Introduzione a GAP

Ilaria Colazzo - Prof. Francesco Catino

Università del Salento

16 Maggio 2019

SOMMARIO

1 Istallazione

2 Iniziamo ad usare GAP

3 Comandi di base

4 I gruppi di matrici

5 Gruppi Liberi

Windows

Scaricare il file di istallazione **.exe** dalla pagina **GAP Download page** (https://www.gap-system.org/Releases/).
Completato il download eseguire il file facendo doppio clic.
È importante assicurarsi che nel percorso di istallazione tutte le cartelle non contengano spazi. Di default il programma verrà istallato in **C:\gap4r8p4**

OS X

Dalla pagina **GAP Download page** (https://www.gap-system.org/Releases/) scaricare uno degli archivi, decomprimerlo e lanciarlo usando il comando **./configure; make** nella cartella decompressa.
Poi è necessrio cambiare la sottocartella **pkg** chiamando **../bin/BuildPackages.sh** e richiamando lo script che istallerà molti dei pacchetti necessari che richiedono di essere compilati.
In alternativa, è possibile istallare GAP usando Homebrew. Dopo aver istallato Homebrew, si apra il terminale e si richiami brew install **homebrew/science/gap –with-packages**. Se Homebrew è già istallato sul vostro Mac, assicurarsi che sia aggionato prima di procedere.

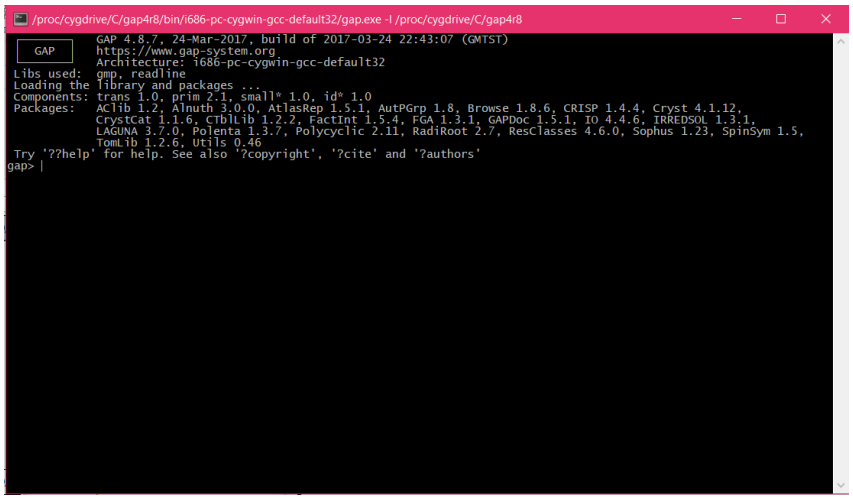
Linux

Dalla pagina **GAP Download page** (`https://www.gap-system.org/Releases/`) scaricare uno degli archivi, decomprimerlo e lanciarlo usando il comando **./configure; make** nella cartella decompressa.
Poi è necessrio cambiare la sottocartella **pkg** chiamando **../bin/BuildPackages.sh** e richiamando lo script che istallerà molti dei pacchetti necessari che richiedono di essere compilati.

Lavorare con GAP da linea di comando (I)

Lanciare GAP, il come dipende dal sistema operativo. Se istallato correttamente GAP inizia con un banner che indica quale versione di GAP e installata e quali componenti sono stati caricati. Dopo è presente la linea di comando `gap>`.
Per uscire da GAP digitare `quit;` nel prompt.

Importante!! Tutte le linee di comando in GAP devono terminare con un punto e virgola!



Lavorare con GAP da linea di comando

Prima di continuare usiamo il comando `ColorPrompt(true);` per distinguere con due colori differenti le richieste di GAP e gli input utente. Possiamo usare GAP come una calcolatrice inseriamo ad esempio

```
( 1 + 2^32 ) \ ( 1 - 2*3*107 );
```

Per tenere traccia di tutto quello che si fa in una sessione di GAP, possiamo creare un file in cui salvare in sequenza gli input e gli output. Per fare questo useremo la funzione `LogTo`.

Alcuni comandi utili (I)

1. Il comando per uscire da GAP è `quit;`
2. Per salvare il proprio lavoro su un file digitare `LogTo(‘‘nome file’’);`
3. Per interrompere il salvataggio digitare `LogTo();`
4. Digitare `<ctrl>+p` per riottenere il comando precedente
5. Se un errore determina l'entrata in un loop di GAP (e fa comparire il prompt `brk>` invece del classico `gap>`) è possibile uscire dal ciclo digitando `quit;` oppure `<ctrl>+d`
6. Digitare `?` seguito dal nome di un argomento per accedere al menù di aiuto.

Alcuni comandi utili (II)

Il comando `?` può essere molto utile nel caso in cui non si ricordi l'esatta sintassi di un comando.

Importante!! GAP distingue tra lettere maiuscole e minuscole. (per esempio le due istruzioni `LogTo` e `Logto` sono diverse).

Importante!! Ricordarsi sempre di mettere un punto e virgola alla fine di ogni istruzione.

La funzione LogTo

Digitando

```
LogTo(‘‘gap01.log’’);
```

verrà creato un file nella cartella predefinita.
Creiamo sulla scrivania una cartella con il nome `LogGap`, salviamo in una variabile chiamata `scirvania` il percorso per arrivare sulla scrivania e in una variabile `file` il percorso e il nome su cui salvare il log nel seguente modo:

```
> srcivania:=DirectoryDesktop();
> file:=Filename(scrivania, ‘‘LogGap/gap01.log’’);
```

Per interrompere la scrittura sul file

```
> LogTo();
```

Facciamo pratica (I)

Nel prompt digitare

```
gap> srcivania:=DirectoryDesktop();
gap> file:=Filename(scrivania, "LogGap/gap01.log");
```

per salvare tutto quello che viene digitato nel file `gap01.log` nella cartella LogGap sul Desktop. Nel prompt digitare $(5+3)*9;$e premere invio.

Sul vostro schermo dovrebbe comparire qualcosa tipo

```
gap> srcivania:=DirectoryDesktop();
gap> file:=Filename(scrivania, "LogGap/gap01.log");
gap> (5+3)*9;
72
```

Facciamo pratica (II)

Ora si provi a digitare nel prompt $(5+3)*9$ senza il punto e virgola e si prema invio. Quello che si ottiene è

```
gap> (5+3)*9
>
```

Si noti che non succede niente perché GAP non sa che avete completato l'istruzione dal momento che ogni istruzione deve termina con un punto e virgola. Se ora si digita il punto e virgola si ottiene 72

```
gap> (5+3)*9
>;
72
gap>
```

Facciamo pratica (III)

Si provi a digitare $(5+3*9;$, premendo invio si ottiene un messaggio di errore

```
gap> (5+3)*9
Syntax error:  )  expected
```

questo è dovuto al fatto che il numero di parentesi aperte non coincide con il numero di parentesi chiuse.

Digitando <ctrl>+p riappare la stringa precedentemente inserita e usando i tasti delle frecce e possibile inserire la parentesi mancante.

Confronti

Il simbolo di uguaglianza, =, verifica se due espressioni sono uguali o no. Ad esempio:

```
gap> 8 = 9;
false
gap> 1^3+12^3=9^3+10^3;
true
```

Esercizio

- Si calcoli 3^{121};
- Si determini se $2^{25} + (45 * 51)$ è più grande di 34 milioni.

Confronti

Il simbolo di uguaglianza, =, verifica se due espressioni sono uguali o no. Ad esempio:

```
gap> 8 = 9;
false
gap> 1^3+12^3=9^3+10^3;
true
```

Esercizio

- Si calcoli 3^{121};
- Si determini se $2^{25} + (45 * 51)$ è più grande di 34 milioni.

Variabili (I)

Un oggetto può essere memorizzato in una variabile. Per le variabili può essere usata una qualsiasi sequenza di lettere e numeri che contenga almeno una lettera e non sia una parola riservata.
L'operatore di assegnamento è un due punti seguito dal simbolo uguale, `:=`. Una volta che un dato è stato registrato all'interno di una variabile questa può essere usata in sua vece per questo è spesso chiamata **identificatore**. Ad esempio:

```
gap> variabile1:=126;
126
gap> variabile1^2;
15876
```

Il simbolo = è l'operatore di confronto. Per assegnare un valore ad una variabile usare il simbolo :=.

```
gap> variabile2=14;
Variable:  'variabile2' must have a value
```

Variabili (II)

In GAP non è necessario dichiarare il tipo di variabili.

Nel seguente esempio a è il nome di una variabile.

```
gap> a:= (10 + 7) * (9 - 6);
51
gap> a;
51
gap> a*(a-1);
2250
gap> a:= 14;;
gap> a*(a-1);
182
```

Quando un valore è assegnato ad una variabile il valore è ripetuto nella linea successiva. Nella riga `gap> a:= 14;;` poiché sono presenti due punti e virgola il valore assegnato alla variabile non viene ripetuto.

Variabili (III)

Esercizio

- Si assegni il valore `4` all'identificatore `b`.
- Si assegni il valore `522456` all'identificatore `numerogrande`.
- Si calcoli `b + numerogrande`

Le matrici

Le **matrici** in GAP sono rappresentate come liste di vettori riga che abbiano tutti la stessa lunghezza e i cui elementi appartengano ad un anello comune.
Considerare in GAP le matrici

$$a := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \text{ e } b := \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}.$$

Per farlo digitiamo nel promt dei comandi

```
gap> a:=[[1,0],[0,-1]];
 [ [ 1, 0 ], [ 0, -1 ] ]
gap> b:=[[1,1],[0,-1]];
 [ [ 1, 1 ], [ 0, -1 ] ]
```

Gruppi di Matrici (I)

Ora vogliamo considerare il gruppo generato da queste due matrici, In altre parole vogliamo considerare il gruppo $g := \langle a, b \rangle$.
Per definire il gruppo generato da una lista di generatori (in questo caso le matrici *a* *b*) in GAP si utilizza il comando `Group( gen, ... )`.
Nel nostro esempio digitiamo sul prompt

```
g:=Group(a,b);
 Group([ [ [ 1, 0 ], [ 0, -1 ] ], [ [ 1, 1 ], [ 0, -1 ] ]
])
```

Gruppi di Matrici (II)

Vogliamo ora considerare in GAP il gruppo h generato dalle matrici

$$c := \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \text{ e } d := \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}.$$

Per prima cosa abbiamo bisogno di considerare in GAP l'unità immaginaria i. Un modo per far questo è quello di usare l'istruzione `E(4)`. Quindi per considerare le matrici c e d digitiamo le seguenti istruzioni

```
gap> c:=[[0,1],[-1,0]];
 [ [ 0, 1 ], [ -1, 0 ] ]
gap> d:=[[0,E(4)],[E(4),0]];
 [ [ 0, E(4) ], [ E(4), 0 ] ]
```

Il gruppo libero in GAP

Per implementare in GAP il gruppo libero si usa il comando `f:=FreeGroup(n);` che genera in GAP il gruppo libero con n generatori. Per accedere agli elementi nella lista dei generatori si usa l'istruzione `nomeGruppo.posizioneNellaLista`, ad esempio per accedere al secondo generatore del gruppo `f` usiamo `f.2`.

Presentazione di un gruppo in GAP

Per implementare in GAP un gruppo gruppo finitamente relazionato si usa l'istruzione `GruppoLibero/[lista relazioni]` Ad esempio, un presentazione del gruppo simmetrico S_3 può essere data a partire dal gruppo libero generato da due elementi i, j tali che $i^2 = 1, j^2 = 1, (ij)^3 = 1$ per implementare questo in GAP possiamo usare il seguente comando.

```
gap> s3_2:=f/[f.1^2,f.2^2,(f.2*f.1)^(3)];
<fp group on the generators [ f1, f2 ]>
```