intersection[l1, l2]`: Torna una llista amb els elements comuns a l1 i l2
- `Complement[l1, l2]`: Torna una llista amb els elements de l1 que no estan en l2.
- `Sort[l1]`: Torna l1 ordenada de menor a major (no actualitza l1).
- `Reverse[l1]`: Torna el revers de l1.
- `First[l1]`: Torna el primer element de la llista.
- `Rest[l1]`: Llista l1 sense el primer element.
- `Take[l1, n]`: Torna els primers n elements de la llista.
- `Take[l1, {n1,n2}]`: Torna la subllista de l1 que conté els elements de la posició n1 a la n2.
- `Append[l1, x]`: Afegeix l'element x al final.
- `Prepend[l1, x]`: Afegeix l'element x al començament.
- `AppendTo[l1, x]`, `PrependTo[l1, x]`: Idèntiques a les anteriors però actualitzen la llista.
- `MemberQ[l1,x]`: Torna True si x pertany a l1 i False en cas contrari.

La funció Cases

Per la seua importància, destaquem la funció següent:

`Cases[llista, patró]`: Torna una llista amb els elements de llista que concorden amb patró. El patró pot contenir el símbol `_` (subratllat), que se substitueix per qualsevol símbol.

Exemple `lista={{a,a},{b,a},{b,b},{a,b}}`

`Cases[lista,{a,_}]` torna `{{a,a},{a,b}}`.

`Cases[lista,{c,_}]` torna `{}` (llista buida).

Operadors lògics i relacionals

Són operadors que donen com a resultat True o False.

Són els següents:

- Negació `!`
- Conjunció `&&`
- Disjunció `||`

- Igualtat ==
- No igualtat !=
- >, <, >=, <=.

Programació

Mathematica du incorporat un llenguatge de programació propi que permet incorporar funcions per realitzar tasques específiques al mateix nivell que les funcions predefinides.

Al ser un intèrpret, el mode de treball pot ser totalment interactiu; així, si executem l'expressió `For[i = 1, i < 10, i++, Print[i]]` s'escriuen en pantalla els dígitos de l'1 al 9 (s'ha introduït la sentència `For`, la sintaxi de la qual és totalment idèntica a la que té en C).

Mòduls

El concepte de mòdul és totalment similar al de procedure en altres llenguatges de programació. El seu esquema genèric és:

```
nom[paràmetres] : = Module [{variables locals separades per comes},
  Accions (separades per );
  Return[nvar] (si el mòdul torna un valor)
]
```

El mòdul, les variables i l'execució del mòdul amb les variables convé que estiguen en cel·les diferents.

Exemple

Mòdul que pren com entrada un nombre positiu n i torna la suma dels n primers nombres enters.

```
suma[n_Integer]:=Module[{i,suma1},
  suma1=0;
  For[i =1,i <= n,i++,suma1 = suma1 + i];
  Return[suma1];
]
```

Per a executar-lo s'escriuria per exemple `suma[3]` (que donarà com a resultat 6).

Estructures condicionals i de repetició

- **Condiciona**l: `If[condició, sentències-veritat, sentències-fals]`
- **Iteració fixa**: `For[començament, test, increment, sentències];` S'avalua *començament* i s'executen *sentències* i *increment* fins que *test* falla.
- **Iteració While**: `While[condició, sentències];` S'executa *sentències* mentre *condició* és certa.

En el que segueix, una paraula es representa como una llista de símbols sobre un determinat alfabet. Així la paraula $x=abbaca$, es representarà com $\{a,b,b,a,c,a\}$; la paraula buida es representa com la llista de longitud 0, és a dir, $\{\}$. Un llenguatge finit és un conjunt finit de paraules. Per tant, un llenguatge es representa com una llista, els elements de la qual (paraules) són llistes. Per exemple, el llenguatge $L=\{abba, bb\}$, es representarà com $\{\{a,b,b,a\},\{b,b\}\}$; el llenguatge buit es representa com $\{\}$ el llenguatge que només té la paraula buida es representa com $\{\{\}\}$.

Exercicis

Exercici 1

Es demana escriure un mòdul Mathematica que amb entrada una cadena x i un enter positiu n , torne una llista amb els elements que apareguen almenys n vegades en x .

Exercici 2

Simulació de la funció `Length`

L'expressió `Length[l]`, on l és una llista, torna la longitud de l . Es demana, sense usar la funció `Length`, un fragment de programa iteratiu que simule dita funció.

Ajudes:

- La llista buida es representa mitjançant `{}`.
- `l = Rest[l]` assigna a la variable `l` (que és una llista) la llista `l` sense el primer element.
- Recordem la estructura repetitiva `While[condició,sentències]`. Cal anar amb compte de no entrar en un bucle sense fi. En cas d'entrar, `Alt + .` aborta l'execució.

Exercici 3

Es demana escriure un mòdul Mathematica que amb entrada una cadena x , torne una cadena que continga primer els elements en posició parella de x seguits dels elements de x en posició imparella. Exemple: donada la cadena `{a,b,c,d,e}`, el mòdul tornarà `{b,d,a,c,e}`.

Exercici 4

Es demana escriure un mòdul Mathematica que, donats una cadena i dos enters i i j , torne la cadena que resulta de considerar els elements entre les posicions i i j en ordre invers. Exemple: donada la cadena `{a,b,c,d,e,f,g,h}` i els enters 3 i 6, el mòdul tornarà la cadena `{a,b,f,e,d,c,g,h}`.

Exercici 5

Es demana escriure un mòdul Mathematica que amb entrada una paraula x i un enter positiu n , obtinga x^n (concatenació de la paraula x amb sí mateix n voltes).

Exercici 6

Es demana escriure un mòdul Mathematica que torne el conjunt de segments d'una paraula x .

Exercici 7

Es demana escriure un mòdul Mathematica que torne el producte de dos llenguatges finits donats.

Exercici 8

Es demana escriure un mòdul Mathematica que torne el quocient per la dreta d'un llenguatge finit per una cadena.

Exercici 9

Es demana escriure un mòdul Mathematica que, amb entrada un llenguatge finit L , i un enter $n > 0$ calcule L^n .

Exercici 10

Es demana escriure un mòdul Mathematica que, donada una cadena x , torne el símbol a tal que existeix en x un segment a^n en x amb n màxim. Exemple: donada la cadena $\{a,b,b,a,c,c,c,a,b,b\}$ el mòdul torna c .

Exercici 11

Es demana escriure un mòdul Mathematica que, donades dues paraules x_n i x_m de longituds n i m , amb $n \leq m$, torne **False** si x_n no és un segment de x_m , o bé, cas que sí ho siga, la posició del primer símbol de x_n en x_m .

Exercici 12

Es demana escriure un mòdul Mathematica que, donats un conjut de paraules M i una paraula x , torne el conjunt de posicions de x on es pot trobar una paraula de M com a segment.