

UNIVERSITA' DEGLI STUDI DI FIRENZE

Facoltà di Ingegneria  
Corso di Laurea in Ingegneria Informatica

***Esercitazioni per l'esame di  
"Intelligenza Artificiale"  
Prof. G. Soda***

***Classic***

di Sauro Menchetti

***A.A.1998-99***

# II CLASSIC

Il CLASSIC è un linguaggio di modellizzazione che si basa sulla descrizione strutturata dei concetti. Un punto di forza di questo approccio è che lo stesso linguaggio serve a definire lo schema, ad aggiornare il database, a formulare ed a rispondere alle query. Tale approccio permette inoltre di descrivere in modo parziale gli elementi del sistema (non è detto si conoscano subito tutte le informazioni di interesse), di rispondere alle query in modo descrittivo e di estendere in modo semplice lo schema del DB. Il CLASSIC esegue in modo automatico inferenze su concetti ed individui in modo tale che tutto il processo inferenza rimane trattabile. I database tradizionali hanno raggiunto una grande efficienza restringendo la potenza del modello, ma sono per lo più inadeguati in quelle situazioni in cui oggetti complessi sono il modo naturale di descrivere il dominio, quando l'informazione non è completa e diviene disponibile in modo incrementale e quando è utile un ruolo più attivo nel dedurre le relazioni tra i dati. Il campo dei database logici o deduttivi soddisfa la maggior parte delle precedenti richieste, ma il suo principale inconveniente è l'intrattabilità computazionale. Il CLASSIC si sforza quindi di fornire un insieme limitato di costrutti in grado di generare un linguaggio espressivo ed allo stesso tempo computazionalmente trattabile.

## 1.1 Dominio da modellare

Il dominio da modellare è una piccola parte di un archivio utilizzato dalla segreteria studenti di una facoltà. È di interesse distinguere le varie classi di studenti iscritti alla facoltà in base al sesso, al tipo di iscrizione ed al fatto se sono studenti a tempo pieno o meno (ad esempio se svolgono anche un'attività lavorativa). Combinando le precedenti alternative, gli studenti possono essere classificati in base a diversi criteri e si possono definire dei nuovi concetti: ad esempio, si può definire il concetto di "studente bravo" come quello di uno studente che è iscritto in corso ed è in pari con gli esami, pur svolgendo anche un'attività lavorativa. Per differenziare ed accomunare certe caratteristiche, si crea una gerarchia di concetti in cui i concetti figli ereditano tutte le caratteristiche dei genitori ed in più ne aggiungono di nuove: le classi foglia sono quindi le più complesse e quelle che semanticamente hanno più significato.

## 1.2 Analisi del dominio

Il primo concetto inserito nel DB è quello di `PERSONA`: tutti gli studenti sono persone ed in quanto tali possiedono un nome, un cognome, un'età ed un recapito. Le persone sono distinte in due classi disgiunte in base al sesso: si parla quindi di uomini e donne. Un altro concetto utile è quello di `OCCUPAZIONE` che viene specializzato in `LAVORATORE` e `STUDENTE`. Il concetto generico di `OCCUPAZIONE` raggruppa le caratteristiche comuni dei due sottoconcetti come ad esempio l'orario, il luogo e la durata (part-time o a tempo pieno). I due sottoconcetti si specializzano aggiungendo attributi e relazioni proprie come la data di assunzione, lo stipendio, il numero di matricola e l'anno di iscrizione. A partire dai due concetti di studente e di lavoratore, è naturale definire il concetto di `STUDENTE-LAVORATORE` che ingloba tutte le caratteristiche degli studenti e dei lavoratori. Infine è opportuno definire il concetto di `ISCRIZIONE` distinguibile nei due sottosistemi disgiunti in corso e fuori corso e caratterizzato da una facoltà, da un anno di immatricolazione, da un anno accademico e da una tassa di iscrizione.

## 1.3 Definizione dello schema

La definizione dello schema consiste nel dare dei nomi ai concetti ed ai ruoli che sono di interesse per i vari utenti, in modo da costruire un vocabolario per interagire con il database. Un DB costruito con il CLASSIC è per lo più un archivio che contiene informazioni su *oggetti individuali* o *individui*. Gli oggetti hanno un'identità intrinseca e sono correlati gli uni agli altri per mezzo di relazioni binarie dette *ruoli* (noti anche come attributi o proprietà). Gli individui sono raggruppati in insiemi detti *concetti* che ne mettono in risalto le caratteristiche comuni. Una caratteristica del CLASSIC è quella di definire i concetti in modo compositivo: alcuni concetti sono ottenuti raggruppando insieme degli individui, altri più complessi raggruppano alcuni concetti già definiti. La definizione dello schema si distingue in una dichiarazione dei ruoli ed in una definizione dei concetti. Il DDL (data definition language) permette la definizione dello schema del DB.

**Notazione:** le parti riguardanti l'utilizzo del CLASSIC sono scritte in Courier New. Più precisamente, i simboli scritti in maiuscolo sono **CONCETTI**, quelli in minuscolo sono *ruoli* o *indici* e quelli misti sono *Individui*. I costruttori dei concetti come **PRIMITIVE** sono scritti con **QUESTI CARATTERI**. Gli operatori sul database sono scritti in **questo stile**, mentre le procedure del linguaggio ospite che sono argomenti del costrutto **TEST** in *questi caratteri*.

### 1.3.1 Dichiarazione dei ruoli

Serve per comunicare al database che un certo identificatore sarà usato come nome di un ruolo, in modo da permettere la rilevazione di alcuni tipi di errore come ad esempio quelli di tipo.

Ruoli per il concetto PERSONA:

```
define-role[nome]
define-role[cognome]
define-role[età]
define-role[recapito]
```

Ruoli per il concetto OCCUPAZIONE:

```
define-role[impiego]
define-role[orario]
define-role[durata]
define-role[luogo]
```

Ruoli per il concetto LAVORATORE:

```
define-role[data-assunzione]
define-role[mestiere]
define-role[qualifica]
define-role[stipendio]
```

Ruoli per il concetto STUDENTE:

```
define-role[matricola]
define-role[anno-iscrizione]
define-role[libretto]
define-role[piano-di-studi]
define-role[lista-esami-sostenuti]
```

Ruoli per il concetto ISCRIZIONE:

```
define-role[facoltà]
define-role[anno-immatricolazione]
define-role[anno-accademico]
define-role[tassa-iscrizione]
```

Ruolo per i concetti ISCRIZIONE-IN-CORSO e ISCRIZIONE-FUORI-CORSO:

```
define-role[numero-di-iscrizioni]
```

Ruoli per il concetto STUDENTE-BRAVO:

```
define-role[esami-previsti].
```

### 1.3.2 Definizione dei concetti

Serve a definire i concetti del dominio in modo composizionale, partendo dai più semplici fino a quelli più complessi. Si distinguono in primitivi e definiti: la descrizione di un concetto primitivo è intesa a rappresentare solo *condizioni necessarie* per l'estensione del concetto e l'appartenenza di un individuo all'estensione *deve* essere stabilita dall'utente; la descrizione di un concetto definito è intesa a rappresentare *condizioni necessarie e sufficienti* per l'estensione del concetto e l'appartenenza di un individuo all'estensione *viene calcolata* dal sistema. Per la definizione dei concetti vengono usati i concetti di STRING ed INTEGER facenti parte dell'host language in cui è implementato il CLASSIC.

Definizione del concetto PERSONA:

```
define-concept [PERSONA,  
               (PRIMITIVE (AND THING  
                             (AT-LEAST 1 nome)  
                             (ALL nome STRING)  
                             (AT-LEAST 1 cognome)  
                             (ALL cognome STRING)  
                             (AT-LEAST 1 età)  
                             (AT-MOST 1 età)  
                             (ALL età (AND INTEGER  
                                       (TEST >=0)  
                                       (TEST <=200)))  
                             (AT-LEAST 1 recapito)  
                             (ALL recapito STRING))  
               persona) ]
```

Definizione dei concetti UOMO e DONNA:

```
define-concept [UOMO,  
               (DISJOINT-PRIMITIVE PERSONA genere maschile)]  
define-concept [DONNA,  
               (DISJOINT-PRIMITIVE PERSONA genere femminile)]
```

Definizione del concetto OCCUPAZIONE:

```
define-concept [OCCUPAZIONE,  
               (PRIMITIVE (AND THING  
                             (AT-LEAST 1 impiego)  
                             (ALL impiego STRING)  
                             (AT-LEAST 1 orario)  
                             (ALL durata (ONE-OF  
                                       part-time  
                                       tempo-pieno))  
                             (ALL luogo (ONE-OF  
                                       ufficio  
                                       scuola  
                                       fabbrica)))  
               occupazione) ]
```

Definizione del concetto LAVORATORE:

```
define-concept [LAVORATORE,  
  (AND PERSONA  
    OCCUPAZIONE  
    (AT-LEAST 1 data-assunzione)  
    (AT-LEAST 1 mestiere)  
    (AT-LEAST 1 qualifica)  
    (AT-LEAST 1 stipendio)  
    (ALL stipendio (AND INTEGER  
      (TEST >=0)))  
    (ALL luogo  
    (ONE-OF ufficio fabbrica)))]
```

Definizione del concetto STUDENTE:

```
define-concept [STUDENTE,  
  (AND PERSONA  
    OCCUPAZIONE  
    (AT-LEAST 1 matricola)  
    (AT-MOST 1 matricola)  
    (ALL matricola (AND INTEGER  
      (TEST >=0)))  
    (AT-LEAST 1 anno-iscrizione)  
    (AT-MOST 1 anno-iscrizione)  
    (ALL anno-iscrizione  
      (AND INTEGER (TEST >=0)))  
    (AT-LEAST 1 libretto)  
    (AT-MOST 1 libretto)  
    (ALL libretto STRING)  
    (AT-LEAST 1 piano-di-studi)  
    (AT-MOST 1 piano-di-studi)  
    (ALL piano-di-studi STRING)  
    (AT-LEAST 1 lista-esami-sostenuti)  
    (ALL lista-esami-sostenuti STRING)  
    (SAME-AS (luogo) (ONE-OF scuola)))]
```

Definizione del concetto STUDENTE-LAVORATORE:

```
define-concept [STUDENTE-LAVORATORE, (AND LAVORATORE STUDENTE)]
```

Definizione del concetto ISCRIZIONE:

```
define-concept [ISCRIZIONE,  
  (PRIMITIVE (AND THING  
    (AT-LEAST 1 facoltà)  
    (AT-MOST 1 facoltà)  
    (ALL facoltà STRING)  
    (AT-LEAST 1 anno-immatricolazione)  
    (AT-MOST 1 anno-immatricolazione)  
    (ALL anno-immatricolazione  
      (AND INTEGER (TEST >=0)))  
    (AT-LEAST 1 anno-accademico)  
    (AT-MOST 1 anno-accademico)  
    (ALL anno-accademico  
      (AND INTEGER (TEST >=0)))  
    (SAME-AS (anno-iscrizione)  
      (anno-accademico)))]
```

```

(AT-LEAST 1 tassa-iscrizione)
(ALL tassa-iscrizione
  (AND INTEGER (TEST >=0)))
iscrizione)]

```

Definizione del concetto ISCRIZIONE-IN-CORSO:

```

define-concept [ISCRIZIONE-IN-CORSO,
  (DISJOINT-PRIMITIVE
    (AND ISCRIZIONE (SAME-AS
      (anno-iscrizione)
      (numero-di-iscrizioni)))
    tipo-iscrizione in-corso)]

```

Definizione del concetto ISCRIZIONE-FUORI-CORSO:

```

define-concept [ISCRIZIONE-FUORI-CORSO,
  (DISJOINT-PRIMITIVE
    (AND ISCRIZIONE
      (TEST anno-iscrizione < numero-iscrizioni))
    tipo-iscrizione fuori-corso)]

```

Definizione del concetto STUDENTE-BRAVO:

```

define-concept [STUDENTE-BRAVO,
  (AND STUDENTE-LAVORATORE
    ISCRIZIONE-IN-CORSO
    (SAME-AS (lista-esami-sostenuti)
      (esami-previsti))) ]

```

Definizione del concetto STUDENTE LAVORATRICE-FUORI-CORSO:

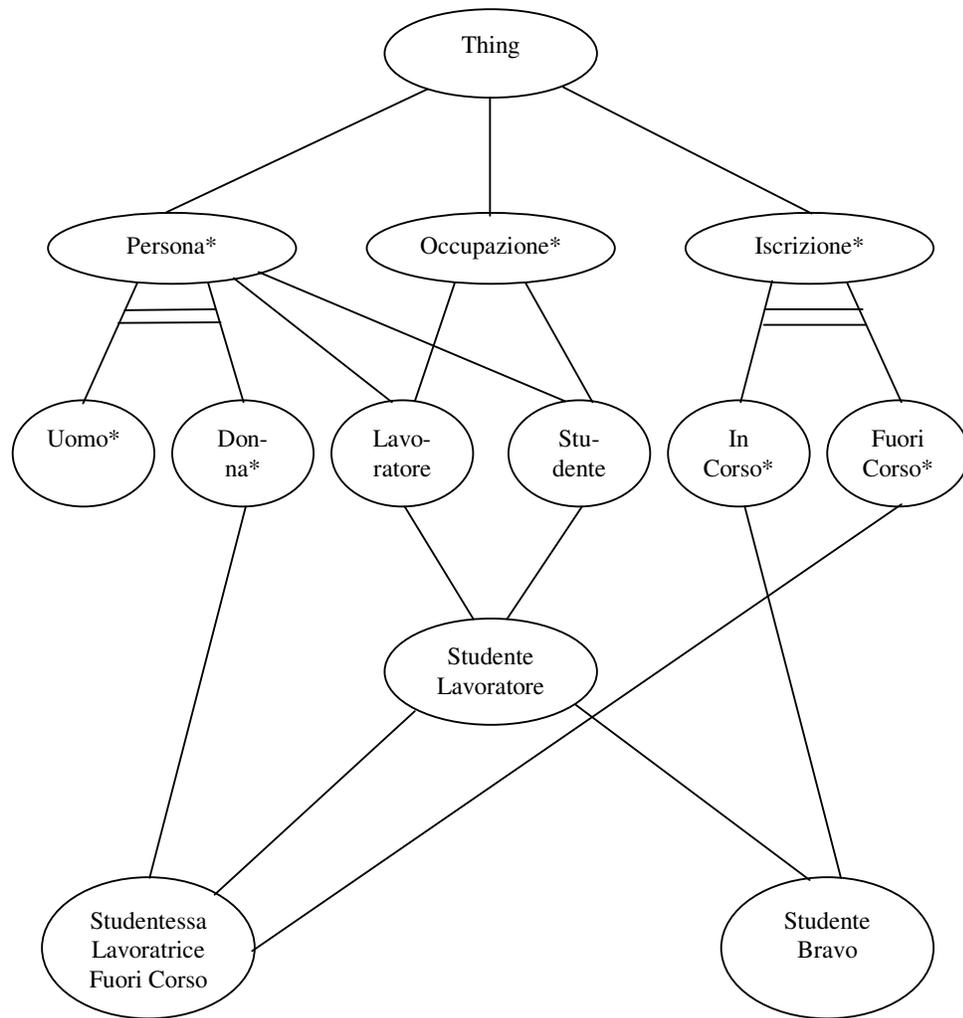
```

define-concept [STUDENTESSA-LAVORATRICE-FUORI-CORSO,
  (AND DONNA
    STUDENTE-LAVORATORE
    ISCRIZIONE-FUORI-CORSO) ].

```

### 1.3.3 Gerarchia tra concetti

La precedente definizione dei concetti induce la seguente gerarchia tra i vari concetti:



**Figura 1:** Gerarchia dei concetti

Il simbolo \* indica i concetti primitivi, mentre la doppia linea = lega le coppie dei concetti primitivi disgiunti. Preoccupiamoci adesso di inserire delle informazioni nel DB.

## 1.4 Aggiornamento del Database

L'aggiornamento del DB avviene in modo incrementale e non è quindi necessario conoscere subito tutte le informazioni riguardanti i vari individui. Il CLASSIC permette di memorizzare una visione incompleta dei fatti e di aggiungere di volta in volta le informazioni che divengono disponibili: questo approccio è particolarmente utile in quei domini in cui vi è un'alta incertezza. Il DML (data manipulation language) permette l'aggiornamento del contenuto del DB.

Inserimento dell'individuo Studente1:

```
create-ind[Studente1]
assert-ind[Studente1, STUDENTE]
assert-ind[Studente1, (FILLS nome Luca)]
assert-ind[Studente1, (CLOSE nome)]
assert-ind[Studente1, (FILLS cognome Rossi)]
assert-ind[Studente1, (CLOSE cognome)]
assert-ind[Studente1, (FILLS età 23)]
assert-ind[Studente1, (FILLS recapito Via Garibaldi 3 Firenze)]
assert-ind[Studente1, (FILLS impiego Studente)]
assert-ind[Studente1, (FILLS orario 8-13)]
assert-ind[Studente1, (FILLS durata tempo-pieno)]
assert-ind[Studente1, (FILLS luogo scuola)]
assert-ind[Studente1, (FILLS matricola 2124345)]
assert-ind[Studente1, (FILLS anno-iscrizione 1°)]
assert-ind[Studente1, (FILLS lista-esami-sostenuti
    Chimica Informatica Geometria)]
define-role[mezzo-di-trasporto]
assert-ind[Studente1, (ALL mezzo-di-trasporto
    (ONE-OF autobus auto motorino bicicletta))]
```

Inserimento dell'individuo StudenteBravo1:

```
create-ind[StudenteBravo1, STUDENTE-BRAVO]
assert-ind[StudenteBravo1,
    (AND (FILLS nome Marco)
        (CLOSE nome)
        (FILLS cognome Bianchi)
        (CLOSE cognome)
        (FILLS età 22)
        (FILLS recapito Via Tevere 6 Roma))]
assert-ind[StudenteBravo1,
    (AND (FILLS impiego Studente)
        (FILLS durata tempo-pieno)
        (FILLS orario 8,30-13,30)
        (FILLS luogo scuola)
        (FILLS impiego Lavoratore)
        (FILLS durata part-time)
        (FILLS orario non-precisato)
        (FILLS luogo ufficio)
        (CLOSE impiego))]
define-role[laureando]
assert-ind[StudenteBravo1, (AND (ALL laureando (ONE-OF Si No)
    (FILLS laureando Si))]
assert-ind[StudenteBravo1,
    (AND (FILLS data-assunzione 5-1-1995)
        (CLOSE data-assunzione))]
```

```

        (FILLS mestiere Impiegato)
        (CLOSE mestiere)
        (FILLS qualifica 1° Livello)
        (CLOSE qualifica)
        (FILLS stipendio 500000)
        (CLOSE stipendio)
        (FILLS matricola 1805698)
        (FILLS anno-iscrizione 5°)
        (FILLS lista-esami-sostenuti tutti))]
assert-ind[StudenteBravol,
        (AND (FILLS facoltà Ingegneria)
        (FILLS anno-immatricolazione 1994)
        (FILLS anno-accademico 1999)
        (FILLS tassa-iscrizione 750000))]
define-role[media]
assert-ind[StudenteBravol,
        (AND (ALL media INTEGER) (FILLS media 107))]

```

Inserimento dell'individuo StudententessaLavoratriceFuoriCorsol:

```

create-ind[StudentessaLavoratriceFuoriCorsol,
        STUDENTESSA-LAVORATRICE-FUORI-CORSO]
assert-ind[StudentessaLavoratriceFuoriCorsol,
        (AND (FILLS nome Daniela)
        (CLOSE nome)
        (FILLS cognome Lunghi)
        (CLOSE cognome)
        (FILLS età 26)
        (FILLS recapito Via Montale 9 Pisa))]
assert-ind[StudentessaLavoratriceFuoriCorsol,
        (AND (FILLS impiego Studente)
        (FILLS impiego Lavoratore)
        (CLOSE impiego))]
assert-ind[StudentessaLavoratriceFuoriCorsol,
        (AND (FILLS data-assunzione 5-1-1995)
        (CLOSE data-assunzione)
        (FILLS stipendio 1500000)
        (CLOSE stipendio)
        (FILLS matricola 1805698)
        (FILLS anno-iscrizione 4°))]
assert-ind[StudentessaLavoratriceFuoriCorsol,
        (AND (FILLS facoltà Farmacia)
        (FILLS anno-immatricolazione 1991)
        (FILLS anno-accademico 1999)
        (FILLS tassa-iscrizione 1450000))].

```

### 1.4.1 Regole

Una regola consiste di due operandi, un antecedente e un conseguente, entrambi concetti. Sono della forma “se un individuo è un <concetto1>, allora è anche un <concetto2>” e aggiungono al database l’informazione che ogni istanza del <concetto1> è anche un’istanza del <concetto2>. Questo è diverso dal definire il <concetto2> come parte del concetto <concetto1>, in quanto in quel caso il sistema non riconosce un individuo facente parte del <concetto1> se non soddisfa anche il <concetto2>. La regola permette invece al DB di dedurre che qualsiasi istanza del <concetto1> è anche un’istanza del <concetto2>. Ecco un esempio di regola:

```
define-role[impegno]
```

```
assert-rule[STUDENTE, (AND (ALL impegno (ONE-OF poco molto))  
(FILLS impegno molto))]
```

significa che ogni individuo che è uno **STUDENTE** ha come riempitore del ruolo **impegno** la stringa **molto** (il ruolo **impegno** può assumere solo i due valori **poco** e **molto**).

## 1.5 Interrogazione del Database

Il **CLASSIC** mette a disposizione una serie di operatori per l'interrogazione del **DB**: tali operatori fanno parte del **Query Language**. Si possono formulare interrogazioni sia sui concetti, sia sui ruoli. Ecco alcuni esempi di query sui concetti:

```
concept-aspect [STUDENTE, AT-LEAST]
```

restituisce tutti gli **AT-LEAST** che fanno parte della definizione del concetto di **STUDENTE**;

```
concept-aspect [STUDENTE-BRAVO, ALL, lista-esami-sostenuti]
```

restituisce il tipo del vincolo sul riempitore del ruolo **lista-esami-sostenuti**;

```
concept-aspect [ISCRIZIONE-IN-CORSO, AT-LEAST, facoltà]
```

restituisce il limite inferiore sul vincolo del ruolo **facoltà**;

```
concept-subsumes [STUDENTE, STUDENTE-LAVORATORE]
```

restituisce vero se ogni individuo che soddisfa **STUDENTE-LAVORATORE** è anche un'istanza di **STUDENTE** (relazione di sussunzione). Ecco delle query sui riempitori dei ruoli:

```
ind-aspect [Studente1, FILLS, nome]
```

restituisce il nome dell'individuo **Studente1**.

```
ind-aspect [StudenteBravo1, CLOSE]
```

restituisce i ruoli dell'individuo **StudenteBravo1** chiusi da **CLOSE**.

```
ind-aspect [StudentessaLavoratriceFuoriCorsol, FILLS, stipendio]
```

restituisce l'ammontare dello stipendio dell'individuo **StudentessaLavoratriceFuoriCorsol**.