UNIVERSITA' DEGLI STUDI DI TORINO

Facoltà di Scienze Matematiche Fisiche e Naturali

Corso di Laurea in Scienze dell'Informazione

TESI DI LAUREA

Alcuni aspetti di Logica Modale

Relatore
Prof. Gabriele Lolli

Candidato
Rossati Giovanni

Anno Accademico 1990/1991

INDICE

0 PREMESSA

La logica ha per oggetto lo studio di quei complessi linguistici, detti proposizioni, ai quali può essere associato un valore di verità.

Nelle proposizioni si individuano uno o più oggetti, di essi è asserito un comportamento o una qualità: una proposizione con un argomento o che coinvolge un solo oggetto è detta predicato, e, in genere, relazione *n*-aria se coinvolge *n* argomenti; se il valore di verità è determinato, si usa il termine enunciato.

Una proposizione elementare, è detta anche variabile proposizionale.

Nella logica Classica si considerano delle relazioni fra proposizioni, di solito una relazione unaria e alcune relazioni binarie, il termine relazione è spesso sostituito dal termine connettivo (verofunzionale), o talvolta, operatore.

Le proposizioni ottenute da altre proposizioni tramite connettivi, hanno un valore di verità che dipende unicamente dal valore di verità delle proposizioni componenti.

Nelle logiche Modali, oltre agli operatori della logica Classica, si ammettono degli operatori che determinano un valore di verità non dipendente, nel senso su specificato, dal valore di verità dei componenti l'enunciato.

Inoltre, ed è ciò che rende interessanti queste logiche, si ammettono sia assiomi che regolano i rapporti fra proposizioni e proposizioni modalizzate, sia assiomi circa la reiterazione degli operatori modali, allo scopo di descrivere certe forme della realtà.

Questo lavoro si occupa di alcuni aspetti di logica Modale, in particolare in esso saranno considerati non solo operatori che formano enunciati da enunciati, ma anche operatori che rendono enunciati degli oggetti linguistici cui non ha senso attribuire valori di verità, come azioni, oggetti, ecc..

Il primo capitolo ha caratteristiche di introduzione e vuol evidenziare alcuni aspetti delle modalità, sia di tipo intensionale che formale, aspetti che saranno trattati in modo più esteso successivamente.

Lo scopo principale del capitolo è di sottolineare la ricchezza espressiva che gli strumenti Modali forniscono, in particolare in campi di interesse per l'Informatica e l'Intelligenza Artificiale.

Il capitolo 2 tratta di Modalità da un punto di vista algebrico: sulla base della possibilità di esprimere algebricamente la Logica Proposizionale, sono introdotti due operatori che agiscono non su variabili proposizionali (variabili in breve), ma su insiemi di variabili.

Gli insiemi sono individuati da Relazioni binarie che intercorrono fra le variabili, più precisamente ad ogni variabile **p** è associato l' insieme composto da tutte le variabili che sono in Relazione con **p**, in simboli:

$$I(p) = \{p_i \mid R(p,p_i)\}$$

Gli operatori introdotti, **Π** e **Σ**, estendono, in pratica, gli operatori binari $\cdot$ e + al caso di più variabili. Si dimostra che se la Relazione ha certe proprietà valgono certe formule, fra le più significative:

$\mathbf{\Pi}p \leq p$ sse la Relazione R è riflessiva,

$\mathbf{\Pi}p \leq \mathbf{\Pi\Pi}p$ sse la Relazione R è transitiva,

$\mathbf{\Sigma}p \leq \mathbf{\Pi\Sigma}p$ sse la Relazione R è simmetrica e transitiva.

Un altro punto messo in luce nel capitolo 2, riguarda la regola di Necessitazione (se α è una Tesi, tale è $\mathbf{\Pi}\alpha$): la regola vale se e solo se gli insiemi **I**(p) hanno la stessa cardinalità.

Il capitolo termina evidenziando che sistemi Algebrici in cui valgono certe formule, coincidono con i sistemi Modali T, S4 ed S5.

Il Capitolo 3, oltre a fornire una breve panoramica di Logica Temporale e di sue estensioni, propone una definizione Modale dei concetti Prima e Dopo, sufficiente per giustificare l'ottenimento, da una lista di precedenze fra eventi, di una formula in cui questi eventi sono espressi in modo "ordinato".

Il Capitolo 4 si occupa di Modalità epistemiche, applicate ai programmi, in particolare viene fornita una definizione di conoscenza acquisita da una macchina di Turing o da un generico programma.

Infine le appendici raccolgono alcune semplici proprietà delle Modalita[1], elenchi di assiomatizzazioni, una breve sintesi della teoria delle Relazioni di Peirce, e una schematizzazione della equivalenza fra il sistema modale S5 e il calcolo dei predicati del primo ordine con una sola variabile.

Quest' ultima proprietà sembrerebbe confermare la convinzione di Russell (e altri), circa la poca utilità di un Calcolo Modale,[2] tuttavia le Modalità, unendo una rigorosa formalizzazione ad uno spiccato contenuto antropocentrico, agevolano lo studio di quei sottosistemi logici che descrivono il comportamento umano.

In questo lavoro si seguirà, a meno di indicazioni contrarie, l'assiomatizzazione del calcolo proposizionale (PC) dei Principia Mathematica (PM) di Whitehead e Russell come riportata in [HC] pag. 33 e seguenti (All. A1).

---

[1] Sono proprietà o semplici teoremi "scoperti" durante lo sviluppo del presente lavoro.

[2] Vedasi l' introduzione di [HC].

1 MODALITÀ

1.1 Modalità, definizioni e considerazioni

Le possibili "operazioni" linguistiche su di una proposizione, il cui risultato è una proposizione, sono molteplici, e, poiché questa è grosso modo la definizione di modalità, molteplici sono le modalità stesse.

Le Modalità sono un arricchimento del Calcolo Proposizionale, (o del Calcolo dei Predicati), arricchimento ottenuto tramite assiomi modali. La scelta fra vari assiomi origina Sistemi Modali diversi, i più noti dei quali sono denominati T S4 ed S5 (v. A2 per la loro assiomatizzazione).

Il riferimento ad una Modalità piuttosto che ad un'altra, è una operazione intensionale, utilizzata per dare un significato intuitivo ad assiomi e formule modali; ciò, pur non essendo necessario, diventa una esigenza concreta quando si assiomatizzano aspetti particolari della realtà.

Infatti il successo di una trattazione modale è dato dalla capacità di descrivere, in modo soddisfacente, un frammento della realtà stessa e, secondo Russell ([R] pag. 90 con riferimento, in realtà, alle teorie logiche), dalla capacità di risolvere dei paradossi.

Analogamente il rigetto di un sistema modale consistente avviene, più che altro, sulla base di contraddizioni intensionali, ad esempio per la modalità **Opinione**, non si accetta l'assioma che afferma che ciò che è creduto è sempre vero, mentre non si ha difficoltà ad accettarlo per la modalità **Conosciuto**.

Le prime modalità studiate sono state le modalità **Necessario** e **Possibile**, dette modalità **Aletiche**. Compaiono, sotto forma di sillogismi modali, nell'Organon di Aristotele.

Con la ripresa degli studi sulla logica modale alla fine del XIX secolo, sono state investigate altre modalità, quali le **Deontiche** (Permesso, Obbligatorio, Vietato, ...), le **Epistemiche** (Conosciuto, Creduto, ...), le **Temporali** (in Passato, in Futuro, ...), ed altre.

Alcune Modalità operano sia su proposizioni che su non proposizioni, ad esempio le modalità Epistemiche: si conosce una proposizione, un oggetto, ecc.; altre modalità sono relative ad azioni come le Deontiche.

Una proposizione modale può essere soggetta ad operatori modali (modalità di modalità), e gli operatori modali coinvolti possono essere di tipo diverso.

Nel caso di modalità di atto, o miste, di proposizione e di atto, la reiterazione di modalità non sempre è possibile: infatti se **F** ed **A** sono operatori modali di proposizioni e di atto rispettivamente, e p ed α sono una proposizione ed un atto, è corretta la formula:

$\mathbf{F}(\mathbf{A}(\alpha))$

ma non lo sono:

$\mathbf{A}(\mathbf{F}(p))$ e $\mathbf{A}(\mathbf{A}(\alpha))$

perché l'argomento di **A** deve essere un atto, mentre $\mathbf{F}(p)$ e $\mathbf{A}(\alpha)$ sono proposizioni.

Ad esempio si considerino le modalità di fatto **Necessario** e di atto **Obbligatorio**, indicate nel seguito rispettivamente con **N** ed **O**, una proposizione come:

$\mathbf{N}(\mathbf{O}(\alpha))$

è corretta, e può essere letta: "È necessario che α sia obbligatorio", ma come si può leggere, supposto sia valida, la formula $\mathbf{O}(\mathbf{N}(\alpha))$?

Una formula del tipo $\mathbf{O}(\mathbf{O}(\alpha))$ non è una proposizione, perché $\mathbf{O}(\alpha)$ è una proposizione e non un atto; si può pensare come von Wright ([vW] pag 12, 13) ad una forma prescrittiva delle formulazioni deontiche di ordine superiore, e quindi leggere: "È obbligatorio obbligare ad eseguire α" ma "obbligare ad eseguire" è un atto e non un fatto.

Dal punto di vista formale o si considerano diverse le due modalità o si introduce, ove necessario, un operatore di attualizzazione.

D' altra parte lo stesso von Wright ([vW] pag. 13 e 23), postula qualcosa di analogo per le formulazioni Deontiche di ordine superiore, che chiama "Principio di trasmissione della volontà".

In Åqvist ([Å] pag. 610) le modalità Deontiche sono espresse in funzione delle modalità Aletiche, di una costante proposizionale (Q) che indica l' assenza di castigo e di una azione α:

$\mathbf{O}\alpha =_{df} \mathbf{L}(Q \Rightarrow \alpha)$

$\mathbf{P}\alpha =_{df} \mathbf{M}(Q \,\&\, \alpha)$

$\mathbf{F}\alpha =_{df} \mathbf{L}(Q \Rightarrow \neg\alpha)$

Ad esempio l' ultima formula si legge: "α è proibito significa che necessariamente vi è o il castigo o non si è compiuto α".

La trasformazione di un atto in un fatto può essere considerata una operazione modale: **obbligatorio**, **permesso**, **prima**, **dopo**, ecc. sono esempi di modalità che agiscono su atti trasformandoli in fatti, e come tali assumono valore di verità.

Le proposizioni su azioni, si possono considerare:

- proposizioni esistenziali, cioè espressioni che asseriscono, ad esempio, l' esistenza di un obbligo, di un permesso, ecc.,
- atti complessi,
- modalità su fatti derivati dall'esecuzione di atti (Åqvist [A] pag. 617),
- modalità di atto.

La prima alternativa ha a che fare sostanzialmente con proposizioni e non con modalità.

La seconda alternativa conduce ad una "logica dell'azione", in cui si possono dare regole ed assiomi, e dove si interpreta semanticamente "fatto" e "non fatto" analogamente a "vero" e "falso", e con l'eventuale introduzione di operatori modali.

La terza alternativa, ha come oggetto, non atti, ma fatti derivati da questi.

Infine l'ultima alternativa tratta di operatori modali su atti, e può essere utilizzata per analizzare i complessi linguistici detti prescrizioni, che coinvolgono fatti ed azioni.

1.2 Modalità prescrittive

1.2.1 Prescrizioni ed azioni condizionate

La logica prescrittiva o normativa o **Deontica**, è la logica che ha per oggetto enunciati su azioni, con degli aspetti di tipo "operativo".

Tali enunciati sono Modalità di atto; se ne possono definire molteplici, in genere sono varianti della Modalità **Obbligatorio** (**O**), e della sua duale **Permesso**:

$\mathbf{P} \equiv \neg\mathbf{O}\neg$.

Nel paragrafo precedente si è accennato ad azioni composte, ottenute da azioni più semplici mediante l'uso di connettivi, l'arricchimento del linguaggio si ottiene tramite l'introduzione di azioni condizionate e di prescrizioni.

Anche queste ultime possono considerarsi come azioni, e sono, in generale, una delle tre possibili interpretazioni delle clausole che le esprimono:

```
┌── prescrizioni
│
│
┤
│                      ┌─ con valore di verità
│                      │
└── asserzioni ────────┤
                       │
                       └─ indeterminate
```

Esempi:

Si consideri la clausola:

L'atto α è da eseguire prima dell'atto ß

Può essere sia una prescrizione (o istruzione o comando) che una asserzione con o senza valore di verità associato.

Considerazioni analoghe si possono fare per la clausola:

Vietato fumare

che è interpretabile come una prescrizione di tipo negativo o come asserzione.

Si possono considerare come azioni anche prescrizioni in cui più azioni sono fra di loro subordinate o sono soggette a condizioni: ad esempio un reticolo i cui nodi, α, ß, ..., rappresentano attività, e gli archi le precedenze fra queste: gli archi sono delle prescrizioni, anzi, il reticolo è descrivibile da una serie di prescrizione del tipo:

D(α,ß) ≡ α dopo ß.

Ancora, in un programma scritto, per esempio, in COBOL, vi sono due tipi di prescrizione; le prime sono rappresentate dalle istruzioni `GO TO`, le seconde sono implicite nella sequenza di esecuzione delle istruzioni. In un linguaggio di programmazione strutturato, vi sono solamente queste ultime.

La differenza fra azioni condizionate e prescrizioni è nel tipo di condizionamento: nel primo caso l'elemento condizionante è un predicato, nel secondo è una relazione fra azioni.

1.2.2 Interpretazione modale delle clausole di Horn.

Le clausole di Horn sono facilmente interpretabili nella logica modale prescrittiva, tramite una semplice trascrizione in cui se p un enunciato ed α una azione:

if p then α (**I**(p,α)), corrisponde a:

p => **O**(α) 1.2.2.1

e **if p then (α, β, ...)** corrispnde a

p => **P**(α,β,...)

se si vuole introdurre il concetto di clausola di Horn nondeterministica.

D'altra parte le Clausole di Horn sono essenzialmente dei "pezzi" di programma, ne segue che si può associare ad un programma una o più proposizioni, e quindi un valore di verità.

La 1.2.2.1 ha valore di verità "Vero", a meno di comportamento scorretto del programma, cioè, a fronte di una istruzione del tipo **I**(p,α), con p "Vero", **O**α è "Falso" perché α non viene eseguito.

Il programma ha valore "Vero" anche nel caso in cui p è "Falso" e **O**α è "Vero", ma ciò non corrisponde necessariamente ad un malfunzionamento, infatti α potrebbe essere stata eseguita in uno stato precedente.

Interpretando la 1.2.2.1 come p & **O**(α), non si ha questa apparente inconsistenza, tuttavia la formula corrispondente al programma, origina il valore "falso" con un comportamento corretto, ad esempio se p è "falso", con un comportamento corretto del programma, sarebbe "falso" **`if p then α`**.

Una interpretazione senza gli inconvenienti precedenti, è: p ≡ **O**(α), tuttavia, assumendo questa interpretazione, da **`if p then if q then α`** non è derivabile **`if p and q then α`**, infatti per la prima formula si ha p ≡ (q ≡ **O**α) e per la seconda (p & q) ≡ **O**α, ma se p = q = **O**α = "Falso", la prima formula è falsa e la seconda è vera.

Tutto ciò suggerisce l'insufficienza della logica classica nel dominare strutture dinamiche, anche se la struttura che meglio a ciò si adatta, sembra essere l'implicazione.

1.2.3 I programmi come formule modali prescrittive

Qualsiasi programma è costruibile, come è noto, con tre strutture: `Let`, `If_then_else` e `While_do`.

Indicando con **p**, **q** dei predicati, con **α**, **ß** delle azioni, le tre strutture di base di un programma, sono esprimibili come:

```
α                   Let α
E(p,α,ß)            If p Then Let α Else Let ß
W(p(x),α(x))        While p(x) do α(x)
```

È possibile introdurre la struttura `If-then`:

```
I(p,α)              If p Let α
```

Anche **I**(**p**,α), E(**p**,**α**,**ß**) e **W**(**p**(x),**α**(x)) sono azioni, l' operatore modale **L** (Let) le trasforma in enunciati:

Lα	"vero" se **α** è stata eseguita
LI(**p**,**α**)	"vero" se **I**(**p**,**α**) è stata eseguita e (**p** => **Lα**)
LE(**p**,**α**,**ß**)	"vero" se **E**(**p**,**α**,**ß**) è stata eseguita e (**p** => **Lα**) & (¬**p** => **Lß**)
LW(**p**(x),**α**(x))	"vero" se **W**(**p**(x),**α**(x)) è stata eseguita e, essendo x_0 il valore di x all'inizio del While:

```
∀(x)((x ε{xi|(xi=x0)v(p(xi)&(xi=α(xi-1))}
=> (p(x) => Lα(x))
```

Con queste notazioni, per esempio un programma di divisione ed uno di ricerca del maggiore in un insieme di interi, possono essere scritti, in forma di enunciato:

```
Lx <-; Ly <-; L(k <- 0); LW(x > y,L(x <- x-y); L(k <- k+1))
Lx <-; La(x) <-;(Lm <- 0);(Lk <- 0);
       LW(k < x, L(k <- k+1);LI(a(k) > m,L(m <- a(k))))
```

Poiché i programmi non sono statici, in qualche modo va indicato l'ordine di esecuzione delle istruzioni, ma una formalizzazione esplicita appesantirebbe la notazione: di fatto le convenzioni sulla direzione della lettura e sull'uso delle parentesi, sono sufficienti sostituti di queste modalità normativo-temporali.

Con la condizione espressa nelle definizioni, cioè che l'enunciato-istruzione è vero se l'istruzione è stata eseguita, e considerando il segno `;` come **and**, risulta che il programma ha valore vero solo se esso termina e nessuna istruzione si è comportata in modo errato, ad esempio una istruzione **I**(**p**,**α**) con **p** "vero" ed **α** non eseguito.

1.2.4 Esempi di trasformazione di programmi

Nel paragrafo precedente sono state associate alle strutture di base di un programma, degli enunciati, qui di seguito, in modo puramente esemplificativo, si dimostra che dei frammenti di programma equivalenti corrispondono a enunciati equivalenti.

Siano definiti **LI**(**p**,**α**) e **LE**(**p**,**α**,**ß**) come:

```
D1 LI(p,α) =df p => L(α)
D2 LE(p,α,ß) =df ((p => L(α)) & (¬p => L(ß)))
```

Si abbiano i due frammenti di programma equivalenti:

Si dimostra che è **LI**(**p**,**I**(**q**,**α**)) ≡ **LI**(**p**&**q**,**α**):

```
LI(p,I(q,α)) = p => LI(q,α)              D1
             = p => (q => L(α))          D1
             = p&q => L(α)           importazione
             = LI(p&q,α)                 D1
LI(p&q,α)  = p&q => L(α)                 D1
           = p => (q => L(α))        esportazione
           = p => LI(q,α)                D1
           = LI(p,I(q,α))                D1          ■
```

Altro esempio: i due diagrammi a blocchi riportati qui di seguito, sono equivalenti:

```
LI(p,E(q,α,ß))                        LE(p&q,α,I(p,ß))
```

```
LI(p,E(q,α,ß)) =
      p => LE(q,α,ß)
      p => ((q => Lα) & (¬q => Lß))                    D2
      ¬p v ((q => Lα) & (¬q => Lß))
      (¬p v (q => Lα)) & (¬p v (¬q => Lß))          distribuzione
      (p => (q => Lα)) & (p => (¬q => Lß))
      (p => (q => Lα)) & ((p & ¬q) => Lß))          importazione
      (p & q => Lα) & (¬p v q v Lß)                 importazione
      (p & q => Lα) & ((¬p v q) & (¬p v p) v Lß)
      (p & q => Lα) & ((¬p v (q & p) v Lß)          distribuzione
      (p & q => Lα) & (¬ (q & p) => (p => Lß))

LE(p&q,α,I(p,ß))
LE(p&q,α,I(p,ß)) =
      (p & q => Lα) & (¬ (p & q) => LI(p,ß))          D2
      (p & q => Lα) & (¬ (p & q) => (p => Lß))        D1
      (p => (q => Lα)) & ((¬ (p & q) & p) => Lß)   esp./imp.
```

$(\mathbf{p} \Rightarrow (\mathbf{q} \Rightarrow \mathbf{L}\alpha))\ \&\ (((\neg p \lor \neg q)\ \&\ p) \Rightarrow \mathbf{L}\beta)$

$(\mathbf{p} \Rightarrow (\mathbf{q} \Rightarrow \mathbf{L}\alpha))\ \&\ ((\neg \mathbf{q}\ \&\ \mathbf{p}) \Rightarrow \mathbf{L}\beta)$

$(\mathbf{p} \Rightarrow (\mathbf{q} \Rightarrow \mathbf{L}\alpha))\ \&\ (\mathbf{p} \Rightarrow (\neg \mathbf{q} \Rightarrow \mathbf{L}\beta))$ esport.

$(\mathbf{p} \Rightarrow ((\mathbf{q} \Rightarrow \mathbf{L}\alpha)\ \&\ (\neg \mathbf{q} \Rightarrow \mathbf{L}\beta))$ distrib.

$\mathbf{LI}(\mathbf{p},\mathbf{E}(\mathbf{q},\alpha,\beta))$ ■

1.3 Modalità epistemiche

Il modo di conoscere i fatti è almeno duplice, in ciò si riflette la dicotomia semantica fra estensione ed intensione.

Per l’operatore epistemico classico, in genere denotato con **K**, si ammette l'assioma **Kp => p**, si conoscono solo dei fatti veri; ciò rappresenta una conoscenza intensionale.

Nella conoscenza estensionale, denotata con C, si conosce o meno, il valore di verità di un fatto, ed è naturale assumere per questa modalità, la relazione:

$\mathbf{Cp} \equiv \mathbf{C}\neg\mathbf{p}$

Se si conosce il valore di verità di un fatto, si conosce anche il valore di verità del fatto negato.

Le procedure di validità relative alla modalità fanno riferimento ai mondi possibili, in particolare questi modelli, nel caso di Necessità e Possibilità, hanno una spiccata connotazione intensionale.

Infatti, se invece di mondi possibili, si considerano stati della realtà, una proposizione è necessariamente vera se è vera in tutti gli stati considerati, possibile se è vera in almeno uno di essi.

Il modello della conoscenza in [HM] è quello dei mondi possibili, e le procedure di validazione sono quelle di **T**, **S4** o **S5**; ciò in virtù della identità di assiomatizzazione di **K** ed **L**.

L'interpretazione che ne viene data, non sembra del tutto soddisfacente, non tanto perché non si distingue estensionalmente da quella della Necessità, ma perché questo modello presuppone una onniscienza logica in forza principalmente della regola di necessitazione ([HM] pag. 489 che cita Hintikka).

In [HM] è messo sotto accusa anche l'assioma di deduzione epistemica: (**K**p & **K**(p => q)) => **K**q, ma esso, esprime solamente un comportamento deduttivo "razionale".

Al paragrafo 2.6.4 sarà proposta una interpretazione modale che ovvia in parte a questi inconvenienti.

1.4 Modalità diadiche

Il termine *Modalità diadica* indica un operatore modale su due enunciati: le modalità diadiche hanno una loro presenza nella logica modale, ad esempio l'operatore Implicita: **L**(p => q), la negazione alternativa stretta: ¬**M**(p & q) ([HC] pag. 47, 339); Åqvist ([Å] pag 688 e seguenti) usa espressamente il termine diadico, per indicare l' obbligo condizionato:

$\mathbf{O}_B(A) =_{df} L(QB \Rightarrow A)$ [3]

Sono possibili più alternative, riconducibili come per Implicita, ad espressioni con operatori unari, o del tutto arbitrarie, in questo caso, però il sistema risultante potrebbe non essere più decidibile ([RU] pag. 49).

Nel caso, piuttosto semplice, in cui l'operatore modale è applicato ad un connettivo vero-funzionale Φ (Φ = and, or, ecc...), e il sistema modale includa T, intercorrono le seguenti relazioni:

$\Phi(p,q) \Rightarrow \mathbf{M}\Phi(p,q)$

$\mathbf{L}\Phi(p,q) \Rightarrow \Phi(p,q)$

(da p => **M**p e **L**p => p per sostituzione p/Φ(p,q)

Si ottengono modalità diadiche, un po' più complesse considerando dei connettivi binari in cui il valore di verità non è del tutto individuato da relazioni tabellari; ad esempio, la seguente tavola, rappresenta la tavola di verità di uno dei molti "implica" modali:

[3] Per il significato di **Q** vedasi pag. 7

p/q	F	V
F	V	V
V	F	?

Φ'(p,q) 1.4.1

Il valore di verità non è del tutto determinato nel mondo reale poiché in un elemento della tavola di verità è presente ? (componente modale); nell'esempio, in tre casi su quattro il valore di verità è verofunzionale, nel quarto caso, la verità di p e q non è sufficiente per stabilire la verità della formula Φ '(p,q).

Si può scegliere di esprimere la componente modale o in termini di Necessità o di Possibilità, ottenendo due operatori diversi; se si sceglie l'espressione in termini di necessità, allora la componente modale assume il valore vero solo se essa è necessariamente vera, ad esempio la 1.4.1 in forma normale disgiuntiva, con la parte modale espressa in termini di Necessità, diventa:

$(\neg p \& \neg q) \lor (\neg p \& q) \lor L(p \& q) = \neg p \lor L(p \& q)$

Nel caso di espressioni in termini di Necessità, vale la seguente proprietà:

Se $\boldsymbol{\Phi}$ è un connettivo, indicando con $\boldsymbol{\Phi}'$ il corrispondente connettivo modale, allora si ha:

$\boldsymbol{\Phi}'(p,q) \Rightarrow \boldsymbol{\Phi}(p,q)$

Infatti la tavola di verità dell'operatore modale differisce dalla tavola dell'operatore verofunzionale, per aver **?** invece di vero, quindi Vero implica Vero oppure Falso implica Vero, ma in entrambi i casi il valore di verità è Vero.
È possibile dare definizioni più complesse delle precedenti, definizioni in cui i valori di verità sono determinati, ad esempio, da una delle seguenti regole:

Sia **p' & q'** la componente modale dell'operatore, ricavata esprimendo l'operatore stesso in forma normale disgiuntiva, allora **p' & q'** è vero se:

i) ovunque **p'** è vero, **q'** è vero, oppure
ii) ovunque **q'** è vero, **p'** è vero.

La scelta di una, piuttosto che dell'altra regola, determina il significato intuitivo dell'operatore modale, per esempio se si indica con **&'** l'equivalente modale di **AND**, le clausole i) e ii), originano rispettivamente le formule:

$p_i \& q_i \& \mathbf{L}(p_i \Rightarrow q_i)$ oppure
$p_i \& q_i \& \mathbf{L}(q_i \Rightarrow p_i)$

Ci si può sbizzarrire nel fornire formulazioni modali su due variabili, esse sono interessanti se descrivono una porzione di realtà; più avanti, al paragrafo 3.6, sarà sviluppata una trattazione modale degli operatori diadici temporali Prima e Dopo.

2 INTERPRETAZIONI ALGEBRICHE

2.1 Premessa

In questi paragrafi è sviluppata una interpretazione algebrica di alcuni sistemi modali; questa tecnica permette di calcolare il valore di una formula piuttosto che dimostrarla, rendendo più stretto il rapporto fra semantica e sintassi.

Un intero capitolo in [HC] è dedicato ad una trattazione algebrica delle Modalità, i numerosi rimandi bibliografici testimoniano la gran mole di lavori in questo campo.

In [KK] nel capitolo VI.5 (pag. 486 e seg.), vi è una esposizione della teoria delle relazioni di Peirce (vedasi A5 per una breve sintesi), con l'introduzione dei simboli **Π** e **Σ** per esprimere il prodotto e la somma logica su tutti gli oggetti di un certo insieme. Anche nella logica modale temporale ([vB] pag. 17) sono utilizzati strumenti algebrici.

La definizione algebrica di modalità presente in questo lavoro, differisce da quella data in [HC] pag. 354-374, pur essendoci qualche riscontro nella trattazione algebrica della T-validità ([HC] pag. 370 - 371) ed ha dei punti di analogia con la Teoria delle Relazioni di Peirce.

2.2 Calcolo proposizionale ed interpretazioni Algebriche

L'algebra si basa su di un insieme non vuoto I di elementi o variabili proposizionali {p, q, r, ...} in quantità eventualmente infinita numerabile, un operatore unario {¬} ed un operatore binario { ·} per generare delle formule, con l'ausilio di parentesi tonde quali simboli sintattici di precedenza:

- ogni variabile proposizionale è una formula,
- se P e Q sono formule, ¬ (P) e (P ·Q) sono formule.

Per comodità si definiscono gli operatori binari derivati + e =>:

$$p + q =_{df} \neg(\neg p \cdot \neg q)$$
$$p \Rightarrow q =_{df} \neg p + q$$

Le formule:

```
A1) (p + p) => p
A2) q => (p + q)
A3) (p + q) => (q + p)
A4) (¬q + r) => ((¬p · ¬q) + p + r)
```

sono dette Teoremi, e Teoremi sono anche le formule che si ottengono da A1 - A4 per sostituzione uniforme, e quelle ottenute applicando la regola del Modus Ponens: se `p` e `p => q` sono Teoremi, `q` è un Teorema.

Come sinonimo di Teorema, si userà il simbolo `1`, mentre il simbolo `0` indicherà il nome di una formula la cui negazione è un Teorema; delle formule, infine, non hanno un nome determinato.

Per inciso, le regole per individuare i nomi di formule ottenute combinando formule tramite operatori, in sostanza le tavole di verità degli operatori, sono deducibili dalle definizioni date, ad esempio, per l'operatore **+**:

```
1 + 0 = 1     da (p + ¬p) + (p · ¬p)
0 + 1 = 1     da (p · ¬p) + (p + ¬p)
1 + 1 = 1     da (p + ¬p) + (p + ¬p)
0 + 0 = 0     da (p · ¬p) + (p · ¬p)
```

Ovviamente ogni formula a destra è un teorema o lo è la sua negazione.

Infine, si definisce la relazione binaria **≤** fra due formule:

$$p \leq q$$

sse `p` e `q` hanno lo stesso nome, oppure il nome di `q` è `1`.

Ne segue che se `p => q` ha nome `1`, cioè è un Teorema, in forza della tavola di verità dell'implicazione, sussiste la relazione `p ≤ q`.

Questa assiomatizzazione è equivalente al calcolo proposizionale (algebra di Lindenbaum), qui di seguito ne sono ripresi i punti fondamentali.

$P \& Q$ equivale a $p \cdot q$ (o pq omettendo il segno $\cdot$)
$P \vee Q$ equivale a $p + q$
$\neg P$ equivale a $\neg p$

ed inoltre:

- $P \Rightarrow Q$ equivale a $\neg p + q$;
- se => è l'operatore principale di un Teorema, questo è associato alla relazione ≤:

$$\vdash p \Rightarrow q \dashrightarrow p \leq q$$

Poiché ad una formula valida corrisponderà il valore 1, il Modus Ponens è consistente con questa Algebra:

Se $p = 1$ e $p \Rightarrow q = 1$ $(p \leq q)$ allora $q = 1$.

Per i 4 assiomi di PC si ha:

A1) $P \vee P \Rightarrow P$
$p + p \leq p$

A2) $Q \Rightarrow P \vee Q$
$q \leq p + q$

A3) $P \vee Q \Rightarrow Q \vee P$
$p + q \leq q + p$

A4) $(Q \Rightarrow R) \Rightarrow ((P \vee Q) \Rightarrow (P \vee R))$

$$\begin{aligned} \neg q + r &\leq \neg p \cdot \neg q + p + r \\ \neg q &\leq \neg p \cdot \neg q + p \\ \neg q \cdot q &\leq (\neg p \cdot \neg q + p) \cdot q \\ 0 &\leq p \cdot q \end{aligned}$$

Se a è una formula che vale 1 a prescindere dal valore dato alle variabili che la costituiscono, allora la sostituzione uniforme di una variabile con una fbf genera una formula che vale ancora 1.

2.3 Estensione degli operatori algebrici e modalità.

(in corso di costruzione)

3 LOGICA TEMPORALE

3.1 Logica temporale e logica classica.

In genere si ammette la scarsa capacità del calcolo dei predicati nel trattare enunciati con riferimenti temporali, ad esempio una deduzione del tipo:

Mario sta scrivendo

Mario avrà scritto

non è derivabile nella logica classica.

Un modo per permettere questo tipo di inferenze è di arricchire la logica classica, di operatori modali temporali, o di opportuni predicati, o di entrambi.

Tuttavia, fino a tempi abbastanza recenti, nel campo della Intelligenza Artificiale sono stati trascurati gli aspetti temporali per due motivi sostanzialmente opposti: il primo è fondato sull'opinione che l'introduzione di aspetti temporali aumenta la complessità dei sistemi senza peraltro arricchirli in proporzione; il seconda deriva, al contrario, al ritenere che l'inclusione di aspetti temporali in una teoria, non richieda significative variazioni concettuali.

Le rappresentazioni del tempo sono essenziali, ad esempio nella diagnostica medica, in cui vi è necessità di rappresentare il sintomo e la sua causa; o nella comprensione di storie, che esige una adeguata rappresentazione del passato; ed infine, nella pianificazione, in cui si deve poter trattare con degli scenari futuri.

Alcuni sistemi utilizzano un meccanismo temporale interno, ma ciò, come rileva Mc Dermott, porta a confondere credenze erronee con asserzioni non più vere; egli illustra ciò con un esempio divertente: un programma risolutore di problemi (Dudley) è posto di fronte ad una situazione in cui una ragazza, (Nelly), è legata a dei binari su cui è in arrivo un treno.

Dudley sa che Nelly sta per essere investita, e deve toglierla dai binari; quindi traccia un piano per raggiungere Nelly e salvarla, il piano è aggiunto al Data Base dei fatti e delle conoscenze.

Il meccanismo che garantisce la consistenza del Data Base osserva che l'intento di Dudley è di realizzare il piano di salvataggio, quindi Nelly non sarà più travolta, allora l'intenzione di salvare la ragazza, è rimossa dal Data Base, assieme al piano. Di conseguenza ritorna vero che Nelly sta per essere travolta.

L'esempio mette in evidenza gli inconvenienti dovuti alla mancanza di un meccanismo per trattare il cambiamento del valore di verità di un asserto, cioè di un meccanismo temporale.

3.2 Elementi di logica temporale

L' apparato modale è sufficientemente stabile, in quanto la semantica kripkiana si adatta in modo naturale alle rappresentazioni del tempo; l'arricchimento del calcolo modale con predicati temporali è un campo di studi rigogliosi, ancora in evoluzione.

Gli operatori temporali sono:

- **F** talvolta in futuro
- **P** talvolta in passato
- **G** sempre in futuro: ¬**F**¬
- **H** sempre in passato: ¬**P**¬

La cornice temporale è data da un insieme non vuoto di punti o istanti temporali (T), una relazione binaria di precedenza (**R**), ed una funzione (h) che associa ad ogni coppia, proposizione atomica ed istante temporale, un valore nell'insieme {0, 1}.

La funzione h è estesa a fbf, ad esempio:

```
h(t,Pq) = 1 sse ∃(esiste) (t')(R(t',t) & (h(t',q)=1))
h(t,Fq) = 1 sse ∃(t')(R(t,t') & (h(t',q)=1))
```

Un semplice sistema di logica temporale detto minimale o **K**, ha i seguenti assiomi:

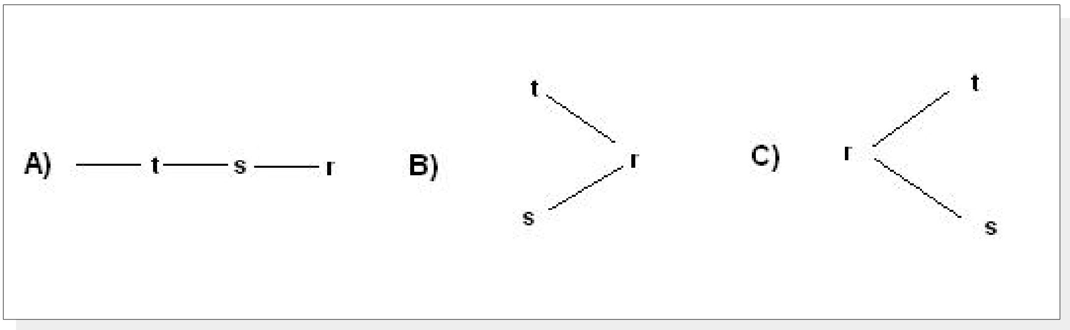
A1) p, con p una tautologia del calcolo proposizionale

A2) $\mathbf{G}\ (p \Rightarrow q) \Rightarrow (\mathbf{U}p \Rightarrow \mathbf{U}q)$
A3) $\mathbf{H}(p \Rightarrow q) \Rightarrow (\mathbf{U}p \Rightarrow \mathbf{U}q)$
A4) $q \Rightarrow \mathbf{H}q$
A5) $q \Rightarrow \mathbf{GP}q$
A6) $\mathbf{G}q$ se q è un assioma
A7) $\mathbf{H}q$ se q è un assioma

e il Modus Ponens come regola di inferenza.

Nel capitolo precedente si è visto quale aspetto algebrico insiemistico A2, A3, A6 e A7 rappresentano; A4 e A5 formalizzano l'esistenza della relazione di precedenza inversa, vale a dire se per a_i e a_j esiste $R(a_i, a_j)$, e quindi $a_j \ \varepsilon \ I_F(a_i)$, allora $a_i \ \varepsilon \ I_P(a_j)$.

Imponendo ulteriori condizioni ad R ed implicitamente anche a T, si ottengono dei sistemi che, nel riflettere tali condizioni, arricchiscono il sistema di base di ulteriori assiomi.



Gli schemi seguenti illustrano possibili grafi di precedenze relativi a tre punti "temporali":

Il caso A) è descritto dalla condizione:

R1) $\forall(t)\forall(s)\forall(r)((R(t,s)\&R(s,r) \Rightarrow R(t,r))$

ed è la transitività fra punti temporali.

Le condizioni:

R2) $\forall(t)\forall(s)\forall(r)((R(t,r)\&R(s,r) \Rightarrow R(t,s) \vee (t=s) \vee R(s,t))$
R2') $\forall(t)\forall(s)\forall(r)((R(r,t)\&R(r,s) \Rightarrow R(t,s) \vee (t=s) \vee R(s,t))$

negano rispettivamente le situazioni B) e C); in particolare la condizione R2) soddisfa l'idiosincrasia verso un passato non univoco. Gli assiomi per un tempo non circolare, con passato univoco e futuro aperto, sono:

A8) $\mathbf{FF}q \Rightarrow \mathbf{F}q$

A9) $(\mathbf{P}q \ \& \ \mathbf{P}r) \Rightarrow (\mathbf{P}(q \ \& \ r) \vee \mathbf{P}(q \ \& \ \mathbf{P}r) \vee \mathbf{P}(\mathbf{P}q \ \& \ r)))$

che esprimono modalmente R1) e R2).

Se vogliamo un futuro certo, ed è il caso in fisica del tempo assoluto Newtoniano, allora occorre imporre la condizione R2') ed un assioma analogo ad A9) con l' operatore **F** al posto dell' operatore **P**.

L' eternità è garantita da:

$\forall(s)\exists(t)R(t,s)$

$\forall(s)\exists(t)R(s,t)$

cui corrispondono gli assiomi:

A11) $\mathbf{G}q \Rightarrow \mathbf{F}q$

A12) $\mathbf{H}q \Rightarrow \mathbf{P}q$

(Questi assiomi sembrano condizioni necessarie ma non sufficienti, infatti A11) è analogo a S5D1) della Logica Deontica, che ammette modelli finiti; in [B] pag. 8 detti assiomi sono sostituiti rispettivamente da **F** "vero" e **P** "vero", che sono indubbiamente più forti, infatti i primi algebricamente diventano $\mathbf{\Pi}q \leq \mathbf{\Sigma}q$, mentre per i secondi è $\mathbf{\Sigma}q = 1$.)

La condizione che il tempo sia denso, cioè tale che fra due "momenti" se ne possa individuare sempre uno intermedio, è data dalla relazione, e relativo assioma:

$\forall(s)\forall(t)\exists(r)(R(s,t) \Rightarrow (R(s,r) \ \& \ (R(r,t)))$

A13) **F**q => **FF**q

Va da se che a questo punto gli elementi di T appartengono o all' insieme dei numeri razionali **Q** o all' insieme dei numeri reali **R**.

Nel caso dei numeri reali, sorgono dei problemi relativi al punto di separazione fra due sottoinsiemi temporali non vuoti contigui; per evitare fenomeni di inconsistenza locale, nel caso in cui il punto appartenga ad entrambi gli intervalli, o la caduta locale del principio del terzo escluso, se il punto di separazione non appartiene né all'uno né all'altro degli intervalli, si conviene che ogni sottoinsieme temporale sia aperto inferiormente e chiuso superiormente (o viceversa).

La prima alternativa è espressa da:

```
∀(T1)∀(T2)((T=T1UT2&(∀(sεT1)∀(tεT2)R(s,t)))
                              =>
∃(s')(∀(sεT1)R(s,s')&(∀(tεT2)R(s',t))))
```

L'assioma è:

```
A14)  G((Gq => PGq) => (Gq => Hq)) &
      H((Gq=> PGq) => (Gq => Hq)) &
       ((Gq => PGq) => (Gq => Hq))
```

L' assioma A14) ha la forma **G**p & **H**p & p che corrisponde alla definizione di Necessità della scuola Aristotelico Megarica ([RU] pag. 125), vale a dire p è Necessario se è vero adesso, sempre vero in futuro e sempre vero in passato; quindi il sistema temporale con A14) "contiene" una parte modale[4].

Fin qui sono stati introdotti solamente assiomi modali, ma per descrivere la realtà in modo più comprensivo, sono state proposte delle formalizzazioni che, su una base modale, o meglio, su un particolare modello di tempo, innestano dei predicati che tentano di catturare la realtà. Una breve descrizione di alcuni di questi sistemi, è l'oggetto dei successivi paragrafi.

3.3 Alcuni esempi di logiche temporali

3.3.1 La logica temporale di Mc Dermott

La logica temporale di Mc Dermott si basa su una relazione di precedenza temporale transitiva, lineare a sinistra (il passato non si può modificare), infinita in entrambe le direzioni, densa e continua.

In sostanza si impongono sull' insieme dei numeri reali, delle relazioni che si esprimono tramite predicati. Su questa "piattaforma" di istanti temporali o stati, vengono inseriti vari predicati del primo ordine per descrivere oggetti temporali, quali "tempi", "stati", "fatti", ed "eventi".

I fatti sono definiti come oggetti che possono cambiare il loro valore di verità nel tempo, ma sono individuati in pratica dall'insieme di stati in cui sono veri ([T] pag.84): (ON A B) è in apparenza un fatto, in realtà è l'insieme di stati in cui A è attualmente in B.

Gli stati sono punti temporali con data, e l'universo può essere contemporaneamente in più di uno stato; gli stati ereditano le precedenze temporali; l'insieme di più stati forma una storia.

Gli eventi sono identificati con gli insiemi di intervalli in cui "qualcosa" è accaduto una volta.

Il lavoro di Mc Dermott sfiora, senza approfondire, argomenti filosofici insidiosi, in particolare sembra errato identificare un evento come un insieme di intervalli, inoltre la mescolanza fra oggetti e metalinguaggio, e il gran numero di assiomi impiegato, fa sorgere dei dubbi sulla consistenza della teoria.

3.3.2 La logica temporale di Allen

(in costruzione)

[4] Una sintesi delle equivalenze fra sistemi temporali e sistemi modali, si trova in [RU] pag, 125-137 e 258.

4 CONOSCENZA DEI PROGRAMMI

4.1 Generalità.

Questo capitolo è dedicato all'indagine del significato della conoscenza di un programma, a partire dalla conoscenza di una Macchina di Turing (MdT), per proseguire con la conoscenza di generici programmi scritti con un linguaggio strutturato di alto livello, linguaggio dotato di sole istruzioni `Let`, `If_Then_else` e `While_do`.

La conoscenza che un programma ha dell' ambiente si riconduce, in sostanza, al contenuto della memoria che esso utilizza, ed è di tipo **C** (5 (conoscenza del valore di verità) per quanto riguarda gli antecedenti delle clausole `I`, `W` ed `E`; è di tipo **K** per quanto riguarda il contenuto delle memorie.
Esemplificando: ad un certo punto del calcolo della divisione di `x = 23` per `y = 7`; il programma sa (**C**) che `9 > 7` è vero, e alla fine del calcolo sa (**K**) che il risultato è `3` ed il resto è `2`, perché le sue memorie contengono appunto questi valori.

La conoscenza dei programmi ha diversi aspetti, uno generico ed è relativo a quanta e quale conoscenza le istruzioni permettono di acquisire e modificare, un altro, è più specifico e riguarda la conoscenza di quei programmi il cui scopo è di agire su Basi di Conoscenza; infine l'ultimo aspetto, forse il più interessante, riguarda l'aspetto intensionale ed estensionale della conoscenza.

4.2 Conoscenza delle Macchine di Turing

(in fase di costruzione)

[5] Vedasi pag. ^E, paragrafo 1.3.

5 ELENCO DI SIMBOLI E ABBREVIAZIONI

operatori logici

=>	implica
&	e
v	o
ª	non
≡	equivale

operatori modali

K	Conosciuto
L	Necessario
M	Possibile
O	Obbligatorio
P	Permesso
F	in Futuro talvolta
G	in Futuro sempre
P	in Passato talvolta
H	in Passato sempre
;	operatore di sequenzializzazione

operatori e simboli algebrici

Σ	sommatoria
Π	produttoria
·	per
+	più
ª	complemento
≤	minore uguale
=	uguale

abbreviazioni, simboli metalogici e insiemistici

Π	tesi
-->	ne segue
ε	appartiene a
U	unione
sse	Se e solo se
fbf	formula ben formata
MdT	Macchina di Turing

B BIBLIOGRAFIA

Allen James F. [A]
Towards a General Theory of Action and Time in Artificial Intelligence
Elsevier Science Publishers
1984 Amsterdam Netherlands

Åqvist Lennart [Å]
Deontic Logic
in Handboock of philosophical logic
Reidel Publishing Company
Doordrecth Holland 1984

van Benthem Johan [vB]
Time, Logic and Computation
in Linear time, Branching Time and partial Order
in Logics and models for concurrency
Lecture Notes in Computer Science n 354
Springer Verlag - Berlin Heidelberg 1989

Maria Luisa Dalla Chiara Scabini [dCS]
Logica
Enciclopedia filosofica ISEDI
ISEDI Milano 1974

Hughes G.E. Cresswell M.J. [HC]
Introduzione alla logica modale
(introduzione di Claudio Pizzi)
Il Saggiatore collana Theoria 5
II edizione Milano 1983

Enciclopedia Feltrinelli Fisher [FF]
Matematica
Feltrinelli Editore
Milano 1967

Halpern J.H Moses Y. [HM]
A guide to the modal logic of Knoweledge and Belief
preliminary draft
Los Angeles 1985

Kneale W. C. Kneale M. [KK]
Storia della logica
A cura di A. Conte
Einaudi Torino 1972

Lewis H.R. Papadimitriou C.H. [LP]
Elements of the theory of computation
Prentice Hall Software series
Englewood cliffs, New Jersey 1981

Rescher Nicolas Urquhart Alasdair [RU]
Temporal Logic
Library of Exact Philosophy
Springer Verlag - Wien New York 1971

Russell Bertrand [R]
Linguaggio e realtà
Antologia a cura di Massimo Bufalini
Editori Laterza
Bari I edizione 1970
Cap. IV traduzione del saggio "On denoting" 1905

Turner [T]
Logics for Artificial Intelligence
Ellis Horwoot

von Wright Georg Henrik [vW]
Norme, verità e logica
tradotto da G. Pezzini, rivisto da A.A. Martino
INFORMATICA E DIRITTO n. 3 Anno IX /sett.-dic. 1983
Le Monnier Firenze