

Linguaggi per la rappresentazione della conoscenza, Logiche Descrittive e NeoClassic

Matteo Barontini

Rappresentazione della conoscenza

- *Base di conoscenza:*
 - rappresentazione esplicita, parziale e compatta
 - in un linguaggio simbolico dichiarativo
 - capacità di inferire nuovi fatti a partire da quelli (esplicitamente) memorizzati.
- *Basi di dati:*
 - solo fatti specifici e positivi
 - assumono una conoscenza completa del mondo
 - nessuna capacità inferenziale

Schemi di R.C.

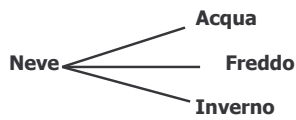
- Diversi tipi di schemi:
 - Procedurali (sistemi di produzione)
 - Basati sulla logica (FOL)
 - Rappresentazioni strutturate:
 - Che enfatizzano le associazioni di concetti (reti semantiche)
 - Che enfatizzano le aggregazioni di concetti (frame)

Approccio psicologico-linguistico alla R.C.

- La logica per formalizzare il ragionamento valido
 - nata per la matematica e poi estesa al ragionamento di "senso comune".
- Il filone psicologico-linguistico pone l'enfasi sui meccanismi di rappresentazione ed uso della conoscenza.

Teorie associazioniste

- Le teorie *associazioniste* si occupano delle relazioni tra simboli e del significato che scaturisce da tali associazioni

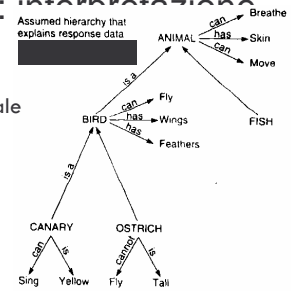


Organizzazione gerarchica dei concetti: interpretazione

Le proprietà sono "attaccate" al concetto più generale a cui si applicano

Treatmento eccezioni

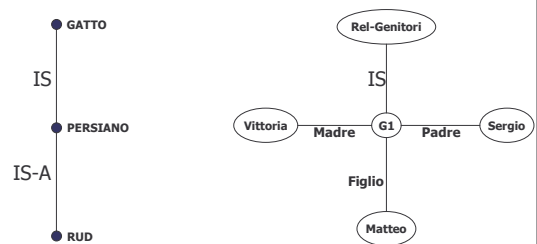
Le eccezioni sono memorizzate direttamente con l'oggetto



Le reti semantiche: definizione

- Le reti semantiche sono una grande famiglia di schemi di rappresentazione.
- Un rete semantica è un grafo in cui:
 - I nodi, etichettati, corrispondono a concetti (individui o classi)
 - Gli archi, etichettati e orientati, a relazioni binarie tra concetti
- Due particolari relazioni "primitive" sono sempre presenti:
 - IS ("è" in inglese), relazione di sotto-classe (\subseteq)
 - IS-A ("è un", sempre in inglese), relazione di appartenenza (\in)

Ereditarietà nelle reti semantiche



Mancanza di semantica

- Incongruenza tra l'uso del primo link e degli altri due
 - Il primo esprime una proprietà che si applica a tutti gli elementi della classe
 - Gli altri esprimono una proprietà della classe



I Frame

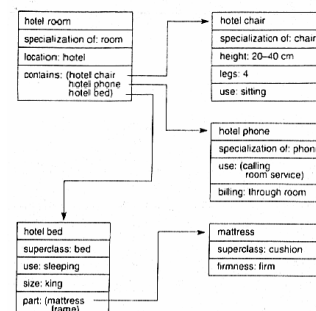
- L'idea è quella di organizzare la conoscenza in strutture mentali complesse (frame) e di collegare tra loro questi frame mediante sistemi di frame
- Essenza della teoria: *“Quando si incontra una situazione nuova, o imprevista, viene evocata dalla memoria una struttura mentale complessa, la quale, mediante un processo di istanziazione, viene adattata alla situazione specifica e fornisce una chiave di interpretazione per essa”*

I frame sono strutture dati...

- Condizioni per l'attivazione
- Collezione di coppie *slot-filler* (attributo-valore)
- I *filler* possono essere di diversi tipi (*face*)
 - valore specifico
 - condizione sul valore, riferimento ad altro *frame*
 - valore default
 - una procedura da attivare quando lo slot riceve un valore (*if-added*) o è richiesto il valore dello slot (*if-needed*) ⇒ *demoni* o *allegati procedurali*
- Slot particolari sono IS e IS-A per organizzazione gerarchica dei frame.

Esempio: una stanza d'albergo

- Frame di una stanza d'albergo è visto come specializzazione di quello della stanza



Un frame particolare: lo Script

- Tipi particolari di *frame* che descrivono situazioni tipiche, sequenze di eventi che normalmente succedono, oggetti e attori normalmente presenti.
- Usati con successo nella comprensione di storie, fornendo un contesto di riferimento.
- Esempio: Script del ristorante:
"Kate andò al ristorante. Fu accompagnata ad un tavolo e ordinò una bistecca alla cameriera. Sedette lì aspettando a lungo. Infine si arrabbiò e uscì"

Le Logiche terminologiche (o descrittive)

Le *Logiche Terminologiche* (LT -- o *Logiche di Descrizione*) possono essere viste come:

1. *Evoluzioni "logiche"* di linguaggi di KR basati sulle nozioni di frame e reti semantiche
2. *Contrazioni* della logica del prim'ordine (FOL) per ottenere migliori proprietà computazionali

Le radici delle LT nelle reti semantiche e nei frame

- I modelli di memoria basati su reti semantiche e *frame* raccolgono due intuizioni fondamentali:
 - *reti semantiche*: la memoria è organizzata secondo un principio di contiguità fisica dell'informazione semanticamente correlata;
 - *frame*: la memoria è organizzata secondo un principio di aggregazione di tutte le informazioni pertinenti a una stessa situazione, o a uno stesso concetto.

Un motivo di interesse...

- Codificare conoscenza sui termini si sta rivelando fondamentale per la prossima generazione dei motori di ricerca (*Semantic Web*)

Formalizzazione

- Il progetto KL-ONE evolve nel filone delle LT [Brachman84]
- Il processo di formalizzazione consiste nel
 - riformulare i costrutti secondo i canoni della logica
 - eliminare i costrutti che non si prestano a tale riformulazione (*default* ed eccezioni)

Espressività delle Logiche Terminologiche

- Le LT sono meno espressive della logica del prim'ordine
 - Non hanno
 - Variabili
 - Quantificatori
 - Connettivi
 - Simboli di funzione (di arietà ≥ 1)
 - Simboli di predicato n-ari (con $n \geq 3$)
 - Predicato di uguaglianza
 - Hanno altri operatori che possono essere visti come combinazioni fisse di essi

Esempio (basato su una delle Logiche Terminologiche)

- La "Mercatone Uno" ha due direttori sportivi
- Con una delle LT...
 - (and squadraDiCiclismo
 - (atmost 2 ds)
 - (atleast 2 ds)) [MercatoneUno]
- In logica...
 - squadraDiCiclismo(MercatoneUno) \wedge
 - $\exists x. ds(\text{MercatoneUno}, x) \wedge$
 - $\exists y. ds(\text{MercatoneUno}, y) \wedge$
 - $x \neq y$
 - $ds(\text{MercatoneUno}, z) \Rightarrow (z=x) \vee (z=y)$

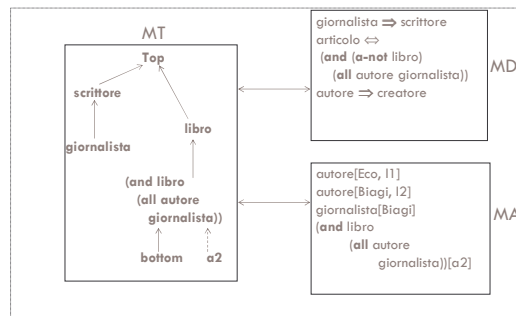
Concetti e Ruoli

- Ogni LT è caratterizzata da operatori per la costruzione di termini di due tipi:
 - *concetti* (corrispondenti a relazioni unarie) con operatori and, or, not ...
 - *ruoli* (corrispondenti a relazioni binarie) con operatori all, some, atleast, atmost, ...

I tre moduli di una LT

- Un sistema KB basato su una logica terminologica è costituito in genere da tre componenti:
 - Un *modulo terminologico* (MT), un linguaggio per la definizione di concetti e ruoli (termini)
 - Un *modulo asserzionale* (MA), per asserzioni *ground* riguardanti questi termini
 - Un *modulo definizionale* (MD), opzionale, per la specifica di relazioni di inclusione fra questi termini
- Il modulo MT è la parte caratterizzante di una LT (MA e MD, se ci sono, sono uguali per tutte le logiche)

Una KB basata su una Logica Terminologica



Problemi decisionali per le LT

- Problemi classici:
 1. Soddisfaccibilità di una KB (KBS)
 2. Conseguenza logica di una KB: il problema di decidere, dato $C[i]$, se $KB \models C[i]$, detto anche instance checking, $IC(KB, C[i])$
- Problemi specifici delle LT:
 - Soddisfaccibilità di un concetto $[CS(C)]$: il problema di decidere se esiste un'interpretazione diversa dall'insieme vuoto

Problemi decisionali per le LT (cont.)

- Sussunzione terminologica, o strutturale (SU):
 - C_1 sussume C_2 $[SU(C_1, C_2)]$
 - sse per ogni interpretazione I , $I(C_2) \subseteq I(C_1)$
 - Es. person sussume (and person (some child))
- Sussunzione ibrida (HSU(KB, C_1, C_2)): come sopra ma si usano anche le definizioni nella KB
 - Es. Se $student \Rightarrow person \in MD$ allora person sussume student

NeoClassic

*Un sistema per la rappresentazione
della conoscenza*

Introduzione

- Classic è una famiglia di sistemi di R.C. basati su Logica Descrittiva (o Terminologica)
- NeoClassic è il più recente membro della famiglia Classic
 - Informazione rappresentata in termini di
 - Descrizioni
 - Concetti
 - Ruoli
 - Regole
 - Individui

Introduzione (cont.)

- Peculiarità di
 - Classic
 - estensione delle nozioni basilari delle logiche terminologiche con regole (di concatenazione in avanti)
 - NeoClassic
 - introduce l'idea di più basi di conoscenza indipendenti

Descrizioni

- Costruite usando espressioni che contengono
 - altre descrizioni (inclusi concetti)
 - ruoli
 - individui
- Non possono usare informazione di altre basi di conoscenza
- Sono applicate ad individui

Descrizioni (cont.)

- Un individuo soddisfa una descrizione se l'informazione conosciuta circa l'individuo implica la descrizione
- Esempio:
 - Descrizione di un uomo con prole
(and UOMO
(atLeast 1 figli))Poiché l'individuo Abramo si sa avere almeno un figlio, questo individuo si dice che soddisfa la descrizione

Descrizioni: *and* e *oneOf*

- *and*
Sintassi: (and Descrizione+)
Esempio: Un addetto al Tempio è un uomo appartenente alla tribù di Levi
(and UOMO LEVITA)
- *oneOf*
Sintassi: (oneOf Individuo+)
Esempio:
(oneOf AnticoTestamento NuovoTestamento)

Descrizioni: *le restrizioni di ruolo*

- *all*
Sintassi: (all Ruolo Descrizione)
Effetto: restrizione in base al tipo
- *atLeast*
Sintassi: (atLeast InteroPositivo Ruolo)
Effetto: limitazione inferiore al numero di riempitori di un ruolo
- *atMost*
Sintassi: (atMost InteroNonNegativo Ruolo)
Effetto: limitazione superiore al numero di riempitori di un ruolo
- *fills*
Sintassi: (fills Ruolo Individuo+)
Effetto: specifica i riempitori (attuali) di un ruolo

Concetti

- Descrizione denominata
- Fanno parte di una base di conoscenza e non devono usare informazione di altre basi di conoscenza
- Esistono due tipi di concetti
 - Primitivi
 - Definito


Concetti primitivi

- Definiamo un personaggio come un qualcuno che sappiamo avere, tra le altre cose, un nome ed un periodo in cui è vissuto:
(createConcept PERSONAGGIO
(all nome STRING)
(all periodo-storico INTEGER) true)
- nome* e *periodo-storico* sono ruoli precedentemente definiti
- PERSONAGGIO fornisce condizioni di appartenenza necessarie ma non sufficienti

Concetti definiti

- Definizione del concetto PROFETA:
(createConcept PROFETA
(and PERSONAGGIO
(fills categoria profeta)))
- La condizione di appartenenza è necessaria e sufficiente
 - ogni PERSONAGGIO con la categoria riempita da profeta sarà classificato come PROFETA
 - ogni PROFETA sarà classificato come PERSONAGGIO ed avrà il ruolo categoria riempito da profeta

Ruoli

- Rappresentano le proprietà degli individui
- Attributi
 - hanno al massimo un riempitore
- Sintassi:*
(createRole NomeDelRuolo Boolean)

- Ruolo chiuso su un individuo: il ruolo, su quell'individuo, non può avere altri riempitori (oltre a quelli già specificati)
(closeRole Individuo Ruolo)

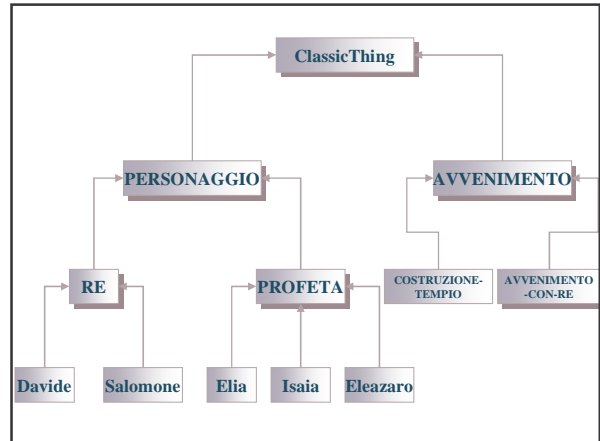
Esempio

- Definiamo i concetti di
 - PERSONAGGIO
 - PROFETA
 - RE
 - AVVENIMENTO
 - AVVENIMENTO-CON-RE
- Definiamo il ruolo di
 - protagonista
- Definiamo l'individuo
 - Costruzione-Tempio
- Analizziamo la tassonomia creata

```

(createConcept PERSONAGGIO ClassicThing true)
(createConcept AVVENIMENTO ClassicThing true)
(createConcept PROFETA (and PERSONAGGIO
  (oneOf Elia Isaia Eleazaro)))
(createConcept RE (and PERSONAGGIO
  (oneOf Davide Salomone)))
(addToldInformation Elia PROFETA)
(addToldInformation Isaia PROFETA)
(addToldInformation Eleazaro PROFETA)
(addToldInformation Davide RE)
(addToldInformation Salomone RE)
(createRole protagonista)
(createConcept AVVENIMENTO-CON-RE (and AVVENIMENTO
  (all protagonista RE)))
(createIndividual Costruzione-Tempio AVVENIMENTO)

```



Aggiungere informazione

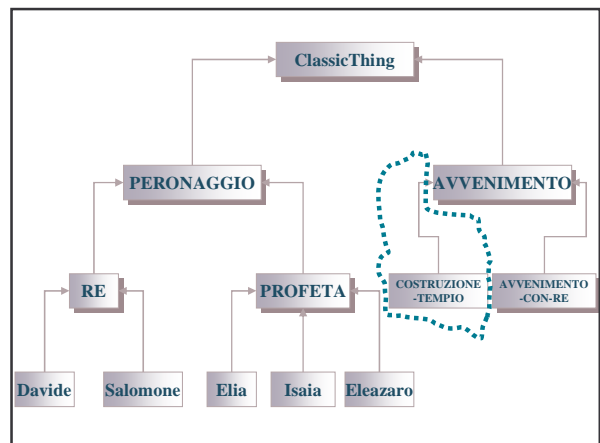
All'individuo Costruzione-Tempio è possibile aggiungere informazione nel modo seguente:

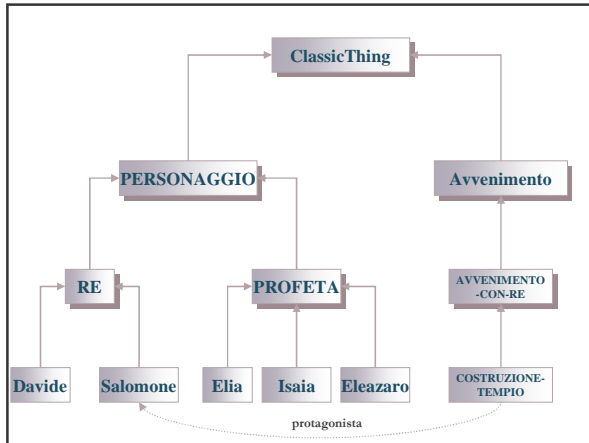
```

(addToldInformation Costruzione-Tempio
  (fills protagonista Salomone))

```

La tassonomia sarà aggiornata classificando Costruzione-Tempio come istanza di AVVENIMENTO-CON-RE.





Chiusura di un ruolo

- Il Tempio di Gerusalemme è stato costruito sotto il regno di Salomone; come specificarlo?
 - Usando `atMost`

```
(createConcept PROTAGONISTA-UNICO
  (and AVVENIMENTO
    (atMost 1 protagonista)))
```
 - Usando `closeRole`

```
(closeRole Costruzione-Tempio protagonista)
```

NeoClassic e C++

- Funzioni C++ che permettono di determinare se un individuo soddisfa una descrizione; può ritornare i seguenti valori:
 - `testFalse` *,inconsistente con la descrizione*
 - `testTrue` *,soddisfa categoricamente la descrizione*
 - `testMaybe` *,soddisfa la descrizione, ma un'aggiunta di informazione può "cambio di stato" provocare una*

NeoClassic e C++: un esempio

- Per trovare i nemici degli ebrei...
 - Definizione di un nuovo ruolo


```
(createRole opinione-autore)
```
 - Uso di `checkNemico` (procedura C++)
 - Definizione di un nuovo concetto


```
(createConcept NEMICO
            (and PERSONAGGIO
              (testC checkNemico opinione-autore))
```

Regole

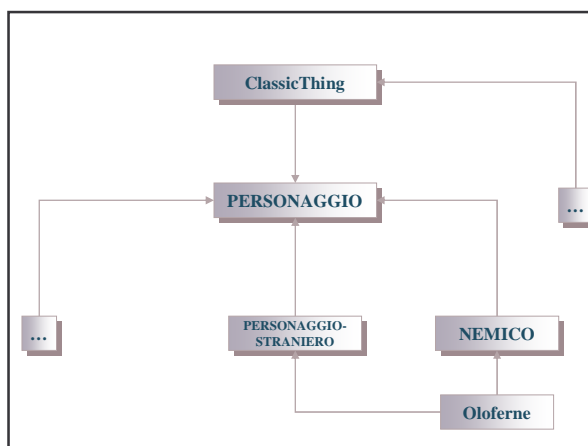
- NeoClassic è deduttivo
- Tre tipi di regole (di concatenazione in avanti)
 - semplici
 - con descrizione calcolata
 - con riempitore calcolato

Regole semplici

- Sono formate da
 - antecedente (deve essere un concetto)
 - conseguente (deve essere una descrizione)
- Sono attivate (fire) per ogni individuo I che soddisfa l'antecedente; in tal caso I si deduce che soddisferà anche il conseguente.

Esempio: uso di una regola semplice

- Ipotesi: un nemico degli ebrei è un egiziano o un babilonese
- Estendiamo l'esempio precedente
 - (createRole popolo-di-appartenenza true)
 - (createRule regola-uno NEMICO (all popolo-di-appartenenza (oneOf Egiziano Babilonese)))
 - (createConcept PERSONAGGIO-STRANIERO (and PERSONAGGIO (all popolo-di-appartenenza (oneOf Egiziano Babilonese Etiope))))
- Creiamo l'individuo...
 - (createIndividual Oloferne PERSONAGGIO)
- Nuova tassonomia...



Regole con descrizione calcolata

- Il conseguente non è specificato al momento della definizione della regola
 - si usa una funzione con parametri che genera, al momento dell'attivazione della regola, il conseguente
 - il conseguente si basa sulle informazioni a disposizione al momento dell'attivazione della regola

Regole con riempitore calcolato

- Simili a quelle con descrizione calcolata
- Oltre ad una funzione prendono un ruolo
 - al momento dell'attivazione, la funzione genera una lista di riempitori per il ruolo

Inferenza in NeoClassic

- NeoClassic supporta tre meccanismi
 - applicazione di regole
 - completamento
 - classificazione e sussunzione

Completamento

- Ereditarietà
 - le restrizioni che si applicano ad un concetto devono applicarsi anche alle sue specializzazioni
- Combinazione
 - ...logica di restrizioni applicate a concetti ed individui al fine di ottenere restrizioni più stringenti
- Propagazione
 - l'informazione associata ad un individuo viene propagata agli altri in relazione con esso
- Individuazione di contraddizioni
 - capacità di riconoscere affermazioni contraddittorie riguardo ad un individuo

Classificazione e Sussunzione

- **Classificazione di concetti**
 - possono essere individuati concetti più generali e/o più specifici di un certo concetto
- **Classificazione di individui**
 - possono essere individuati tutti i concetti che un certo individuo soddisfa
- **Sussunzione**
 - è possibile individuare se un concetto sia più generale o più specifico di un altro
 - usata nella *classificazione di concetti*

Considerazioni

- Orientato agli oggetti
- Terminologico
 - adatto alla descrizione di oggetti complessi
 - NON adatto a gestire asserzioni complesse (con quantificatori e disgiunzioni)
- Deduttivo
- Incrementale
- Ritrattazione della conoscenza
 - tiene traccia delle dipendenze tra fatti
 - permette che certi fatti siano ritrattati
- Supporta le regole
- Supporta test procedurali

Quando NeoClassic è adatto

- Domini con un gran numero di oggetti naturalmente rappresentabili in termini di caratteristiche e ruoli
- Applicazioni di recupero di informazione
 - se gli oggetti hanno una descrizione complessa
 - se le interrogazioni si presentano come descrizioni di oggetti con una certa struttura
- Applicazioni su basi di dati con schema che evolve nel tempo
- Applicazioni che coinvolgono descrizioni che evolvono incrementalmente nel tempo, a partire da descrizioni parziali
- Asserire a quanti oggetti un individuo è associato

Quando NeoClassic è adatto (cont.)

- Descrivere i riempitori dei ruoli senza conoscerli
 - esempio: “tutti i riempitori del ruolo popolo-di-appartenenza sono cittadinanze”
- Informazione raffinabile
 - esempio: la cittadinanza di un personaggio era inizialmente straniera, più tardi si scopre essere egiziana
- Individui riclassificabili
 - Costruzione-Tempio era classificato come AVVENIMENTO, dopo è stato classificato come AVVENIMENTO-CON-RE

Quando NeoClassic NON è adatto

- Applicazioni che richiedono manipolazione di entità matematiche
- Semplice recupero di informazioni da basi di dati
- Applicazioni che richiedono condizioni complesse nell'antecedente delle regole
- Applicazioni che richiedono aggiornamenti frequenti della KB

Bibliografia

- Appunti e lucidi delle lezioni di Intelligenza Artificiale 1 (Simi)
- NeoClassic user's guide version 1.0 (Weixelbaum, Abrahams, Borgida)
- NeoClassic, un sistema per la rappresentazione della conoscenza (Giovannetti)