LOGICHE DESCRITTIVE

NOTA BENE.
Le pagine seguenti sono sostanzialmente tratte dal secondo capitolo di The Description Logic Handbook, disponibile al seguente link http://www.inf.unibz.it/%7Efranconi/dl/course/dlhb/dlhb-02.pdf.
Vogliono solo costituire una traccia degli argomenti sull'argomento trattati a lezione.

Le logiche descrittive sono una famiglia di formalismi atti a rappresentare le conoscenze su un dato dominio definendo prima i concetti relativi al dominio (terminologia) e poi usando questi concetti per specificare proprietà degli individui appartenenti al dominio (asserzioni).
Come tutte le logiche hanno una "parte" sintattica ed una "parte" semantica e sono caratterizzate dalla presenza di servizi di reasoning che permettono di inferire conoscenze implicitamente racchiuse dalle conoscenze esplicitamente contenute nella base di conoscenza.
Ovviamente è importante che questi formalismi di rappresentazione della conoscenza siano in grado di rispondere in tempi ragionevoli alle query di un utente e perciò è importante (anche se non sempre indispensabile per avere una logica di interesse pratico) che i problemi di inferenza siano decidibili e appartengano ad una classe di bassa complessità computazionale.
Tuttavia logiche descrittive con efficienti procedure di reasoning sono spesso poco espressive, per cui è importante trovare un buon trade off fra espressività e complessità di logiche descrittive.
Un sistema di rappresentazione di conoscenza ( che di seguito indicheremo con KR) basato sulle logiche descrittive ha una base di conoscenze (KD) costituita da due componenti: il Tbox che introduce la terminologia e l'Abox che contiene le asserzioni sugli individui.
Il linguaggio delle logiche descrittive per la parte terminologica è basato su concetti atomici, spesso indicati con A e B, ruoli atomici, indicati con R, e costruttori che servono a comporre concetti e ruoli atomici secondo precise regole sintattiche per formare i concetti composti o descrizione di concetti. Per le asserzioni servono anche le costanti. Le logiche descrittive si distinguono una dall'altra sulla base dei costruttori ammessi dal linguaggio. Oltre alla KB il sistema offre servizi per fare inferenza su di essa ed è inserito in un contesto più ampio che dispone di regole che ad esempio permettono di decidere se e come aggiornare la terminologia etc.

Per i Tbox i principali servizi di reasoning sono:
- determinare se una descrizione è non contraddittoria
- determinare se una descrizione è più generale di un'altra
- determinare se una descrizione è equivalente ad un'altra

per gli Abox sono:
- decidere se l'insieme delle asserzioni è consistente
- decidere se un dato individuo è un'istanza di un dato concetto
- descrivere tutti gli oggetti che sono istanza di un dato concetto.

Il linguaggio di qualche interesse pratico col minor numero di costruttori è $\mathcal{AL}$ (linguaggio attributivo).
Le descrizioni di concetti di $\mathcal{AL}$ sono definiti con le seguenti regole:
- ogni concetto atomico è una descrizione di concetto
- $\bot$ e $\top$ sono descrizioni di concetti (detti rispettivamente concetto vuoto e concetto universale)
- se A è un concetto atomico, $\neg$A è una descrizione di concetto (negazione di concetti atomici)
- se C e D sono descrizioni di concetto, C$\sqcap$D è una descrizione di concetto (intersezione o congiunzione di concetti)

- se R è un ruolo (atomico) e C è una descrizione di concetto, $\forall$R.C è una descrizione di concetto (quantificazione universale qualificata)
- se R è un ruolo (atomico), $\exists$R.$\top$ è una descrizione di concetto (quantificazione esistenziale limitata)
- niente altro è una descrizione di concetto.

Il linguaggio della logica $\mathcal{AL}$ si può ampliare ammettendo ulteriori costruttori:

($\mathcal{U}$) unione di concetti: se C e D sono descrizione di concetti, C $\sqcup$ D è una descrizione di concetto

($\mathcal{C}$) negazione di concetti arbitrari: se C è una descrizione di concetti, $\neg$C è una descrizione di concetto

($\mathcal{E}$) quantificazione esistenziale qualificata: se R è un ruolo e C è una descrizione di concetto, $\exists$R.C è una descrizione di concetto

($\mathcal{N}$) restrizioni numeriche: se R è un ruolo ed n è un intero naturale $\geq$nR e $\leq$nR sono descrizioni di concetto.

Aggiungendo ad $\mathcal{AL}$ un insieme qualunque dei precedenti costruttori si ottiene un linguaggio della famiglia $\mathcal{AL}$, il cui nome si ottiene facendo seguire $\mathcal{AL}$ dai nomi dei costruttori introdotti.

Vediamo ora come dare una semantica a queste famiglie di logiche descrittive.
Dobbiamo introdurre una *interpretazione* $\mathcal{I} = <\Delta^{\mathcal{I}}, \bullet^{\mathcal{I}}>$, dove
$\Delta^{\mathcal{I}}$ è un insieme non vuoto detto dominio dell'interpretazione e
$\bullet^{\mathcal{I}}$ è una funzione di interpretazione che assegna

- ad ogni concetto atomico A un sottoinsieme $A^{\mathcal{I}}$ di $\Delta^{\mathcal{I}}$
- ad ogni ruolo atomico R una relazione binaria $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ su $\Delta^{\mathcal{I}}$
- ad ogni costante a un elemento $a^{\mathcal{I}}$ di $\Delta^{\mathcal{I}}$.

Questa funzione di interpretazione può essere estesa alle descrizione di concetti nel seguente modo:

- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
- $\bot^{\mathcal{I}} = \emptyset$
- $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} - C^{\mathcal{I}}$
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
- $(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \Rightarrow b \in C^{\mathcal{I}}\}$
- $(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b: (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- $(\leq nR)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid |\{b|(a,b) \in R^{\mathcal{I}}\}| \leq n\}$
- $(\geq nR)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid |\{b|(a,b) \in R^{\mathcal{I}}\}| \geq n\}$.

Due concetti C e D i dicono *equivalenti* e si scrive C$\equiv$D se per ogni interpretazione $\mathcal{I}$ si ha $C^{\mathcal{I}} = D^{\mathcal{I}}$.
Dalla semantica segue subito la seguente
Proposizione: Le logiche $\mathcal{ALUE}$ e $\mathcal{ALC}$ hanno la stessa espressività.
Infatti se abbiamo i costrutti $\mathcal{U}$ ed $\mathcal{E}$ possiamo esprimere $\mathcal{C}$ per ricorsione in questo modo

- se C è un concetto atomico o una sua negazione abbiamo subito che $\neg$C è una descrizione di concetto
- se C è una descrizione di concetto della forma D$\sqcap$E allora $\neg$C è equivalente a $\neg$D $\sqcup$ $\neg$E
- se C è una descrizione di concetto della forma $\forall$R.D allora $\neg$C è equivalente a $\exists$R.$\neg$D
- se C è una descrizione di concetto della forma $\exists$R.D allora $\neg$C è equivalente a $\forall$R.$\neg$D

come facilmente si può osservare nelle formule equivalenti figura spesso una (o più) descrizione di concetto negata, tuttavia si tratta di una descrizione più semplice di quella da cui si era partiti e dunque in un numero finito di passi si riporta la negazione sui concetti atomici

Viceversa se abbiamo il costruttore $\mathcal{C}$ possiamo esprimere facilmente $\mathcal{U}$ ed $\mathcal{E}$, infatti $C \sqcup D$ è equivalente a $\neg(\neg C \sqcap \neg D)$ e $\exists R.C$ è equivalente a $\neg(\forall R. \neg C)$.

La precedente semantica mette in evidenza anche uno stretto legame fra le logiche descrittive e le logiche del I ordine, infatti una interpretazione associa ad ogni concetto atomico un predicato unario e ad ogni ruolo atomico un predicato binario. Quindi ogni concetto A può essere visto come una formula atomica $\mathcal{A}(x)$ ed ogni ruolo R come una formula atomica $\mathcal{R}(x,y)$, ed è immediato vedere a che formule corrispondono i concetti che coinvolgono solo costrutti di intersezione, unione e negazione (che vengono tradotti nell'omonimo connettivo logico). Il costrutto $\exists R.C$ viene tradotto nella formula $\exists y(\mathcal{R}(x,y) \wedge \mathcal{C}(y))$, mentre il costrutto $\forall R.C$ viene tradotto in $\forall y(\mathcal{R}(x,y) \Rightarrow \mathcal{C}(y))$. Le logiche descrittive in cui sono ammesse restrizioni numeriche possono essere tradotti in formule di logiche del primo ordine con identità, infatti $\leq nR$ può essere tradotta in

$\forall y_1 \forall y_2 \ldots \forall y_{n+1}(\mathcal{R}(x,y_1) \wedge \ldots \wedge \mathcal{R}(x,y_{n+1}) \Rightarrow \bigvee_{i<j} (y_i=y_j))$

mentre $\geq nR$ si traduce in

$\exists y_1 \exists y_2 \ldots \exists y_n(\mathcal{R}(x,y_1) \wedge \ldots \wedge \mathcal{R}(x,y_n) \bigwedge_{i<j} (y_i \neq y_j))$.

Va osservato che mentre ogni descrizione di concetto della famiglia $\mathcal{AL}$ può essere espressa come formula del I ordine il viceversa non vale, quindi stiamo parlando di un frammento delle logiche del I ordine, la scelta di introdurre un nuovo linguaggio, invece di servirsi direttamente di un linguaggio del I ordine, è dovuta alla sua maggiore concisione che facilita lo sviluppo di algoritmi.
Il frammento che viene preso in considerazione è un frammento decidibile, infatti $\mathcal{ALC}$ ad esempio corrisponde ad un linguaggio logico del I ordine che non fa uso di lettere funzionali e utilizza solo due variabili ed è stato dimostrato che tale linguaggio è decidibile, inoltre sempre $\mathcal{ALC}$ può essere vista come variante sintattica della logica preposizionale multimodale K che pure è noto essere decidibile. Infatti basta introdurre per ogni ruolo R un connettivo modale [R] e tradurre le formule del tipo $\forall R.C$ in $[R]C$ ed $\exists R.C$ in $\langle R \rangle C$.

## *Tbox*

Gli *assiomi terminologici* possono assumere una delle seguenti forme:
$C \equiv D$ (uguaglianza), $C \sqsubseteq D$ (inclusione) ($R \equiv S$ , $R \sqsubseteq S$).
Notate che nell'ambito della famiglia delle logiche $\mathcal{AL}$ assiomi che stabiliscono uguaglianze o inclusioni di relazioni non occorrono in quanto non sono stati introdotti costruttori per formare relazioni non atomiche.
Una interpretazione $\mathcal{I}$ *soddisfa una inclusione* $C \sqsubseteq D$ se $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ , soddisfa una uguaglianza $C \equiv D$ se $C^{\mathcal{I}} = D^{\mathcal{I}}$.
$\mathcal{I}$ *soddisfa un insieme di assiomi terminologici* (ovvero è un modello per un insieme di assiomi) se soddisfa ogni assioma dell'insieme.
Due insiemi di assiomi sono *equivalenti* se hanno gli stessi modelli.
Una uguaglianza si dice una *definizione* se la sua parte sinistra è un concetto atomico.

*Esempio*

PERSONA$\sqcap$FEMMINA$\sqcap\exists$HAUNFIGLIO.$\top \equiv$ UMANO$\sqcap\neg$MASCHIO$\sqcap \geq 1$ HAUNFIGLIO
è una uguaglianza ma non è una definizione,
MADRE$\equiv$ PERSONA$\sqcap$FEMMINA$\sqcap\exists$HAUNFIGLIO.$\top$
è una definizione perché la parte sinistra è un concetto atomico e non una descrizione di concetti.

Un insieme di definizioni non deve presentare ambiguità, un insieme di definizioni è una *terminologia* o un *Tbox* se per ogni concetto atomico A c'è al più definizione che ha quel concetto come parte sinistra.

I concetti atomici di un Tbox si possono dividere in simboli di base che occorrono solo nella parte destra delle definizioni e simboli nominali che occorrono nella parte sinistra (non necessariamente solo in quella) delle definizioni, in altre parole i simboli di base sono concetti primitivi mentre i simboli nominali sono concetti definiti.

*Esempio*
L'insieme di definizioni
MADRE≡ UMANO⊓¬MASCHIO⊓≥1 HAUNFIGLIO
MADRE≡ PERSONA⊓FEMMINA⊓∃HAUNFIGLIO.⊤
non è un Tbox.
L'insieme di definizioni
PADRE≡ UMANO⊓MASCHIO⊓≥1 HAUNFIGLIO
MADRE≡ PERSONA⊓FEMMINA⊓∃HAUNFIGLIO.⊤
GENITORE≡PADRE ⊔ MADRE
FEMMINA≡ ¬MASCHIO
è un Tbox in $\mathcal{ALNU}$
I simboli di base di questo Tbox sono UMANO, MASCHIO, PERSONA, HAUNFIGLIO, invece PADRE, MADRE, GENITORE, FEMMINA sono simboli nominali.

Una *interpretazione di base* $\mathcal{I}$ per un Tbox è una interpretazione che interpreta soltanto i simboli di base. Una interpretazione $\mathcal{I}'$ che interpreta tutti i simboli atomici è una *estensione* di $\mathcal{I}$ se ha lo stesso dominio di $\mathcal{I}$ e coincide con $\mathcal{I}$ quando è ristretta ai simboli di base.
Un Tbox $\mathcal{T}$ si dice *definitorio* se ogni interpretazione di base ha un'unica estensione che è modello di $\mathcal{T}$.
Ovviamente se $\mathcal{T}'$ è equivalente a $\mathcal{T}$, $\mathcal{T}'$ è definitorio sse lo è $\mathcal{T}$.
Il fatto che un Tbox sia definitorio è strettamente legato al fatto che siano presenti simboli nominali che richiamano se stessi; per definire correttamente questa nozione introduciamo la relazione di "usare direttamente in $\mathcal{T}$" sui concetti atomici in questo modo: un concetto A usa direttamente B in $\mathcal{T}$ se A è un nome simbolico in $\mathcal{T}$ e nella sua definizione occorre B. Sia "usare in $\mathcal{T}$" la chiusura transitiva di "usare direttamente in $\mathcal{T}$". $\mathcal{T}$ è un Tbox *aciclico* se nessun concetto usa se stesso in $\mathcal{T}$.

*Esempio*
Nel Tbox
PADRE≡ UMANO⊓MASCHIO⊓≥1 HAUNFIGLIO
MADRE≡ PERSONA⊓FEMMINA⊓∃HAUNFIGLIO.⊤
GENITORE≡PADRE ⊔ MADRE
FEMMINA≡ ¬MASCHIO
GENITORE usa direttamente PADRE e MADRE, ed usa UMANO, MASCHIO,PERSONA etc…
Si può facilmente verificare che questo Tbox è aciclico.
Invece il Tbox
PADRE≡ UMANO⊓MASCHIO⊓GENITORE
MADRE≡ PERSONA⊓FEMMINA⊓∃HAUNFIGLIO.⊤
GENITORE≡PADRE ⊔ MADRE
è ciclico perché la definizione di GENITORE usa il concetto GENITORE.

E'chiaro che sussiste la seguente
Proposizione: Un Tbox aciclico è definitorio.
Basta infatti espandere $\mathcal{T}$ sostituendo iterativamante nella parte destra della definizione ogni occorrenza di nome simbolico con la sua definizione e cancellando poi la definizione di quel nome

simbolico, si ottiene così una terminologia $\mathcal{T}'$ equivalente a $\mathcal{T}$ in cui i nomi simbolici non occorrono mai nelle parti destre delle definizioni.

*Esempio*
Il Tbox
PADRE≡ UMANO⊓MASCHIO⊓≥1 HAUNFIGLIO
MADRE≡ PERSONA⊓FEMMINA⊓∃HAUNFIGLIO.⊤
GENITORE≡PADRE ⊔ MADRE
FEMMINA≡ ¬MASCHIO
si può espandere in
GENITORE≡ (UMANO⊓MASCHIO⊓≥1 HAUNFIGLIO) ⊔ ( PERSONA⊓¬MASCHIO⊓∃HAUNFIGLIO.⊤)
che si ottiene sostituendo nella definizione di MADRE il nome simbolico FEMMINA con la sua definizione ¬MASCHIO e poi cancellando l'assioma FEMMINA≡ ¬MASCHIO e successivamente sostituendo nella definizione di GENITORE la definizione di PADRE e la nuova definizione di MADRE e cancellando gli assiomi che definiscono PADRE e MADRE. Nel nuovo Tbox espanso nessun simbolo nominale occorre nella parte destra delle definizioni.

Appare abbastanza evidente che un Tbox ciclico è spesso non definitorio, pensiamo ad esempio al seguente Tbox
UMANO≡ ANIMALE ⊓∃HAGENITORE.UMANO
i simboli di base sono ANIMALE e HAGENITORE, ma una volta interpretati in modo naturale questi due simboli, possiamo estendere in vari modi l'interpretazione (ad esempio umano può avere il suo significato naturale o essere interpretato come insieme dei cani o come insieme dei gatti, etc…).
Tuttavia il viceversa del precedente enunciato non vale (basta considerare un Tbox formato dall'unico assioma A≡B⊓∃R.(A ⊔ ¬A), ovviamente si tratta di un Tbox ciclico, ma essendo A ⊔ ¬A equivalente a ⊤ si intuisce subito che l'interpretazione di A dipende solo da quella di B ed R). Una specie di inversione debole è data dalla seguente
Proposizione: Ogni Tbox definitorio del linguaggio $\mathcal{ALC}$ è equivalente ad un Tbox aciclico.

Per Tbox essenzialmente ciclici (ovvero per Tbox che non ammettono alcun Tbox equivalente non ciclico) la semantica che abbiamo appena introdotto (e che viene detta semantica descrittiva) non è adatta, viene quindi utilizzata la semantica del punto fisso che non tratteremo.

Fino ad ora ci siamo limitati a considerare terminologie i cui unici assiomi sono uguaglianze fra concetti , ma potremmo anche avere assiomi terminologici che sono inclusioni, le inclusioni le cui parti sinistre sono concetti atomici si dicono *specializzazioni*.
Ad esempio NONNA⊑DONNA⊓∃HAUNFIGLIO.⊤ è una specializzazione.

Un insieme di assiomi terminologici si dice una terminologia generalizzata se la parte sinistra di ogni assioma è un concetto atomico e per ogni concetto atomico c'è al più un assioma che lo ammette come parte sinistra (N.B. Le terminologie generalizzate possono contenere specializzazioni).
Ogni terminologia generalizzata può essere trasformata in una terminologia contenente solo definizioni, introducendo per ogni specializzazione A⊑C un nuovo simbolo di base $\overline{A}$ e sostituendo A⊑C con A≡ $\overline{A}$ ⊓C. Ad esempio si può trasformare la specializzazione
NONNA⊑DONNA⊓∃HAUNFIGLIO.⊤ nell'uguaglianza NONNA≡ $\overline{N}$ ⊓DONNA⊓∃HAUNFIGLIO.⊤, dove $\overline{N}$ è un nuovo simbolo di base.
Si ottiene così dalla terminologia generalizzata $\mathcal{T}$ una terminologia composta di sole uguaglianze $\mathcal{T}^*$ detta normalizzazione di $\mathcal{T}$. Questa terminologia è equivalente a $\mathcal{T}$.

Gli assiomi di inclusione che possono essere visti come conoscenze incomplete non aggiungono espressività alle terminologie.
Tuttavia avere informazioni su possibili inclusioni introduce una gerarchia di concetti che può rendere più efficienti gli algoritmi di reasoning.

## *Abox*

Gli Abox contengono asserzioni di uno dei due tipi C(a), R(a,b) con a,b individui e C ed R rispettivamente concetti e ruoli del nostro linguaggio descrittivo. Può cioè contenere asserzioni del tipo
MADRE(SANDRA), DOCENTE(SANDRA), HAFIGLIA(SANDRA,GIULIA) etc…
Gli Abox possono essere visti come una istanza di un data base relazionale che contenga solo predicati binari o unari, ma a differenza della semantica del mondo chiuso solitamente utilizzata per i data base, per gli Abox si utilizza la *semantica del mondo aperto* per cui non si sa nulla sulle asserzioni non contenute negli Abox.
Si adotta invece, se non differentemente precisato, in analogia a quanto avviene nelle basi di dati l'ipotesi di *unicità del nome*, in altre parole per due individui a, b con nomi diversi si ha in ogni interpretazione $\mathcal{I}$ $a^{\mathcal{I}} \neq b^{\mathcal{I}}$.
Una interpretazione $\mathcal{I}$ *soddisfa un'asserzione* C(a) (R(a,b)) se $a^{\mathcal{I}} \in C^{\mathcal{I}}$ ($(a^{\mathcal{I}},b^{\mathcal{I}}) \in R^{\mathcal{I}}$), *soddisfa un* Abox se soddisfa tutte le asserzioni dell'Abox. In tal caso $\mathcal{I}$ si dice anche modello dell'Abox, infine si dice cha $\mathcal{I}$ *soddisfa un* Abox $\mathcal{A}$ *rispetto ad un* Tbox $\mathcal{T}$ se oltre ad essere un modello per $\mathcal{A}$ è un modello per $\mathcal{T}$.

Osservazione.
In alcune logiche descrittive sono permessi costruttori che portano ad inserire le costanti anche nei Tbox; i più importanti di questi costruttori sono

- il costruttore "insieme" che viene scritto come $\{a_1,a_2,\ldots,a_n\}$ ed interpretato come $\{a_1,a_2,\ldots,a_n\}^{\mathcal{I}} = \{a_1^{\mathcal{I}},a_2^{\mathcal{I}},\ldots,a_n^{\mathcal{I}}\}$
- il costruttore "fill" scritto come R:a ed interpretato come $(R{:}a)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} | (d,a^{\mathcal{I}}) \in R^{\mathcal{I}}\}$.

Notiamo inoltre che sono possibili anche costruttori fra i ruoli come la composizione di due ruoli, l'introduzione dell'inversione di un ruolo, etc…Logiche che permettono tali costruttori saranno introdotte ad esercitazioni.

Una base di conoscenza formata da Tbox e Abox è equivalente ad un insieme di assiomi propri di una teoria del I ordine e quindi come nella teoria del I ordine dagli assiomi ricavavamo i teoremi della teoria (che sono formule vere in ogni modello della teoria) utilizzando le regole di inferenza, qui dovremo stabilire tecniche di inferenza che ci permettano di ricavare "teoremi" ovvero conoscenze implicitamente contenute nella nostra base di conoscenza.
Nel seguito quindi tratteremo le possibili inferenze riguardanti Tbox e Abox.

Le più importanti inferenze su un Tbox $\mathcal{T}$ sono:

- *soddisfacibilità*: un concetto C è soddisfacibile rispetto a $\mathcal{T}$ se esiste un modello $\mathcal{I}$ di $\mathcal{T}$ tale che $C^{\mathcal{I}}$ non sia vuoto, o in altre parole se c'è un modello di $\mathcal{T}$ che è anche modello per C.
- *sussunzione*: un concetto C è sussunto da un concetto D (o D sussume C) rispetto a $\mathcal{T}$ se $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ per ogni modello $\mathcal{I}$ di $\mathcal{T}$. Scriveremo in tal caso $\mathcal{T} \models C \sqsubseteq D$ o $C \sqsubseteq_{\mathcal{T}} D$.
- *equivalenza*: un concetto C è equivalente ad un concetto D rispetto a $\mathcal{T}$ se $C^{\mathcal{I}} = D^{\mathcal{I}}$ per ogni modello $\mathcal{I}$ di $\mathcal{T}$. Scriveremo in tal caso $\mathcal{T} \models C \equiv D$ o $C \equiv_{\mathcal{T}} D$.
- *disgiunzione*: un concetto C è disgiunto da un concetto D rispetto a $\mathcal{T}$ se $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ per ogni modello $\mathcal{I}$ di $\mathcal{T}$.

Ad esempio nel Tbox

PADRE≡ UMANO⊓MASCHIO⊓≥1 HAUNFIGLIO
MADRE≡ PERSONA⊓FEMMINA⊓∃HAUNFIGLIO.⊤
GENITORE≡PADRE ⊔ MADRE
FEMMINA≡ ¬MASCHIO
UMANO sussume PADRE, FEMMINA e MASCHIO sono disgiunti etc…

Le precedenti nozioni possono essere date rispetto ad un Tbox vuoto e allora trascuriamo la dicitura "rispetto a $\mathcal{T}$".

L'inferenza principale fornito da KR basati su DL è la sussunzione ed un procedimento di sussunzione ci permette di implementare anche le altre inferenze, infatti abbiamo il seguente
Teorema: Siano C e D due concetti:

- C è insoddisfacibile (rispetto a $\mathcal{T}$) se e solo se è sussunto da ⊥ (rispetto a $\mathcal{T}$),
- C e D sono equivalenti (rispetto a $\mathcal{T}$) sse C è sussunto da D e D è sussunto da C (rispetto a $\mathcal{T}$),
- C e D sono disgiunti (rispetto a $\mathcal{T}$) sse C⊓D è sussulto da ⊥ (rispetto a $\mathcal{T}$).

Questa riduzione degli altri meccanismi di reasoning alla sussunzione può essere fatta già in $\mathcal{AL}$ perché il linguaggio di $\mathcal{AL}$ permette la intersezione di concetti e prevede il concetto ⊥.
In linguaggi in cui è permessa la negazione di concetti anche non atomici tutti i precedenti meccanismi di reasoning possono essere ridotti alla soddisfacibilità. Abbiamo infatti
Teorema: Siano C e D due concetti:

- C è sussunto da D (rispetto a $\mathcal{T}$) sse C⊓¬D è insoddisfacibile (rispetto a $\mathcal{T}$),
- C e D sono equivalenti (rispetto a $\mathcal{T}$) sse sia C⊓¬D sia ¬C⊓D sono insoddisfacibili (rispetto a $\mathcal{T}$),
- C e D sono disgiunti (rispetto a $\mathcal{T}$) sse C⊓D è insoddisacibile (rispetto a $\mathcal{T}$).

Inoltre l'insoddisfacibilità è un caso particolare degli altri meccanismi, infatti abbiamo
Teorema: Per un concetto C sono equivalenti:

- C è insoddisfacibile (rispetto a $\mathcal{T}$),
- C è sussunto da ⊥ (rispetto a $\mathcal{T}$),
- C e ⊥ sono equivalenti (rispetto a $\mathcal{T}$),
- C e ⊥ sono disgiunti (rispetto a $\mathcal{T}$).

Questi teoremi permettono di dire che un upper bound per la complessità della sussunzione è un upper bound per ogni altra inferenza e un lower bound per la complessità della insoddisfacibilità è un lower bound per tutte le altre inferenze.

Osserviamo inoltre che se $\mathcal{T}$ è un Tbox aciclico le inferenze rispetto ad un Tbox possono essere trasformate in inferenza rispetto a un Tbox vuoto, e questa riduzione semplifica lo sviluppo delle procedure di reasoning.
Abbiamo già visto che un Tbox aciclico $\mathcal{T}$ può essere espanso ad un Tbox $\mathcal{T}'$ in cui i simboli nominali non occorrono nelle parti destre delle definizioni tale che $\mathcal{T}'$ è equivalente a $\mathcal{T}$, a questo punto ogni concetto C può essere espanso in un concetto C' che contiene solo simboli di base, infatti basta sostituire ogni simbolo nominale che occorre in C con la sua definizione in $\mathcal{T}'$.
E' immediato osservare che

- $C \equiv_{\mathcal{T}} C'$,
- C è soddisfacibile rispetto a $\mathcal{T}$ sse C' è soddisfacibile,
- $\mathcal{T} \models C \sqsubseteq D$ sse $\models C' \sqsubseteq D'$,

- $\mathcal{T}\models C\equiv D$ sse $\models C'\equiv D'$,
- C e D sono disgiunti rispetto a $\mathcal{T}$ sse C' e D' sono disgiunti.

Vediamo ora un algoritmo di sussunzione.
Partiamo dal caso del linguaggio $\mathcal{FL}_0$ che è una restrizione di $\mathcal{AL}$ ottenuta eliminando la negazione dei concetti atomici e la quantificazione esistenziale (non qualificata) e $\bot$.
L'algoritmo parte da concetti in forma normale ovvero nella forma
$C\equiv A_0\sqcap A_1\sqcap\ldots\sqcap A_m\sqcap\forall R_1.C_1\sqcap\ldots\sqcap\forall R_n.C_n$
dove $A_i$ ed $R_j$ sono concetti e ruoli atomici distinti e $C_j$ sono concetti in forma normale. E' facile osservare che ogni concetto può essere portato in forma normale usando associatività, commutatività e idempotenza di $\sqcap$ e l'equivalenza fra $\forall R.(C\sqcap D)$ e $\forall R.C\sqcap\forall R.D$.
A questo punto si ha il seguente
Teorema: Siano $A_0\sqcap A_1\sqcap\ldots\sqcap A_m\sqcap\forall R_1.C_1\sqcap\ldots\sqcap\forall R_n.C_n$ e $B_0\sqcap B_1\sqcap\ldots\sqcap B_k\sqcap\forall S_1.D_1\sqcap\ldots\sqcap\forall S_h.D_h$ le forme normali dei due concetti C e D rispettivamente. Allora $C\sqsubseteq D$ se e solo se

- per ogni i ($1\leq i\leq k$) esiste j ($1\leq j\leq m$) tale che $B_i= A_j$
- per ogni i ($1\leq i\leq h$) esiste j ($1\leq j\leq n$) tale che $S_i= R_j$ e $C_j\sqsubseteq D_i$

Se si estende $\mathcal{FL}_0$ con $\bot$, diciamo che un concetto è in forma normale se è $\bot$ oppure ha la forma
$A_0\sqcap A_1\sqcap\ldots\sqcap A_m\sqcap\forall R_1.C_1\sqcap\ldots\sqcap\forall R_n.C_n$
dove $A_i$ sono concetti atomici distinti fra loro e da $\bot$ ed $R_j$ sono ruoli atomici distinti e $C_j$ sono concetti in forma normale. Ovviamente anche in questo caso è facile ottenere dalla descrizione di un concetto la forma normale procedendo come prime e ricordandosi che se compare fra i concetti in congiunzione un $\bot$ allora l'intero concetto è $\bot$ e l'algoritmo di sussunzione è del tutto analogo al precedente tenendo conto che $\bot$ è sussunto da ogni concetto.
Se si permette anche la negazione di concetti atomici la forma normale ammette la presenza di letterali (ovvero di concetti atomici e loro negazioni) in congiunzione ed in tal caso bisogna solo prestare attenzione al fatto che se A e $\neg$A occorrono allo stesso livello vanno sostituiti da $\bot$.
Infine se lavoriamo su $\mathcal{ALN}$ si procede ancora nello stesso modo tenendo conto che in congiunzione possiamo avere anche dei concetti della forma $\geq nR$ e $\leq mR$. Bisogna allora tener conto che se occorrono allo stesso livello $\geq nR$ e $\leq mR$ con $n>m$ oppure $\geq nR$ con $n>0$ e $\forall R.\bot$ questi vanno sostituiti con $\bot$, inoltre nell'algoritmo di sussunzione bisogna tener conto che $\geq nR\sqsubseteq\geq mR$ sse $n\geq m$.
Algoritmi di questo genere non possono trattare la sussunzione in linguaggi che ammettono il costruttore di disgiunzione $\sqcup$, la negazione di un qualsiasi concetto e la quantificazione esistenziale qualificata.
Per tali linguaggi si usano algoritmi basati sull'insoddisfacibilità e quindi su tableaux che verranno presentati ad esercitazioni.

Trattiamo ora le inferenze su un Abox $\mathcal{A}$ . Centrale è il concetto di consistenza di $\mathcal{A}$. $\mathcal{A}$ è consistente rispetto ad un Tbox $\mathcal{T}$ se esiste una interpretazione che è un modello di $\mathcal{A}$ e di $\mathcal{T}$, è consistente se è consistente rispetto al Tbox vuoto. E' abbastanza semplice trovare un esempio di Abox consistente che non è consistente rispetto ad un dato Tbox. Basta considerare l'Abox formato dalle asserzioni MADRE(SANDRA), PADRE(SANDRA), si tratta di un Abox consistente perché si può sempre trovare una interpretazione in cui madre e padre non siano concetti disgiunti, ma l'Abox non è consistente rispetto ad un Tbox del tipo
GENITORE≡∃HAUNFIGLIO.⊤
PADRE≡GENITORE⊓MASCHIO
MADRE≡GENITORE⊓FEMMINA
FEMMINA≡¬MASCHIO

perché padre e madre sono concetto disgiunti in tale Tbox e non possono avere un elemento in comune.
Se il Tbox $\mathcal{T}$ è aciclico in modo del tutto analogo a quanto fatto per i Tbox possiamo ridurre la verifica della consistenza di un Abox $\mathcal{A}$ rispetto a $\mathcal{T}$ alla verifica della consistenza di una estensione $\mathcal{A}$' di $\mathcal{A}$ rispetto al Tbox vuoto.
Su un Abox normalmente vengono fatte delle query sulle relazioni fra concetti, ruoli e individui. Queste query sono basate sull'instance check, che consiste nel verificare se una asserzione $\alpha$ è implicata da un Abox $\mathcal{A}$ (rispetto a $\mathcal{T}$) ovvero se ogni modello di $\mathcal{A}$ (e di $\mathcal{T}$) soddisfa l'asserzione.
In tal caso scriviamo $\mathcal{A}\models\alpha$ ($\mathcal{A}\models_{T}\alpha$).
Nel caso di asserzioni su ruoli l'instance check è una semplice verifica di appartenenza perché non abbiamo (fino a questo momento) introdotto regole per costruzione di ruoli non atomici, nel caso di asserzioni su concetti, ovvero di asserzioni di tipo C(a), possiamo ridurre la query a una verifica di consistenza infatti abbiamo

- $\mathcal{A}\models C(a)$ sse $\mathcal{A}\cup\{\neg C(a)\}$ è inconsistente

Inoltre la soddisfacibilità di un concetto può essere ridotta alla consistenza dell'Abox, infatti è evidente che

- C è soddisfacibile sse $\{C(a)\}$ è consistente dove a è una costante arbitraria,

la consistenza di un Abox può essere ridotta alla soddisfacibilità di concetti solo in presenza di linguaggi che ammettono i costruttori "insieme" e "fill".
Altre inferenze su Abox sono

- retrieval: trovare dato un Abox $\mathcal{A}$ e un concetto C tutti gli individui a tali che $\mathcal{A}\models C(a)$
- realizzazione: trovare dato un individuo a e un insieme di concetti, i concetti più specifico C (cioè i concetti minimali rispetto alla $\sqsubseteq$ ) per cui $\mathcal{A}\models C(a)$

Osserviamo che il fatto che negli Abox si usi la semantica di mondo aperto rende la risposta alle query più complessa che nei data base, perché un Abox può avere varie (anche infinite) interpretazioni.

Da ultimo accenniamo brevemente alle regole. Le più semplici hanno la forma $C\Rightarrow D$, dove C e D sono concetti. Il significato di queste regole è " se si è provato che un individuo è un'istanza di C allora quell'individuo è un'istanza di D". La presenza di queste regole permette di estendere la base di conoscenze arricchendo ogni Abox in cui sia presente una asserzione C(a) con l'asserzione D(a). Notate che la regola $C\Rightarrow D$ è diversa dall'inclusione $C\sqsubseteq D$, basta pensare alle due inclusioni $C\sqsubseteq D$ e $\neg C\sqsubseteq D$ che implicano che ogni oggetto appartiene a D mentre le due regole $C\Rightarrow D$ e $\neg C\Rightarrow D$ non danno alcuna informazione su individui a per cui non può essere provato C(a) o $\neg$C(a).
Per tradurre una regola $C\Rightarrow D$ in una inclusione si può introdurre un operatore **K** (detto operatore epistemico) che può essere visto come ciò che la base di conoscenza sa sul dominio, in altri termini **K**C per un concetto C indica l'insieme degli oggetti chela base di conoscenza sa appartenere a C. Allora la regola $C\Rightarrow D$ può essere tradotta in $\mathbf{K}C\sqsubseteq D$ dove l'operatore **K** posto davanti a C sta ad indicare che l'assioma si applica solcagli individui che appaiono nell'Abox e per i quali l'Abox e il Tbox dicono che sono istanze di C. Tralasciamo l'introduzione di una semantica formale per le inclusioni che coinvolgono **K** .