

# AMPL Sintassi ed Esempi

F. Rinaldi

Dipartimento di Matematica  
Università di Padova

Corso di Laurea Informatica

# Outline

## AMPL Sintassi ed Esempi

Sintassi

Esempi di Modellazione in AMPL

# Elementi di un problema

- ▶ Insiemi
- ▶ Parametri
- ▶ Variabili
- ▶ Funzione Obiettivo
- ▶ Vincoli
- ▶ Dati

# Insiemi

- ▶ Gli insiemi definiscono gli indici di variabili, parametri e vincoli del modello;
- ▶ Dichiarazione (file.mod): `set NomeInsieme;`
- ▶ Definizione (file.dat): `set NomeInsieme:= e1 e2 ... eM;`

Diversi tipi di insieme:

- ▶ insiemi non ordinati:

```
set NomeInsieme;
```

- ▶ insiemi numerici:

```
set NomeInsieme:= inizio .. fine by distanza;
```

- ▶ insiemi ordinati:

```
set NomeInsieme ordered;
```

- ▶ insiemi circolari:

```
set NomeInsieme circular;
```

# Insiemi

- ▶ Gli insiemi definiscono gli indici di variabili, parametri e vincoli del modello;
- ▶ Dichiarazione (file.mod): `set NomeInsieme;`
- ▶ Definizione (file.dat): `set NomeInsieme:= e1 e2 ... eM;`

Diversi tipi di insieme:

- ▶ insiemi non ordinati:  
`set NomeInsieme;`
- ▶ insiemi numerici:  
`set NomeInsieme:= inizio .. fine by distanza;`
- ▶ insiemi ordinati:  
`set NomeInsieme ordered;`
- ▶ insiemi circolari:  
`set NomeInsieme circular;`

# Insiemi Numerici

- ▶ Gli insiemi numerici non devono necessariamente essere interi
- ▶ AMPL cerca di memorizzare ogni elemento dell'insieme utilizzando il fpn più vicino

- ▶ Provare le seguenti istruzioni:

```
set a:= -5/3 .. 5/3 by 1/3;  
display a;
```

- ▶ 0 e 1 non sono elementi dell'insieme!
- ▶ meglio usare interi

# Insiemi Numerici

- ▶ Gli insiemi numerici non devono necessariamente essere interi
- ▶ AMPL cerca di memorizzare ogni elemento dell'insieme utilizzando il fpn più vicino

- ▶ Provare le seguenti istruzioni:

```
set a:= -5/3 .. 5/3 by 1/3;  
display a;
```

- ▶ 0 e 1 non sono elementi dell'insieme!
- ▶ meglio usare interi

# Insiemi Numerici

- ▶ Gli insiemi numerici non devono necessariamente essere interi
- ▶ AMPL cerca di memorizzare ogni elemento dell'insieme utilizzando il fpn più vicino

- ▶ Provare le seguenti istruzioni:

```
set a:= -5/3 .. 5/3 by 1/3;  
display a;
```

- ▶ 0 e 1 non sono elementi dell'insieme!
- ▶ meglio usare interi



# Operazioni tra Insiemi

Dati due insiemi A e B possibile in AMPL effettuare operazioni elementari:

Operazione	Significato
A union B	insieme di elementi che stanno in A o B
A inter B	insieme di elementi che stanno sia in A che in B
A diff B	insieme di elementi che stanno in A ma non in B
A symdiff B	insieme di elementi che stanno in A o in B ma non in entrambi
card(A)	numero di elementi che stanno in A

`within`: serve a dichiarare un insieme come sottoinsieme di un altro:

```
set A;  
set B within A;
```

**IMPORTANTE:** Nella definizione A e B devono essere coerenti!

# Operazioni tra Insiemi

Dati due insiemi A e B possibile in AMPL effettuare operazioni elementari:

Operazione	Significato
$A \cup B$	insieme di elementi che stanno in A o B
$A \cap B$	insieme di elementi che stanno sia in A che in B
$A \setminus B$	insieme di elementi che stanno in A ma non in B
$A \text{ symdiff } B$	insieme di elementi che stanno in A o in B ma non in entrambi
$\text{card}(A)$	numero di elementi che stanno in A

`within`: serve a dichiarare un insieme come sottoinsieme di un altro:

```
set A;  
set B within A;
```

**IMPORTANTE:** Nella definizione A e B devono essere coerenti!

# Esercizio

Generate due insiemi numerici

- ▶  $A \in [40, 80]$  con intervallo 4
- ▶  $B \in [50, 100]$  con intervallo 5

Calcolare  $A \text{ union } B$ ,  $A \text{ inter } B$ ,  $B \text{ diff } A$ ,  $A \text{ symdiff } A$ ,  $\text{card}(A \text{ union } B)$ .

Ricorda:

- ▶ le operazioni tra insiemi all'interno di una espressione complessa vengono effettuate da sinistra verso destra
- ▶ gerarchia: inter seguito da union, diff, symdiff (stessa priorità)
- ▶  $A \text{ union } B \text{ inter } C \text{ diff } D$  equivale a  $(A \text{ union } (B \text{ inter } C)) \text{ diff } D$

# Esercizio

Generate due insiemi numerici

- ▶  $A \in [40, 80]$  con intervallo 4
- ▶  $B \in [50, 100]$  con intervallo 5

Calcolare  $A \text{ union } B$ ,  $A \text{ inter } B$ ,  $B \text{ diff } A$ ,  $A \text{ symdiff } A$ ,  $\text{card}(A \text{ union } B)$ .

Ricorda:

- ▶ le operazioni tra insiemi all'interno di una espressione complessa vengono effettuate da sinistra verso destra
- ▶ gerarchia: inter seguito da union, diff, symdiff (stessa priorità)
- ▶  $A \text{ union } B \text{ inter } C \text{ diff } D$  equivale a  $(A \text{ union } (B \text{ inter } C)) \text{ diff } D$

## Operazioni tra Insiemi II

Dato un insieme ordinato, possibile in AMPL effettuare operazioni elementari:

Operazione	Significato
<code>first(A)</code>	primo elemento di A
<code>last(A)</code>	ultimo elemento di A
<code>next(a,A)</code>	elemento di A dopo di a
<code>prev(a,A)</code>	elemento di A prima di a
<code>next(a,A,k)</code>	k-esimo elemento di A dopo di a
<code>prev(a,A,k)</code>	k-esimo elemento di A prima di a
<code>ord(a,A)</code>	posizione di a in A
<code>ord0(a,A)</code>	come <code>ord(a,A)</code> , ma restituisce 0 se a non è in A
<code>member(k,A)</code>	elemento di A in k-esima posizione

# Insiemi Ordinati

- ▶ Provare le seguenti istruzioni:

```
set a ordered;  
data;  
set a:= a1 a2 a3 a4 a5 a6 a7;  
display first(a), last(a);  
display next('a3',a), next('a3',a,2);  
display ord('a3',a);
```

# Insiemi a più dimensioni

`dimen`: Permette di specificare la dimensione di un insieme contestualmente alla sua dichiarazione:

- ▶ dichiarazione(file.mod): `set TUPLE dimen 3 ;`
- ▶ definizione (file.dat): `set TUPLE:= (1,7,1) (7,1,1) (10,5,3) (pippo, pluto, paperino) ;`

`cross`: modo alternativo per costruire insiemi di dimensione >1

```
set INS1;  
set INS2;  
set INS3;  
set INS:= INS1 cross INS2 cross INS3;
```

`setof` effettua procedimento inverso rispetto a `cross`(file.mod):

```
set INS dimen 3;  
set INS1:=setof{(i,j,h) in INS} i ;  
set INS2:=setof{(i,j,h) in INS} j ;  
set INS3:=setof{(i,j,h) in INS} h ;
```

# Parametri

- ▶ `param`: per creare i parametri del modello
- ▶ dichiarazione(`file.mod`): `param T;`
- ▶ definizione (`file.dat`): `param T:= 10;`
- ▶ possibile anche specificare delle restrizioni: `param T>1;`



## Parametri a più dimensioni II

### Dichiarazione(file.mod):

```

set INS1;
set INS2;
set INS3;
param p1{INS1};
param q1{INS1};
param p2{INS1,INS2};
param p3{INS1, INS2, INS3};

```

### Definizione(file.dat):

```

set INS1:=A1 A2;
set INS2:=B1 B2 B3;
set INS3:=C1 C2 C3;
param : p1 q1:=
    A1    1  5
    A2    2  3;
param p2: B1 B2 B3:=
    A1    3  5  7
    A2    1  4  8;
param p3:=
[A1,*,*]: C1 C2 C3:=
    B1    3  6  5
    B2    1  4  8
    B3    3  2  6
[A2,*,*]: C1 C2 C3:=
    B1    4  6  5
    B2    2  5  8
    B3    3  7  6;

```

# Variabili

Variabili sono le incognite del modello:

```
set INS;  
param u{INS};  
  
var prova;  
  
var x{INS} integer;  
var y{INS} binary;  
  
var z{INS}>=0, <=M;  
var w{p in INS}>=0, <=u[p];
```

# Funzione Obiettivo e Vincoli

La f.o. rappresenta ciò che vogliamo massimizzare o minimizzare(file.mod):

```
minimize(o maximize) nomeobiettivo: espressionearitmetica;
```

I vincoli rappresentano le specifiche che la soluzione deve soddisfare(file.mod):

```
subject to nomevincolo: espressionelogica;
```

```
s.t. nomevincolo: espressionelogica;
```

# Funzioni Aritmetiche e Operatori Indicizzati

Funzioni Aritmetiche:

Funzione	Significato
<code>abs(x)</code>	valore assoluto di x
<code>floor(x)</code>	approssimazione intera per difetto di x
<code>ceil(x)</code>	approssimazione intera per eccesso di x
<code>sqrt(x)</code>	radice quadrata x
<code>log(x)</code>	logaritmo naturale di x
<code>exp(x)</code>	esponenziale di x
<code>sqrt(x)</code>	radice quadrata di x

Operatori Indicizzati:

Funzione	Significato
Sommatoria	$\text{sum}\{ a \text{ in } A \}$
Produttoria	$\text{prod}\{ a \text{ in } A \}$
Massimo	$\text{max}\{ a \text{ in } A \}$
Minimo	$\text{min}\{ a \text{ in } A \}$

Possibile utilizzare sommatorie multiple e condizionate

```
sum{r in MPRIM, f in PROD: r<>3}
```



# Esempio 1

## Pianificazione Produzione

Un'azienda produce fibra di vetro al metro cubo e desidera pianificare la produzione per le prossime sei settimane. La capacità produttiva è limitata e tale limite varia al variare del periodo considerato. La domanda settimanale è nota per l'intero periodo. I costi di produzione ed immagazzinamento variano al variare del periodo. I dati sono riportati in Tabella 1:

Tabella 1

Settimana	Cap.	Domanda	Costo Prod.	Costo Imm.
1	140	100	5	0.2
2	100	120	8	0.3
3	110	100	6	0.2
4	100	90	6	0.25
5	120	120	7	0.3
6	100	110	6	0.4

Pianificare la produzione in modo da minimizzare il costo totale di produzione e stoccaggio.