

## COMPITO DI APPLICAZIONI DI INTELLIGENZA ARTIFICIALE

23 Giugno 2004 (Tot. 30/30) Tempo: 2h

### Esercizio 1 (punti 8)

Dato il seguente training set S:

A1	A2	Classe
1	0	A
1	1	A
3	0	B
2	1	B
2	0	A
2	1	B
?	0	A
3	1	B
3	0	B

- Si calcoli l'entropia del training set rispetto all'attributo Classe
- Si calcoli il gain ratio dei due attributi rispetto a questi esempi di training.
- si costruisca un albero decisionale ad un solo livello per il training set dato, indicando le etichette delle foglie (numero di esempi finiti nella foglia/numero di esempi finiti nella foglia non appartenenti alla classe della foglia).

### Esercizio 2 (punti 6)

Si scrive un predicato CLP(FD) che impone che una lista di variabili FD, date in ingresso, sia ordinata in senso decrescente stretto.

Esempio:

```
?- [A,B,C]::0..10, ordina([A,B,C,D]).  
yes, A:: 2..10  
      B:: 1..9  
      C:: 0..8
```

**Facoltativo:** Si mostrino poi i passi eseguiti dall'algoritmo AC3 (come vengono modificati i domini delle variabili ad ogni imposizione e risveglio di un vincolo) invocando il goal:

```
?- [A,B,C]::0..10, D::5..8, ordina([A,B,C,D]).
```

### Esercizio 3 (punti 8)

È dato lo stato iniziale descritto dalle seguenti formule atomiche:

**[ontable(b), ontable(d), on(a,b), on(c,d), clear(a), clear(c), handempty]**

(a,b, c,d rappresentano dei blocchi e si suppone ci siano infinite posizioni occupabili del tavolo  
- le azioni sono modellate opportunamente come segue:

#### **pickup(X)**

PRECOND: ontable(X), clear(X), handempty

DELETE: ontable(X), clear(X), handempty

ADD: holding(X)

#### **putdown(X)**

PRECOND: holding(X)

DELETE: holding(X)

ADD: ontable(X), clear(X), handempty

#### **stack(X,Y)**

PRECOND: holding(X), clear(Y)

DELETE: holding(X), clear(Y)

ADD: handempty, on(X,Y), clear(X)

#### **unstack(X,Y)**

PRECOND: handempty, on(X,Y), clear(X)

DELETE: handempty, on(X,Y), clear(X)

ADD: holding(X), clear(Y)

e il goal **ontable(a) and on(b,c)**

si descriva come l'algoritmo lineare backward STRIPS trova un piano per questo goal. Si descriva lo stato e lo stack dei goal passo passo.

### Esercizio 4 (punti 6)

Discutere le modalità di ragionamento in backward e forward chaining, le differenze e le opportunità di applicazione.

## SOLUZIONE

### Esercizio 1

$$a) \text{info}(S) = -4/9 * \log_2 4/9 - 5/9 * \log_2 5/9 = 0,991$$

b) Per calcolare il guadagno dell'attributo A1 non si usa l'entropia calcolata su tutto il training set ma solo sugli esempi che hanno A1 noto (insieme F):

$$\text{info}(F) = -2/8 * \log_2 2/8 - 3/8 * \log_2 3/8 - 3/8 * \log_2 3/8 = 1,561$$

$$\text{info}_{A1}(F) = 2/8 * (-0/2 * \log_2 0/2 - 2/2 * \log_2 2/2) + 3/8 * (-0/3 * \log_2 0/3 - 3/3 * \log_2 3/3) + 3/8 * (-2/3 * \log_2 2/3 - 1/3 * \log_2 1/3) =$$

$$= 0,25 * 0 + 0,375 * 0 + 0,375 * 0,918 = 0,344$$

$$\text{gain}(A1) = 8/9 * (1,561 - 0,344) = 1,082$$

$$\text{splitinfo}(A1) = -2/9 * \log_2(2/9) - 3/9 * \log_2(3/9) - 3/9 * \log_2(3/9) - 1/9 * \log_2(1/9) = 1,891$$

$$\text{gainratio}(A1) = 1,082 / 1,891 = 0,572$$

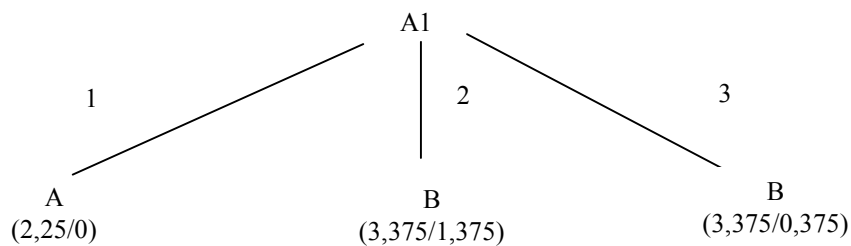
$$\text{info}_{A2}(S) = 5/9 * (-3/5 * \log_2 3/5 - 2/5 * \log_2 2/5) + 4/9 * (-1/4 * \log_2 1/4 - 3/4 * \log_2 3/4) =$$

$$= 0,556 * 0,971 + 0,444 * 0,811 = 0,900$$

$$\text{gain}(A2) = 0,991 - 0,900 = 0,091$$

$$\text{splitinfo}(A2) = -5/9 * \log_2(5/9) - 4/9 * \log_2(4/9) = 0,991$$

$$\text{gainratio}(A2) = 0,091 / 0,991 = 0,092$$



### Esercizio 2

```
ordina([_]).
ordina([A,B|T]) :-
    A #> B,
    ordina([B|T]).
```

Esecuzione del goal:

```
?- [A,B,C]::0..10, D::5..8, ordina([A,B,C,D]).
```

Viene invocato A#>B:

```
A::1..10, B::0..9
```

Viene invocato B#>C:

```
B::1..9, C::0..8
```

Risveglio A#>B:

```
A::2..10, B::1..9
```

Viene invocato C#>D:

```
C::6..8, D::5..7
```

Risveglio B#>C:

B::7..9      C::6..8

Risveglio A#>B

A:: 8..10,      B::7..9

### Esercizio 3

**Stato:**

ontable(b)  
ontable(d)  
on(a,b)  
on(c,d)  
clear(a)  
clear(c)  
handempty

**Stack:**

ontable(a) and on(b,c)

Split del goal:

**Stato:**

ontable(b)  
ontable(d)  
on(a,b)  
on(c,d)  
clear(a)  
clear(c)  
handempty

**Stack:**

ontable(a)  
on(b,c)  
ontable(a) and on(b,c)

Goal regression con l'azione **putdown(X)**:

**Stato:**

ontable(b)  
ontable(d)  
on(a,b)  
on(c,d)  
clear(a)  
clear(c)  
handempty

**Stack:**

holding(a)  
**putdown(a)**  
on(b,c)

ontable(a) and on(b,c)

Goal regression con l'azione **unstack(X,Y)**:

**Stato:**

ontable(b)  
ontable(d)  
on(a,b)  
on(c,d)  
clear(a)  
clear(c)  
handempty

**Stack:**

handempty and on(a,b) and clear(a)  
**unstack(a,b)**  
**putdown(a)**  
on(b,c)  
ontable(a) and on(b,c)

Split del goal:

**Stato:**

ontable(b)  
ontable(d)  
on(a,b)  
on(c,d)  
clear(a)  
clear(d)  
handempty

**Stack:**

handempty  
on(a,b)  
clear(a)  
handempty and on(a,b) and clear(a)  
**unstack(a,b)**  
**putdown(a)**  
on(b,c)  
ontable(a) and on(b,c)

Tutti i primi 4 goal sono verificati nello stato corrente:

**Stato:**

ontable(b)  
ontable(d)  
on(a,b)  
on(c,d)  
clear(a)  
clear(c)  
handempty

**Stack:**

**unstack(a,b)**  
**putdown(a)**  
on(b,c)  
ontable(a) and on(b,c)

Esecuzione dell'azione unstack(a,b):

**Stato:**  
ontable(b)  
ontable(d)  
on(c,d)  
clear(c)  
holding(a)  
clear(b)

**Stack:**  
**putdown(a)**  
on(b,c)  
ontable(a) and on(b,c)

Esecuzione dell'azione putdown(a):

**Stato:**  
ontable(b)  
ontable(d)  
on(c,d)  
clear(c)  
clear(b)  
ontable(a)  
clear(a)  
handempty

**Stack:**  
on(b,c)  
ontable(a) and on(b,c)

Goal regression con l'azione **stack(X,Y)**:

**Stato:**  
ontable(b)  
ontable(d)  
on(c,d)  
clear(c)  
clear(b)  
ontable(a)  
clear(a)  
handempty

**Stack:**  
holding(b) and clear(c)  
**stack(b,c)**  
on(b,c)  
ontable(a) and on(b,c)

Split del goal:

**Stato:**  
ontable(b)  
ontable(d)  
on(c,d)  
clear(c)  
clear(b)  
ontable(a)  
clear(a)  
handempty

**Stack:**  
holding(b)  
clear(c)  
holding(b) and clear(c)  
**stack(b,c)**  
on(b,c)  
ontable(a) and on(b,c)

Goal regression con l'azione **pickup(X)**:

**Stato:**  
ontable(b)  
ontable(d)  
on(c,d)  
clear(c)  
clear(b)  
ontable(a)  
clear(a)  
handempty

**Stack:**  
ontable(b) and clear(b) and handempty  
**pickup(b)**  
clear(c)  
holding(b) and clear(c)  
**stack(b,c)**  
on(b,c)  
ontable(a) and on(b,c)

Split del goal, poi i primi 4 goal sono verificati nello stato corrente.

**Stato:**  
ontable(b)  
ontable(d)  
on(c,d)  
clear(c)  
clear(b)  
ontable(a)  
clear(a)

handempty

**Stack:**

**pickup(b)**

clear(c)

holding(b) and clear(c)

**stack(b,c)**

on(b,c)

ontable(a) and on(b,c)

Esecuzione di pickup(b):

**Stato:**

ontable(d)

on(c,d)

clear(c)

ontable(a)

clear(a)

holding(b)

**Stack:**

clear(c)

holding(b) and clear(c)

**stack(b,c)**

on(b,c)

ontable(a) and on(b,c)

Primi due goal soddisfatti nello stato corrente:

**Stato:**

ontable(d)

on(c,d)

clear(c)

ontable(a)

clear(a)

holding(b)

**Stack:**

**stack(b,c)**

on(b,c)

ontable(a) and on(b,c) and on(c,d)

Esecuzione di stack(b,c)

**Stato:**

ontable(d)

on(c,d)

on(b,c)

ontable(a)

clear(a)

clear(b)



handempty

**Stack:**

on(b,c)

ontable(a) and on(b,c)

Tutti i goal nello stack sono soddisfatti nello stato corrente.