Consiglio Nazionale delle Ricerche

Istituto di Elaborazione dell'Informazione



# Sulla complessita' della sussunzione in linguaggi a frame con restrizioni numeriche

Fabrizio Sebastiani & Umberto Straccia

Nota Interna IEI-B4-38-1990

Via S. Maria, 46 - 56126 Pisa (Italy)
Phone : +39.(0)50.553159
Fax : +39.(0)50.554342
Telex : 590305 IEICNR I

# Sulla complessità della sussunzione in linguaggi a frame con restrizioni numeriche

Fabrizio Sebastiani
Istituto di Elaborazione dell'Informazione
Consiglio Nazionale delle Ricerche
Via S. Maria, 46 - 56126 Pisa (Italy)

Umberto Straccia
Dipartimento di Informatica
Università degli Studi di Pisa
Corso Italia, 40 -56126 Pisa (Italy)

Nota Interna IEI-B4-38-1990

Maggio 1990

**Abstract**

Tra i linguaggi di rappresentazione della conoscenza più popolari oggi in uso vi sono certamente quelli basati sulla nozione di *frame*, detti *Frame Representation Languages* (FRLs); tali linguaggi permettono di rappresentare conoscenza di natura tassonomica mediante costrutti fortemente strutturati. Ultimamente sono apparsi nella letteratura svariati risultati relativi alla complessità computazionale del calcolo della *sussunzione* nei FRLs. La maggior parte di questi risultati sono purtroppo "negativi", nel senso che mostrano che i FRLs presi in considerazione sono computazionalmente intrattabili. In questa nota, dopo una breve rassegna riguardante i concetti fondamentali che stanno alla base dei FRLs e i risultati attualmente noti e relativi al calcolo della sussunzione, mostriamo che tale operazione è trattabile nel linguaggio $\mathcal{FL}^{\mathcal{N}-}$, un'estensione del linguaggio $\mathcal{FL}-$ di Brachman e Levesque. Il linguaggio $\mathcal{FL}^{\mathcal{N}-}$ è dunque il linguaggio più espressivo fra quelli fino a oggi noti per essere computazionalmente trattabili.

## 1 Introduzione

Tra i linguaggi di rappresentazione della conoscenza più popolari oggi in uso vi sono certamente quelli basati sulla nozione di *frame*, detti *Frame Representa-*

*tion Languages* (FRLs). In questi linguaggi si hanno tipicamente due costrutti di base: le *costanti individuali* e i *frame*. Le costanti individuali descrivono individui del mondo reale, mentre i frame descrivono collezioni di tali individui. Attraverso gli *slot* si possono associare alle costanti individuali delle informazioni. Ogni slot associa a una costante individuale un insieme di valori, detti *slot-filler* dello slot di quella costante individuale. I valori di uno slot sono in genere costanti individuali, o numeri. Per esempio, gli slot-filler della costante individuale *giuseppe* relativamente allo slot *Figlio* possono essere *alex* e *maria*, intendendo così descrivere il fatto che *alex* e *maria* sono figli di *giuseppe*. D'altra parte lo slot *età*, che intende descrivere la relazione tra una persona e la sua età, avrà come slot-filler un numero.

Tipicamente nei FRLs un frame specifica delle condizioni che un individuo deve soddisfare per essere un'*istanza* di esso. Quindi i frame non hanno contributo "asserzionale" (non denotano cioè un valore di verità), ma servono solo come *descrizione* di un insieme di individui. Vi sono generalmente due tipi di condizioni che possono essere specificate in un frame: condizioni del tipo *superframe* e condizioni del tipo *restrizione*. Un superframe è semplicemente un altro frame. Una costante individuale soddisfa le condizioni sui superframe di un frame se è un'istanza di ogni superframe del frame. Una restrizione è invece una condizione sugli slot-filler di uno slot. Una costante individuale soddisfa la restrizione se gli slot-filler della costante individuale soddisfano la condizione. Quindi, affinché una costante individuale sia istanza di un frame è necessario che essa sia istanza di tutti i superframe del frame e che i sui slot-filler soddisfano tutte le restrizioni.

Alcuni frame sono specificati come frame *completamente definiti*, e in tal caso le condizioni sono sia necessarie che sufficienti all'istanziazione, mentre per gli altri frame, detti frame *primitivi*, le condizioni sono solo necessarie. Questa distinzione si rende necessaria per distinguere i cosiddetti *tipi naturali*, insiemi per cui non esiste un insieme di condizioni sufficienti per determinare l'appartenenza di un individuo all'insieme, per esempio elefante, albero, da altri insiemi per cui tale insieme di condizioni sufficienti esiste, per esempio triangolo.

I vantaggi offerti dai linguaggi a frame sono notevoli: essi riproducono il modo con il quale gli esseri umani tipicamente organizzano gran parte delle loro conoscenze, appunto in tassonomie; offrono una concisa rappresentazione strutturata delle relazioni che intercorrono tra diverse entità e supportano una precisa tecnica di definizione per *specializzazione* che può essere utilizzata facilmente nella formalizzazione di svariati domini d'applicazione. Inoltre sono stati sviluppati degli algoritmi di deduzione orientati a tali linguaggi che sfruttano le caratteristiche strutturali dei frame per realizzare "rapidamente" una serie di deduzioni richieste nelle applicazioni più comuni.

Un notevole successo è stato anche raggiunto nei sistemi in cui i linguaggi a frame sono stati integrati ai linguaggi basati su regole di produzione. Questi sistemi hanno dimostrato come i linguaggi a frame possano servire come potente fondamento per i linguaggi basato su regole; i frame offrono infatti, un ricco linguaggio strutturato per descrivere gli oggetti a cui si fa riferimento nelle regole. E infine, recentemente (vedi CLASSIC [3]), il concetto di frame è stato introdotto nelle basi di dati. Il vantaggio che questo sembra di offrire è notevole, in quanto visto che i frame permettono di specificare condizioni sia necessarie che sufficienti affinché un individuo sia istanza di un frame, diventa possibile dedurre informazioni "implicite" non ottenibili mediante i sistemi di gestione di basi di dati tradizionali. Per esempio, se con il frame completamente definito *Padre* intendiamo descrivere la classe di tutte le persone che hanno almeno un figlio, allora non appena per una costante individuale, per esempio *carlo*, si specifica che essa è un'istanza del frame *Persona* e che *carlo* ha come figlio *giuseppe*, si specifica *implicitamente* che *carlo* è un'istanza del frame *Padre*. Si osservi che ciò non deve essere esplicitato, ma è invece determinato implicitamente dalla forma del frame *Padre*. Questo tipo di informazione implicita non è presente nei DB tradizionali, e quindi rappresenta un interessante campo di ricerca e punto di incontro tra KB e DB.

I primi FRLs, per esempio KL-ONE[15,9,2], erano certamente lacunosi per quanto riguardava l'aspetto semantico; tali linguaggi non possedevano infatti una precisa *truth theory*. Infatti in questi, così come in molti linguaggi orientati a oggetti di oggi, la nozione di *ereditarietà* era una nozione "procedurale". Il suo significato dipendeva cioè strettamente da un particolare algoritmo che veniva eseguito sopra una particolare struttura dati.

Notazionalmente, un frame viene descritto mediante la lista dei suoi superframe seguita dalla lista delle sue restrizioni. Per esempio il frame completamente definito

$$
\begin{array}{lll}
(\text{DEF } \mathit{Persona} & & \\
 & \text{WITH} & \mathit{Sesso} : \mathit{maschile} \\
 & \text{WITH} & \mathit{Figlio} : \geq 1)
\end{array}
$$

intende descrivere la classe di tutte le persone di sesso maschile che hanno almeno un figlio, ovvero la classe dei padri. Se questo frame ha un nome, diciamo *Padre*, allora possiamo esprimere il suo significato mediante una equivalente formula FOL:

$$
\begin{array}{lll}
\forall x\ \mathit{Padre}(x) & \Leftrightarrow & (\mathit{Persona}(x) \\
 & & \wedge\ \mathit{Sesso}(x, \mathit{maschile}) \\
 & & \wedge\ \exists y\ \mathit{Figlio}(x, y))
\end{array}
$$

Osserviamo che le restrizioni sul frame *Padre* stanno a indicare che ogni istanza di *Padre* dev'essere tale che esista almeno uno slot-filler relativamente allo slot *Figlio*, e che lo slot-filler dello slot *Sesso* sia *maschile*. Teniamo presente che *Padre* è un frame completamente definito, nel senso che un particolare individuo che, tra le altre cose, è una persona di sesso maschile e con almeno un figlio, è automaticamente un'istanza del frame *Padre*.

Similmente si possono descrivere anche classi di individui più complesse, come per esempio quella delle "persone i cui figli sono dottori, i figli dei quali sono a loro volta studenti d'informatica", che è descritta mediante il frame $F$ completamente definito

(DEF *Persona*
    WITH *Figlio*:
        (*Dottore*
            WITH *Figlio*:
                (*Studente*
                    WITH *Dipartimento*: *Informatica*)))

La relazione esistente tra diversi frame dà luogo a una tassonomia. Questa non deve essere necessariamente esplicitata, ma è determinata *implicitamente* dalle relazioni di contenimento fra le classi degli individui descritti dai frame; ovvero esiste una relazione di *sussunzione* tra tali frame. Informalmente, un frame $F_2$ è sussunto da un frame $F_1$ se e solo se, in virtù delle rispettive forme di $F_1$ e $F_2$, ogni istanza di $F_2$ è anche un'istanza di $F_1$, ovvero la classe degli individui descritta dal primo è una sottoclasse della classe degli individui descritta dal secondo. Va sottolineato che tale relazione è determinabile facendo unicamente riferimento alla forma dei frame (si veda appendice A).

Il calcolo della sussunzione è sicuramente un'operazione basilare nei FRLs (si veda paragrafo 3). Difatto, una delle interrogazioni che vengono rivolte più frequentemente a un FRL è quella di determinare se un frame $F_2$ è sussunto da un frame $F_1$. Il calcolo della sussunzione è anche strumentale all'esecuzione di un'altra operazione di fondamentale importanza nei FRLs, detta *classificazione*. In generale, in tutti i sistemi che permettono a una tassonomia di essere aggiornata dinamicamente, si usa "classificare" un frame nel momento in cui questo viene definito: "classificare" vuol dire porre virtualmente il frame nella "giusta" posizione nella gerarchia; ciò corrisponde a calcolare tutte le relazioni di sussunzione che intercorrono tra il frame e tutti gli altri frame già precedentemente definiti. Osserviamo che una tassonomia "statica" potrebbe essere costruita "manualmente" una volta per sempre, mentre una tassonomia che richieda continui aggiornamenti, come in genere si verifica nelle applicazioni, richiede invece, da parte del sistema, l'abilità di classificare automaticamente i nuovi frame nella tassonomia preesistente.

Un altra possibile interrogazione[1] che è strettamente legata al calcolo della sussunzione è la determinazione delle condizioni (superframe e restrizioni) specificate relativamente a un frame . Infatti, se un frame $A$ è sussunto da un frame $B$ (ovvero tutte le istanze di $A$ sono istanze di $B$), allora per il frame $A$ valgono anche tutte le condizioni che valgono per $B$. Diremo che A *eredita* le condizioni di B.

[1] Altre interrogazioni tipiche dei FRLs sono la determinazione della (*disgiunzione*) tra frame e la determinazione (*inconsistenza*). Anche queste interrogazioni sono riconducibili alla sussunzione.

In generale, la maggior parte delle interrogazioni che sono di interesse relativamente a un sistema basato su un FRL sono strettamente dipendenti dal calcolo della sussunzione. È auspicabile quindi avere un algoritmo efficiente per calcolare la sussunzione. Purtoppo non sempre ciò è possibile. Come vedremo, ciò dipende fortemente dall'espressività del linguaggio; in certi casi, come ad esempio nel linguaggio NIKL[13], determinare se un frame $F_1$ è sussunto da un frame $F_2$ è addirittura indecidibile.

# 2 Descrizione formale del FRL $\mathcal{SFL}$

## 2.1 Sintassi di $\mathcal{SFL}$

Il FRL formale che consideriamo, $\mathcal{SFL}$ (Sample Frame Language), ha la seguente sintassi:

$$
\begin{array}{lcl}
< \mathit{frame} > & ::= & < \mathit{atom} > \\
 & & \mid (\mathbf{and}\ < \mathit{frame} >^{+}) \\
 & & \mid (\mathbf{all}\ < \mathit{slot} > < \mathit{frame} >) \\
 & & \mid (\mathbf{atleast}\ < \mathit{minimum} > < \mathit{slot} >) \\
 & & \mid (\mathbf{atmost}\ < \mathit{maximum} > < \mathit{slot} >) \\
< \mathit{minimum} > & ::= & < \mathit{positive\ integer} > \\
< \mathit{maximum} > & ::= & < \mathit{positive\ integer} > \\
< \mathit{slot} > & ::= & < \mathit{atom} > \\
 & & \mid (\mathbf{andslot}\ < \mathit{slot} >^{+}) \\
 & & \mid (\mathbf{restr}\ < \mathit{slot} > < \mathit{frame} >)
\end{array}
$$

Per descrivere il significato dei vari costrutti del linguaggio daremo prima una descrizione informale e corredata da alcuni esempi. Nel prossimo paragrafo ne daremo poi una specifica formale in stile *model-theoretic*. Il costrutto (**and** $F_1$ ... $F_n$) rappresenta la congiunzione di frame. Per esempio, (**and** *Donna Studente*) descrive l'intersezione fra la classe delle donne e quella degli studenti (quindi la classe delle studentesse). In generale un individuo $x$ è un (**and** $F_1$ $F_2 \ldots$ $F_n$) se e solo se $x$ è sia un $F_1$, che un $F_2$, ..., che un $F_n$. Questo costrutto ci permette di specificare sia i superframe che le restrizioni, di un dato frame.

Il costrutto (**all** $S$ $F$) ci permette di specificare una restrizione sui valori assumibili dagli slot-filler dello slot $S$; secondo tale restrizione tutti questi slot-filler devono essere degli $F$. Per esempio, (**all** *Figlio Maschio*) descrive la classe degli individui i cui figli sono tutti maschi. In generale un $x$ è un (**all** $S$ $F$) se e solo se tutti gli $S$ di $x$ sono degli $F$.

I costrutti (**atleast** *min* $S$) e (**atmost** *max* $S$) ci permettono di specificare una restrizione sul numero di slot-filler che uno slot può avere. Per esempio, (**atleast** 3 *Figlio*) descrive la classe di tutti gli individui che hanno almeno tre figli. In generale $x$ è un (**atleast** *min* $S$) se e solo se $x$ ha almeno *min* $S$. Analogamente, (**atmost** 3 *Figlio*) descrive la classe di tutti gli individui che hanno al più tre figli. In generale $x$ è un (**atmost** *max* $S$) se e solo se $x$ ha al più *max* $S$. Osserviamo che nel caso di *min* = 1 nella restrizione **atleast** questo costrutto si riduce al costrutto "esistenziale" **some** presente, per esempio, in FL[4,8].

Il costrutto (**andslot** $S_1 \ldots S_n$) è un costrutto di congiunzione tra slot simile al costrutto di congiunzione tra frame. Per esempio, (**and** *Ama Sposato*) rappresenta la congiunzione della relazione *Ama* e *Sposato*, corrispondente all'insieme delle coppie che sono sia nella relazione *Ama* che nella relazione *Sposato*. In generale $< x, y >$ è un (**andslot** $S_1 \ldots S_n$) se e solo se $< x, y >$ è un $S_i$, per $1 \leq i \leq n$.

Il costrutto (**restr** $S$ $F$) permette di restringere la relazione descritta dallo slot $S$ a un suo sottinsieme. Così, per esempio, (**restr** *Figlio Maschio*) corrisponde allo slot "figlio maschio di". In generale $< x, y >$ è un (**restr** $S$ $F$) se e solo se $x$ ha un $S$ che è $y$ e $y$ è un $F$. Ad esempio, il frame che avevamo preso ad esempio nella parte introduttiva di questo capitolo e che intendeva descrivere la classe delle persone di sesso maschile che hanno almeno un figlio, può essere descritto in $\mathcal{SFL}$ con:

```
(and Persona
     (all Sesso Maschile)
     (atleast 1 Figlio))
```

Alternativamente, la classe di tutte le persone i cui figli sono dottori, i figli dei quali sono a loro volta studenti d'informatica, può essere descritta attraverso il frame:

```
(and Persona
     (all Figlio
        (and Dottore
             (all Figlio
                ( and Studente
                      (all Dipartimento Informatica))))))
```

Volendo poi usare anche l'operatore **restr**, possiamo descrivere la classe di tutte le persone i cui figli adulti sono sposati con:

```
(and Persona
     (all (restr Figlio Adulto) Sposato))
```

## 2.2 Semantica formale di $\mathcal{SFL}$

Abbiamo già descritto informalmente quale sia il significato dei costrutti di $\mathcal{SFL}$. Tale specifica, nonostante sia sufficientemente intuitiva, non è però rigorosa.

Dare una semantica formale per formalismi di rappresentazione si dimostra utile per molte ragioni. Ad esempio, essa costituisce un mezzo per discutere e criticare un formalismo di rappresentazione ed è un termine di confronto per l'implementazione. A tal fine la semantica deve essere data in uno stile che sia comunemente accettato e compreso, come per esempio quello denotazionale. Un modo alternativo per dare la semantica di un formalismo $\mathcal{F}_1$ è di tradurre questo in un altro formalismo $\mathcal{F}_2$ per il quale già esiste una semantica ben definita[2].

Vediamo di descrivere brevemente i due principali approcci generalmente usati per dare la semantica di FRLs[3].

**Approccio assiomatico:** In questo approccio la semantica $S_1$ di un formalismo $\mathcal{F}_1$ viene data mediante una traduzione $T$ delle espressioni di $\mathcal{F}_1$ in espressioni di un altro formalismo $\mathcal{F}_2$ per il quale già esiste una semantica ben definita. In questo modo si fa implicito riferimento alla semantica di $S_2$ di $\mathcal{F}_2$, e $S_1 = S_2(T)$. Questo significa che la semantica viene data indirettamente. Il vantaggio di questo approccio è la sua semplicità, soprattutto quando $\mathcal{F}_2$ è un formalismo universalmente noto (come ad esempio FOL).

**Approccio model-theoretic:** In questo approccio ci si propone di attribuire a ciascuna espressione del linguaggio una *denotazione* (o *estensione*); in particolare, nei FRLs si attribuiscono estensioni a frame e slot. Dato un dominio di individui $\mathcal{D}$, a ogni frame viene associato un sottinsieme di $\mathcal{D}$ che corrisponde all'insieme dei particolari individui descritti dal frame, mentre a ogni slot viene associato un sottinsieme di $\mathcal{D} \times \mathcal{D}$. Questo è l'approccio più diretto, in quanto specifichiamo il significato di un'espressione in termini della sua estensione senza passare attraverso altri formalismi.

Diamo qui di seguito la semantica di $\mathcal{SFL}$ usando l'approccio model-theoretic. In questa semantica a ogni espressione viene associato l'insieme di individui (individui per frame e coppie di individui per gli slot) che essa descrive nell'interpretazione data (*estensione* dell'espressione). Questa semantica è di tipo *composizionale*, nel senso che l'estensione di un'espressione complessa è solamente funzione delle estensioni delle espressioni componenti e della semantica degli operatori ivi contenuti. Per esempio, l'estensione di (**and** $F_1$ $F_2$) sarà data dall'intersezione delle estensioni di $F_1$ e di $F_2$. Quando l'estensione di un frame è inclusa nell'estensione di un altro, diremo che il primo è *sussunto* dal secondo. Formalmente abbiamo la

**Definizione 1**
*Sia $\mathcal{D}$ un insieme di individui e $\mathcal{E}$ una funzione da frame a sottinsiemi di $\mathcal{D}$, da slot a sottinsiemi di $\mathcal{D} \times \mathcal{D}$, da costanti per individui in $\mathcal{D}$. $\mathcal{E}$ è una* **funzione di estensione** *su $\mathcal{D}$ se e solo se :*

1. $\mathcal{E}[(\textbf{and}\ F_1 \ldots\ F_n)] \ = \ \{x \in \mathcal{D} | x \in \bigcap_{i=1}^{n} \mathcal{E}[F_i]\}$
2. $\mathcal{E}[(\textbf{all}\ S\ F)] \ = \ \{x \in \mathcal{D} | \forall y < x, y > \in \mathcal{E}[S] \Rightarrow y \in \mathcal{E}[F]\}$

[2] A rigore potremmo dire che il formalismo $\mathcal{F}_1$ è una variazione notazionale o un sottinsieme del formalismo $\mathcal{F}_2$.

[3] Un terzo approccio è quello *proof-theoretic*, che costituisce, in un certo modo, una semantica "implicita".

3. $\mathcal{E}[(\textbf{atleast}\ min\ S)] = \{x \in \mathcal{D} | \ \| \{y \in \mathcal{D} | \ <x,y> \in \mathcal{E}[S]\} \| \geq min\}$

4. $\mathcal{E}[(\textbf{atmost}\ max\ S)] = \{x \in \mathcal{D} | \ \| \{y \in \mathcal{D} | \ <x,y> \in \mathcal{E}[S]\} \| \leq max\}$

5. $\mathcal{E}[(\textbf{restr}\ S\ F)] = \{<x,y> \in \mathcal{D} \times \mathcal{D} | \ <x,y> \in \mathcal{E}[S] \ \wedge \ y \in \mathcal{E}[F]\}$

6. $\mathcal{E}[(\textbf{andslot}\ S_1 \ldots S_n)] = \{<x,y> \in \mathcal{D} \times \mathcal{D} | \ <x,y> \in \bigwedge_{i=1}^{n} \mathcal{E}[S_i]\}$.

La sussunzione può essere definita attraverso la nozione di estensione con la

**Definizione 2**
*Siano $F_1$ e $F_2$ due frame. Diremo che $F_2$ è* **sussunto** *da $F_1$ se e solo se per ogni insieme di individui $\mathcal{D}$ e ogni funzione di estensione $\mathcal{E}$ su $\mathcal{D}$ vale: $\mathcal{E}[F_2] \subseteq \mathcal{E}[F_1]$*

Vediamo un esempio di sussunzione tra frame. Consideriamo il frame

$F_1$ =(**and** *Persona*
(**all** (**restr** *Amico Dottore*)
(**atleast** 1 *Specializzazione*)))

che intende descrivere la classe di tutte le persone i cui amici dottori hanno almeno una specializzazione, e consideriamo

$F_2$=(**and** *Persona*
(**all** *Amico* (**atleast** 1 *Specializzazione*)))

che intende descrivere la classe di tutte le persone i cui amici hanno almeno una specializzazione.

Chiaramente una persona i cui amici hanno almeno una specializzazione è anche una persona i cui amici dottori hanno almeno una specializzazione, ovvero ci aspettiamo che $F_2$ sia sussunto da $F_1$. Possiamo verificare che ciò infatti accade osservando che se $\mathcal{D}$ è un qualsiasi insieme di individui e $\mathcal{E}$ una qualsiasi funzione di estensione su $\mathcal{D}$, accade che

$$\begin{aligned}
\mathcal{E}[F_2] &= \{x \in \mathcal{D} | x \in \mathcal{E}[Persona] \\
&\quad \cap \mathcal{E}[(\textbf{all}\ Amico\ (\textbf{atleast}\ 1\ Specializzazzione))]\} \\
&\subseteq \{x \in \mathcal{D} | x \in \mathcal{E}[Persona] \\
&\quad \cap \mathcal{E}[(\textbf{all}\ (\textbf{restr}\ Amico\ Dottore)(\textbf{atleast}\ 1\ Specializzazzione))]\} \\
&= \mathcal{E}[F_1]
\end{aligned}$$

Quindi $F_2$ è sussunto da $F_1$[4]. Si osservi che in questo caso siamo in presenza di sussunzione "implicita" anziché "esplicita", come spiegato nel paragrafo 1. Inoltre $F_1$ non è sussunto da $F_2$ in quanto un'istanza di $F_1$ potrebbe avere un amico non dottore senza specializzazione.

[4]Si osservi infatti che $\mathcal{E}[(\textbf{restr}\ Amico\ Dottore)] \subseteq \mathcal{E}[Amico]$. Questo esempio mostra come esista anche una tassonomia tra slot.

Teniamo presente che l'uso di restrizioni sul numero di slot-filler di uno slot può portare alla definizione di frame che hanno estensione sempre vuota. Infatti basta considerare

$F$=(**and**(**atleast** 2 S)
(**atmost** 1 S))

la cui estensione è data da

$$\begin{aligned} \mathcal{E}[F] &= \{x \in \mathcal{D} | \ \| \{y \in \mathcal{D} | \ < x, y > \in \mathcal{E}[S]\} \| \leq 1\} \\ &\cap \ \{x \in \mathcal{D} | \ \| \{y \in \mathcal{D} | \ < x, y > \in \mathcal{E}[S]\} \| \geq 2\} = \emptyset \end{aligned}$$

## 3 FRLs e trattabilità della sussunzione

Abbiamo visto come la determinazione della relazione di sussunzione giochi un importante ruolo nei FRLs. Pertanto è auspicabile avere un algoritmo che la calcoli in tempo ragionevole. Sfortunatamente, determinare se un frame è sussunto da un altro sembra essere un'operazione inerentemente complessa. Questo fatto non era ancora noto quando vennero sviluppati algoritmi di sussunzione per i primi FRLs. Ad esempio, l'algoritmo di sussunzione per KL-ONE, sviluppato da Lipkis[9], era un semplice algoritmo di complessità polinomiale e che inizialmente era creduto completo[5]. Successivamente, quando venne data una semantica formale per KL-ONE (Schmolze e Israel [15]), si scoprì che l'algoritmo di Lipkis era corretto ma non completo.

Recentemente, Levesque e Brachman [4,8] hanno mostrato che la sussunzione può essere intrattabile anche in FRLs piuttosto poveri in espressività. Difatto, essi hanno considerato il linguaggio $\mathcal{FL}$ = {**and**, **all**, **some**, **restr**}[6] e il linguaggio $\mathcal{FL}^-$ ={**and**, **all**, **some**}. La differenza tra $\mathcal{FL}^-$ e $\mathcal{FL}$ sembra piccola dal punto di vista dell'espressività. Ma dal punto di vista della complessità vi è un'enorme differenza. Infatti in [8] viene dato un algoritmo corretto e completo per $\mathcal{FL}^-$, che calcola la sussunzione in O($n^2$), dove $n$ è la lunghezza dell'argomento più lungo[7]. Il problema della sussunzione in $\mathcal{FL}$ è invece co-NP-Hard. $\mathcal{FL}^-$ può essere esteso aggiungendo gli operatori **atleast** e **atmost**; chiameremo questo linguaggio $\mathcal{FL}^{\mathcal{N}-}$. Il risultato del nostro lavoro è che anche per $\mathcal{FL}^{\mathcal{N}-}$ esiste un algoritmo di sussunzione corretto e completo di complessità O($n^2$), e che costituisce uno dei pochi risultati "positivi" relativamente al calcolo della sussunzione nei FRLs. Infatti, abbiamo dimostrato (per la dimostrazione si veda la sezione A) il seguente

[5] Se SUBS?[$F_1, F_2$] è un algoritmo di decisione, esso si dirà *corretto* se e solo se da SUBS?[$F_1, F_2$]=*true* segue che $F_2$ è sussunto da $F_1$; mentre si dirà *completo* se e solo se dal fatto che $F_2$ è sussunto da $F_1$ segue che SUBS?[$F_1, F_2$]=*true* .

[6] Per semplicità, in questa sede identifichiamo un linguaggio con l'insieme dei suoi operatori.

[7] Per la precisione l'algoritmo determina la sussunzione tra due frame $F_1$ e $F_2$ in $O(|F_1||F_2|)$ dove $|F_i|$ è la lunghezza del frame $F_i$.

**Teorema 1**
*Dati due frame $F_1$ e $F_2$ nel linguaggio $\mathcal{FL}^{\mathcal{N}-}$, la relazione di sussunzione tra $F_1$ e $F_2$ è determinabile in $O(|F_1||F_2|)$.*

Questo teorema conferma anche l'ipotesi di von Luck e di Owsnicki-Klewe, i quali congetturavano che $\mathcal{FL}^{\mathcal{N}-}$ avesse complessità quadratica [19]. Notiamo come $\mathcal{FL}^{\mathcal{N}-}$ sia il linguaggio più espressivo fra quelli a tutt'oggi conosciuti come trattabili. Il problema della sussunzione in $\mathcal{FL}$ è invece co-NP-Hard. Questo mostra come piccole variazioni espressive possono avere grosse ripercussioni sulla trattabilità.

Teniamo presente che $\mathcal{FL}$ è anche un sottinsieme dei FRLs KL-ONE, NIKL, e KL-TWO, e quindi anche in questi linguaggi la sussunzione è co-NP-Hard. A un per certi versi risultato simile è giunto Nebel [10], mostrando che la sussunzione è co-NP-Hard per i linguaggi KANDOR [14] e BACK [18]. La dimostrazione si basa sul fatto che, se consideriamo il FRL $\mathcal{FL}^{\mathcal{N}} = \mathcal{FL}^{\mathcal{N}-} \cup$ {**andslot**}, che è un sottinsieme di KANDOR [14] e BACK [18], anche questo risulta essere co-NP-Hard[8]. Se poi aumentiamo l'espressività fino ad ottenere il linguaggio descritto in [13], otterremmo che la sussunzione è indecidibile. Osserviamo che in tutti questi casi la sussunzione si basa sulla semantica tradizionale a due valori di verità. Si può guadagnare in espressività senza perdere in trattabilità usando una *semantica a quattro valori di verità*[9], come discusso in [11,12]. In questo caso però la nozione di sussunzione è più "debole", nel senso che se un frame è sussunto da un altro, allora lo è anche rispetto alla semantica tradizionale, mentre non vale necessariamente l'inverso. È inoltre discutibile l'utilità di un linguaggio che si basi su una semantica fondamentalmente diversa da quelle, quasi universalmente accettate, a due valori.

Determinare la sussunzione in un linguaggio ragionevolmente espressivo si dimostra quindi alquanto difficoltoso. Vi è una specie di "barriera" tra espressività e trattabilità computazionale: se la sussunzione dev'essere trattabile, allora il linguaggio dev'essere comparabilmente povero dal punto di vista espressivo. Potremmo quindi dire che la scelta di un FRL per una specifica applicazione comporta la scelta di una delle opzioni seguenti:

- Limitare l'espressività del linguaggio. Questa scelta può essere fatta solo nel caso in cui il dominio d'applicazione non richieda un notevole potere espressivo. Vanno in questo caso usati solo operatori che non danno origine a intrattabilità, come, per esempio, quelli in $\mathcal{FL}^{\mathcal{N}-}$.

- Avere solo un algoritmo di sussunzione parziale, nel senso che l'algoritmo di sussunzione a cui si fa riferimento è corretto ma non completo. Certamente la correttezza è una condizione necessaria, mentre in certe appli-

[8] Si noti ancora una volta come piccole variazioni del linguaggio possono avere notevoli ripercussioni sulla complessità.

[9] Semantiche a quattro valori di verità sono discusse in [1,6,5]. Applicazioni si trovano per esempio in [11,12,17,16,7].

| *Linguaggi* | *Complessità* | *Fonte* |
|---|---|---|
| KL-ONE | co-NP-Hard | [4,8] |
| NIKL | indecidibile | [13] |
| KANDOR | co-NP-Hard | [10] |
| BACK | co-NP-Hard | [10] |
| $\mathcal{FL}^-$ | $O(n^2)$ | [4,8] |
| $\mathcal{FL}$ | co-NP-Hard | [4,8] |
| $\mathcal{FL}^{\mathcal{N}-}$ | $O(n^2)$ | questa nota |
| $\mathcal{FL}^{\mathcal{N}}$ | co-NP-Hard | [10] |

Table 1: Complessità della sussunzione.
Notiamo che la maggior parte dei linguaggi sufficientemente espressivi ha una complessità non indifferente.

cazioni può essere accettabile che l'algoritmo non necessariamente calcoli tutte le istanze della relazione di sussunzione. Questa soluzione viene usata per esempio in KL-ONE, NIKL e BACK.

- Usare una semantica più debole, come quella a quattro valori; ciò può portare a un comportamento scarsamente intuitivo da parte del sistema, ma offre il vantaggio di poter aumentare l'espressività senza perdere troppo in trattabilità. In un certo senso, questo approccio è un caso particolare del precedente.

La tabella riassuntiva 1 descrive la complessità della sussunzione nei vari linguaggi KL-ONE, NIKL, KRYPTON, KANDOR, BACK, $\mathcal{FL}^-$, $\mathcal{FL}$, $\mathcal{FL}^{\mathcal{N}-}$ e $\mathcal{FL}^{\mathcal{N}}$.

# 4 Conclusione

Nella recente letteratura riguardante i FRLs sono apparsi svariati risultati che riguardano la complessità computazionale del calcolo della sussunzione. Questi risultati hanno mostrato che la complessità di tale operazione sembra crescere drasticamente al crescere del potere espressivo di un FRL, nel senso che è talvolta sufficiente aggiungere un solo operatore per passare da una situazione di trattabilità a una situazione di intrattabilità. Purtroppo questi risultati sono in massima parte di natura negativa, in quanto mostrano che, a meno che $P = NP$, i linguaggi presi di volta in volta in considerazione sono intrattabili. Pochi sono invece i risultati "positivi", risultati cioè che indichino come determinati linguaggi abbiano algoritmi polinomiali. In questa nota abbiamo descritto un risultato "positivo" interessante: abbiamo cioè dimostrato che il linguaggio $\mathcal{FL}^{\mathcal{N}-}$, un soprainsieme proprio del linguaggio $\mathcal{FL}^{-}$ di Brachman e Levesque (che prima del nostro risultato era il linguaggio più espressivo fra quelli noti per essere trattabili), ha complessità quadratica. Il risultato è notevole perchè $\mathcal{FL}^{\mathcal{N}-}$ estende $\mathcal{FL}^{-}$ con due fra i costrutti più interessanti presenti nella letteratura sui FRLs, i costrutti **atmost** e **atleast**. In un prossimo lavoro ci proponiamo di investigare la complessità del linguaggio ottenuto aggiungendo il ben noto operatore **slot-chain** a $\mathcal{FL}^{\mathcal{N}-}$; congetturiamo che il linguaggio risultante abbia ancora complessità polinomiale.

# A La complessità computazionale della sussunzione nel linguaggio $\mathcal{FL}^{\mathcal{N}-}$

Il linguaggio $\mathcal{FL}^{\mathcal{N}-}$ ha la seguente sintassi:

$$
\begin{array}{lcl}
< \mathit{frame} > & ::= & < \mathit{atom} > \\
 & & \mid (\textbf{and} \ < \mathit{frame} >^{+}) \\
 & & \mid (\textbf{all} \ < \mathit{slot} > < \mathit{frame} >) \\
 & & \mid (\textbf{atleast} \ < \mathit{minimum} > < \mathit{slot} >) \\
 & & \mid (\textbf{atmost} \ < \mathit{maximum} > < \mathit{slot} >) \\
< \mathit{minimum} > & ::= & < \mathit{positive\ integer} > \\
< \mathit{maximum} > & ::= & < \mathit{positive\ integer} > \\
< \mathit{slot} > & ::= & < \mathit{atom} >
\end{array}
$$

La semantica di $\mathcal{FL}^{\mathcal{N}-}$ è la seguente:

**Definizione 3**
*Sia $\mathcal{D}$ un insieme di individui ed $\mathcal{E}$ una funzione da frame a sottoinsiemi di $\mathcal{D}$ e da slot a sottoinsiemi di $\mathcal{D} \times \mathcal{D}$. $\mathcal{E}$ è una* **funzione di estensione** *su $\mathcal{D}$ se e solo se :*

1. $\mathcal{E}[(\textbf{and}\ F_1 \dots\ F_n)] \ = \ \{x \in \mathcal{D} | x \in \bigcap_{i=1}^{n} \mathcal{E}[F_i]\}$
2. $\mathcal{E}[(\textbf{all}\ S\ F)] \ = \ \{x \in \mathcal{D} | \forall y < x, y > \in \mathcal{E}[S] \Rightarrow y \in \mathcal{E}[F]\}$
3. $\mathcal{E}[(\textbf{atleast}\ min\ S)] \ = \ \{x \in \mathcal{D} | \ \| \{y \in \mathcal{D} | \ < x, y > \in \mathcal{E}[S]\} \| \geq min\}$
4. $\mathcal{E}[(\textbf{atmost}\ max\ S)] \ = \ \{x \in \mathcal{D} | \ \| \{y \in \mathcal{D} | \ < x, y > \in \mathcal{E}[S]\} \| \leq max\}$

La sussunzione può essere definita attraverso la nozione di estensione con la

**Definizione 4**
*Siano $F_1$ e $F_2$ due frame. Diremo che $F_2$ è* **sussunto** *da $F_1$ se e solo se per ogni insieme di individui $\mathcal{D}$ e ogni funzione di estensione $\mathcal{E}$ su $\mathcal{D}$ vale: $\mathcal{E}[F_2] \subseteq \mathcal{E}[F_1]$.*

Mostriamo che esiste un algoritmo che calcola in tempo polinomiale se fra due frame $F_1$ e $F_2$ in $\mathcal{FL}^{\mathcal{N}-}$ esiste una relazione di sussunzione. Questo risultato conferma la congettura di von Luck *et al.* [19] e migliora sostanzialmente il risultato di Brachman & Levesque [4], i quali avevano dato un algoritmo polinomiale per il sottinsieme di $\mathcal{FL}^{\mathcal{N}-}$ (detto $\mathcal{FL}^-$) consistente dei soli operatori **and**, **all** e **some**.

Fondamentalmente, l'algoritmo per il calcolo della sussunzione che considereremo si suddivide in due parti: la prima parte effettua la riduzione di un frame in una forma normale, mentre la seconda parte valuta se esiste una relazione di sussunzione fra due frame in forma normale. L'algoritmo si basa essenzialmente sull'algoritmo di sussunzione per $\mathcal{FL}^-$ presentato in [8].

Nella riduzione di un frame in forma normale, faremo uso di due altri operatori **void** e **card**, la cui sintassi è data estendendo la definizione di frame in modo tale che

- (**void**) è un frame;
- se $C$ un insieme di coppie di numeri naturali $< n_1, n_2 >$, dove $n_2$ può eventualmente assumere il valore $\infty$, e $S$ uno slot, allora (**card** $C$ $S$) è un frame

e la cui semantica è ottenuta modificando la definizione 3 mediante le seguenti clausole:

- $\mathcal{E}[(\textbf{void})] = \emptyset$;
- $\mathcal{E}[(\textbf{card}\ C\ S)] = \{x \in \mathcal{D} | \bigwedge_{<n_1,n_2>\in C} (n_1 \leq \| \{y \in \mathcal{D} | < x, y > \in \mathcal{E}[S]\} \| \leq n_2)\}$

Quindi (**void**) corrisponde al "frame vuoto". Invece (**card** $\{< n_1, n_2 >\}$ $S$) denota l'insieme degli individui che hanno almeno $n_1$ e al più $n_2$ $S$. Si osservi che questi due operatori sono solo strumentali alla dimostrazione del risultato di complessità, e non fanno parte del linguaggio di rappresentazione vero e proprio.

**Definizione 5**
*La* **profondità** *di un frame è data dal massimo numero di annidamenti dell'operatore* **all**.

**Definizione 6**
*Un frame $F$ è in* **forma normale** *se e solo se è della forma* (**void**) *oppure della forma* (**and** $a_1 \ldots a_n$) *e ogni $a_i$ è un fattore normale; un frame è un* **fattore normale** *se e solo se è un atomo, o se è della forma* (**all** $S$ $a$) *dove $a$ è un frame in forma normale diversa da* (**void**), *oppure se è della forma* (**card** $\{< n_1, n_2 >\}$ $S$) *con $n_1 \leq n_2$ e tale che se $n_2 = \infty$, allora $n_1 > 0$. Inoltre i fattori normali $a_1, \ldots, a_n$ devono essere tali che: se $i \neq j$ e $a_i, a_j$ sono atomi, allora $a_i \neq a_j$; se $a_i$=*(**all** $S$ $a$) *e $a_j$=*(**all** $S'$ $a'$) *e $i \neq j$, allora $S' \neq S$; se*

$a_i$=(**card** $C$ $S$) *e* $a_j$=(**card** $C'$ $S'$) *e* $i \neq j$, *allora* $S' \neq S$ *e se* $a_i$=(**all** $S$ $a$), *allora non esiste* $j$ *tale che* $a_j$=(**card** $\{<0,0>\}$ $S$)[10].

**Lemma 1** *Riduzione*
*Se* $F$ *è un frame, allora esiste un algoritmo* $O(|F|^2)$ *che riduce* $F$ *in una forma normale* $F'$ *tale che* $\mathcal{E}[F] = \mathcal{E}[F']$ *per ogni funzione di estensione* $\mathcal{E}$, *e tale che* $|F'|$ *è* $O(|F|)$.

**Dimostrazione:**
L'algoritmo è il seguente:

1. si racchiuda $F$ e ogni sottoframe di $F$ in un **and**. Questo si ottiene attraverso la semplice procedura:
   **procedure Add-and**($F$);
   **case**
   $F \in \{$(**atleast** $n$ $S$), (**atmost** $n$ $S$), $P$ $\}$: **RETURN** (**and** $F$);
   $F$= (**all** $S$ $F'$): **RETURN** (**and** (**all** $S$ **Add-and**($F'$)));
   $F$= (**and** $F_1 \ldots F_n$): **RETURN** (**and Add-and**($F_1$) ...**Add-and**($F_n$));
   **endcase**
   **endprocedure**
2. si semplifichino tutti gli **and** annidati e gli atomi ridondanti; per esempio,
   (**and** $A$ (**and** $B$ $F$ $A$) $D$)
   si riduce a
   (**and** $A$ $B$ $F$ $D$);
3. si semplifichino tutti gli **all** che si riferiscono allo stesso slot e che sono componenti di top-level dello stesso **and**; per esempio,
   (**and** (**all** $S$ $A$) (**all** $S$ (**and** $B$ $F$)))
   si riduce a
   (**and** (**all** $S$ (**and** $A$ $B$ $F$)));
4. si semplifichino tutti gli **atleast** che si riferiscono allo stesso slot e che sono componenti di top-level dello stesso **and**; per esempio,
   (**and** (**atleast** $n_1$ $S$) (**atleast** $n_2$ $S$))
   si riduce a
   (**and** (**card** $\{<n_1,\infty>,<n_2,\infty>\}$ $S$ ));
   si osservi che tale operazione viene fatta anche nel caso si abbia un solo **atleast**.
5. si semplifichino tutti gli **atmost** che si riferiscono allo stesso slot e che sono componenti di top-level dello stesso **and**; per esempio,
   (**and** (**atmost** $n_1$ $S$) (**atmost** $n_2$ $S$))
   si riduce a

[10] Si osservi che $\mathcal{E}$[(**all** $S$ (**void**))] $= \{x \in \mathcal{D} | \neg\exists y <x,y> \in \mathcal{E}[S]\} = \mathcal{E}$[(**card** $\{<0,0>\}$ $S$)].

(**and** (**card** $\{< 0, n_1 >, < 0, n_2 >\}$ $S$ ));
si osservi che tale operazione viene fatta anche nel caso si abbia un solo **atmost**.

6. si semplifichino tutti i **card** che si riferiscono allo stesso slot e che sono componenti di top-level dello stesso **and**; per esempio,
(**and** (**card** $C$ $S$) (**card** $C'$ $S$))
si riduce a
(**and** (**card** $C \cup C'$ $S$ ));
7. per ogni (**card** $C$ $S$) si sostituisca l'insieme degli intervalli $C = \{< n_{1_1}, n_{2_1} >, \ldots, < n_{1_m}, n_{2_m} >\}$ con $\{< n_1, n_2 >\}$, dove $n_1 = max\{n_{1_i}\}$ e $n_2 = min\{n_{2_i}\}$ per $i = 1, \ldots, m$. Se $n_2 < n_1$, allora si sostituisca (**card** $\{< n_1, n_2 >\}$ $S$) con (**void**); per esempio,
(**card** $\{< 0,6 >, < 0,4 >, < 0,7 >, < 1,\infty >, < 2,\infty >\}$ $S$)
si riduce a
(**card** $\{< 2,4 >\}$ $S$)
mentre
(**card** $\{< 0,2 >, < 3,\infty >\}$ $S$)
si riduce a
(**void**);
8. si riducano tutti gli **and** che hanno (**void**) come fattore e quelli la cui congiunzione è inconsistente al frame vuoto (**void**); per esempio,
(**and** (**all** $S$ (**and** (**all** $P$ $F$) $B$ (**void**))))
si riduce a
(**and** (**all** $S$ (**void**)))
così come
(**and** (**all** $S$ (**void**)) (**card** $\{< n_1, n_2 >\}$ $S$))
con $n_1 > 0$ si riduce al frame vuoto[11].
9. Si riducano tutte le forme (**all** $S$ (**void**)) nella forma equivalente (**card** $\{< 0,0 >\}$ $S$); per esempio
(**and** (**all** $S$ (**void**)))
si riduce a
(**and** (**card** $\{< 0,0 >\}$ $S$))
così come
(**and** (**all** $S$ (**void**)) (**card** $\{< 0, n_2 >\}$ $S$))
si riduce a
(**and** (**card** $\{< 0,0 >\}$ $S$))[12]

Valutiamo dapprima la complessità di questo algoritmo di riduzione. Il passo 1 viene fatto in $O(|F|)$. Poiché a ogni sottoframe aggiungiamo un **and** di lunghezza costante, la lunghezza del frame ottenuto dal passo 1

[11] Non esiste individuo che contemporaneamente non deve avere e deve avere almeno un $S$.

[12] Si osservi che a questo punto non si possono più avere situazioni del tipo (and (all $S$ (void)) (card $\{< n_1, n_2 >\}$ $S$)) con $n_1 > 0$ in quanto già trattate al passo 8.

è $O(|F|)$. Il passo 2 può essere fatto in $O(|F|)$ in quanto basta scorrere il frame ottenuto al passo 1 e la lunghezza del frame ottenuto è $O(|F|)$. Il passo 3 può essere fatto in $O(|F|^2)$ in quanto si deve scorrere il frame ottenuto al passo 2 per ogni slot $S$ e la lunghezza del frame ottenuto è $O(|F|)$. Analogamente, i passi 4, 5, 6 possono essere fatti in $O(|F|^2)$ e la lunghezza del frame ottenuto è $O(|F|)$. Il passo 7 può essere fatto in $O(|F|)$ e la lunghezza del frame ottenuto è $O(|F|)$. Il passo 8 può essere fatto in $O(|F|)$ in quanto basta scorrere il frame ottenuto al passo 7 e ridurre (**and** $F_1$, ..., (**void**), ...,$F_n$) così come, per $n_1 > 0$, il frame (**and** (**all** $S$ (**void**)) (**card** $\{< n_1, n_2 >\}$ $S$)) al frame vuoto (**void**) e la lunghezza del frame ottenuto è $O(|F|)$. Infine, si vede facilmente che il passo 9 può essere fatto in $O(|F|^2)$ e la lunghezza del frame ottenuto è $O(|F|)$. Quindi la riduzione può essere fatta in $O(|F|^2)$ e $|F'|$ è $O(|F|)$.

Risulta inoltre evidente che i passi 1, ..., 8 riducono un frame $F$ in una forma normale $F'$. Mostriamo, ora, che la riduzione è invariante rispetto all'estensione di $F$, ovvero $\mathcal{E}[F] = \mathcal{E}[F']$ per ogni funzione di estensione $\mathcal{E}$. Chiaramente i passi 1 e 2 preservano banalmente l'estensione. Anche il passo 3 ha questa proprietà; infatti

$$\begin{array}{l}
\mathcal{E}[(\textbf{and}\ (\textbf{all}\ S\ F_1)\ (\textbf{all}\ S\ F_2)\ \ldots\ (\textbf{all}\ S\ F_n)\ F] = \\
\{x \in \mathcal{D} | \forall y < x, y > \in \mathcal{E}[S] \Rightarrow y \in \mathcal{E}[F_1]\} \cap \\
\{x \in \mathcal{D} | \forall y < x, y > \in \mathcal{E}[S] \Rightarrow y \in \mathcal{E}[F_2]\} \cap \ldots \\
\{x \in \mathcal{D} | \forall y < x, y > \in \mathcal{E}[S] \Rightarrow y \in \mathcal{E}[F_n]\} \cap \mathcal{E}[F] = \\
\{x \in \mathcal{D} | \forall y < x, y > \in \mathcal{E}[S] \Rightarrow y \in \bigcap_{i=1}^{n} \mathcal{E}[F_i]\} \cap \mathcal{E}[F] = \\
\mathcal{E}[(\textbf{and}\ (\textbf{all}\ S\ (\textbf{and}\ F_1\ F_2\ \ldots\ F_n))\ F)]
\end{array}$$

In modo del tutto analogo si dimostra che i passi 4, ..., 7 preservano l'estensione. Ciò vale anche per il passo 8 in quanto

$$\begin{array}{l}
\mathcal{E}[(\textbf{and}\ F_1\ \ldots\ F_{k-1}\ (\textbf{void})\ F_{k+1}\ \ldots\ F_n)] = \\
\mathcal{E}[F_1] \cap \ldots \cap \mathcal{E}[F_{k-1}] \cap \mathcal{E}[(\textbf{void})] \cap \mathcal{E}[F_{k+1}] \ldots \cap \mathcal{E}[F_n] = \\
\mathcal{E}[F_1] \cap \ldots \cap \mathcal{E}[F_{k-1}] \cap \emptyset \cap \mathcal{E}[F_{k+1}] \ldots \cap \mathcal{E}[F_n] = \emptyset = \\
\mathcal{E}[(\textbf{void})]
\end{array}$$

e inoltre per $n_1 > 0$

$$\begin{array}{l}
\mathcal{E}[(\textbf{and}\ \mathrm{F}\ (\textbf{all}\ S\ (\textbf{void}))\ (\textbf{card}\ \{< n_1, n_2 >\}\ S))] = \\
\mathcal{E}[F] \cap \{x \in \mathcal{D} | \neg\exists y < x, y > \in \mathcal{E}[S]\} \cap \\
\{x \in \mathcal{D} |\ (0 < n_1 \leq \|\ \{y \in \mathcal{D} |\ < x, y > \in \mathcal{E}[S]\}\ \| \leq n_2\} = \\
\emptyset = \mathcal{E}[(\textbf{void})]
\end{array}$$

Infine, l'analogo vale per il passo 9 in quanto

$$\mathcal{E}[(\textbf{all}\ S\ (\textbf{void}))] = \mathcal{E}[(\textbf{card}\ \{< 0, 0 >\}\ S)]$$

—◇—

Possiamo ora dare l'algoritmo di sussunzione vero e proprio. La procedura SUBS? prende come input due frame in forma normale e decide se esiste fra essi una relazione di sussunzione.

**procedure SUBS?**($F_1$,$F_2$)

**input** Due frame $F_1$, $F_2$ in forma normale;

**output** *true* se $F_2$ è sussunto da $F_1$; *false* altrimenti;

1. se $F_2 \equiv$ (**void**) restituisci *true*; altrimenti
2. se $F_1 \equiv$ (**void**) restituisci *false*; altrimenti, siano $F_1$ della forma (**and** $a_1 \ldots a_n$) e $F_2$ della forma (**and** $b_1 \ldots b_m$); restituisci *true* se e solo se per ogni $1 \leq i \leq n$ sono soddisfatte le seguenti condizioni:
3. se $a_i$ è un atomo, allora uno dei $b_j$ è $a_i$;
4. se $a_i$ è un (**all** $S$ $x$), allora uno dei $b_j$ è un (**all** $S$ $y$) e SUBS?($x$,$y$)=*true* oppure uno dei $b_j$ è un (**card** $\{< 0, 0 >\}$ $S$);
5. se $a_i$ è un (**card** $\{< n_1, n_2 >\}$ $S$), allora uno dei $b_j$ è un (**card** $\{< n_3, n_4 >\}$ $S$), e $n_1 \leq n_3$ e $n_4 \leq n_2$.

**endprocedure**

**Lemma 2** *Complessità*
*La procedura* SUBS?($F_1$, $F_2$) *ha complessità* $O(|F_1||F_2|)$.

**Dimostrazione:**
Dimostriamo il lemma per induzione sulla massima profondità $k$ dei frame $F_1$ e $F_2$.

**caso** $k = 0$**:**
Le condizioni 1 e 2 possono essere verificate in tempo costante. Le condizioni 3 e 5 possono essere verificate in tempo $O(|F_1||F_2|)$ in quanto per ogni $a_i$ si deve scorrere il frame $F_2$ per trovare il $b_j$. Si noti che la condizione 4 è verificata in quanto $k = 0$.

**passo induttivo** $k > 0$**:**
Sia vero l'enunciato per profondità $< k$. Poiché $k > 0$, i frame $F_1$ e $F_2$ sono quindi della forma (**and** $a_1 \ldots a_n$) e (**and** $b_1 \ldots b_m$). Supponiamo che fra gli $a_1, \ldots, a_n$, vi siano $\ell_1$ atomi, $\ell_2$ frame di tipo **all** e $\ell_3$ frame di tipo **card**; quindi $\ell_1 + \ell_2 + \ell_3 = n$. Il passo 3 può essere eseguito in tempo $\ell_1|F_2|$ in quanto per ogni atomo dobbiamo scorrere $F_2$. Analogamente, il passo 5 può essere eseguito in tempo $\ell_3|F_2|$. Vediamo il passo 4. Sia $a_i$ del tipo (**all** $S$ $x_i$). Innanzitutto, dobbiamo scorrere $F_2$ per trovare un $b_j$= (**all** $S$ $y_j$) oppure un $b_j$=(**card** $\{< 0, 0 >\}$ $S$), il che richiede tempo al più $|F_2|$. Il secondo caso non richiede ulteriore passi, mentre

nel primo caso, per ipotesi induttiva su $x_i$ e $y_j$, si ha che SUBS?($x_i$, $y_j$) termina in tempo $|x_i||y_j|$. Considerando che $|x_i||y_j| \leq |x_i||F_2|$, il passo 4 può essere fatto in tempo $\sum_{i=1}^{\ell_2}(|F_2|+|x_i||F_2|)$. Considerando che $|F_2|(\sum_{i=1}^{\ell_2}(1+|x_i|)) \leq |F_1||F_2|$ e che $\ell_1+\ell_3 \leq |F_1|$ si ha che SUBS?($F_1$, $F_2$) richiede al più tempo

$$\begin{aligned}
&\ell_1|F_2| + \textstyle\sum_{i=1}^{\ell_2}(|F_2|+|x_i||F_2|) + \ell_3|F_2| \\
&= \ell_1|F_2| + |F_2|(\textstyle\sum_{i=1}^{\ell_2}(1+|x_i|)) + \ell_3|F_2| \\
&\leq |F_2|(\ell_1+\ell_3) + |F_1||F_2| \\
&\leq |F_1||F_2| + |F_1||F_2| = 2|F_1||F_2|
\end{aligned}$$

che è $O(|F_1||F_2|)$.

—◇—

**Lemma 3** *Correttezza*
*Se* SUBS?($F_1$, $F_2$)=*true, allora* $F_2$ *è sussunto da* $F_1$.

**Dimostrazione:**

Supponiamo che SUBS? ritorni *true* e sia $\mathcal{E}$ una funzione di estensione. Mostriamo per induzione sulla massima profondità $k$ di $F_1$ e $F_2$ che $\mathcal{E}[F_2] \subseteq \mathcal{E}[F_1]$.

**caso k=0:**

1. se $F_2$=(**void**), allora $\emptyset = \mathcal{E}[F_2] \subseteq \mathcal{E}[F_1]$. Altrimenti, $F_1$ è della forma (**and** $a_1 \ldots a_n$) e $F_2$ è della forma (**and** $b_1 \ldots b_m$);
2. se $a_i$ è un atomo, allora esiste un $b_j = a_j$ e quindi $\mathcal{E}[b_j] \subseteq \mathcal{E}[a_i]$;
3. se $a_i$ =(**card** $\{< n_1, n_2 >\}$ $S$), allora esiste $b_j$ =(**card** $\{< n_3, n_4 >\}$ $S$) con $n_1 \leq n_3$ e $n_4 \leq n_2$. Si vede facilmente che $\mathcal{E}[b_j] \subseteq \mathcal{E}[a_i]$;

Quindi per ogni $1 \leq i \leq n$ esiste un $1 \leq j \leq m$ tale che $\mathcal{E}[b_j] \subseteq \mathcal{E}[a_i]$, e quindi che $\mathcal{E}[F_2] = \mathcal{E}[(\textbf{and}\ b_1 \ldots b_m)] \subseteq \mathcal{E}[(\textbf{and}\ b_{f(1)} \ldots b_{f(n)})] \subseteq \mathcal{E}[(\textbf{and}$ $a_1 \ldots a_n) = \mathcal{E}[F_1]$, dove $f$ è una funzione $f : \{1, \ldots, n\} \longrightarrow \{1, \ldots, m\}$ tale che $f(i) = j$.

**passo induttivo $k > 0$:**

Sia vero l'enunciato per profondità $< k$.

1. se $F_2$=(**void**), allora $\emptyset = \mathcal{E}[F_2] \subseteq \mathcal{E}[F_1]$. Altrimenti, $F_1$ è della forma (**and** $a_1 \ldots a_n$) e $F_2$ è della forma (**and** $b_1 \ldots b_m$); i casi in cui $a_i$ è un atomo oppure $a_i$ =(**card** $\{< n_1, n_2 >\}$ $S$) si dimostrano come nel caso $k = 0$.

2. se $a_i$ è un (**all** $S$ $x$), allora se esiste un $b_j$ che è un (**all** $S$ $y$) e SUBS?($x$,$y$)=*true*, allora per ipotesi induttiva su $x$ e $y$ vale $\mathcal{E}[y] \subseteq \mathcal{E}[x]$ e quindi $\mathcal{E}[b_j] = \mathcal{E}[(\textbf{all}\ S\ y)] \subseteq \mathcal{E}[(\textbf{all}\ S\ x)] = \mathcal{E}[a_i]$. Altrimenti esiste un $b_j$=(**card** $\{< 0,0 >\}$ $S$). Si vede facilmente che anche in questo caso vale $\mathcal{E}[b_j] \subseteq \mathcal{E}[a_i]$. Ne consegue che $\mathcal{E}[F_2] \subseteq \mathcal{E}[F_1]$ per gli stessi motivi del caso $k = 0$.

—◇—

Passiamo ora alla dimostrazione della completezza di SUBS? Mostriamo innanzitutto, che dato un insieme di individui $\Phi$, un frame $F$ non vuoto in forma normale e una funzione di estensione $\mathcal{E}$ definita su un dominio $\mathcal{D}$ tale che $\Phi \cap \mathcal{D} = \emptyset$, allora si può effettivamente espandere la $\mathcal{E}$ costruendo una funzione di estensione $\mathcal{E}'$ con dominio $\mathcal{D}'$ tale che $\Phi \subseteq \mathcal{D}'$ e $\Phi \subseteq \mathcal{E}'[F]$. Questo ci sarà utile in quanto mostreremo che se SUBS? ritorna *false*, allora esiste una funzione di estensione $\mathcal{E}$ e un insieme di individui $\Phi$ tale che $\Phi \subseteq \mathcal{E}[F_2]$ e $\Phi \not\subseteq \mathcal{E}[F_1]$ e quindi $F_2$ non è sussunto da $F_1$.

**Definizione 7**
*Date due funzioni di estensione $\mathcal{E}$ e $\mathcal{E}'$ definite sui domini $\mathcal{D}$ e $\mathcal{D}'$, rispettivamente. Allora $\mathcal{E}'$ è un'***espansione** *di $\mathcal{E}$ se e solo se $\mathcal{D} \subseteq \mathcal{D}'$, $\mathcal{E}[A] \subseteq \mathcal{E}'[A]$ per ogni atomo $A$, $\mathcal{E}[S] \subseteq \mathcal{E}'[S]$ per ogni slot $S$; inoltre per ogni slot $S$ e atomo $A$, per ogni $d, d' \in \mathcal{D}'$ vale*

1. *se $< d, d' > \in \mathcal{E}'[S] \setminus \mathcal{E}[S]$, allora $d, d' \in \mathcal{D}' \setminus \mathcal{D}$;*
2. *se $d \in \mathcal{E}'[A] \setminus \mathcal{E}[A]$, allora $d \in \mathcal{D}' \setminus \mathcal{D}$.*

Fondamentalmente, $\mathcal{E}'$ differisce da $\mathcal{E}$ solo per quanto riguarda gli elementi $d \in \mathcal{D}' \setminus \mathcal{D}$. Valgono i seguenti due lemmi.

**Lemma 4**
*Sia $F$ un frame in forma normale non vuoto, $\mathcal{E}$ e $\mathcal{E}'$ due funzioni di estensione con dominio $\mathcal{D}$ e $\mathcal{D}'$, rispettivamente, $\mathcal{E}'$ espansione di $\mathcal{E}$ e $\Psi \subseteq \mathcal{D}$ un insieme di individui. Se $\Psi \subseteq \mathcal{E}[F]$, allora $\Psi \subseteq \mathcal{E}'[F]$.*

**Dimostrazione:**

Se $\Psi$ è vuoto, allora si ha banalmente la tesi, altrimenti si dimostra per induzione sulla profondità $k$ di $F$. Sia $F$ della forma (**and** $a_1 \ldots a_n$) e $\Psi \subseteq \mathcal{E}[F]$. Mostriamo che per ogni $1 \leq i \leq n$, $\Psi \subseteq \mathcal{E}'[a_i]$, da cui la tesi.

**caso** $k = 0$:

1. se $a_i$ è un atomo, allora $\Psi \subseteq \mathcal{E}[a_i] \subseteq \mathcal{E}'[a_i]$ per definizione di $\mathcal{E}'$;

2. se $a_i$ è della forma (**card** $\{< n_1, n_2 >\}$ $S$), allora sia $d \in \Psi$. Poiché $d \in \mathcal{D}$ e $\mathcal{E}'$ espansione di $\mathcal{E}$ ne segue che non esiste un $d' \in \mathcal{D}'$ tale che $< d, d' > \in \mathcal{E}'[S] \setminus \mathcal{E}[S]$ e quindi $\{d' \in \mathcal{D} :< d, d' > \in \mathcal{E}[S]\} = \{d' \in \mathcal{D}' :< d, d' > \in \mathcal{E}'[S]\}$. Ne segue che

$$\begin{aligned} d \in \mathcal{E}[a_i] &= \{d \in \mathcal{D} | \ (n_1 \leq \| \{d' \in \mathcal{D} | \ < d, d' > \in \mathcal{E}[S]\} \| \leq n_2)\} \\ &= \{d \in \mathcal{D} | \ (n_1 \leq \| \{d' \in \mathcal{D}' | \ < d, d' > \in \mathcal{E}'[S]\} \| \leq n_2)\} \\ &\subseteq \{d \in \mathcal{D}' | \ (n_1 \leq \| \{d' \in \mathcal{D}' | \ < d, d' > \in \mathcal{E}'[S]\} \| \leq n_2)\} \\ &= \mathcal{E}'[a_i] \end{aligned}$$

Dall'arbitrarietà di $d \in \Psi$ segue $\Psi \subseteq \mathcal{E}'[a_i]$.

**passo induttivo $k > 0$:**

Sia vero il lemma per profondità $< k$. Sia $a_i$ della forma (**all** $S$ $a$) e $d \in \Psi$. Mostriamo che $d \in \mathcal{E}'[a_i]$.

1. se non esiste $d' \in \mathcal{D}'$ tale che $< d, d' > \in \mathcal{E}'[S]$, allora banalmente $d \in \mathcal{E}'[a_i]$;
2. se esiste $d' \in \mathcal{D}'$ tale che $< d, d' > \in \mathcal{E}'[S]$, allora siccome $d \in \mathcal{D}$ ne segue per definizione di $\mathcal{E}'$ che $< d, d' > \notin \mathcal{E}'[S] \setminus \mathcal{E}[S]$. Quindi $< d, d' > \in \mathcal{E}[S]$ e $d' \in \mathcal{D}$. Ma $d \in \mathcal{E}[a_i]$ e quindi $d' \in \mathcal{E}[a]$. Pertanto, per ipotesi induttiva su $a$, $d' \in \mathcal{E}'[a]$ e quindi $d \in \mathcal{E}'[a_i]$.

—◇—

**Lemma 5**

*Sia $F$ un frame in forma normale non vuoto, $\mathcal{E}$ e $\mathcal{E}'$ due funzioni di estensione con dominio $\mathcal{D}$ e $\mathcal{D}'$, rispettivamente, $\mathcal{E}'$ espansione di $\mathcal{E}$ e $\Psi \subseteq \mathcal{D}$ un insieme di individui non vuoto. Se $\Psi \not\subseteq \mathcal{E}[F]$, allora $\Psi \not\subseteq \mathcal{E}'[F]$.*

**Dimostrazione:**

Sia $F$ della forma (**and** $a_1 \ldots a_n$). Se $\Psi \not\subseteq \mathcal{E}[F]$, allora $\Psi \not\subseteq \bigcap_{i=1}^{n} \mathcal{E}[a_i]$. Quindi per un $d \in \Psi$ esiste un $1 \leq i \leq n$ tale che $d \notin \mathcal{E}[a_i]$. Dimostriamo per induzione sulla massima profondità $k$ dei fattori normali $a_i$ che $d \notin \mathcal{E}'[a_i]$ da cui la tesi.

**caso $k = 0$:**

1. sia $a_i$ è un atomo. Siccome $d \in \mathcal{D}$ ne segue che $d \notin \mathcal{E}'[a_i]$, in quanto altrimenti avremmo $d \in \mathcal{E}'[a_i]$, $d \notin \mathcal{E}[a_i]$ e quindi $d \in \mathcal{E}'[a_i] \setminus \mathcal{E}[a_i]$ che è assurdo per definizione di $\mathcal{E}'$;
2. sia $a_i$ è della forma (**card** $\{< n_1, n_2 >\}$ $S$). Supponiamo per assurdo che $d \in \mathcal{E}'[a_i]$. In questo caso $n_1 \leq \| \{d' \in \mathcal{D}' | \ < d, d' > \in \mathcal{E}'[S]\} \| \leq n_2$. Ma $d \in \mathcal{D}$ e quindi, come visto nel lemma precedente, $\{d' \in \mathcal{D} :< d, d' > \in \mathcal{E}[S]\} = \{d' \in \mathcal{D}' :< d, d' > \in \mathcal{E}'[S]\}$. Ne segue $n_1 \leq \| \{d' \in \mathcal{D} | \ < d, d' > \in \mathcal{E}[S]\} \| \leq n_2$, ovvero $d \in \mathcal{E}[a_i]$, assurdo.

**passo induttivo** $k > 0$:

Sia vero il lemma per profondità $< k$. Sia $a_i$ della forma (**all** $S$ $a$) e $d \notin \mathcal{E}[a_i]$. Allora esiste un $d' \in \mathcal{D} \subseteq \mathcal{D}'$ tale che $< d, d' > \in \mathcal{E}[S]$ e $d' \notin \mathcal{E}[a]$. Ma $\mathcal{E}[S] \subseteq \mathcal{E}'[S]$ e per ipotesi induttiva su $a$, $d' \notin \mathcal{E}'[a]$. Pertanto esiste un $d' \in \mathcal{D}'$ tale che $< d, d' > \in \mathcal{E}'[S]$, $d' \notin \mathcal{E}'[a]$ e quindi $d \notin \mathcal{E}'[a_i]$.

—◇—

Dai lemmi 4 e 5 si ha il

**Corollario 1**
*Sia $F$ un frame in forma normale non vuoto, $\mathcal{E}$ e $\mathcal{E}'$ due funzioni di estensione con dominio $\mathcal{D}$ e $\mathcal{D}'$, rispettivamente, $\mathcal{E}'$ espansione di $\mathcal{E}$ e $\Psi \subseteq \mathcal{D}$ un insieme di individui non vuoto. Allora $\Psi \subseteq \mathcal{E}[F]$ se e solo se $\Psi \subseteq \mathcal{E}'[F]$.*

Diamo qui la descrizione della procedura che dato un insieme di individui $\Phi$, un frame $F$ non vuoto in forma normale e una funzione di estensione $\mathcal{E}$ definita su un dominio $\mathcal{D}$ tale che $\Phi \cap \mathcal{D} = \emptyset$, espande la $\mathcal{E}$ costruendo una funzione di estensione $\mathcal{E}'$ con dominio $\mathcal{D}'$ tale che $\Phi \subseteq \mathcal{D}'$ e $\Phi \subseteq \mathcal{E}'[F]$.

**procedure Build_Extension+**($F$,$\Phi$,$\mathcal{E}$,$\mathcal{D}$)

**input:** Un frame $F$ non vuoto in forma normale, un insieme di individui $\Phi$ e una funzione di estensione $\mathcal{E}$ con dominio $\mathcal{D}$ tale che $\Phi \cap \mathcal{D} = \emptyset$.

**output:** Una funzione di estensione $\mathcal{E}'$ con dominio $\mathcal{D}'$ tale che $\mathcal{E}'$ è un'espansione di $\mathcal{E}$, $\Phi \subseteq \mathcal{D}'$ e $\Phi \subseteq \mathcal{E}'[F]$.

**beginprocedure**

Sia $\mathcal{D}' := \mathcal{D} \cup \Phi$, $\mathcal{E}'[A] := \mathcal{E}[A]$ per ogni atomo $A$ e $\mathcal{E}'[S] := \mathcal{E}[S]$ per ogni slot $S$. Sia $F$ della forma (**and** $a_1 \ldots a_n$), allora per ogni $1 \leq i \leq n$ si esegua

1. se $a_i$ è un atomo, allora sia $\mathcal{E}'[a_i] := \mathcal{E}'[a_i] \cup \Phi$;
2. se $a_i$ è della forma (**all** $S$ $u$) e nessuno degli $a_1, \ldots, a_{i-1}$ è della forma (**card** $\{< n_1, n_2 >\}$ $S$), allora
   (a) se uno degli $a_{i+1}, \ldots, a_n$ è della forma (**card** $\{< n_1, n_2 >\}$ $S$) con $n_1 > 0$, allora per dei $d_1^*, \ldots, d_{n_1}^* \notin \mathcal{D}'$ sia

   $$\Phi' := \{d_1^*, \ldots, d_{n_1}^*\}$$

   si richiami Build_Extension+($u$,$\Phi'$,$\mathcal{E}'$,$\mathcal{D}'$) e sia

   $$\mathcal{E}'[S] := \mathcal{E}'[S] \bigcup_{d \in \Phi} \{< d, d_1^* >, \ldots, < d, d_{n_1}^* >\} \qquad (1)$$

**Commento:**
L'estensione che vorremmo costruire dev'essere tale che $\Phi \subseteq \mathcal{E}'[(\textbf{all}\ S\ u)]$. Se nessuno degli $a_1, \ldots, a_n$ è un $(\textbf{card}\{< n_1, n_2 >\}\ S)$ con $n_1 > 0$, allora non siamo costretti ad associare degli $S$ ai $\Phi$. Altrimenti (caso 2), ciascuno dei $d \in \Phi$ deve avere almeno $n_1$ $S$ (formula 1) e i $d^*_h$ devono essere degli $u$ (Build_Extension+($u$,$\Phi'$,$\mathcal{E}'$,$\mathcal{D}'$)).

3. se $a_i$ è della forma $(\textbf{card}\ \{< n_1, n_2 >\}\ S)$ con $n_1 > 0$ e nessuno degli $a_1, \ldots, a_{i-1}$ è della forma $(\textbf{all}\ S\ u)$, allora
   (a) se nessuno degli $a_{i+1}, \ldots, a_n$ è della forma $(\textbf{all}\ S\ u)$, allora per dei $d^*_1, \ldots, d^*_{n_1} \notin \mathcal{D}'$ sia
   $$\mathcal{E}'[S] := \mathcal{E}'[S] \bigcup_{d \in \Phi} \{< d, d^*_1 >, \ldots, < d, d^*_{n_1} >\}$$
   $$\mathcal{D}' := \mathcal{D}' \cup \{d^*_1, \ldots, d^*_{n_1}\}$$
   (b) se uno degli $a_{i+1}, \ldots, a_n$ è della forma $(\textbf{all}\ S\ u)$, allora per dei $d^*_1, \ldots, d^*_{n_1} \notin \mathcal{D}'$ sia
   $$\Phi' := \{d^*_1, \ldots, d^*_{n_1}\}$$
   si richiami Build_Extension+($u$,$\Phi'$,$\mathcal{E}'$,$\mathcal{D}'$) e sia
   $$\mathcal{E}'[S] := \mathcal{E}'[S] \bigcup_{d \in \Phi} \{< d, d^*_1 >, \ldots, < d, d^*_{n_1} >\}$$

**endprocedure**

Vediamo un esempio. Sia $F$ definito da
$(\textbf{and}\ (\textbf{all}\ S\ A)\ (\textbf{card}\ \{< 2, 3 >\}\ S\ )\ B)$
e sia $\Phi = \{d\}$ e $\mathcal{E}$ la funzione di estensione:

1. $\mathcal{D} = \{a, b\}$;
2. $\mathcal{E}[B] = \{a\}$;
3. $\mathcal{E}[A] = \{a, b\}$ e
4. $\mathcal{E}[S] = \{< a, a >, < a, b >\}$

Si osservi che $a \in \mathcal{E}[F]$, mentre $b \notin \mathcal{E}[F]$. Allora, eseguendo la Build_Extension+($F$,$\Phi$,$\mathcal{E}$,$\mathcal{D}$) otterremo la seguente funzione di estensione $\mathcal{E}'$:

1. $\mathcal{D}' = \{a, b, d, d^*_1, d^*_2\}$;
2. $\mathcal{E}'[B] = \{a, d\}$;

3. $\mathcal{E}'[A] = \{a, b, d_1^*, d_2^*\}$ e

4. $\mathcal{E}'[S] = \{< a, a >, < a, b >, < d, d_1^* >, < d, d_2^* >\}$

Si vede facilmente che $\Phi \subseteq \mathcal{E}'[F]$ e poiché $\mathcal{D} \subseteq \mathcal{D}'$, $\mathcal{E}[A] \subseteq \mathcal{E}'[A]$ per ogni atomo $A$, $\mathcal{E}[S] \subseteq \mathcal{E}'[S]$ per ogni slot $S$, e per ogni $x, y \in \mathcal{D}'$ vale

1. se $< x, y > \in \mathcal{E}'[S] \setminus \mathcal{E}[S] = \{< d, d_1^* >, < d, d_2^* >\}$, allora $x, y \in \mathcal{D}' \setminus \mathcal{D} = \{d, d_1^*, d_2^*\}$;
2. se $x \in \mathcal{E}'[A] \setminus \mathcal{E}[A] = \{d_1^*, d_2^*\}$, allora $x \in \mathcal{D}' \setminus \mathcal{D} = \{d, d_1^*, d_2^*\}$;
3. se $x \in \mathcal{E}'[B] \setminus \mathcal{E}[B] = \{d\}$, allora $x \in \mathcal{D}' \setminus \mathcal{D} = \{d, d_1^*, d_2^*\}$.

e quindi $\mathcal{E}'$ è un'**espansione** di $\mathcal{E}$. Si osservi che vale ancora $a \in \mathcal{E}'[F]$ e $b \notin \mathcal{E}'[F]$, in accordo con il corollario 1.

**Lemma 6**
*Siano $F$=(**and** $a_1 \ldots a_n$) una forma normale, $\Phi$ un insieme di individui, $\mathcal{E}$ una funzione di estensione definita su un dominio $\mathcal{D}$ tale che $\Phi \cap \mathcal{D} = \emptyset$ e $\mathcal{E}'$ data dalla procedura Build_Extension+($F$,$\Phi$,$\mathcal{E}$,$\mathcal{D}$). Allora $\mathcal{E}'$ è un'espansione di $\mathcal{E}$.*

**Dimostrazione:**

Sia $\mathcal{E}'$ definita sul dominio $\mathcal{D}'$ data dalla procedura. In virtù degli assegnamenti nella procedura Build_Extension+, sia ha che $\mathcal{D} \subseteq \mathcal{D}'$, $\mathcal{E}[A] \subseteq \mathcal{E}'[A]$ per ogni atomo $A$, $\mathcal{E}[S] \subseteq \mathcal{E}'[S]$ per ogni slot $S$. Poiché, $\Phi \cap \mathcal{D} = \emptyset$ e negli assegnamenti del tipo (1) $d_j^* \notin \mathcal{D}$, risulta che per ogni slot $S$ e atomo $A$, per ogni $d, d' \in \mathcal{D}'$ vale

1. se $< d, d' > \in \mathcal{E}'[S] \setminus \mathcal{E}[S]$, allora $d, d' \in \mathcal{D}' \setminus \mathcal{D}$;
2. se $d \in \mathcal{E}'[A] \setminus \mathcal{E}[A]$, allora $d \in \mathcal{D}' \setminus \mathcal{D}$.

Pertanto $\mathcal{E}'$ è un'espansione di $\mathcal{E}$.

—◇—

**Lemma 7**
*Siano $F$=(**and** $a_1 \ldots a_n$) una forma normale, $\mathcal{E}$ una funzione di estensione definita su un dominio $\mathcal{D}$, $\Phi$ e $\Phi'$ due insiemi di individui tali che $\Phi \subseteq \mathcal{D}$ e $\Phi' \cap \mathcal{D} = \emptyset$, e $\mathcal{E}'$ data dalla procedura Build_Extension+($F$,$\Phi'$,$\mathcal{E}$,$\mathcal{D}$). Allora per ogni $d \in \Phi'$ non esiste una sequenza di slot $S_1, \ldots, S_n$ e una sequenza di individui $d_1, \ldots, d_{n+1}$ tali che per ogni $1 \leq i \leq n$ $< d_i, d_{i+1} > \in \mathcal{E}'[S_i]$ con $d = d_1$ e $d_{n+1} \in \Phi$ o $d_{n+1} \in \Phi'$.*

**Dimostrazione:**

Si dimostra facilmente, in quanto a ogni passo della procedura Build_Extension+ si scelgono dei nuovi $d_j^* \notin \mathcal{D}$ e quindi $d_j^* \notin \Phi$. Pertanto attraverso le equazioni del tipo (1) non si possono creare coppie del tipo $< d_j^*, d >$ con $d \in \Phi$ o $d_{n+1} \in \Phi'$, da cui si ha facilmente la tesi.

—◇—

**Lemma 8**

*Sia $\mathcal{E}$ e $\mathcal{E}'$ due funzioni di estensione con dominio $\mathcal{D}$ e $\mathcal{D}'$, rispettivamente, $\mathcal{D} \subseteq \mathcal{D}'$, $\Phi$ e $\Phi^* \subseteq \mathcal{D}'$ due insiemi di individui, $S$ uno slot e*

$$\mathcal{E}'[X] = \begin{cases} \mathcal{E}[X] & \textit{per ogni atomo } X; \\ \mathcal{E}[X] & \textit{per ogni slot } X \neq S; \\ \mathcal{E}[X] \bigcup_{d \in \Phi, d^* \in \Phi^*} \{< d, d^* >\} & \textit{per } X = S. \end{cases}$$

*Allora per ogni frame $F$=(**and** $a_1 \ldots a_n$) in forma normale e per ogni insieme di individui $\mathcal{I} \subseteq \mathcal{D}'$ tale che*

1. *per ogni $d \in \mathcal{I}$ non esiste una sequenza di slot $S_1, \ldots, S_n$ e una sequenza di individui $d_1, \ldots, d_{n+1}$ tali che per ogni $1 \leq i \leq n$ $< d_i, d_{i+1} > \in \mathcal{E}'[S_i]$ con $d = d_1$ e $d_{n+1} \in \Phi$;*
2. *se un $a_i$ è del tipo (**card** $\{< n_1, n_2 >\}$ $S$), allora per ogni $d \in \mathcal{I} \cap \Phi$ risulta $d \in \mathcal{E}[a_i]$ se e solo se $d \in \mathcal{E}'[a_i]$;*
3. *se un $a_i$ è del tipo (**all** $S$ $u$) e $\mathcal{I} \cap \Phi \neq \emptyset$, allora $\Phi^* \subseteq \mathcal{E}[u]$*

*risulta: per ogni $d \in \mathcal{I} \cap \mathcal{D}$ $d \in \mathcal{E}[F]$ se e solo se $d \in \mathcal{E}'[F]$.*

**Dimostrazione:**

Mostriamo per induzione sulla profondità $k$ di $F$ che per ogni $1 \leq i \leq n$ e ogni $d \in \mathcal{I} \cap \mathcal{D}$ e $\mathcal{I}$ che soddisfa le condizioni (1), (2) e (3):
$\Rightarrow$.) se $d \in \mathcal{E}[a_i]$, allora $d \in \mathcal{E}'[a_i]$;
$\Leftarrow$.) se $d \notin \mathcal{E}[a_i]$, allora $d \notin \mathcal{E}'[a_i]$,
da cui la tesi. Sia $d \in \mathcal{I} \cap \mathcal{D}$ e $d \in \mathcal{E}[a_i]$.

**caso $k = 0$:**

$\Rightarrow$.)

1. se $a_i$ è un atomo, allora $d \in \mathcal{E}[a_i] = \mathcal{E}'[a_i]$;
2. se un $a_i$ è del tipo (**card** $\{< n_1, n_2 >\}$ $X$) con $X \neq S$, allora $d \in \mathcal{E}[a_i] = \mathcal{E}'[a_i]$;
3. se un $a_i$ è del tipo (**card** $\{< n_1, n_2 >\}$ $S$), allora se $d \in \Phi$, allora per la condizione (3) $d \in \mathcal{E}'[a_i]$. Se invece $d \notin \Phi$, allora $\{d' \in \mathcal{D} : <$

$d, d' >\in \mathcal{E}[S]\} = \{d' \in \mathcal{D}' : < d, d' >\in \mathcal{E}'[S]\}$, in quanto altrimenti esiste un $< d, d^* >\in \mathcal{E}'[S] \setminus \mathcal{E}[S]$ e quindi $d \in \Phi$, assurdo. Ma allora

$$\begin{aligned}
d \in \mathcal{E}[a_i] &= \{d \in \mathcal{D} |\ (n_1 \leq || \{d' \in \mathcal{D} |\ < d, d' >\in \mathcal{E}[S]\} || \leq n_2)\} \\
&= \{d \in \mathcal{D} |\ (n_1 \leq || \{d' \in \mathcal{D}' |\ < d, d' >\in \mathcal{E}'[S]\} || \leq n_2)\} \\
&\subseteq \{d \in \mathcal{D}' |\ (n_1 \leq || \{d' \in \mathcal{D}' |\ < d, d' >\in \mathcal{E}'[S]\} || \leq n_2)\} \\
&= \mathcal{E}'[a_i]
\end{aligned}$$

$\Leftarrow$.) Si dimostra come $\Rightarrow$.).

**passo induttivo $k > 0$:**
Sia vero l'enunciato per profondità $< k$.

$\Rightarrow$.)

1. se $a_i$ è della forma (**all** $X$ $u$) e $X \neq S$, allora supponiamo per assurdo che $d \notin \mathcal{E}'[a_i]$. Allora esiste un $d' \in \mathcal{D}'$ tale che $< d, d' >\in \mathcal{E}'[X]$ e $d' \notin \mathcal{E}'[u]$. Si osservi che $d' \in \mathcal{D}$ in quanto $\mathcal{E}'[X] = \mathcal{E}[X]$. In virtù della condizione (1) si ha che $d' \notin \Phi$ e la (1) è soddisfatta per $\mathcal{I}' = \{d'\}$ in quanto $d \in \mathcal{I}$. Inoltre sono soddisfatte le condizioni (2) e (3) in quanto $\mathcal{I}' \cap \Phi = \emptyset$. Pertanto per ipotesi induttiva su $u$ si ha che $d' \notin \mathcal{E}[u]$. Poiché $\mathcal{E}'[X] = \mathcal{E}[X]$, esiste un $d' \in \mathcal{D}$ tale che $< d, d' >\in \mathcal{E}[X]$ e $d' \notin \mathcal{E}[u]$. Quindi $d \notin \mathcal{E}[a_i]$; assurdo.
2. se $a_i$ è della forma (**all** $S$ $u$), allora se non esiste $d' \in \mathcal{D}'$ tale che $< d, d' >\in \mathcal{E}'[S]$, allora $d \in \mathcal{E}'[a_i]$. Altrimenti sia $d' \in \mathcal{D}'$ tale che $< d, d' >\in \mathcal{E}'[S]$. In virtù della condizione (1) si ha che $d' \notin \Phi$ e la (1) è soddisfatta per $\mathcal{I}' = \{d'\}$ in quanto $d \in \mathcal{I}$. Inoltre sono soddisfatte le condizioni (2) e (3) in quanto $\mathcal{I}' \cap \Phi = \emptyset$.
   (a) se $< d, d' >\in \mathcal{E}[S]$, allora $d' \in \mathcal{D}$ e $d' \in \mathcal{E}[u]$ in quanto $d \in \mathcal{E}[a_i]$. Ma allora per ipotesi induttiva su $u$ e $\mathcal{I}'$ si ha che $d' \in \mathcal{E}'[u]$ e quindi $d \in \mathcal{E}'[a_i]$.
   (b) se $< d, d' >\notin \mathcal{E}[S]$, allora $d \in \Phi$ e $d' \in \Phi^*$. Pertanto $\mathcal{I} \cap \Phi \neq \emptyset$ e quindi per la condizione (3) si ha che $d' \in \mathcal{E}[u]$ e quindi $d' \in \mathcal{D}$. Ma allora per ipotesi induttiva su $u$ e $\mathcal{I}'$ si ha che $d' \in \mathcal{E}'[u]$ e quindi $d \in \mathcal{E}'[a_i]$.

$\Leftarrow$.) si dimostra in modo analogho a $\Rightarrow$.).

—◇—

**Lemma 9** *Esistenza*
*Siano $F$=(***and** *$a_1 \ldots a_n$) una forma normale, $\Phi$ un insieme di individui, $\mathcal{E}$ una funzione di estensione definita su un dominio $\mathcal{D}$ tale che $\Phi \cap \mathcal{D} = \emptyset$. Allora si può effettivamente espandere $\mathcal{E}$ costruendo una funzione di estensione $\mathcal{E}'$ definita su un dominio $\mathcal{D}'$ tale che $\Phi \subseteq \mathcal{D}'$ e $\Phi \subseteq \mathcal{E}'[F]$. $\mathcal{E}'$ è data dalla procedura Build_Extension+.*

**Dimostrazione:**

$\mathcal{E}'$ è un'espansione di $\mathcal{E}$ per il lemma 6. Se $\Phi = \emptyset$, allora si ha banalmente la tesi. Altrimenti dimostriamo, per induzione sulla profondità $k$ di $F$, che la procedura Build_Extension+($F$,$\Phi$,$\mathcal{E}$,$\mathcal{D}$) costruisce una tale $\mathcal{E}'$.

**caso $k = 0$:**

In tal caso non si hanno operatori **all** in $F$. Dopo l'esecuzione dell'algoritmo si ha che $\Phi \subseteq \mathcal{E}'[a_i]$ per ogni atomo $a_i$ (passo 1 dell'algoritmo). Inoltre dal passo 3 dell'algoritmo si ha che $\Phi \subseteq \mathcal{E}'[a_i]$ per ogni operatore **card** e quindi $\Phi \subseteq \bigcap_{i=1}^{n} \mathcal{E}'[a_i] = \mathcal{E}'[F]$.

**passo induttivo $k > 0$:**

Sia vero il lemma per profondità $< k$.

1. Sia $a_i$ della forma (**all** $S$ $u$) e supponiamo che non vi sia un $a_j$=(**card** $\{< n_1, n_2 >\}$ $S$) con $n_1 > 0$. La procedura Build_Extension+ in questo caso non associa a $d \in \Phi$ nessun $S$ e siccome $d$ non è vincolato ad avere degli $S$, $d \in \mathcal{E}'$[(**all** $S$ $u$)];
2. sia $a_i$ della forma (**all** $S$ $u$) e supponiamo che vi sia un un $a_j$=(**card** $\{< n_1, n_2 >\}$ $S$) con $n_1 > 0$. Dal passo 2 dell'algoritmo si ha che $\Phi' = \{d_1^*, \ldots, d_{n_1}^*\}$, $\Phi' \cap \mathcal{D}' = \emptyset$, $\Phi \subseteq \mathcal{D}'$. Sia $\mathcal{E}''$ data da Build_Extension+($u$,$\Phi'$,$\mathcal{E}'$,$\mathcal{D}'$). Quindi vale la tesi del lemma 7. Sia $\mathcal{E}^*$ data dall'assegnamento (1). Posto $\mathcal{I} = \Phi' = \Phi^*$, allora sono soddisfatte le condizioni del lemma 8. Pertanto per ipotesi induttiva su $u$, $\Phi' \subseteq \mathcal{E}''[u]$ e quindi per il lemma 8 vale $\Phi' \subseteq \mathcal{E}^*[u]$; pertanto $\Phi \subseteq \mathcal{E}^*[a_i]$. Infine $\mathcal{E}^*$ è tale che $\Phi \subseteq \mathcal{E}^*[a_j]$, in quanto per ogni $d \in \Phi$ vale $< d, d' > \notin \mathcal{E}''[S]$ per qualsiasi $d'$;
3. Sia $a_i$ della forma (**card** $\{< n_1, n_2 >\}$ $S$) con $n_1 > 0$ e supponiamo che non vi sia un (**all** $S$ $u$). In tal caso per un $d \in \Phi$ vale $< d, d' > \notin \mathcal{E}[S]$ per qualsiasi $d'$ e, quindi, un $d \in \Phi$ ha esattamente $n_1$ $S$ (passo 3). Pertanto, $d \in \mathcal{E}'[a_i]$;
4. Sia $a_i$ della forma (**card** $\{< n_1, n_2 >\}$ $S$) con $n_1 > 0$ e supponiamo che vi sia un (**all** $S$ $u$). Si dimostra come in 2;
5. in tutti gli altri casi ($a_i$=(**card** $\{< 0, n_2 >\}$ $S$), ecc.) non si hanno vincoli relativamente allo slot per un $d \in \Phi$ e quindi risulta $d \in \mathcal{E}'[a_i]$.

Ridenominando $\mathcal{E}^*$ in $\mathcal{E}'$ si ha la tesi.

—◇—

In modo del tutto analogo, dato un insieme di individui $\Phi$ non vuoto e un frame $F$ non vuoto in forma normale e una funzione di estensione $\mathcal{E}$ con $\Phi \cap \mathcal{D} = \emptyset$, si può effettivamente espandere la $\mathcal{E}$ costruendo una funzione di estensione $\mathcal{E}'$ con dominio $\mathcal{D}'$ tale che $\Phi \subseteq \mathcal{D}'$ e $\Phi \not\subseteq \mathcal{E}'[F]$.

**procedure Build_Extension-**($F$,$\Phi$,$\mathcal{E}$, $\mathcal{D}$)

**input:** Un frame $F$ non vuoto in forma normale, un insieme di individui $\Phi$ non vuoto e una funzione di estensione $\mathcal{E}$ definita sul dominio $\mathcal{D}$ tale che $\Phi \cap \mathcal{D} = \emptyset$.

**output:** Una funzione di estensione $\mathcal{E}'$ definita sul dominio $\mathcal{D}'$ tale che $\mathcal{E}'$ è un'espansione della $\mathcal{E}$ tale che $\Phi \subseteq \mathcal{D}'$ e $\Phi \not\subseteq \mathcal{E}'[F]$[13].

**beginprocedure**

Sia $\mathcal{D}' := \mathcal{D} \cup \Phi$, $\mathcal{E}'[A] := \mathcal{E}[A]$ per ogni atomo $A$ e $\mathcal{E}'[S] := \mathcal{E}[S]$ per ogni slot $S$. Sia $F$ è della forma (**and** $a_1 \ldots a_n$), allora per ogni $1 \leq i \leq n$

1. se $a_i$ è della forma (**all** $S$ $u$), allora per un $d^* \notin \mathcal{D}'$ sia
$$\Phi' = \{d^*\}$$
si richiami Build_Extension-($u$,$\Phi'$,$\mathcal{E}'$,$\mathcal{D}'$) e sia
$$\mathcal{E}'[S] := \mathcal{E}'[S] \bigcup_{d \in \Phi} \{< d, d^* >\}$$
**Commento:**
L'estensione che vorremmo costruire dev'essere tale che $\Phi \not\subseteq \mathcal{E}'[(\textbf{all}\ S\ u)]$. Allora associamo un nuovo $S$ $d^*$ ai $\Phi$ tale che $d^*$ non è un $u$ (Build_Extension-($u$,$\Phi'$,$\mathcal{E}'$,$\mathcal{D}'$)).
2. se $a_i$ è della forma (**card** $\{< n_1, n_2 >\}$ $S$), allora per dei $d_1^*, \ldots, d_k^* \notin \mathcal{D}'$ sia
$$\mathcal{E}'[S] := \mathcal{E}'[S] \bigcup_{d \in \Phi} \{< d, d_1^* >, \ldots, < d, d_k^* >\}$$
tale che non vale $n_1 \leq k \leq n_2$[14];
**Commento:** L'estensione che vorremmo costruire dev'essere tale che $\Phi \not\subseteq \mathcal{E}'[(\textbf{card}\ \{< n_1, n_2 >\}\ S)]$. Questo lo otteniamo, associando un numero non legale ($k$) di $S$ ai $\Phi$.

**endprocedure**

Si osservi che neppure gli atomi vengono aggiornati e quindi $\Phi \not\subseteq \mathcal{E}'[a_i]$ se $a_i$ atomo.

Si dimostrano facilmente i seguenti

[13] In realtà sarà $\Phi \cap \mathcal{E}'[F] = \emptyset$.

[14] Si osservi che nel caso $n_1 = 1$ e $n_2 = \infty$ vale $k = 0$.

**Lemma 10**
*Siano $F$=(**and** $a_1 \ldots a_n$) una forma normale, $\Phi$ un insieme di individui, $\mathcal{E}$ una funzione di estensione definita su un dominio $\mathcal{D}$ tale che $\Phi \cap \mathcal{D} = \emptyset$ e $\mathcal{E}'$ data dalla procedura Build_Extension-($F$,$\Phi$,$\mathcal{E}$,$\mathcal{D}$). Allora $\mathcal{E}'$ è un'espansione di $\mathcal{E}$.*

—◇—

**Lemma 11**
*Siano $F$=(**and** $a_1 \ldots a_n$) una forma normale, $\mathcal{E}$ una funzione di estensione definita su un dominio $\mathcal{D}$, $\Phi$ e $\Phi'$ due insiemi di individui tali che $\Phi \subseteq \mathcal{D}$ e $\Phi' \cap \mathcal{D} = \emptyset$, e $\mathcal{E}'$ data dalla procedura Build_Extension-($F$,$\Phi'$,$\mathcal{E}$,$\mathcal{D}$). Allora per ogni $d \in \Phi'$ non esiste una sequenza di slot $S_1, \ldots, S_n$ e una sequenza di individui $d_1, \ldots, d_{n+1}$ tali che per ogni $1 \leq i \leq n$ $< d_i, d_{i+1} >\in \mathcal{E}'[S_i]$ con $d = d_1$ e $d_{n+1} \in \Phi$ o $d_{n+1} \in \Phi'$.*

—◇—

**Lemma 12** *Esclusione*
*Siano $F$=(**and** $a_1 \ldots a_n$) una forma normale, $\Phi$ un insieme di individui non vuoto, $\mathcal{E}$ una funzione di estensione definita su un dominio $\mathcal{D}$ tale che $\Phi \cap \mathcal{D} = \emptyset$. Allora si può effettivamente espandere $\mathcal{E}$ costruendo una funzione di estensione $\mathcal{E}'$ definita su un dominio $\mathcal{D}'$ tale che $\Phi \subseteq \mathcal{D}'$ e $\Phi \not\subseteq \mathcal{E}'[F]$. $\mathcal{E}'$ è data dalla procedura Build_Extension-.*

—◇—

Passiamo ora alla completezza. In seguito, quando faremo riferimento al lemma di esistenza e di esclusione, intenderemo le espansioni costruite secondo le procedure Build_Extension+ e Build_Extension-.

**Lemma 13** *Completezza*
*Siano $F_1$ e $F_2$ due frame in forma normale. Se* SUBS?($F_1$, $F_2$)=false, *allora $F_2$ non è sussunto da $F_1$.*

**Dimostrazione:**

Dimostriamo, per induzione sulla massima profondità $k$ di $F_1$ e $F_2$, che se SUBS? ritorna *false*, allora esiste una funzione di estensione $\mathcal{E}$ e un insieme di individui $\Phi$ non vuoto tale che $\Phi \subseteq \mathcal{E}[F_2]$ e $\Phi \not\subseteq \mathcal{E}[F_1]$ da cui la tesi.

**caso $k = 0$:**

Se SUBS? ritorna *false*, allora vale uno dei seguenti casi:

1. $F_2 \neq$ (**void**) e $F_1$ = (**void**). Per il lemma di esistenza, dato un insieme $\Phi$ non vuoto di individui, esiste una funzione di estensione $\mathcal{E}$ tale che $\Phi \subseteq \mathcal{E}[F_2]$ e quindi $\Phi \not\subseteq \mathcal{E}[F_1] = \emptyset$. Nei successivi casi sia $F_1$ della forma (**and** $a_1 \ldots a_n$) e $F_2$ della forma (**and** $b_1 \ldots b_m$).

2. Esiste un atomo $a_i$ che non è nessuno dei $b_j$. Sia $\Phi$ un insieme non vuoto di individui. Allora, per il lemma di esistenza, esiste una funzione di estensione $\mathcal{E}$ tale che $\Phi \subseteq \mathcal{E}[F_2]$. Ma siccome $a_i$ non è nessuno dei $b_j$, per costruzione della $\mathcal{E}$, $\Phi \not\subseteq \mathcal{E}[a_i]$. Quindi, $\Phi \not\subseteq \mathcal{E}[F_1]$.
3. Esiste un $a_i$ della forma (**card** $\{< n_1, n_2 >\}$ $S$) e
   (a) non esiste un $b_j$ della forma (**card** $\{< n_3, n_4 >\}$ $S$). Sia $\Phi$ un insieme di individui non vuoto. Allora, per il lemma di esistenza, esiste una funzione di estensione $\mathcal{E}$ definita su un dominio $\mathcal{D}$ tale che $\Phi \subseteq \mathcal{E}[F_2]$. Ma siccome non esiste un $b_j$ della forma (**card** $\{< n_3, n_4 >\}$ $S$), né esiste un $b_j$ della forma (**all** $S$ $u$) in quanto $k = 0$, si ha per costruzione di $\mathcal{E}$ che per ogni $d \in \Phi$ e per ogni $d' \in \mathcal{D}$ vale $< d, d' > \notin \mathcal{E}[S]$. Se $n_1 > 0$, allora $\Phi \not\subseteq \mathcal{E}[a_i]$ e quindi $\Phi \not\subseteq \mathcal{E}[F_1]$. Se $n_1 = 0$, in tal caso $n_2 \neq \infty$, allora estendiamo $\mathcal{E}$, ottenendo $\mathcal{E}'$, nel seguente modo[15]: siano $d_1, \ldots, d_{n_2+1} \notin \mathcal{D}$,

   $$\mathcal{E}'[X] = \begin{cases} \mathcal{E}[X] & \text{per ogni atomo } X; \\ \mathcal{E}[X] & \text{per ogni slot } X \neq S; \\ \mathcal{E}[X] \bigcup_{d \in \Phi} \{< d, d_1 >, \ldots, < d, d_{n_2+1} >\} & \text{per } X = S. \end{cases}$$

   e posto $\mathcal{D}' := \mathcal{D} \cup \{d_1, \ldots, d_{n_2+1}\}$. Posto $\mathcal{I} = \Phi$, poiché $\Phi \subseteq \mathcal{E}[F_2]$, per il lemma 8 segue $\Phi \subseteq \mathcal{E}'[F_2]$. Per costruzione di $\mathcal{E}'$ si ha che $\Phi \not\subseteq \mathcal{E}'[a_i]$ e quindi $\Phi \not\subseteq \mathcal{E}'[F_1]$.

   (b) Esiste un $b_j$ della forma (**card** $\{< n_3, n_4 >\}$ $S$), ma non vale $n_1 \leq n_3$ e $n_4 \leq n_2$. Sia $\Phi$ un insieme di individui non vuoto. Allora, per il lemma di esistenza esiste una funzione di estensione $\mathcal{E}$, definita sul un dominio $\mathcal{D}$, per (**and** $b_1 \ldots b_{j-1}$ $b_{j+1} \ldots b_m$) tale che $\Phi \subseteq \mathcal{E}$ [(**and** $b_1 \ldots b_{j-1}$ $b_{j+1} \ldots b_m$)][16]. Estendiamo $\mathcal{E}$, ottenendo $\mathcal{E}'$, nel seguente modo: siano $d_1, \ldots, d_k \notin \mathcal{D}$ tali che $n_3 \leq k \leq n_4$ e non $n_1 \leq k \leq n_2$,

   $$\mathcal{E}'[X] = \begin{cases} \mathcal{E}[X] & \text{per ogni atomo } X; \\ \mathcal{E}[X] & \text{per ogni slot } X \neq S; \\ \mathcal{E}[X] \bigcup_{d \in \Phi} \{< d, d_1 >, \ldots, < d, d_k >\} & \text{per } X = S \end{cases}$$

   Si osservi che può essere $k = 0$. Posto $\mathcal{D}' = \mathcal{D} \cup \{d_1, \ldots, d_k\}$, $\mathcal{I} = \Phi$, poiché $\Phi \subseteq \mathcal{E}[($**and** $b_1 \ldots b_{j-1}$ $b_{j+1} \ldots b_m)]$, per il lemma 8 segue $\Phi \subseteq \mathcal{E}'[($**and** $b_1 \ldots b_{j-1}$ $b_{j+1} \ldots b_m)]$. Per costruzione di $\mathcal{E}'$ si ha che $\Phi \subseteq \mathcal{E}'[b_j]$, pertanto $\Phi \subseteq \mathcal{E}'[F_2]$ e inoltre $\Phi \not\subseteq \mathcal{E}'[a_i]$ e quindi $\Phi \not\subseteq \mathcal{E}'[F_1]$.

**passo induttivo $k > 0$:**

Supponiamo che l'enunciato sia vero per profondità $< k$.

[15] Si osservi che $\mathcal{E}'$ non è un'espansione di $\mathcal{E}$.

[16] Si osservi che per $d \in \Phi$ e $d' \in \mathcal{D}$ vale $< d, d' > \notin \mathcal{E}[S]$, per costruzione di $\mathcal{E}$.

1. Sia $a_i$ del tipo (**all** $S$ $u$) e non esiste un $b_j$ del tipo (**all** $S$ $v$).

   (a) Non esiste un $b_h$ del tipo (**card** $\{< n_1, n_2 >\}$ $S$). Per il lemma di esclusione esiste funzione di estensione $\mathcal{E}$, defınta su un dominio $\mathcal{D}$, per $u$, tale che per un $d^* \in \mathcal{D}$ vale $d^* \notin \mathcal{E}[u]$. Per il lemma di esistenza, dato un insieme non vuoto di individui $\Phi$ tale che $\Phi \cap \mathcal{D} = \emptyset$, possiamo espandere $\mathcal{E}$, ottenendo $\mathcal{E}'$, definita su un dominio $\mathcal{D}'$, tale che $\Phi \subseteq \mathcal{D}$ e $\Phi \subseteq \mathcal{E}'[F_2]$. Estendiamo ora $\mathcal{E}'$, ottenendo $\mathcal{E}^*$, nel seguente modo: $\mathcal{D}^* = \mathcal{D}'$

$$\mathcal{E}^*[X] = \begin{cases} \mathcal{E}'[X] & \text{per ogni atomo } X; \\ \mathcal{E}'[X] & \text{per ogni slot } X \neq S; \\ \mathcal{E}'[X] \bigcup_{d \in \Phi} \{< d, d^* >\} & \text{per } X = S. \end{cases}$$

   Da $d^* \notin \mathcal{E}[u]$ segue $d^* \notin \mathcal{E}'[u]$, per il corollario 1. Quindi, per il lemma 8 con $\mathcal{I} = \{d^*\}$ si ha $d^* \notin \mathcal{E}^*[u]$ e quindi $\Phi \not\subseteq \mathcal{E}^*[($**all** $S$ $u)]$. Pertanto, $\Phi \not\subseteq \mathcal{E}^*[F_1]$. Infine, poiché $\Phi \subseteq \mathcal{E}'[F_2]$ e posto $\mathcal{I} = \Phi$, per il lemma 8 risulta $\Phi \subseteq \mathcal{E}^*[F_2]$.

   (b) Esiste un $b_h$ del tipo (**card** $\{< n_1, n_2 >\}$ $S$). Siccome SUBS?$[F_1,F_2]$=*false* si ha che non vale $n_1 = n_2 = 0$. In questo caso possiamo costruire in modo analogo al caso 1a e $k > 0$ un'estensione $\mathcal{E}^*$ scegliendo, anziché un $d^*$, dei $d_1^*, \ldots, d_k^*$ tale che risulta $n_1 \leq k \leq n_2$ e i $d_1^*, \ldots, d_k^*$ non appartengono a $\mathcal{E}^*[u]$, osservando che la $\mathcal{E}'$ va costruita per (**and** $b_1 \ldots b_{j-1}$ $b_{j+1} \ldots b_m$) anziché per $F_2$ (quindi come nel caso 3b e $k = 0$);

2. Sia $a_i$ del tipo (**all** $S$ $u$) e esiste un $b_j$ del tipo (**all** $S$ $v$) e SUBS?($u$, $v$)=*false*[17].

   (a) Non vi sia un $b_h$ del tipo (**card** $\{< n_1, n_2 >\}$ $S$). Per ipotesi induttiva su $u$ e $v$, per un $\Phi^* = \{d^*\}$ esiste una funzione di estensione $\mathcal{E}$, definita su un dominio $\mathcal{D}$, tale che $\Phi^* \subseteq \mathcal{E}[v]$ e $\Phi^* \not\subseteq \mathcal{E}[u]$. Sia $\Phi$ un insieme di individui non vuoto tale che $\Phi \cap \mathcal{D} = \emptyset$, allora per il lemma dell'esistenza, possiamo espandere $\mathcal{E}$, ottenendo $\mathcal{E}'$ definita su un dominio $\mathcal{D}'$, tale che $\Phi \subseteq \mathcal{E}'[F_2]$. Estendiamo ora $\mathcal{E}'$, ottenendo $\mathcal{E}^*$, nel seguente modo: $\mathcal{D}^* = \mathcal{D}'$

$$\mathcal{E}^*[X] = \begin{cases} \mathcal{E}'[X] & \text{per ogni atomo } X; \\ \mathcal{E}'[X] & \text{per ogni slot } X \neq S; \\ \mathcal{E}'[X] \bigcup_{d \in \Phi, d^* \in \Phi^*} \{< d, d^* >\} & \text{per } X = S. \end{cases}$$

   Poiché $\Phi^* \not\subseteq \mathcal{E}[u]$, per il corollario 1, ne segue che $\Phi^* \not\subseteq \mathcal{E}'[u]$ e quindi, per il lemma 8 con $\mathcal{I} = \Phi^*$, $\Phi^* \not\subseteq \mathcal{E}^*[u]$. Pertanto, $\Phi \not\subseteq \mathcal{E}^*[a_i]$ e quindi $\Phi \not\subseteq \mathcal{E}^*[F_1]$. Siccome $\Phi^* \subseteq \mathcal{E}[v]$, per il corollario 1, ne segue che $\Phi^* \subseteq \mathcal{E}'[v]$ e quindi, per il lemma 8

[17] Si osservi che in tal caso non esiste un $b_j$ del tipo (**card** $\{< 0, 0 >\}$ $S$).

con $\mathcal{I} = \Phi^*$, $\Phi^* \subseteq \mathcal{E}^*[v]$. Infine, per il lemma 8 con $\mathcal{I} = \Phi$, da $\Phi \subseteq \mathcal{E}'[F_2]$ segue $\Phi \subseteq \mathcal{E}^*[F_2]$;

(b) Esiste un $b_h$ del tipo (**card** $\{< n_1, n_2 >\}$ $S$). In tal caso possiamo procedere come nel caso 2a e $k > 0$, costruendo la $\mathcal{E}$ per un $\Phi^*$ tale che $n_1 \leq |\Phi^*| \leq n_2$ e la $\mathcal{E}'$, anziché per $F_2$, per (**and** $b_1 \ldots b_{h-1}\ b_{h+1} \ldots b_m$)[18] e la $\mathcal{E}^*$ come sopra;

3. sia $a_i$ del tipo (**card** $\{< n_1, n_2 >\}$ $S$) e non vi sia un $b_j$ del tipo (**card** $\{< n_3, n_4 >\}$ $S$).
   (a) Non esiste un $b_h$ del tipo (**all** $S$ $u$). Tale caso si dimostra come per $k = 0$ nel caso 3a;
   (b) Esiste un $b_h$ del tipo (**all** $S$ $u$). Sia $\Phi^* = \{d_1^*, \ldots, d_l^*\}$ e tale che non vale $n_1 \leq l \leq n_2$. Per il lemma di esistenza esiste una funzione di estensione $\mathcal{E}$, definita su un dominio $\mathcal{D}$ tale che $\Phi^* \subseteq \mathcal{E}[u]$. Per un insieme non vuoto di individui $\Phi$ e il lemma di esistenza si può espandere $\mathcal{E}$, ottenendo $\mathcal{E}'$ definita su un dominio $\mathcal{D}'$ tale che $\Phi \subseteq \mathcal{E}'[F_2]$. Si osservi che vale ancora $\Phi^* \subseteq \mathcal{E}'[u]$ per il corollario 1. Estendiamo ora, $\mathcal{E}'$, ottenendo $\mathcal{E}^*$ come nel caso $k > 0$ e 2a. Tenendo presente che quindi $\Phi \not\subseteq \mathcal{E}^*[a_i]$ si dimostra facilmente la tesi come nel caso $k > 0$ e 2a.
4. sia $a_i$ del tipo (**card** $\{< n_1, n_2 >\}$ $S$) e vi sia un $b_j$ del tipo (**card** $\{< n_3, n_4 >\}$ $S$).
   (a) Non esiste un $b_h$ del tipo (**all** $S$ $u$). Si dimostra analogamente al caso $k = 0$ e 3b.
   (b) Esiste un $b_h$ del tipo (**all** $S$ $u$). Si dimostra come il caso $k > 0$ e 3b scegliendo l'indice $l$ tale che $n_3 \leq l \leq n_4$ e non $n_1 \leq l \leq n_2$ l'estensione $\mathcal{E}'$, anziché per $F_2$, per (**and** $b_1 \ldots b_{j-1}\ b_{j+1} \ldots b_m$)[19].

—◇—

Considerando il lemma di riduzione, di complessità, di correttezza e di completezza si ha il

**Teorema 2**
*Dati due frame $F_1$ e $F_2$, allora è determinabile in $O(|F_1||F_2|)$ se $F_2$ è sussunto da $F_1$.*

—◇—

[18] Si osservi che per $d \in \Phi$ e $d' \in \mathcal{D}$ vale $< d, d' > \notin \mathcal{E}'[S]$.
[19] Si osservi che può essere $l = 0$; si veda caso $k = 0$ e 3b.

# Riferimenti


[1] N. Belnap. A useful four-valued logic. In G. Epstein and J. Dumm, editors, *Modern Uses of Multiple-valued logic*, pages 8–37, D. Reidel Publishing Company, Boston, 1977.

[2] R. J. Brachman and J. G. Schmolze. *An Overview of the KL-ONE Knowledge Representation System.* Technical Report 655, Fairchild, Palo Alto, CA, 1984.

[3] R.J. Brachman, A. Borgida, D. L. McGuinness, and L. A. Resnick. CLASSIC: a structural data model for objects. In *Proceedings 1989 ACM SIGMOD International Conference on Management of Data*, pages 59–67, Portland, OR, 1989.

[4] R.J. Brachman and H.J. Levesque. The tractability of subsumption in frame-based description languages. In *Proceedings of AAAI-84*, pages 34–37, Austin, TX, 1984.

[5] M. L. Ginsberg. Multi-valued logics. In M. L. Ginsberg, editor, *Readings in Non-monotonic Reasoning*, Morgan Kaufmann, Los Altos, CA, 1987.

[6] M. L. Ginsberg. Multivalued logics: a uniform approach to reasoning in artificial intelligence. *Computational Intelligence*, 4:265–316, 1988.

[7] H. J. Levesque. *A Logic of Implicit and Explicit Belief.* Technical Report 653, Fairchild, Palo Alto, CA, 1984.

[8] H.J. Levesque and R.J. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3(2):78–93, 1987.

[9] T. Lipkis. A KL-ONE classifier. In J. G. Schmolze and R. J. Brachman, editors, *Proceedings of the 1981 KL-ONE Workshop*, pages 128–145, Jackson, NH, 1982.

[10] B. Nebel. Computational complexity of terminological reasoning in BACK. *Artificial Intelligence*, 34:371–383, 1988.

[11] P. F. Patel-Schneider. A four-valued semantics for terminological logics. *Artificial Intelligence*, 38:319–351, 1989.

[12] P. F. Patel-Schneider. A hybrid, decidable, logic-based knowledge representation system. *Computational Intelligence*, 3(2):64–77, 1987.

[13] P. F. Patel-Schneider. Undecidability of subsumption in NIKL. *Artificial Intelligence*, 39:263–272, 1989.

[14] P.F. Patel-Schneider. *Small can be Beautiful in Knowledge Representation.* Technical Report 660, Fairchild, Palo Alto, CA, October 1984.

[15] J. G. Schmolze and D. J. Israel. *KL-ONE: Semantics and Classification.* Technical Report 5421, BBN Laboratories, Cambridge, MA, 1983.

[16] R. H. Thomason, J. F. Horty, and D. S. Touretzky. *A Calculus for Inheritance in Monotonic Semantic Nets.* Technical Report CMU-CS-86-138, Department of Computer Science, Pittsburgh, PA, 1986.

[17] D. S. Touretzky. *The Mathematics of Inheritance Systems.* Pitman, London, GB, 1986.

[18] K. von Luck, B. Nebel, C. Peltason, and A. Schmiedel. *The Anatomy of the BACK System.* KIT Report 41, Fachbereich Informatik, Technische Universität Berlin, Berlin, 1987.

[19] K. von Luck and B. Owsnicki-Klewe. *New AI Formalisms for Knowledge Representation: a Case Study.* KIT Report 57, Fachbereich Informatik, Technische Universität Berlin, Berlin, 1987.