Logica

Giacomo Fantoni
Telegram: @GiacomoFantoni
Github: Github Repository

18 settembre 2020

Indice

Capitolo 1

Logica Proposizionale

Una proposizione è una frase riguardante cosa succede nel mondo, possono avere diverse forme ma unicamente vere o false e formano le formule atomiche del linguaggio proposizionale.

1.1 Linguaggio

L'alfabeto si separa in tre componenti:

- Simboli logici: $\neg, \wedge, \vee, \supset, \equiv$.
- Simboli non logici: un insieme $\mathcal{P}\nabla\wr_{\surd}$ di simboli P detto variabili proposizionali.
- Simboli separatori (e).

Si definisce formula ben formata:

- Ogni $P \in \mathcal{P}\nabla\wr_{\surd}$ è una formula atomica.
- Ogni formula atomica è ben formata.
- Se A e B sono formule ben formate allora lo sono anche $\neg A$, $A \wedge B$, $A \vee B$, $A \supset B$, $A \equiv B$.

Si dicono costanti proposizionali le proposizioni con un dato valore di verità. Le variabili proposizionali sono segni che rappresentano una possibile costante proposizionale, non hanno valore di verità definito. I simboli hanno una scala di priorità, in ordine decrescente: $\neg, \wedge, lor, \supset, \equiv$.

1.1.1 Formule come alberi

Una formula può essere considerata come un albero dove i nodi foglia sono variabili o costanti proposizionali mentre gli altri connettivi logici.

1.1.2 Sottoformule

Si definiscono sottoformule come:

- A è sottoformula di sè stessa.

- A e B sono sottoformule di $A \wedge B$, $A \vee B$, $A \supset B$, $A \equiv B$.
- A è sottoformula di $\neg A$.
- Se A è sottoformula di B e B è sottoformula di C allora A è sottoformula di C.
- A si dice sottoformula di B se A è sottoformula di B e A è diversa da B.

Le sottoformule sono tutti i sottoalberi di una formula rappresentata come albero, uno per ogni nodo.

1.2 Soddisfacibilità

Si definisce interpretazione proposizionale una formula $I : \mathcal{P}\nabla\wr_{\surd} \to \{True, False\}$. Esistono $2^{|\mathcal{P}\nabla\wr_{\surd}|}$ interpretazioni. Si può considerare come un sottoinsieme $S \subset \mathcal{P}\nabla\wr_{\surd}$ tale che $A \in S \Leftrightarrow I(A) = True$. Una formula A si dice soddisfatta per l'interpretazione I o $I \models A$ se:

- $I \models A$ se $I(A) = True,\ A \in \mathcal{P}\nabla\wr_{\surd}$.
- $I \models \neg A$ se non $I \models A$.
- $I \models A \wedge B$ se $I \models A$ e $I \models B$.
- $I \models A \vee B$ se $I \models A$ o $I \models B$.
- $I \models A \supset B$ se quando $I \models A$ allora $I \models B$.
- $I \models A \equiv B$ se $I \models A$ se e solo se $I \models B$.

Per ogni variabile proposizionale P che appaia in A se $I(P) = I'(P)$ allora $I \models A$ se e solo se $I' \models A$.

1.3 Formule valide, soddisfacibili e insoddisfacibili

Una formula A si dice valida se per ogni sua interpretazione I allora $I \models A$. Una formula si dice soddisfacibile se esiste almeno una interpretazione I tale che $I \models A$. Una formula si dice insoddisfacibile non esiste interpretazione che la soddisfi. Per controllare la validità di una formula si possono enumerare tutte le interpretazioni rilevanti per essa e controllare se la soddisfano. Un insieme $\Gamma = \{A_1, \ldots, A_n\}$ è valido, soddisfacibile o insoddisfacibile se e solo se lo è $A_1 \wedge \cdots \wedge A_n$.

1.4 Conseguenza e equivalenza logica

Una formula A si dice conseguenza di Γ o $\Gamma \models A$ se e solo se per ogni interpretazione I tale che $I \models \Gamma$ allora $I \models A$. Due formule F e G si dicono equivalenti se e solo se hanno lo stesso valore per ogni interpretazione: $F \equiv G \Leftrightarrow I(G) = I(F) \forall I$.

1.4.1 Proprietà della conseguenza logica proposizionale

Siano Γ e Σ due insiemi di formule e A e B due formule ben formate, allora :

- Riflessività: $\{A\} \models A$.
- Monotonicità: se $\Gamma \models A$ allora $\Gamma \cup \Sigma \models A$.
- Taglio: se $\Gamma \models A$ e $\Sigma \cup A \models B$ allora $\Gamma \cup \Sigma \models B$.
- Compattezza: Se $\Gamma \models A$ esiste un sottoinsieme finito $\Gamma_0 \subset \Gamma$ tale che $\Gamma_0 \models A$.
- Teorema della deduzione: se $\Gamma, A \models B$ allora $\Gamma \models A \supset B$.
- Principio di refutazione: $\Gamma \models A$ se e solo se $\Gamma \cup \{\neg A\}$ è insoddisfacibile.

Dimostrazione riflessività

Per ogni interpretazione I se $I \models A$ allora $I \models A$.

Dimostrazione monotonicità

Per ogni I se $I \models \Gamma \cup \Sigma$ allora $I \models \Gamma$. Essendo per ipotesi che $\Gamma \models A$ si può inferire che $I \models A$, pertanto $\Gamma \cup \Sigma \models A$.

Dimostrazione Taglio

Per definizione di conseguenza logica si capisce dall'ipotesi che $I \models \Gamma$ e $I \models \Sigma$. Essendo che $I \models \Gamma$ allora $I \models A$, essendo che $I \models \Sigma$, $I \models \Sigma \cup \{A\}$ essendo per ipotesi $\Sigma \cup \{A\} \models B$ allora $I \models B$. Si può pertanto concludere che $\Gamma \cup \Sigma \models B$.

Dimostrazione compattezza

La dimostrazione è banale se A è una tautologia o se Γ è finito. Sia P_A l'insieme di formule atomiche in A, un insieme finito e siano $I_1, \ldots, I_n$ con $n \leq 2^{P_A}$ tutte le interpretazioni di P_A che non soddisfano A, ovvero $I_i \not\models A$. Esistono in quanto A non è una tautologia. Dal fatto che $\Gamma \models A$ si estendano $I_1, \ldots, I_n$ in $I'_1, \ldots, I'_n$ tali che $I'_i \not\models \Gamma_k \in \Gamma$. Sia ora $\Gamma_0 = \{\Gamma_1, \ldots, \Gamma_k\}$ allora $\Gamma_0 \models A$ in quanto dal falso segue ogni cosa. Pertanto se $I \models \Gamma_0$ I è un'estensione di J di P_A che soddisfa A, pertanto $I \models A$.

Dimostrazione del teorema della deduzione

Si assuma per ipotesi che $I \models \Gamma$ ci sono due casi: se $I \models A$ allora $I \models B$ per ipotesi, pertanto $I \models A \Rightarrow B$, se $I \not\models A$ allora, siccome $(FALSE) \models B$ allora $I \models A \Rightarrow B$ in quanto per ogni interpretazione da premessa falsa segue ogni cosa. Si può pertanto concludere che $I \models A \Rightarrow B$.

Principio di refutazione

Si supponga per contraddizione che $\Gamma \cup \{\neg A\}$ sia soddisfacibile: questo implica che esiste un'interpretazione I tale che $I \models \Gamma$ e $I \models \neg A$, ovvero $I \not\models A$, questo crea una contraddizione con l'ipotesi che $\Gamma \models A$. Si consideri ora $I \models \Gamma$, pertanto dal fatto che $\Gamma \cup \{\neg A\}$ è insoddisfacibile si ha che $I \not\models \neg A$ pertanto $I \models A$ e si può concludere che $\Gamma \models A$.

1.5 Assiomi e teorie

1.5.1 Teoria proposizionale

Si definisce teoria un insieme di formule chiuso per la conseguenza logica, ovvero T è una teoria se e solo se $T \models A$ implica che $A \in T$. Una teoria proposizionale contiene un insieme di formule infinito: ogni teoria T contiene almeno tutte le formule valide.

1.5.2 Assiomatizzazione

Insieme di assiomi per una teoria

Un insieme di formule Ω è un insieme di assiomi per una teoria T se per ogni $A \in T$, $\Omega \models A$. Una teoria si dice assiomatizzabile finitamente se ha un insieme finito di assiomi. Si definisce chiusura logica per ogni insieme Γ, la funzione cl tale che $cl(\Gamma) = \{A|\Gamma \models A\}$. Per ogni insieme Γ, la sua chiusura logica $cl(\Gamma)$ è una teoria e Γ è un insieme di assiomi per la sua chiusura logica ma non l'unico.

1.5.3 Rappresentazione compatta della conoscenza

L'assiomatizzazione di una teoria è un modo compatto di rappresentare un insieme di interpretazioni e pertanto un gruppo di stati del mondo possibili e accettabili, ovvero di rappresentare tutta la conoscenza posseduta del mondo. Gli assiomi di una teoria costituiscono la conoscenza base e tutto il resto può essere ottenuto attraverso la conseguenza logica. Da questo deriva il fatto che nessun assioma deve essere derivabile da altri.

1.5.4 Sistemi basati sulla logica

Un sistema basato sulla logica per rappresentare e ragionare sulla conoscenza è composto da una conoscenza di base e un sistema di ragionamento: la prima consiste un una collezione finita di formule in un linguaggio logico in modo da permettere di risolvere queries sottomesse al sistema di ragionamento.

1.6 Tavole di verità

Due formule F e G si dicono logicamente equivalenti se $I(F) = I(G)$ per ogni I e conseguenza logica se ogni interpretazione di F soddisfa G F si dice valida se è soddisfatta da ogni interpretazione, soddisfacibile se è soddisfatta da qualche interpretazione, insoddisfacibile se non esiste alcuna interpretazione che la soddisfa. Le tavole di verità esplicitano ogni possibile interpretazione in modo da decidere il tipo di formula.

1.7 Problemi di ragionamento e decisione

Si dice controllo del modello o $MC(I, \phi)$ il calore di verità di ϕ in I, ovvero se I soddisfa o no ϕ. Soddisfacibilità o insoddisfacibilità $SAT/UNSAT(\phi) : \exists I : I \models \phi$ ovvero l'esistenza di un modello che soddisfa ϕ. Conseguenza logica$(\Gamma, \phi) : \Gamma \models \phi$ se ϕ è soddisfatta da tutti i modelli I che soddisfino tutte le formule in Γ. Una procedura di decisione del modello o MCDP è un algoritmo che controlla se una formula ϕ è soddisfatta da un'interpretazione I, ovvero $MCDP(\phi, I) = TRUE)$

se e solo se $I \models \phi$ e falsa altrimenti. La procedura ritorna per ogni input vero o falso. Un modo semplice per implementare questo algoritmo è sostituire nella formula ogni valore delle formule atomiche attraverso le tavole di verità di I e poi applicare ricorsivamente la vero-funzione degli algoritmi. Una semplice ottimizzazione è rappresentata dalla lazy evaluation che permette, se il valore viene deciso univocamente dal primo elemento di un connettivo di non considerare il secondo. Una procedura di decisione della soddisfacibilità o SDP è un algoritmo che riceve in input una formula ϕ e controlla la sua soddisfacibilità. Una procedura di decisione della validità o VDC è un algoritmo che prende in input una forma e controlla la sua validità. Possono essere basati su una decisione di soddisfacibilità in quanto ϕ è valida se e solo se $\neg\phi$ è non soddisfacibile. Quando $SDP(\neg\phi)$ ritorna un'interpretazione si dice contromodello di ϕ. Una procedura di decisione per la conseguenza logica LCPD è un algoritmo che controlla se una formula sia logica conseguenza di un insieme finito di formula, può essere implementato attraverso una SDP in quanto $\Gamma \models \phi$ se e solo se $\Gamma \cup \{\neg\phi\}$ è insoddisfacibile. Quando $SDP(\gamma_1 \wedge \cdots \wedge \gamma_n \wedge \neg\phi)$ ritorna un'interpretazione I è un'intepretazione per il modello e un contromodello per ϕ.

1.8 Decidire soddisfacibilità

1.8.1 Conjunctive normal form (CNF)

Si definisce come literal una variabile proposizionale o una sua negazione e clause una disgiunzione di literals, una formula in conjunctive normal form è una congiunzione di clauses. Una formula in CNF ha pertanto la forma $(I_{1,1} \vee \cdots \vee I_{1,n_1}) \wedge \cdots \wedge (I_{m1} \vee \cdots \vee I_{m,n_m})$ o, in maniera equivalente $\bigwedge_{i=1}^{m} (\bigvee_{j=1}^{n_j} I_{j,i})$. $\wedge$ e $\vee$ godono delle proprietà della commutatività, e dell'assorbimento: se sono composte dallo stesso elemento si può sostituire con l'elemento.

Proprietà delle clauses

Tutte le clauses ottenute riordinando i literals sono equivalenti. Occorrenze multiple dello stesso literal possono essere elimitate tranne una. Si possono rappresentare pertanto come un insieme di literals, lasciando implicita la disgiunzione.

Proprietà delle formule in CNF

Tutte le formule ottenute riordinando clauses sono equivalenti e occorrenze multiple delle stesse clauses possono essere eliminate tranne una. Si possono pertanto rappresentare le formule CNF come insiemi di insiemi di literals. Inoltre ogni formula può essere ridotta in CNF, ovvero $\models CNF(\phi) \equiv \phi$.

Riduzione in CNF

La funzione CNF che trasforma una formula proposizionale nella sua forma CNF è definita ricorsivamente come:

- $CNF(p) = p$ se $p \in \mathcal{P}\nabla\wr_{\surd}$.
- $CNF(\neg p) = \neg p$ se $\neg p \in \mathcal{P}\nabla\wr_{\surd}$.
- $CNF(\phi \Rightarrow \psi) = CNF(\neg\phi) \otimes CNF(\psi)$.

- $CNF(\phi \wedge \psi) = CNF(\phi) \wedge CNF(\psi)$.
- $CNF(\phi \vee \psi) = CNF(\phi) \otimes CNF(\psi)$.
- $CNF(\phi \equiv \psi) = CNF(\phi \Rightarrow \psi) \wedge CNF(\psi \Rightarrow \phi)$.
- $CNF(\neg\neg\phi) = CNF(\phi)$.
- $CNF(\neg(\phi \Rightarrow \psi)) = CNF(\phi) \otimes CNF(\neg\psi)$.
- $CNF(\neg(\phi \wedge \psi)) = CNF(\neg\phi) \otimes CNF(\neg\psi)$.
- $CNF(\neg(\phi \vee \psi)) = CNF(\neg\phi) \wedge CNF(\neg\psi)$.
- $CNF(\neg(\phi \equiv \psi)) = CNF(\phi \wedge \neg\psi) \otimes CNF(\psi \wedge \neg\phi)$.

Dove $(C_1 \wedge \cdots \wedge C_n) \otimes (D_1 \wedge \cdots \wedge D_m)$ è definito come $(C_1 \vee D_1) \wedge \cdots \wedge (C_1 \vee D_m) \wedge \cdots \wedge (C_n \vee D_1) \wedge \cdots \wedge (C_n \vee D_m)$.

Costo del CNF

Il CNF nella sua forma normale permette di avere un algoritmo di soddisfacibilità semplice ma il costo della trasformazione diventa esponenziale.

1.8.2 La procedura di decisione DPLL SAT

Soddisfacibilità di un insieme di clauses

Sia $N = C_0, \ldots, C_n = CNF(\phi)$ allora $I \models \phi$ se e solo se $I \models C_i$ per ogni $i = 0, \ldots, n$ e $I \models C_i$ se e solo se per qualche literal $P \in C, I \models P$. Per controllare che un modello I soddisfi N non si devono conoscere i valori di verità che I assegna a tutti i literals che appaiono in N.

Valutazione parziale

Una valutazione parziale è una funzione parziale che associa a qualche variabile proposizionale di $\mathcal{P}\nabla\wr\surd$ un valore di verità e può essere indefinita per gli altri. Grazie ad essa si possono costruire modelli per un insieme di clauses N in maniera incrementale. La DPLL comincia con una valutazione vuota e prova ad espanderla passo per passo a tutte le lettere proposizionali in N. Una clause può essere così vera se uno dei suoi literals è vero, falsa se sono tutti falsi o altrimenti indecisa o non risolta.

DPLL

Per ogni formula CNFϕ e la formula atomica p $\phi|_p$ è la formula ottenuta da ϕ sostituendo tutte le occorrenze di p con T e semplificando il risultato rimuovendo tutte le clause contenenti il termine disgiuntivo T e tutti i literals $\neg T$ nelle clause rimanenti. Se una formula contiene una clause che consiste di un singolo literal è detta unit clause. Se ϕ contiene una unit clause allora per soddisfarla si deve soddisfare la unit clause e pertanto il suo literal deve essere valutato VERO. Attraverso questo metodo una formula risulta soddisfacibile se è rappresentata da un insieme vuoto, se invece la formula contiene un insieme vuoto allora è insoddisfacibile. Se la unit propagation non termina si deve decidere un valore di verità per un literal per poterla applicare. Questo verrà fatto attraverso una scelta euristica.

Capitolo 2

Logica del primo ordine

A differenza della logica proposizionale la logica del primo ordine considera un universo costituito da costanti nominali, predicati e funzioni. Una formula atomica viene costruita applicando predicati a costanti. O funzioni a costanti e un predicato sull'oggetto corrispondente. Inoltre vengono introdotti il quantificatore universale $\forall$ ed esistenziale $\exists$ in modo da quantificare arbitrari oggetti dell'universo.

2.1 Linguaggio

2.1.1 Sintassi

L'alfabeto della logica del primo ordine è costituito da:

- Simboli logici: costante logica, connettivi proposizionali, quantificatori e un insieme infinito di simboli di variabili.
- Simboli non logici: un insieme di simboli costanti, un'insieme di simboli funzionali ad ognuno dei quali è associta un'arity (numero di argomenti), un insieme di simboli relazionali ad ognuno dei quali è associata un'arity.

I simboli non logici dipendono dall'universo che si vuole modellare.

2.1.2 Terms e formule

Ogni costante, variabile logica è ogni simbolo funzionale pieno sono term. Le formule sono formate dai term attraverso connettivi logici, quantificatori e simboli relazionali pieni.

2.2 Funzione di interpretazione

Un'interpretazione di primo ordine per il linguaggio $L = (c_1, \ldots, c_n, f_1, \ldots, f_n, P1, \ldots, P_n)$ è una coppia (Δ, I) dove Δ è un insieme non vuoto detto dominio di interpretazione e I è la funzoine di interpretazione tale che $I(c_i) \in \Delta$, $I(f_i) : \Delta^n \to \Delta$ e $I(P_i) \subset \Delta^n$ dove n è l'arity di f_i e P_i.

2.3 Soddisfacibilità rispetto ad un'assegnazione

Si definisce un assegnazione a una funzione dall'insieme di variabili al dominio dell'interpretazione Δ. $a[x/d]$ dentoa l'assegnazione che coincide con a su tutte le variabili x che è associata a d. L'interpretazione di un termine con l'assegnazione a o $I(t)[a]$ è definita come:

- $I(x_i)[a] = a(x_i)$.
- $I(c_i)[a] = I(c_i)$.
- $I(f(t_1, \dots, t_n))[a] = I(f)(I(t_1)[a], \dots, I(t_n)[a])$.

2.4 Variabili libere

Un'occorrenza libera di una variabile x è un'occorrenza di x la quale non è legata a un quantificatore: si definisce come ogni occorrenza di x in t_k è libera in $P(t_1, \dots, t_k, \dots, t_n)$, ogni occorrenza libera di x in ϕ o ψ è libera anche in $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \Rightarrow \psi$ e $\neg\phi$, ogni occorrenza libera di x in ϕ è libera in $\forall y.\phi$ e $\exists y.\phi$ se y è distinta da x. Una formula ϕ si dice ground se non contiene nessuna variabile, è chiusa se non contiene occorrenze libere di una variabile. Una variabile x è libera in ϕ se almeno esiste una sua occorrenza libera, si denota con $\phi(x)$. Un termine t è libero per una variabile x in formula ϕ se e solo se ogni occorrenza di x in ϕ non occorre nello scopo del quantificatore di qualche variabile che occorre in t.

2.5 Soddisfacibilità e validità

Un interprentazione I è un modello di ϕ sotto l'assegnazione a se $I \models \phi[a]$. Una formula si dice soddisfacibile se esiste qualche I e qualche a tali che $I \models \phi[a]$, è insoddisfacibile se non è soddisfacibile ed è valida se ogni I e ogni a $I \models \phi[a]$. Una formula ϕ si dice conseguenza logica di un insieme di formule $\Gamma \models \phi$ se per ogni interpretazione ed assegnazione $I \models \Gamma[a] \Rightarrow I \models \phi[a]$ dove il primo vuol dire che I soddisfa tutte le formule in Γ sotto a. Si noti pertanto che se la formula è chiusa essa non dipende dall'assegnazione e $I \models \phi[a]$ se e solo se $I \models \phi[a']$ dove $[a]$ e $[a']$ coincidono sulle variabili libere in ϕ.

2.6 Domini finiti con nomi per ogni elemento

UNA o unique name assumption è l'assunzione secondo la quale il linguaggio contiene un nome per ogni elemento nel dominio, ovvero $\phi_{\Delta=\{c_1,\dots,c_n\}} = (\bigwedge_{i\neq j=1}^{n} c_i \neq c_j \wedge \forall x(\bigvee_{i=1}^{n} c_i = x))$. Quest'assunzione dice che un predicato P è vero solo per un insieme finito di oggetti per i quali il linguaggio contiene un nome è formalizzato come $\forall x.(P(x) \Leftrightarrow (x = c_1 \vee \dots \vee x = c_n))$. Sotto l'ipotesi di un dominio finito con nomi costanti per ogni elemento le formule del primo ordine possono essere proposizionalizzate o groundet secondo le analogie tra i $\forall$ e $\wedge$ e $\exists$ e $\vee$.

Capitolo 3

Logica modale

3.1 Intuizione

3.1.1 Modalità

Una modalità è un'espressione che è utilizzata per qualificare la verità di un giudizio o un operatore che esprime un modo in cui una proposizione è vera. Può essere considerata come un operatore che prende una proposizione e ritorna un'altra proposizione più complessa. Sono formalizzate attraverso operatori come $\Box$ e $\Diamond$ che possono essere applicati ad una formula ϕ per ottenerne un'altra $\Box\phi$. Una modalità è un espressione utilizzata per qualificare la verità di un giudizio. Le prime modalità formalizzate sono quelle aletiche in cui $\Diamond$ esprime possibilità e $\Box$ necessità.

3.2 Linguaggio

Se P è un insieme di proposizioni primitive l'insieme di formule definite dalla logica modale base sono definite:

- Ogni $p \in P$ è una formula.
- Se A e B sono formule allora $\neq A, A \wedge B, A \vee B, A \Rightarrow B$ e $A \equiv B$ sono formule.
- Se A è una formula $\Box A$ e $\Diamond A$ sono formule.

3.2.1 Interpretazione intuitiva

La formula $\Box\phi$ può essere interpretata intuitivamente in molti modi:

- ϕ è necessariamente vera (logica modale classica).
- Si sa che ϕ è vera (logica epistemologica).
- ϕ è dimostrabile in una teoria (logica della dimostrabilità).
- ϕ sarà sempre vera (logica temporale).

In tutti questi casi $\Diamond\phi$ può essere interpretato come $\neg\Box \neq \phi$, ovvero ϕ è possibilmente vera.

3.3 Strutture relazionali

La semantica estensionale della logica modale è data attraverso strutture relazionali, tuple $< W, R_{a_1}, \ldots, R_{a_n} >$ dove $R_{a_i} \subseteq W \times \cdots \times W$. Ogni $w \in W$ è chiamato punto o mondo e ogni R_{a_i} è detta relazione di accessibilità.

3.3.1 Esempi di strutture relazionali

- Strict partial order (SPO): $< Q, <>$, $<$ è transitiva e irreflessiva.
- Strict linear order: $< W, <>$, SPO e per ogni $v \neq u \in W$, $v < w$ o $w < v$.
- Partial order (PO): $< W, \leq>$. $\leq$ è transitiva, riflessiva e antisimmetrica.
- Linear order: $< W, \leq>$, PO e per ogni $v \neq u \in W$, $v \leq w$ o $w \leq v$.
- Labeled transition system (LTS): $< W, R_a >_{a \in A}$ e $R_a \subseteq W \times W$.
- XML document: $< W, R_I >_{I \in L}$, W contiene i componenti di un documento XML e L è l'insieme delle labels che appaiono nel documento.

3.4 Semantica

Un basic frame è una struttura algebrica $F =< W, R >$ dove $E \subseteq W \times W$. Un interpretazione I di un linguaggio modale in un frame F è una funzione $I : P \to 2^W$. Intuituvamente $w \in I(p)$ vuol dire che p è vera in w o che w è di tipo p. Un modello $\mathcal{M}$ è formato dal paio di frame e interpretazione: $\mathcal{M} =< F, I >$. La verità è relativa al mondo, pertanto si definisce la relazione di $\models$ tra un mondo in un modello e una formula.

- $\mathcal{M}, w \models p$ se e solo se $w \in I(p)$.
- $\mathcal{M}, w \models \phi \wedge \psi$ se e solo se $\mathcal{M}, w \models \phi$ e $\mathcal{M}, w \models \psi$.
- $\mathcal{M}, w \models \phi \vee \psi$ se e solo se $\mathcal{M}, w \models \phi$ o $\mathcal{M}, w \models \psi$.
- $\mathcal{M}, w \models \phi \Rightarrow \psi$ se e solo se $\mathcal{M}, w \models \phi$ implica $\mathcal{M}, w \models \psi$.
- $\mathcal{M}, w \models \phi \equiv \psi$ se e solo se $\mathcal{M}, w \models \phi$ se e solo se $\mathcal{M}, w \models \psi$.
- $\mathcal{M}, w \models \neg\phi$ se e solo se non $\mathcal{M}, w \models \phi$.
- $\mathcal{M}, w \models \Box\phi$ se e solo se per ogni w' tale che wRw', $\mathcal{M}, w' \models \phi$.
- $\mathcal{M}, w \models \Box\phi$ se e solo se esiste un w' tale che wRw', $\mathcal{M}, w' \models \phi$.

ϕ è soddisfatta globalmente in un modello $\mathcal{M} \models \phi$ se e solo se $\mathcal{M}, w \models \phi$ per ogni $w \in W$.

3.4.1 Esprimere proprietà sulle strutture

Formula vera a w	Proprietà di w
$\diamond T$	w ha un punto successore.
$\diamond \diamond T$	w ha un punto successore con un punto successore.
$\diamond_1 \cdots \diamond_n T$	w esiste un cammino di lunghezza n che comincia a w.
$\Box\bot$	w non ha nessun punto successore.
$\Box\Box\bot$	Ogni successore di w non ha un punto successore.
$\Box_1 \ldots \Box_n \bot$	Ogni cammino che comincia da w ha lunghezza minore di n.

Formula vera a w	Proprietà di w
$\diamond p$	w ha un punto successore che è p.
$\diamond \diamond p$	w ha un punto successore con un punto successore che è p.
$\diamond_1 \cdots \diamond_n p$	w esiste un cammino di lunghezza n che comincia a w e finisce con un punto che è p.
$\Box p$	Ogni successore di w è p.
$\Box\Box p$	Ogni successore del successore di w è p.
$\Box_1 \ldots\ Box_n p$	Ogni di lunghezza n che comincia da w finisce in un punto che è p.

3.4.2 Proprietà della relazione di accessibilità

- Logica temporale: la relazione di accessibilità deve rappresentare una relazione temporale e wRw' vuol dire che w' è un mondo futuro. Esistono due tipi di strutture: alberi o insiemi di cammini nella direzione del tempo per futuro e passato con come radice il mondo presente.
- Logica epistemica: la relazione di accessibilità è utilizzata per rappresentare le credenze dell'agente A e wRw' rappresenta il fatto che w' è una situazione possibile coerente con la situazione attuale w. Utilizzano strutture gerarchiche radicate.
- Logica descrittiva: la relazione di accessibilità rappresenta una relazione tra due concetti o classi. La struttura è un grafo di conoscenza senza restrizioni sulla sua forma.

3.5 Analisi di proposizioni

3.5.1 Relazione di validità su frames

Una formula ϕ è valida in un mondo w di un frame F, ovvero $F, w \models \phi$ se e solo se $\mathcal{M}, w \models \phi$ per ogni I con $\mathcal{M} =< F, I >$. Una formula è valida in un frame F, ovvero $F \models \phi$ se e solo se $F, w \models \phi$ per ogni $w \in W$. Detto C una classe di frames allora una formula ϕ 'e valida nella classe di frames C, ovvero $\models_C \phi$ se e solo se $F \models \phi$ per ogni $F \in C$. Una formula si dice valida $\models \phi$ se e solo se $F \models \phi$ per ogni frame F.

3.5.2 Conseguenza logica

ϕ è conseguenza logica di Γ , $\Gamma \models \phi$ se per ogni modello $\mathcal{M} =< F, I >$ e ogni punto $w \in W$, $\mathcal{M}, w \models \Gamma$ implica che $\mathcal{M}, w \models \phi$. ϕ si dice conseguenza logica di Γ in una classe di frame C, $\Gamma \models_C \phi$ se per ogni modello $\mathcal{M} =< F, I >$ con $F \in C$ e ogni punto $w \in W$, $\mathcal{M}, w \models \Gamma$ implica che $\mathcal{M}, w \models \phi$. Insoddisfacibilità ed equivalenza logica sono definiti come al solito.

3.6 K modale o analisi di Hilbert

Gli assiomi che formano **K** caratterizzano le logiche modali con relazione di validità chiusa al loro interno chiamate logiche modali normali. Gli assiomi sono:

$\mathbf{A_1}$ $\phi \Rightarrow (\psi \Rightarrow \phi)$.

$\mathbf{A_2}$ $(\phi \Rightarrow (\psi \Rightarrow \theta)) \Rightarrow ((\phi \Rightarrow \psi) \Rightarrow (\phi \Rightarrow \theta))$.

$\mathbf{A_3}$ $(\neq \psi \Rightarrow \neq \phi) \Rightarrow ((\neq \psi \Rightarrow \phi) \Rightarrow \phi)$.

MP $\dfrac{\phi, \phi \Rightarrow \psi}{\psi}$.

K $\Box(\phi \Rightarrow \psi) \Rightarrow (\Box\phi \Rightarrow \Box\psi)$.

Nec $\dfrac{\phi}{\Box\phi}$ o regola di necessità.

Si noti come $\phi \Rightarrow \Box\phi$ non è equivalente a Neq, in quanto non valida. Si nota come nell'analisi di Hilbert esiste una differenza tra deduzione e prova in quanto il teorema di deduzione non è vero.

3.7 Relazione di accessibilità

Le formule possono essere utilizzate per dare forma alla struttura imponendo proprietà all'accessibilità di R. Siano esempi:

- Logica temporale: se la relazione di accessibilità deve rappresentare una relazione temporale e wRw' vuol dire che w' è un mondo futuro, allora R deve essere una relazione transitiva, ovvero se w' è un mondo futuro di w allora ogni mondo futuro di w' è un mondo futuro di w:
- Logica della conoscenza: se la relazione di accessibilità è utilizzata per rappresentare la conoscenza di un agente A e wRw' rappresenta il fatto che w' è una possibile situaizone coerente con la situazione attuale w allora R deve essere riflessiva in quanto w è sempre coerente con sè stesso.

3.7.1 Proprietà tipiche di R

- Riflessività: $\forall w.R(w, w)$.
- Transitività: $\forall w, v, u.((R(w, v) \wedge R(v, u)) \Rightarrow R(w, u))$.
- Simmetria: $\forall w, v.(R(w, v) \Rightarrow R(v, w))$.
- Euclidea: $\forall w, v, u.((R(w, v) \wedge R(w, u)) \Rightarrow R(v, u))$.
- Seriale: $\forall w.\exists v.R(w, v)$.
- Funzionale: $\forall w.\exists! v.R(w, v)$.

3.8 KT modale

In questa logica modale la relazione R è riflessiva e vale l'assioma **T** secondo cui se un frame è riflessivo allora vale la formula $\Box\phi \Rightarrow \phi$ o alternativamente $\phi \Rightarrow \diamond\phi$.

3.8.1 Correttezza

Sia $\mathcal{M}$ un modello su un frame riflessivo $F = (W, R)$ e $w \in E$. Si prova come $\mathcal{M}, w \models \Box\phi \Rightarrow \phi$. Siccome R è riflessiva allora wRw. Si supponga che $M, w \models \Box\phi$, dalla condizione di soddisfacibilità di $\Box$, $\mathcal{M}, w \models \Box\phi$ e wRw implica che $\mathcal{M}, w \models \phi$. Siccome dall'ipotesi si è derivata la tesi si può concludere che $\mathcal{M}, w \models \Box\phi \Rightarrow \phi$.

3.8.2 Completezza

Si supponga che un frame $F = (W, R)$ non sia riflessivo: se R non è riflessiva allora esiste un $w \in W$ tale che non accede a sè stesso, ovvero wRw non sussiste. Sia $\mathcal{M}$ un qualunque modello di F e ϕ una formula proposizionale p. Sia V l'insieme p vero in tutti i mondi di W tranne w dove è falsa. Dal fatto che w non accede a sè stesso si ha che in tutti i mondi w' accessibili da w p è vera: $\forall w', wRw', \mathcal{M}, w' \models p$. Dalla condizione di soddisfacibilità di $\Box$ si ha che $\mathcal{M}, w \models \Box p$. Siccome $\mathcal{M}, w \models \Box p$ si ha che $w \models \Box p \Rightarrow p$.

3.9 KB modale

In questa logica modale la relazione R è simmetrica e vale l'assioma **B** secondo cui se un frame è riflessivo allora vale la formula $\phi \Rightarrow \Box \diamond \phi$.

3.9.1 Correttezza

SIa $\mathcal{M}$ un modello su un frame simmetrico $F = (W, R)$ e $w \in W$. Si provi che $\mathcal{M}, w \models \phi \Rightarrow \Box \diamond \phi$. Per ipotesi $\mathcal{M}, w \models \phi$ e si vuole dimostrare che $\mathcal{M}, w \models \phi$. Dalle condizioni di soddisfacibilità di $\Box$ si deve provare che per ogni mondo w' accessibile da w, $w' \models \diamond\phi$. Sia w' un mondo accessibile da w: wRw'. Dal fatto che R è simmetrica si ha che $w'Rw$ e dalle condizioni di soddisfacibilità di $\diamond$ dal fatto che $w'Rw$ e che $\mathcal{M}, w \models \phi$ si ha che $\mathcal{M}, w' \models \diamond\phi$. Dalle condizioni di soddisfacibilità di $\Box$ si inferisce che $\mathcal{M}, w \models \Box \diamond \phi$. Essendo che dall'ipotesi abbiamo derivato la tesi si conclude che $\mathcal{M}mw \models \phi \Rightarrow \Box \diamond \phi$.

3.9.2 Completezza

Si supponga che un frame $F = (W, R)$ non sia simmetrico. Se non lo è esistono due mondi $w, w' \in W$ trali che wRw' e non $w'Rw$. Sia $\mathcal{M}$ un qualsiasi modello di F e ϕ la formula proposizionale p. Sia V l'insieme di mondi in cui p sia falsa e p vera in w. Dal fatto che w' non accede a w si ha che in tutti i mondi accessibili da w' p è falsa. Dalle condizioni di soddisfacibilità di $\diamond$ si ha che $\mathcal{M}, w' \not\models \diamond p$. Essendo che esiste un mondo w' accessibile da w con $\mathcal{M}mw \not\models \diamond p$ dalle condizoni di soddisfacibilità di $\Box$ si ha che $\mathcal{M}mw \not\models \Box \diamond p$. Essendo $\mathcal{M}, w \models p$ e $\mathcal{M}mw \not\models \Box \diamond p$ si ha che $\mathcal{M}, w \not\models p \Rightarrow \Box \diamond p$.

3.10 KD modale

In questa logica modale la relazione R è seriale e vale l'assioma **D** secondo cui se un frame è seriale allora vale la formula $\Box\phi \Rightarrow \diamond\phi$.

3.10.1 Correttezza

Sia $\mathcal{M}$ un modello su un frame seriale $F = (W, R)$ e w un mondo in W. Si provi che $\mathcal{M}mw \models \Box\phi \Rightarrow \Diamond\phi$. Siccome R è seriale esiste un mondo $w' \in W$ tale che wRw'. Si suppone per ipotesi che $\mathcal{M}, w \models \Box\phi$. Dalla condizione di soddisfacibilità di $\Box$ $\mathcal{M}, w \models \Box\phi$ implica che $\mathcal{M}, w' \models \phi$. Siccome esiste un mondo w' accessibile da w che soddisfa ϕ dalla condizione di soddisfacibilità di $\Diamond$ si ha che $\mathcal{M}, w \models \Diamond\phi$. Siccome dall'ipotesi si è derivata la tesi si può concludere che $\mathcal{M}, w \models \Box\phi \Rightarrow \Diamond\phi$.

3.10.2 Completezza

Si supponga che un frame $F = (W, R)$ non sia seriale. Se non è seriale allora esiste un $w \in W$ che non accede a nessun mondo: per ogni w' non vale wRw'. Sia $\mathcal{M}$ un modello su F. Dalla condizione di soffisfacibilità di $\Box$ e dal fatto che w non accede a nessun mondo si ha che $\mathcal{M}, w \models \Box\phi$. Dalla condizione di soddifacibilità di $\Diamond$ e dal fatto che w non accede a nessun mondo si ha che $\mathcal{M}, w \models \phi$ che implica che $\mathcal{M}, w \models \Box\phi \Rightarrow \Diamond\phi$.

3.11 KT$_4$ modale

In questa logica modale la relazione R è riflessiva e transitiva e vale l'assioma 4 secondo cui se un frame è riflessivo e transitivo allora vale la formula $\Box\phi \Rightarrow \Box\Box\phi$.

3.11.1 Correttezza

Sia $\mathcal{M}$ un modello su un frame transitivo $F = (W, R)$ e $w \in W$ un mondo qualsiasi. Si provi che $\mathcal{M}, w \models \Box\phi \Rightarrow \Box\Box\phi$. Si supponga per ipotesi che $\mathcal{M}, w \models \Box\phi$ e si deve provare che $\mathcal{M}, w \models \Box\Box\phi$. Dalla condizione di soddifacibilità di $\Box$ è equivalente a provare che per ogni mondo w' accessibile da w, $\mathcal{M}, w' \models \Box\phi$. Sia w' un qualsiasi mondo accessibile da w. Per provare che $\mathcal{M}, w' \models \Box\phi$ si deve provare che per tutti i mondo w'' accessibili da w', $\mathcal{M}, w'' \models \phi$. Sia w'' un mondo accessibile da w', ovvero $w'Rw''$. Dal fatto che wRw' e che $w'Rw''$ e che R è transitiva si ha che wRw''. Siccome $\mathcal{M}, w'' \models \phi$, dalla condizione di soddifacibilità di $\Box$ si ha che $\mathcal{M}, w'' \models \phi$. Siccome $\mathcal{M}, w'' \models \phi$ per ogni mondo w'' accessibile da w' si ha che $\mathcal{M}, w' \models \Box\phi$ e pertanto $\mathcal{M}, w \models \Box\Box\phi$. Siccome dall'ipotesi si è derivata la tesi si può concludere che $\mathcal{M}, w \models \Box\phi \Rightarrow \Box\Box\phi$.

3.11.2 Completezza

Si supponga che un frame $F = (W, R)$ non sia transitivo. Se non lo è esistono tre mondi $w, w', w'' \in W$ tali che wRw' e $w'Rw''$ ma non wRw''. Sia $\mathcal{M}$ un modello qualsiasi su F e sia ϕ la formula proposizionale p. Sia V l'insieme di mondi con p vero in tutti i mondi di W ma w'' dove p falsa. Dal fatto che w non accede a w'' e dal fatto che w'' è l'unico mondo in cui p è falsa si ottiene che in tutti i mondi accessibili da w p è vera. Questo implica che $\mathcal{M}, w \models \Box p$. Si ha che $w'Rw''$ e $w'' \models p$ implica che $\mathcal{M}, w' \models \Box\phi$. Siccome wRw' si ha che $\mathcal{M}, w \models \Box\Box p$. Riassumendo $\mathcal{M}, w \models \Box\Box p$ e $\mathcal{M}mw \models \Box p$ da cui si ha che $\mathcal{M}, w \models \Box p \Rightarrow \Box\Box p$.

3.12 KT$_5$ modale

In questa logica modale la relazione R è euclidea e riflessiva (relazione di equivalenza) e vale l'assioma 4 secondo cui se un frame è euclideo e riflessivo allora vale la formula $\Diamond\phi \Rightarrow \Box \Diamond \phi$.

3.12.1 Correttezza

Sia $\mathcal{M}$ un modello su un frame euclideo $F = (W, R)$ e w un mondo qualsiasi in W. Si provi che $\mathcal{M}, w \models \diamond\phi \Rightarrow \Box \diamond \phi$. Si supponga per ipotesi che $\mathcal{M}, w \models \diamond\phi$. La condizione di soddisfacibilità di $\diamond$ implica che esiste un mondo w' accessibile da w tale che $\mathcal{M}mw' \models \phi$. Si deve provare che $\mathcal{M}, w \models \Box \diamond \phi$. Dalla condizione di soddifacibilità di $\Box$ è equivalente a provare che per ogni mondo w'' accessibile da w, $\mathcal{M}, w'' \models \diamond\phi$. SIa w'' un qualsiasi mondo accessibile da w. Dal fatto che R è euclidea il fatto che wRw' implica che $w''Rw'$. Siccome $\mathcal{M}, w' \models \phi$, la condizione di soddifacibilità di $\diamond$ implica che $\mathcal{M}, w'' \models \diamond\phi$ e pertanto $\mathcal{M}, w \models \Box \diamond \phi$. Siccome dall'ipotesi si è derivata la tesi si può concludere che $\mathcal{M}, w \models \Box\phi \Rightarrow \Box \diamond \phi$.

3.12.2 Completezza

Si supponga che un frame $F = (W, R)$ si non euclideo. Se R è non euclidea allora esistono tre mondo $w, w', w'' \in W$ tali che wRw' e wRw'' e non $w'Rw''$. Sia $\mathcal{M}$ un modello qualsiasi su F e ϕ la formula proposizionale p. Sia V l'insieme di mondi in cui p è falsa in tutti i mondi di W ma in w' vera. Dal fatto che w'' non accede a w' e in tutti gli altri mondi p è falsa si ha che $w'' \not\models \diamond p$ che implica $\mathcal{M}, w \not\models \Box \diamond p$. Però si ha che wRw' e $w' \models p$ e pertanto $\mathcal{M}, w \models \diamond p$, $\mathcal{M}, w \not\models \Box p \Rightarrow \Box\Box p$. Riassumendo $\mathcal{M}, w \not\models \Box \diamond p$ e $\mathcal{M}, w \models \diamond p$ da cui si ha che $\mathcal{M}, w \not\models \diamond p \Rightarrow \Box \diamond p$.

3.13 Logica multi modale

Tutte le definizioni date per la logica modale base possono essere generalizzate nel caso in cui si hanno n $\Box$-operatori $\Box_1, \ldots, \Box_n$ e i rispettivi $\diamond$ interpretati nel frame $F = (w, R_1, \ldots, R_n)$ che interpreta il corrispettivo operatore. Tale logica è detta multi modale. Sono spesso utilizzate per modellare sistemi multi-agente dove la modalità è espressa per esprimere conoscenza o credenza.

3.14 Tableaux

Si definisce un tableau un albero finito con i nodi marcati da una delle seguenti asserzioni: $w \models \phi$, $w \not\models \phi$, wRw' costruiti secondo delle regole di espansione. Si definisce ramo o branch di un tableaux una sequenza $n_1, \ldots, n_k$ dove n_1 è la radice dell'albero e n_k una foglia e n_{i+1} un figlio di n_i per $1 \leq i < k$. Un ramo chiuso è un ramo che contiene nodi marcati con $w \models \phi$ e $w \not\models \phi$- Tutti gli altri rami sono aperti. Se tutti i rami sono chiusi, il tableaux è chiuso.

3.14.1 Regole di espansione

$\dfrac{w \models \phi \wedge \psi}{\begin{array}{c} w \models \phi \\ w \models \psi \end{array}}$	$\dfrac{w \not\models (\phi \wedge \psi)}{w \not\models \phi \mid w \not\models \psi}$
$\dfrac{w \models \phi \vee \psi}{w \models \phi \mid w \models \psi}$	$\dfrac{w \not\models \phi \vee \psi}{\begin{array}{c} w \not\models \phi \\ w \not\models \psi \end{array}}$
$\dfrac{w \models \neg\phi}{w \not\models \phi}$	$\dfrac{w \not\models \neg\phi}{w \models \phi}$
$\dfrac{w \models \phi \Rightarrow \psi}{w \not\models \phi \mid w \models \psi}$	$\dfrac{w \not\models \phi \Rightarrow \psi}{\begin{array}{c} w \models \phi \\ w \not\models \psi \end{array}}$
$\dfrac{w \models \Box\phi}{w' \models \phi}$ Se wRw' è già sul ramo.	$\dfrac{w \not\models \Box\phi}{\begin{array}{c} wRw' \\ w' \not\models \phi \end{array}}$ Dove w' è nuova sul ramo.
$\dfrac{w \models \Diamond\phi}{wRw' \; w' \models \phi}$ Dove w' è nuova sul ramo.	$\dfrac{w \not\models \Diamond\phi}{w' \not\models \phi}$ Se wRw' è già sul ramo.

Applicazione

Se un ramo $\beta = n_1, \ldots, n_k$ contiene un nodo n_i etichettato con la premessa di una delle regole e tale regola non è stata ancora applicata sul nodo allora può essere applicata e il ramo è espanso nel modo sequente:

- Se ha una sola conseguenza β è espanso in $n_1, \ldots, n_k, n_{k+1}$ dove n_{k+1} è etichettato come la conseguenza della regola.
- Se ha due conseguenze una sopra l'altra β è espanso in $n_1, \ldots, n_k, n_{k+1}, n_{k+2}$ dove n_{k+1} e n_{k+2} sono etichettati come le conseguenze della regola.
- Se ha due conseguenze alternative β è espanso in due nodi $n_1, \ldots, n_k, n_{k+1}$ e $n_1, \ldots, n_k, n_{k+2}$ dove n_{k+1} e n_{k+2} sono etichettati come le conseguenze alternative della regola.

Capitolo 4

Logica descrittiva

Le logiche descrittive sono una famiglia di logiche che permettono di parlare riguardo un dominio composto da un insieme di oggetti organizzati in concetti e messi in relazione attraverso relazioni binarie.

4.1 Knowledge Graphs

Queste logiche permettono di costruire predicati riguardo grafi diretti etichettati chiamati Knowledge graph (KGs), i cui nodi rappresentano le classi di oggetti reali e gli archi le proprietà che intercorrono tra paia di oggetti.

4.1.1 Proprietà

- Lo schema del grafo non è fisso e può evolvere.
- Non è necessario conoscere tutti i valori dei dati per tutti gli oggetti.

4.1.2 Livelli nei Knowledge graph

Livello di schema

- Tipi di entità (etypes): descrizioni dei tipi di oggetti contenuti nei nodi.
- Proprietà:
  - Attributi (data properties): relazioni che associano ad ogni oggetto un tipo.
  - Relazioni (Object properties): relazioni che sussistono tra gli oggetti.

Livello dei dati

- Entità: associano valori ai tipi di entità.
- Valori di proprietà:
  - Valori di attributi: associano valori agli attributi della oggetti.
  - Valori di Relazioni: associano relazioni a valori degli oggetti.

4.2 Caratterizzazione di una logica descrittiva

Una logica descrittiva è caratterizzata da quattro elementi:.

Linguaggio

Un linguaggio descrittivo L che determina come formare concetti (i nodi dei KG) e i ruoli (gli archi del KG e i constraints). È pertanto la formalizzazione di etypes, entità, proprietà e valori di proprietà.

Specificare conoscenza

Un meccanismo per specificare la conoscenza riguardante etypes (concetti) e proprietà (ruoli), come per esempio una $TBox$, la formalizzazione dello schema del KG con etypes e properties con i constraints.

Proprietà degli oggetti

Un meccanismo per specificare le proprietà degli oggetti o entità come un $ABox$, una formalizzazione delle entità e valori di proprietà del KG.

Servizi di ragionamento

Un insieme di servizi di ragionamento che permettono di inferire nuove proprietà su concetti, ruoli e oggetti che sono le conseguenze logiche di quelle asserite esplicitamente nelle $TBox$ e $ABox$. Sono equivalenti al ragionamento logico, ovvero un estensione delle formalizzazione del ragionamento KG.

4.3 Architettura di un sistema di ragionamento di una logica descrittiva

La logica descrittiva è una famiglia di formalismi KR, dove rappresentazione e ragionamento puntano a un sistema formato da $TBox$ e $ABox$. Esiste un alfabeto di simboli formato da quello della logica proposizionale con l'aggiunta di:

- $\forall R$ value restriction.
- $\exists R$ existential quantification.

E R sono nomi di ruolo atomici. Si può dire che la base della conoscenza è formata da una $TBox$ che garantisce una conoscenza terminologica e un $ABox$ che fornisce conoscenza riguardante gli oggetti.

4.4 Sintassi TBox - ALC (AL con il concetto di negazione)

Regole di formazione:

- $< Atomic >:= A|B|\dots|P|Q|\dots|\bot|T.$
- $< wwf >:=< Atomic > |\neg < wwf > | < wwf > \sqcap < wwf > | < wwf > \sqcup < wwf >$

4.4.1 Interpretazione AL$*$

La funzione di interpretazione $I :< wwf > \rightarrow \Delta$.

- $I(\bot) = \emptyset$ e $I(T) = \Delta$ (dominio pieno universo).
- Per ogni nome concettuale A di L, $I(A) \subseteq \Delta$.
- $I(\neg C) = \Delta \backslash I(C)$.
- $I(C \sqcap D) = I(C) \cap I(D)$.
- $I(C \sqcup D) = I(C) \cup I(D)$.
- Per ogni nome di ruolo di R, $I(R) \subseteq \Delta \times \Delta$.
- $I(\forall R.C) = \{a \in \Delta | per\ ogni\ b,\ se\ (a,b) \in I(R)\ allora\ b \in I(C)\}$.
- $I(\exists R.T) = \{a \in \Delta | esiste\ b\ tale\ che\ (a,b) \in I(R)\}$.
- $I(\exists R.C) = \{a \in \Delta | esiste\ b\ tale\ che\ (a,b) \in I(R), b \in I(C)$.
- Aggiunte esterne a ALC:
  - $I(\leq nR) = \{a \in \Delta || \{b | (a,b) \in I(R)\}| \leq n\}$.
  - $I(\geq nR) = \{a \in \Delta || \{b | (a,b) \in I(R)\}| \geq n\}$.

4.4.2 Assiomi terminologici

Vengono introdotti due nuovi simboli logici:

- $\sqsubseteq$ (sottoassunzione) con $I \models C \sqsubseteq D$ se e solo se $I(C) \subseteq I(D)$.
- $\equiv$ (equivalenza) con $I \models C \equiv D$ se e solo se $I(C) = I(D)$:

Assioma di inclusione

$C \sqsubseteq D$ deve essere letto C è sottoassunto da (più specifico, meno generale rispetto a) D. Lo stesso concetto può essere espresso come $D \sqsupseteq C$.

Assioma di uguaglianza

$C \equiv D$ deve essere letto come C è equivalente a S. Vale se e solo se valgono insieme $C \sqsubseteq D$ e $D \sqsubseteq C$

Definizione

Una definizione è un'equivalenza con un concetto atomico sulla parte sinistra.

Specializzazione

Una specializzazione è un'inclusione con un concetto atomico sulla parte sinistra.

Terminologia

Una terminologia o $TBox$ è un insieme di definizioni e specializzazioni. Gli assiomi terminologici esprimono constraints sui concetti del linguaggio limitando così i modelli possibili. Il $TBox$ è l'insieme di tutti i constraints sul modello possibile.

4.4.3 TBox - ragionamento

Date due proposizione di classe P e Q si presentano i seguenti problemi di ragionamento:

- Soddisfacibilità: $T \models P$.
- Sottoassunzione: $T \models P \sqsubseteq Q$ o $T \models Q \sqsubseteq P$.
- Equivalenza: $T \models P \sqsubseteq Q$ e $T \models Q \sqsubseteq P$.
- Disgiuntezza: $T \models P \sqcap Q \sqsubseteq \bot$.

Soddisfacibilità

Un concetto P è soddisfacibile rispetto a una terminologia T se esiste almeno un'interpretazione I con $I \models \theta$ per ogni $\theta \in T$ tale che $I \models P$ ovvero $I(P) \neq \emptyset$. In questo caso si dice che I è un modello per P. In altre parole l'interpretazione I non solo soddisfa P ma soddisfa tutti i constraints in T.

Sottoassunzione

Sia T una $TBox$. Un problema di sottoassuzione rispetto a T è definito come $T \models P \sqsubseteq Q (P \sqsubseteq_T Q)$. Un concetto P è sottoassunto da un concetto Q con rispetto a T se $I(P) \subseteq I(Q)$ per ogni modello I di T. Questo problema si riduce a derivazione logica e validità quando T è vuota.

Equivalenza

Sia T una $TBox$. Un problema di equivalenza rispetto a T è definito come $(T \models P \equiv Q)(P \equiv_T Q)$. Due concetti P e Q sono equivalenti rispetto a T se $I(P) = I(Q)$ per ogni modello I di T.

Disgiuntezza

Sia T una $TBox$. Un problema di disgiuntezza rispetto a T è definito come $T \models P \sqcap Q \sqsubseteq \bot (P \sqcap Q \sqsubseteq_T \bot)$. Due concetti P e Q sono disgiunti rispetto a T se la loro intersezione è vuota, $I(P) \cap I(Q) = \emptyset$ per ogni modello I di T.

4.5 ABox

4.5.1 Sintassi

Il secondo componente della base della conoscenza è la descrizione del mondo o $ABox$. Un'$ABox$ riguarda individui con nomi e asserisce proprietà riguardo essi. Si denotano nomi individuali con lettere minuscoli e un asserzione con concetto C è chiamata asserzione di concetto nella forma $C(a)$.

4.5.2 Semantica

Si dà semantica alle *ABoxes* estendendo l'interpretazione ai nomi individuali. Un'interpretazione $I : L \to \Delta'$ non mappa concetti atomici a insiemi ma mappa ogni nome individuale a un elemento di Δ'. ovvero $I(a) = a' \in \Delta'$. Con la Unique name assumption (UNA) si assume che nomi individuali distinti denotano oggetti distinti nel dominio.

4.5.3 Individui nella TBox

A volte può essere conveniente permettere nomi individuali detti nominali nelle $TBox$. Sono usati per costruire concetti come per esempio con un costruttore di insieme che definisce un concetto senza dargli un nome enumerando i suoi elementi con $I(\{a_1, \ldots, a_n\}) = \{a'_1, \ldots, a'_n\}$.

4.5.4 Problemi di ragionamento

- Soddisfacibilità, consistenza: Un $ABox$ A è consistente rispetto a T se esiste almeno un'interpretazione I che è modello sia di A che di T.
- Controllo delle istanze: controllare se un asserzione $C(a)$ è derivata da un'$ABox$, ad esempio controllare se a appartiene a C. $A \models C(a)$ se ogni interpretazione che soddisfa A soddisfa anche $C(a)$:
- Recupero dell'istanza: dato un concetto C recuperare tutte le istanze a che soddisfano C.
- Realizzazione dei concetti: dato un insieme α di concetti e un individuo a, trovare i concetti più specifici C tali che $A \models C(a)$.

4.5.5 Ragionamento attraverso l'espansione di un ABox

I servizi di ragionamento su un $ABox$ e una $TBox$ aciclica possono essere ridotti a controllare un'$ABox$ espansa. Si definisce l'espansione di un'$ABox$ A rispetto a T l'$ABox$ A' che è ottenuta da A sostituendo ogni asserzione di concetto $C(a)$ con l'asserzione $C'(a)$ con C' l'espansione di C rispetto a T. L'espansione $C'(a)$ di $C(a)$ è l'insieme dei concetti $C''(a)$ tali che $C''(a)$ sono usati per definire $C(a)$. L'espansione A' di A rispetto a T contiene solo primitive.

- A è consistente rispetto a T se e solo se la sua espansione A' è consistente.
- A è consistente se e solo se A è soddisfacibile, ovvero non contraddittoria.

4.5.6 Consistenza

Un'$ABox$ A si dice consistente rispetto a T se esiste un interpretazione I che è un modello sia di A che di T. Si dice che A è consistente se è consistente rispetto alla $TBox$ vuota.

4.5.7 Controllo d'istanza

Si deve controllare se un'istanza $C(a)$ è derivata da un'$ABox$, ovvero se a appartiene a C. $A \models C(a)$ se ogni interpretazione che soddisfa A soddisfa anche $C(a)$. $A \models C(a)$ se e solo se $A \cup \{\neg C(a)\}$ è inconsistente.

Recupero di istanza

Dato un concetto C si devono recuperare tutte le istanze di a che soddisfano C. Una possibile implementazione consiste nel controllare l'istanza per tutte le istanze presenti.

Realizzazione dei concetti

Dato un insieme di concetti e un individuo a si devono trovare i concetti più specifici C ordinati con le sottoassunzioni tali che $A \models C(a)$. È il problema opposto rispetto al recupero di istanza. Una possibile implementazione è rappresentata nel fare un controllo di istanza per ogni concetto.

4.6 World assumptions

4.6.1 Closed world assumption CWA

Tutto ciò che non è asserito esplicitamente è falso, viene utilizzata nei database.

4.6.2 Open world assumption OWA

Tutti ciò che non è specificatamente asserito è indecidibile, viene utilizzata nelle $ABox$.

4.7 Tableaux

4.7.1 Logica descrittiva come logica multimodale

Si può pensare alla logica descrittiva come ad una logica multimodale in cui le entità sono i mondi e sono capaci di ragionamento proposizionale. Una relazione è una relazione di accessibilità, mentre un data type è un'insieme di entità il cui comportamento è conosciuto a priori e computato dal ragionatore. Un attributo è una relazione di accessibilità a un data type. Tutti le relazioni e gli attributi che hanno un'entità nel loro dominio definiscono tutte le maniere possibili in cui quale entità ha accesso alle altre entità di mondo. Un etype è un insieme di valori di mondi che hanno lo stesso insieme di relazioni di accessibilità. $\exists R.C$ vuol dire accessibilità ad almeno un'altra entità o etype C. $\forall R.C$ vuol dire accessibilità a tutte le entità di etype C.

Mappatura del linguaggio

- $\neg$ equivalente a $\neg$.
- $\sqcap, \sqcup$ equivalente a $\wedge, \vee$.
- $\sqsubseteq$ equivalente a $\Rightarrow$.
- Le entità sono mondi.
- $TBox$ sono mondi resi espliciti nel linguaggio come variabili:
  - Etype C scritto come $C(x)$
  - Relazione R scritta come $R(x, y)$.
  - $\exists R.C$ equivalente a $\diamond_R C$ e scritto come $\exists R.C(x)$.
  - $\forall R.C$ equivalente a $\Box_R C$ scritto come $\forall R.C(x)$.

- *ABox* sono mondi resi espliciti nel linguaggio come costanti.
  - Etype C scritto come $C(a)$
  - Relazione R scritta come $R(a, b)$.
  - $\exists R.C$ equivalente a $\diamond_R C$ e scritto come $\exists R.C(a)$.
  - $\forall R.C$ equivalente a $\Box_R C$ scritto come $\forall R.C(a)$.

4.7.2 L'algoritmo dei tableaux

La formula C di input è trasformata in NNN (negation normal form) con la negazione abbassata per negare le formule atomiche in modo da evitare le regole di negazione. Una *ABox* A è costruita incrementalmente secondo i constraints in C identificati attraverso decomposizione seguendo delle precise regole di trasformazione. Ogni volta che si ha più di un'opzione si divide lo spazio delle soluzioni come in un albero di decisione. Quando si è trovata una contraddizione si deve cercare un altro cammino nello spazio delle soluzioni. L'algoritmo si ferma quando trova una A consistente che soddisfa tutti i constraints in C (soddisfacibile) o quando non esiste una A consistente (la formula è insoddisfacibile).

4.7.3 Regole di trasformazione

$\sqcap$-rule

La condizione è che A contenga $(C_1 \sqcap C_2)(x)$ ma non $C_1(x)$ e $C_2(x)$ contemporaneamente. Allora si crea $A' = A \cup \{C_1(x), C_2(x)\}$

$\sqcup$-rule

La condizione è che A contenga $(C_1 \sqcup C_2)(x)$ ma nè $C_1(x)$ nè$C_2(x)$. Allora si crea $A' = A \cup \{C_1(x)\}$ e $A'' = A \cup \{C_2(x)\}$.

$\exists$-rule

La condizione è che A contenga $(\exists R.C)(x)$ ma non esiste z tale che sia $C(z)$ che $R(x, z)$ siano in A. Allora si crea $A' = A \cup \{C(z), R(x, z)\}$.

$\forall$-rule

La condizione è che A contenga $(\forall R.C)(x)$ ma non contiene $C(z)$. Allora si crea $A' = A \cup \{C(z)\}$.

4.8 Tableaux

- Per ogni restrizione esistenziale si introduce un nuovo individuo come un role filler e questo individuo deve soddifare il constraint espresso dalla restrizione.
- L'algoritmo utilizza la restrizione di valore in interazione con ruoli già definiti per imporre nuovi constraints sugli indiviudi.
- Per i constraints per le disgiunzioni si provano entrambe le possibilità in ptrove successive.