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, $\mathbf{\Pi}$ e $\mathbf{\Sigma}$, 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:

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

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:

$\forall(x)((x\ \varepsilon\{x_i|(x_i=x_0)\vee(\mathbf{p}(x_i)\&(x_i=\boldsymbol{\alpha}(x_{i-1}))\}$
$\Rightarrow\ (\mathbf{p}(x)\ \Rightarrow\ \mathbf{L}\boldsymbol{\alpha}(x))$

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 $\mathbf{I}(\mathbf{p},\boldsymbol{\alpha})$ con $\mathbf{p}$ "vero" ed $\boldsymbol{\alpha}$ non eseguito.

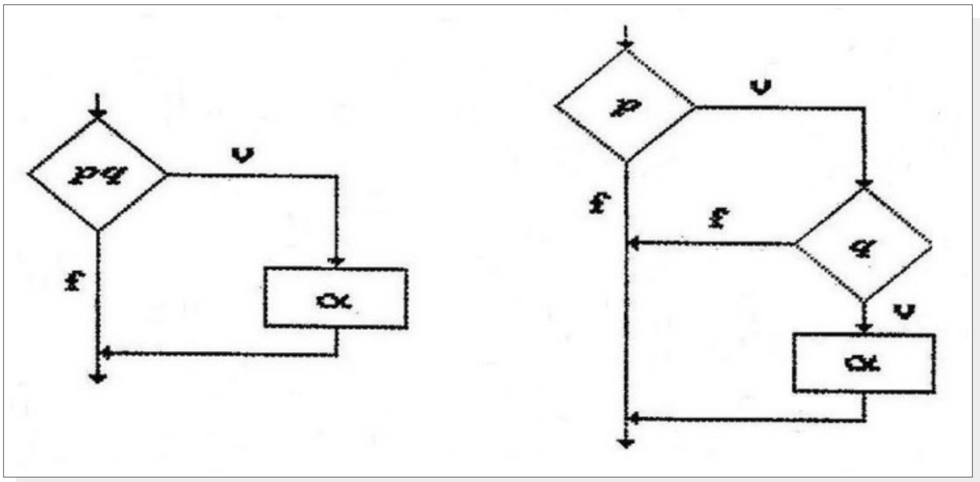
1.2.4 Esempi di trasformazione di programmi

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 $\mathbf{LI}(\mathbf{p},\boldsymbol{\alpha})$ e $\mathbf{LE}(\mathbf{p},\boldsymbol{\alpha},\boldsymbol{\beta})$ come:

D1 $\mathbf{LI}(\mathbf{p},\boldsymbol{\alpha}) =_{df} \mathbf{p} \Rightarrow \mathbf{L}(\boldsymbol{\alpha})$
D2 $\mathbf{LE}(\mathbf{p},\boldsymbol{\alpha},\boldsymbol{\beta}) =_{df} ((\mathbf{p} \Rightarrow \mathbf{L}(\boldsymbol{\alpha}))\ \&\ (\neg\mathbf{p} \Rightarrow \mathbf{L}(\boldsymbol{\beta})))$

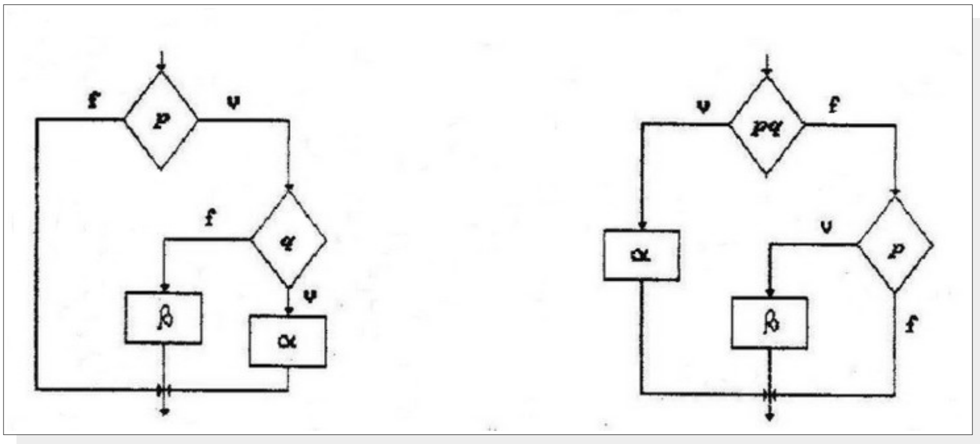
Si abbiano i due frammenti di programma equivalenti:



Si dimostra che è $\mathbf{LI}(\mathbf{p},\mathbf{I}(\mathbf{q},\boldsymbol{\alpha})) \equiv \mathbf{LI}(\mathbf{p}\&\mathbf{q},\boldsymbol{\alpha})$:

$\mathbf{LI}(\mathbf{p},\mathbf{I}(\mathbf{q},\boldsymbol{\alpha}))$	$= \mathbf{p} \Rightarrow \mathbf{LI}(\mathbf{q},\boldsymbol{\alpha})$	D1
	$= \mathbf{p} \Rightarrow (\mathbf{q} \Rightarrow \mathbf{L}(\boldsymbol{\alpha}))$	D1
	$= \mathbf{p}\&\mathbf{q} \Rightarrow \mathbf{L}(\boldsymbol{\alpha})$	importazione
	$= \mathbf{LI}(\mathbf{p}\&\mathbf{q},\boldsymbol{\alpha})$	D1
$\mathbf{LI}(\mathbf{p}\&\mathbf{q},\boldsymbol{\alpha})$	$= \mathbf{p}\&\mathbf{q} \Rightarrow \mathbf{L}(\boldsymbol{\alpha})$	D1
	$= p \Rightarrow (q \Rightarrow \mathbf{L}(\boldsymbol{\alpha}))$	esportazione
	$= p \Rightarrow \mathbf{LI}(\mathbf{q},\boldsymbol{\alpha})$	D1
	$= \mathbf{LI}(\mathbf{p},\mathbf{I}(\mathbf{q},\boldsymbol{\alpha}))$	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.
        (p => (q => Lα)) & (((¬p v ¬q) & p) => Lß)
        (p => (q => Lα)) & ((¬q & p) => Lß)
        (p => (q => Lα)) & (p => (¬q => Lß))        esport.
        (p => ((q => Lα) & (¬q => Lß)               distrib.
LI(p,E(q,α,ß))
```
■

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:

Cp ≡ C¬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 $p \Rightarrow \mathbf{M}p$ e $\mathbf{L}p \Rightarrow p$ per sostituzione $p/\Phi(p,q)$)

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:

Φ'(p,q)

p/q	F	V
F	V	V
V	F	?

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:

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

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

Se **Φ** è un connettivo, indicando con **Φ'** il corrispondente connettivo modale, allora si ha:

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

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.

[3] Per il significato di **Q** vedasi fine paragrafo 1.1 Modalità, definizioni e considerazioni

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 $\leq$:

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

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 questo paragrafo l'algebra sarà arricchita con due operazioni sugli elementi dell'insieme I; esse si basano sull'associazione ad ogni elemento appartenente ad I, di un sottoinsieme di I:

$I(p_i) = \{p_{i1}, p_{i2}, \ldots\}$

I due operatori, indicati con **Π** e **Σ**, sono interdefinibili e sono applicabili a qualsiasi fbf; ciò si esprime aggiungendo alla definizione di formula posta all' inizio del paragrafo 2.2:

- se P è una fbf, tali sono ΠP e ΣP.

Si definisce Πp_i come il prodotto di tutti i $p_{ij} \in I(p_i)$:

$$\Pi p_i = p_{i1} \cdot p_{i2} \cdot \ldots \cdot p_{in} \cdot \ldots \qquad p_{ij} \in I(p_i)$$

e **Σ**:

$$\Sigma p =_{df} \neg\Pi\neg p$$

Le regole che seguono indicano come calcolare una formula con gli operatori **Π** e **Σ**, di fatto trasformano delle fbf contenenti **Π** e **Σ**, in fbf senza di essi.

L' estensione di **Π** a fbf qualsiasi presuppone che gli elementi di ogni $I(p_i)$ siano considerati sempre nello stesso ordine; inoltre, in vista dell' applicazione del metodo ai Sistemi Modali "tradizionali", si richiede che l'ordine sia tale da soddisfare il seguente requisito: se la variabile p appartiene sia ad $I(q)$ che ad $I(r)$, p deve avere in questi insiemi la stessa posizione. Una conseguenza di ciò è che l'indice delle variabili proposizionali può non essere strettamente sequenziale, in questo caso gli insiemi sono lacunosi.

L'esemplificazione che segue vuol dare un'idea delle corrispondenze fra insiemi: il segno - indica la mancanza di una variabili proposizionale:

$I(p) = \{p_1, -, p_3, -, p_5, p_6, p_7, -\}$

$I(q) = \{q_1, q_2, -, q_4, -, q_6, -, q_8\}$

Poiché l'operazione è effettuata su variabili proposizionali appartenenti ad un insieme ordinato, e per qualche indice queste possono mancare, per tali indici le variabili proposizionali mancanti saranno sostituite dal valore `0` o `1` a seconda se l' operatore è **Π** o **Σ**; ciò rende non influente la presenza di "lacune", ad esempio con riferimento a `I(p)` ed `I(q)` dell' esempio si ha:

$$\Pi(p + q) = (p_1+q_1) \cdot q_2 \cdot p_3 \cdot q_4 \cdot p_5 \cdot (p_6+q_6) \cdot p_7 \cdot q_8$$

Sia **Φ** l'operatore che precede una fbf `P` non contenente **Π** o **Σ**, sia $\Phi' = \neg\Phi\neg$, `p` e `q` siano variabili proposizionali, eventualmente negate, si hanno i seguenti casi:

i) $P = \neg P'$, allora ΦP diventa $\neg\Phi P'$

ii) se $P = (p \cdot q)$, allora ΦP è, tenendo conto di quanto convenuto circa le lacune:

$$\Pi(p \cdot q) = (p_1 \cdot q_1) \cdot \ldots \cdot (p_m \cdot q_m) \qquad 2.3.1$$

$$\Sigma(p \cdot q) = (p_1 \cdot q_1) + \ldots + (p_m \cdot q_m)$$

Le definizioni valgono anche per formule con più lettere enunciative, ad esempio $\Pi(p + q + r + \ldots)$.

Dalla definizione di **Σ** in funzione di **Π**, si ricavano analoghe relazioni per $\Pi(p + q)$ e $\Sigma(p + q)$:

Si ricava, inoltre, dalle definizioni e dall'applicazione degli operatori, che vale:

$$\Pi(p \cdot q) = \Pi p \cdot \Pi q \qquad 2.3.1a$$

$$\Sigma(p + q) = \Sigma p + \Sigma q \qquad 2.3.1b$$

iii) se `P` è una fbf qualsiasi, essa va trasformata in forma normale congiuntiva se l'operatore è **Π**, altrimenti in forma normale disgiuntiva; quindi si applica 2.3.1a o 2.3.1b rispettivamente, ad esempio:

$$\Pi(p \cdot q + \neg r) \equiv \Pi((p + \neg r) \cdot (q + \neg r)) = \Pi(p + \neg r) \cdot \Pi(q + \neg r)$$

iv) nel caso in cui più operatori sono applicati ad una fbf `P` che non ne contiene:

$$\Phi_1\Phi_2 \ldots \Phi_n P$$

a `P` viene applicato per primo l' operatore più distante, nell'esempio Φ_1.

Esempio:

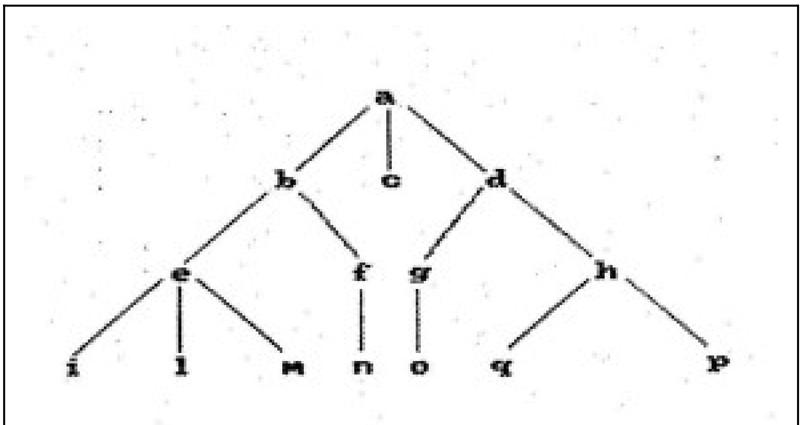


fig. 2.3.1

12

(Il grafo rappresenta le relazioni intercorrenti fra gli elementi di un insieme, ma per semplicità, non è stata segnata la relazione di un elemento con se stesso; la relazione è, quindi, riflessiva.)

Data questa struttura di Relazioni, gli insiemi sono:

```
I(a) = {a, b, c, d}
I(b) = {b, e, f}
I(c) = {c}
I(d) = {d, g, h}
......
I(e) = {e, i, l, m}
......
I(p) = {p}
```

E alcune equivalenze:

$\Sigma a \ = a+b+c+d$
$\Pi a \ = a \cdot b \cdot c \cdot d$
$\Pi\Sigma a = \Sigma(a \cdot b \cdot c \cdot d) = (a \cdot b \cdot c \cdot d)+(b \cdot e \cdot f)+c+(d \cdot g \cdot h)$

Sulle caratteristiche dell' insieme associato a p, non sono state fatte, finora, delle ipotesi oltre quella dell'ordine, d'altra parte, dato un qualsiasi, arbitrario sottoinsieme di I, ed un qualsiasi, arbitrario elemento p_j ε I, esiste, di fatto una relazione binaria $R(p_j, p_k)$ fra p_j e ogni p_k ε I. Il significato della relazione qui interessa relativamente, in quanto è un aspetto intensionale, interessano invece gli aspetti estensionali, cioè le proprietà formali delle Relazione, quali la riflessività, la simmetria, la transitività, ecc..

Prima di esaminare i riflessi algebrici di alcuni tipi di Relazione, saranno analizzate delle proprietà relative alla ordinalità e cardinalità degli insiemi $I(p)$.

L'assioma $\Pi(\neg p + q) \le (\neg\Pi p + \Pi q)$

$\Pi(\neg p + q) \le (\neg\Pi p + \Pi q)$ è un assioma se la corrispondenza fra gli elementi degli insiemi $I(p)$ e $I(q)$, utilizzata per applicare **Π** e **Σ**, è univoca. L'assioma vale per qualsiasi corrispondenza univoca, ma nei casi che più interessano, esiste un ordine, e la dimostrazione ne terrà conto.

In generale gli insiemi $I(p)$ ed $I(q)$ avranno delle variabili proposizionali in corrispondenza (si assumerà per comodità che siano le prime **n**), e rispettivamente **r** e **t** variabili non in corrispondenza, con riferimento all'esempio di pagina 12, avremo p_1, q_1 e p_6, q_6, e poi p_3, p_5, p_7 e q_2, q_4, q_8.

Per non appesantire la notazione, sarà indicato con lettere maiuscole indicizzate, il prodotto di variabili proposizionali:

$$P_n \equiv p_1 \cdot \ldots \cdot p_n \text{ e } \neg P_n \equiv \neg p_1 \cdot \ldots \cdot \neg p_n$$

(La seconda notazione, in particolare, non è rigorosa, ma nel contesto non dovrebbe generare confusione.)

Secondo le convenzioni adottate, il primo e secondo membro di $\Pi(\neg p + q) \le (\neg\Pi p + \Pi q)$ sono formati da:

$\Pi(\neg p + q)$	$\neg\Pi p + \Pi q$
i) $\neg P_n \cdot \neg P_r \cdot Q_t$	$\Sigma\neg p$
ii) $\neg P_n \cdot Q_n \cdot Q_t$	$Q_n \cdot Q_t$
~~iii) $\neg P_i \cdot \neg P_r \cdot Q_j \cdot Q_t$~~	
iii) $\neg p_i \cdot \neg P_r \cdot q_j \cdot Q_t$ (*)	

(*) iii) sono 2^2- 2 addendi, con i + j = n

i) è minore o uguale (implicato) dal secondo membro, infatti ogni $\neg p_i$ del prodotto è a destra come somma; ii) è implicato da $Q_n \cdot Q_t$; per ognuno degli iii) vale quanto detto per i). ■

La dimostrazione ha stabilito che l'assioma è coerente con l'abbinamento univoco fra gli elementi di $I(p)$ ed $I(q)$, il viceversa può essere dimostrato con un controesempio siano:

$$I(p) = \{ p_i, p_2\}, \ I(q) = \{q_1, q_2\}$$

e siano gli abbinamenti:

(p_i, q_i) e (p_2, q_i),

allora $\Pi(p \Rightarrow q) \Rightarrow (\Pi p \Rightarrow \Pi q)$ diventa:

$$(\neg p_1 + q_1) \cdot (\neg p_2 + q_1) \Rightarrow \neg p_1 + \neg p_2 + q_1q_2$$
$$\neg p_1\neg p_2 + \neg p_2q_1 + \neg p_1q_1 + q_1 \Rightarrow \neg p_1 + \neg p_2 + q_1q_2$$

che non è un teorema (si ponga $p_1 = p_2 = q_1 = 1$ e $q_2 = 0$).

Necessitazione

La regola di Necessitazione è legata alla cardinalità degli insiemi $I(p_1)$:

se per tutte le variabili proposizionali p, q è $|I(p)| = |I(q)|$, e se esiste p_1, esiste q_1, allora:

se $\vdash a \dashrightarrow \vdash \Pi a$

e viceversa.

In altre parole, sotto la condizione enunciata, se P è una formula che vale 1 a prescindere dal valore dato alle variabili p, q, .., che la costituiscono, allora vale:

$$\Pi P = 1$$

È sufficiente far vedere che ciò vale per gli assiomi di PM, in particolare ad esempio per A4, che nell'ipotesi di cui sopra, $\Pi((q \Rightarrow r) \Rightarrow ((p \vee q) \Rightarrow (p \vee r)))$ diventa $\Pi(1 \cdot 1 \cdot \ldots 1)$ in quanto:

$((q \Rightarrow r) \Rightarrow ((p \vee q) \Rightarrow (p \vee r))) \cdot$
$((q_1 \Rightarrow r_1) \Rightarrow ((p_1 \vee q_1) \Rightarrow (p_1 \vee r_1))) \cdot$
....
$((q_n \Rightarrow r_n) \Rightarrow ((p_n \vee q_n) \Rightarrow (p_n \vee r_n)))$ ■

Per il viceversa è sufficiente un controesempio: dato il teorema:

$p \Rightarrow p \vee q$

se $m = |I(p)| < n = |I(q)|$

allora $\Pi(\neg p + p + q)$ può non valere 1, se esiste un q_j $(m < j \leq n)$, per cui:

$\Pi(\neg p+p+q) =$

$(\neg p_1+p_1+q_1) \cdot \ldots \cdot (\neg p_m+p_m+q_m) \cdot q_{m+1} \cdot \ldots \cdot q_j \cdot \ldots q$

Assegnando il valore 0 a q_j si ha la $\Pi(\neg p+p+q) = 0$ ■

2.4 *Aspetti algebrici delle Relazioni*

2.4.1 Relazioni Riflessive

Se la relazione **R** è riflessiva, è verificata la seguente formula:

$\Pi p \leq p$

Viceversa se vale $\Pi p \leq p$, allora la Relazione è riflessiva.

Il primo membro è:

$p_1 \cdot p_2 \cdot \ldots.$

Se R è riflessiva è $R(p,p)$, quindi $p \ \varepsilon \ I(p)$ ed esiste un $p_i = p$, per cui è verificata (PC15):

$p_1 \cdot p_2 \cdot \ldots \cdot p_i \cdot \ldots \leq p$ ■

Viceversa, se vale $\Pi p \leq p$, e p, per assurdo, non appartiene a $I(p)$, si può dare una assegnazione di valori, per cui ad ogni p_i si associa il valore 1, e a p stesso il valore 0, contraddicendo l' ipotesi.

2.4.2 Relazioni Transitive

Se la relazione **R** è transitiva, allora vale:

$\Pi p \leq \Pi\Pi p$ 2.4.2

e vale anche il viceversa, cioè se vale $\Pi p \leq \Pi\Pi p$ la Relazione è transitiva.

Il primo membro è:

$p_1 \cdot \ldots \cdot p_n \cdot \ldots$

ed il secondo:

$p_{11} \cdot p_{21} \cdot \ldots. \cdot p_{12} \cdot p_{22} \cdot \ldots \cdot p_{1n} \cdot p_{1n} \cdot \ldots$

Poiché la relazione è transitiva, si ha:

$R(p, p_i)$ per ogni $p_i \ \varepsilon \ I(p)$
$R(p_i, p_{ij})$ per ogni $p_{ij} \ \varepsilon \ I(p_i)$
$R(p, p_{ij})$

da cui si deduce che esiste un $p_k \ \varepsilon \ I(p)$ tale che $p_{ij} = p_k$, quindi ogni elemento del secondo membro coincide con uno del primo membro, in conclusione si ha una formula riconducibile a:

$p \cdot q \leq p$ (PC15) ■

Viceversa se è $\Pi p \leq \Pi\Pi p$, allora la Relazione fra gli elementi di $I(p)$ è transitiva, si ha che:

$R(p,p_i)$ per ogni $p_i \ \varepsilon \ I(p)$
$R(p_i,p_{ij})$ per ogni $p_{ij} \ \varepsilon \ I(p_i)$

poiché vale la 2.4.2, ogni p_{ij} del secondo membro, deve coincidere con un p_k del primo membro:

$R(p_i,p_{ij}) = R(p_i,p_k)$
$R(p,p_k)$ perché $p_k \ \varepsilon \ I(p)$
$R(p,p_{ij})$ ■

Se la relazione è anche riflessiva, si può dimostrare che vale: $\Pi p = \Pi\Pi p$

2.4.3 Relazioni Simmetriche e Transitive

Se la relazione **R** è simmetrica e transitiva allora vale:

$\Sigma p \leq \Pi\Sigma p$ 2.4.3

Vale anche il viceversa: se è verificato $\Sigma p \leq \Pi\Sigma p$, allora la relazione è simmetrica e transitiva.

Se vale $\Sigma p \leq \Pi\Sigma p$ cioè:

$p_1 + p_2 + \ldots \leq$

$(p_{11} \cdot p_{21} \cdot \ldots) + (p_{12} \cdot p_{22} \cdot \ldots) + \ldots + (p_{1j} \cdot p_{2j} \cdot \ldots) + \ldots$

Si deduce che ogni p_i del primo membro, deve essere $\leq$ ad un $(p_{1j} \cdot p_{2j} \cdot \ldots)$ e deve essere $p_{1j} = p_{2j} = \ldots = p_i$.

Da $R(p,p_j)$ e $R(p_j,p_{jk})$ poiché per ipotesi esiste un p_i tale che $p_i = p_{jk}$, si ha $R(p_j,p_{jk})$, ma essendo anche $R(p, p_i)$, ne segue che la relazione è transitiva.

Da $R(p_j,p_{jk})$ e $p_{jk} = p_i$ si ha $R(p_j, p_i)$; poiché p_j è presente nel secondo membro, esiste un indice h tale che:

$p_j = p_{1h} = p_{2h} = \ldots$

In particolare è:

$p_j = p_{1h}$, cioè $R(p_i, p_{1h}) = R(p_i,p_j)$

ciò basta per concludere che la relazione è simmetrica. ■

Se la relazione R è simmetrica e transitiva, tutti i p_i del primo membro compaiono nel secondo membro, ed è:

$p_{1h} = p_{2h} = \ldots = p_{kh} = \ldots = p_i$

per cui vale:

$p_1 + p_2 + \ldots \leq$

$(p_{11} \cdot p_{21} \cdot \ldots) + (p_{12} \cdot p_{22} \cdot \ldots) + \ldots + (p_{1j} \cdot p_{2j} \cdot \ldots) + \ldots$

da $R(p,p_i)$ e $R(p_i, p_{ij})$ per transitività si ottiene $R(p, p_{ij})$, vale a dire che p_{ij} equivale ad un $p_k \; \varepsilon \; I(p)$, inoltre da $R(p_i,p_k)$ per simmetria si ha:

$R(p_h,p_i)$, cioè $p_i \equiv p_{hk} \; \varepsilon \; I(p_h)$.

Poiché è:

$R(p_i, p_{i1})$ e $R(p_{i1}, p_i)$
$R(p_i, p_{i2})$ e $R(p_{i2}, p_i)$
...
$R(p_i, p_{ij})$ e $R(p_{ij}, p_i)$
...

Si ha che p_i appartiene a tutti gli insiemi $I(p_{ij}) \equiv I(p_k)$ (tranne al più a $I(p_i)$ se la Relazione non è riflessiva), ed è esattamente l' i-esimo elemento, ma il generico addendo del secondo membro è il prodotto degli elementi di pari indice appartenenti ai vari $I(p_k)$, quindi esisterà un h-esimo addendo tale che:

$p_{1h} = p_{2h} = \ldots = p_{kh} = \ldots = p_i$ ■

2.5 Rapporti fra relazioni, condizioni e sistemi Modali

Le condizioni poste sulla cardinalità e ordinalità degli insiemi, le proprietà formali delle Relazioni, si riflettono sulla validità di certe formule, in particolare il contenuto dei paragrafi precedenti si può riassumere nei seguenti punti:

a) $\Pi p \leq p$ è soddisfatta sse la Relazione è transitiva, quindi se $p \; \varepsilon \; I(p)$,
b) la regola di Necessitazione presuppone $|I(p)| = |I(q)|$ per ogni p e q e per ogni p_i esiste il corrispondente q_i,
c) $\Pi(\neg p + q) \leq (\neg\Pi p + \Pi q)$ presuppone l'accoppiamento univoco fra gli elementi di $I(p)$ e $I(q)$,
d) $\Pi p \leq \Pi\Pi p$ è soddisfatta sse la relazione è transitiva,
e) $\Sigma p \leq \Pi\Sigma p$ è soddisfatta sse la relazione è simmetrica, e transitiva.

Le condizioni a) b) c) individuano il sistema modale **T**, e, queste più d) o e) individuano rispettivamente **S4** e **S5**.

2.6 Applicazione a modalità composte e non standard.

Questo paragrafo delinea alcune applicazioni del metodo algebrico alle Modalità fornendo qualche esempio concreto, senza aver la pretesa di essere una trattazione completa dei singoli argomenti.

2.6.1 Non riflessività.

Per un discreto numero di Modalità non è un assioma `Lp => p`, ciò vale ad esempio, per la modalità Deontica **Obbligatorio**, per la modalità Epistemica **Creduto** e per gli operatori modali Temporali "**nel passato**" e "**nel futuro**".

Per questi, se `p` è una variabile enunciativa ε I, `p` non appartiene a I(p), come nelle modalità temporali "in Passato" e "in Futuro", o può non appartenervi come nelle modalità Deontiche o Doxastiche (ciò che è un obbligo non sempre si verifica, e si possono credere cose false); in tutti questi casi la Relazione $R(p,p_i)$ non è riflessiva e poiché `p` può non appartenere ad I(p), può non valere algebricamente:

$$\Pi p \leq p$$

2.6.2 Modalità miste.

Un sistema con più modalità presuppone relazioni distinte fra gli elementi dell' insieme delle variabili, ad esempio le modalità temporali "talvolta nel passato" (**P**) e "talvolta nel futuro" (**F**), presuppongono, data una variabile proposizionale p, due diversi insiemi: uno è il passato di `p`, e l' altro, il suo futuro[4].

Ogni variabile enunciativa avrà tanti Insiemi corrispondenti quante sono le Modalità; essi saranno individuati da un pedice, ad esempio $I_A(p)$ e $I_D(p)$ sono gli insiemi corrispondenti di p rispettivamente per le Modalità Aletiche e Deontiche.

Un esempio di assioma di Modalità miste, Deontiche ed Aletiche è il seguente:

$$\mathbf{O}p \Rightarrow \mathbf{M}p$$

Ciò che è obbligatorio è possibile.
Questo assioma garantisce che la norma legislativa non imponga qualcosa di impossibile.

Gli insiemi relativi a p sono:

$$I_A(p) = \{p,\ p_1,\ p_2,\ \dots\}$$
$$I_D(p) = \{p_1',\ p_2',\ \dots \mid p_i\ \varepsilon\ I_A(p)\}$$

cioè $I_D(p)$ è totalmente incluso in $I_A(p)$; sotto questa ipotesi, infatti vale:

$$\Pi_D p \leq \Sigma_A p$$
$$p_1' \cdot p_2' \cdot \dots \leq p + p_1 + p_2 + \dots$$

2.6.3 Modalità Deontiche

Il sistema di logica Deontica S5D ([RU] pag. 136-137) è assiomatizzato come:

```
├ A  --> ├ OA
S5D1) Op => Pp
S5D2) O(p => q) => (Op => Oq)
S5D3) POp => Op
```

S5D3 è, in altra forma `Pp => OPp`.

Questo sistema è parzialmente riflessivo, in quanto S5D1) vale anche se la relazione non è riflessiva, tuttavia nel sistema è derivabile `OOp => Op`, da S5D1 per sostituzione `p/Op` e sillogismo con S5D3, che, in S5 classico è derivabile direttamente da `Op => p` per sostituzione `Op/p`, ma `OOp => Op` dice che l' insieme `I(Op)` è riflessivo.

In altre parole, mentre `I(p)` non è riflessivo, è riflessivo $I(p_i)$ con $p_i\ \varepsilon\ I(p)$.

In [vW] pag. 27 è data una interessante interpretazione di S5D3, nell' ottica del considerare la Modalità di Modalità, come dettati legislativi di un superiore livello, questo espresso come ¬(**PO**p & **P**¬p) vieta la deroga,

4 Il capitolo 3 tratta in modo più sistematico di logica temporale.

cioè se p, ad esempio, è "esigere un tributo", non può essere che l'autorità legislativa superiore (il Ministero delle Finanze) permetta ad un legislatore subordinato (il comune) di esigere un tributo (**PO**p) e contemporaneamente consente di evaderlo (**P**¬p).

2.6.4 Modalità Epistemiche

Nel paragrafo 2.3 è stato evidenziato che la regola di Necessitazione presuppone la equicardinalità degli insiemi associati alle variabili enunciative, ed è essenzialmente questa regola la responsabile della "omniscienza" logica della Modalità Epistemica. Questo inconveniente cade, parzialmente, se si sostituisce la regola di necessitazione con la regola più debole:

$\vdash (p \Rightarrow q) \dashrightarrow \vdash (\mathbf{K}p \Rightarrow \mathbf{K}q)$

Questa è derivabile dalla Necessitazione stessa (è la DR1 di [HC] pag. 50), ed è essenziale nella dimostrazione di:

$\mathbf{K}(p\&q) \equiv (\mathbf{K}p\&\mathbf{K}q)$ ([HC] pag. 52)

Il sistema modale costituito da una base assiomatica per PC e da:

A1) $\mathbf{K}p \Rightarrow p$,

A2) $\mathbf{K}(p \Rightarrow q) \Rightarrow (\mathbf{K}p \Rightarrow \mathbf{K}q)$

R1) $\vdash (\alpha \Rightarrow ß) \dashrightarrow \vdash (\mathbf{K}\alpha \Rightarrow \mathbf{K}ß)$

è il sistema E2 di Lemmon ([HC] pag. 344).

Il sistema di Lemmon sembra quindi più adatto a descrivere la conoscenza, che non sistemi contenenti la regola di Necessitazione, tuttavia R1) è ancora una regola troppo forte.

La regola R1) ha come condizione che gli I(p) non siano vuoti, in questo caso la Relazione si dice **seriale** ([HM] pag. 484): per ogni variabile proposizionale p esiste una variabile proposizionale r tale che esiste la relazione R(p,r).

Se I(p) è vuoto si porrà per convenzione:

$\mathbf{\Pi}p \equiv p$ 2.6.4

e di conseguenza $\mathbf{\Sigma}p \equiv p$, infatti $\neg(\mathbf{\Pi}p) \equiv \neg p$ e $\mathbf{\Pi}\neg p \equiv \neg p$

quindi $\neg\mathbf{\Pi}\neg p \equiv p \equiv \mathbf{\Sigma}p$.

Quindi, se esistono I(p) vuoti, DR1) non sussiste, infatti, si supponga $I(p) = \{\Phi\}$ e $I(q) = \{q_1\}$

1)	$p \Rightarrow p \vee q$	A2
2)	$\mathbf{K}p \Rightarrow \mathbf{K}(p \vee q)$	DR1)
3)	$\mathbf{K}p \equiv p$	2.6.4
4)	$\mathbf{K}(p \vee q) \equiv q_1$	applicazione di 2.3.1
5)	$p \Rightarrow q_1$	3 e 4

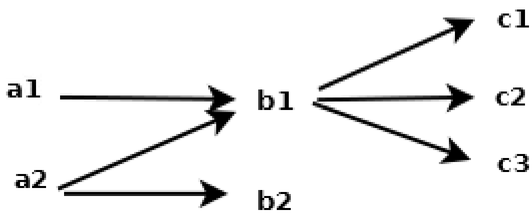
5) non è un teorema. ■

Per inciso ciò non vale per gli altri componenti la base assiomatica, infatti questi sopportano la "soppressione uniforme" di variabili, cioè eliminando uniformemente alcune variabili la fbf risultante è ancora un Teorema; ad esempio da A4:

$(q \Rightarrow r) \Rightarrow ((p \vee q) \Rightarrow (p \vee r))$	
$q \vee (\neg p \& \neg q) \vee p$	eliminazione di r
$(\neg p \& \neg q) \vee (p \vee q)$	
$\neg(p \vee q) \vee (p \vee q)$	De Morgan
$(p \vee q) \Rightarrow (p \vee q)$	A3 ■

2.6.5 Modalità Temporali

Si abbia il frammento temporale seguente:



In relazione alla proposizione b_1, si hanno i seguenti insiemi:

$I_F(b_1) = \{c_1, c_2, c_3\}$

$I_P(b_1) = \{a_1, a_2\}$

ed inoltre:

$I_F(a_2) = \{b_1, b_2\}$

$I_F(a_1) = \{b_1\}$

Supponendo che l'adesso coincida con la proposizione b_1, i vari simboli che compaiono, devono essere considerati la proposizione b_1 in diversi stati temporali: come si vede dall'esempio, si è in presenza di un tempo non lineare, vale a dire che nel futuro, nel passato, e nel presente di b_1, sono possibili stati diversi.

Nel calcolo temporale minimale **K** si ammette l'assioma: (v. 3.2):

p => **HF**p (**H** $=_{df}$ ¬**P**¬p)

esso può essere letto: "se p allora sempre in passato p è in qualche futuro".

L' applicazione algebrica a b_1 diventa:

$b_1 \leq \mathbf{\Pi}_P\mathbf{\Sigma}_F(b_1)$

$b_1 \leq \mathbf{\Sigma}_F(a_1 \cdot a_2)$

$b_1 \leq b_1 \cdot b_1 + b_2$

che è ovviamente vero.

In questo caso, ad esempio, non vale l' assioma che indica la transitività, nel futuro, del tempo,

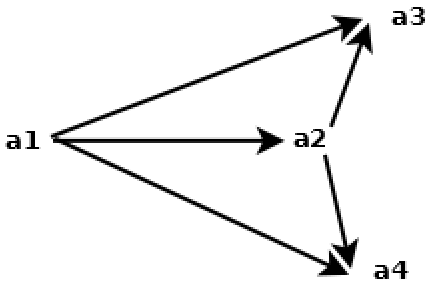
FFp => **F**p 2.6.5.1

infatti:

$\mathbf{\Sigma}_F\mathbf{\Sigma}_F(a_2) = \mathbf{\Sigma}_F(b_1 + b_2) = c_1 + c_2 + c_3$

$\mathbf{\Sigma}_F(a_2) = b_1 + b_2$

Questo invece vale nella situazione di transitività espressa dalla seguente figura:



FFp => **F**p è algebricamente:

$\mathbf{\Sigma}_F\mathbf{\Sigma}_F(a_1) \leq \mathbf{\Sigma}pFa_1$

$\mathbf{\Sigma}_F(a_2 + a_3 + a_4) \leq a_2 + a_3 + a_4$

$a_3 + a_4 \leq a_2 + a_3 + a_4$

La 2.6.5.1, espressa algebricamente è $\mathbf{\Sigma\Sigma}p \leq \mathbf{\Sigma}p$, essa è equivalente a $\mathbf{\Pi}p \leq \mathbf{\Pi\Pi}p$, che si è visto essere valida se e solo se la relazione **R** è transitiva (v. paragrafo 2.4.2).

2.7 I Mondi possibili e il punto di vista algebrico.

La Logica Modale si avvale del paradigma dei Mondi possibili(5 per rendere più intuitivo l'aspetto semantico senza peraltro ledere il rigore formale con cui è stabilito il modello di verifica ([HC] pag. 95 e seg.).

Questo si basa su di una tripla ordinata $<W,R,V>$, in cui W è un insieme di "Mondi", **R** una relazione diadica definita fra i membri di W, e V una funzione per assegnare uno dei due valori $\{0,1\}$ ad ogni coppia (w_i, p) formata da un "Mondo" e da una fbf p.

Il significato, intensionale e puramente intuitivo, della relazione **R** è l'accessibilità fra i vari mondi.

La relazione **R** è riflessiva per il sistema T, riflessiva e simmetrica per il sistema S4 e riflessiva, simmetrica transitiva per il sistema S5.

L' approccio algebrico alle Modalità è riconducibile alla trattazione classica, e viceversa; qui di seguito se ne schematizzano i passaggi.

Nella trattazione algebrica, l'insieme delle variabili proposizionali è suddiviso in sottoinsiemi disgiunti, ognuno di questi corrisponde ad un Mondo w_i.

Per ogni variabile proposizionale p_i, è determinato un insieme di variabili proposizionali $p_{ij} \varepsilon I(p_i)$, ad essa associate, ognuna di queste appartiene a Mondi diversi.

Il Mondo w_t è accessibile dal Mondo w_r se esiste almeno una variabile proposizionale p_{it} nel Mondo w_t, che appartiene all' insieme $I(p_{ir})$.

Gli operatori modali **L** ed **M** hanno il loro equivalente in Π e Σ rispettivamente.

Le caratteristiche formali della Relazione $R(p_i,p_{ij})$, come visto nei paragrafi precedenti, determinano certe proprietà algebriche cui corrispondono degli Assiomi Modali.

Per quanto riguarda la corrispondenza fra paradigma Modale e trattazione algebrica, si ha che l'insieme di tutte le variabili proposizionali appartenenti a tutti i Mondi, corrisponde all'insieme delle variabili proposizionali su cui si fonda l'algebra. Gli insiemi $I(p_i)$ contengono tutte le immagini di p_i appartenenti ai Mondi accessibili dal Mondo cui appartiene p_i. Infine la Relazione di accessibilità fra Mondi corrisponde alla Relazione che associa ad ogni p_i gli elementi di $I(p_i)$.

[5] Il concetto dei Mondo Possibili è dovuto a Leibniz; la formalizzazione che segue deriva sostanzialmente dai lavori di Kripke.

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,**P**q) = 1 sse ∃(esiste) (t')(R(t',t) & (h(t',q)=1))

h(t,**F**q) = 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) $q \Rightarrow \mathbf{GP}q$

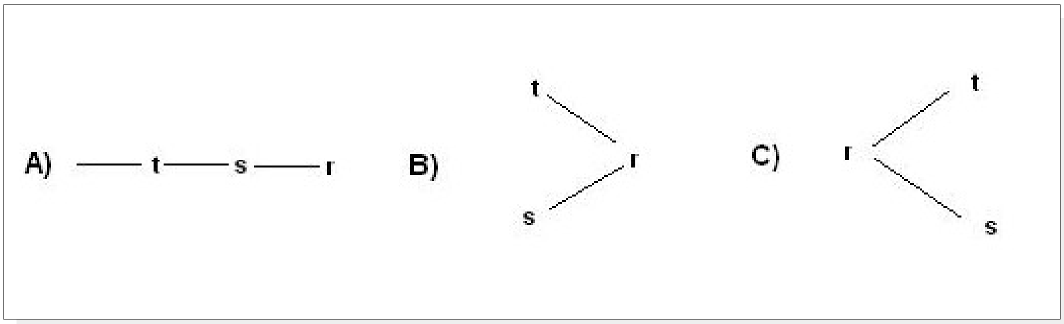
A6) $\mathbf{G}q$ se q è un assioma

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 $a_j \in I_F(a_i)$, allora $a_i \in I_P(a_j)$.

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[6].

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.

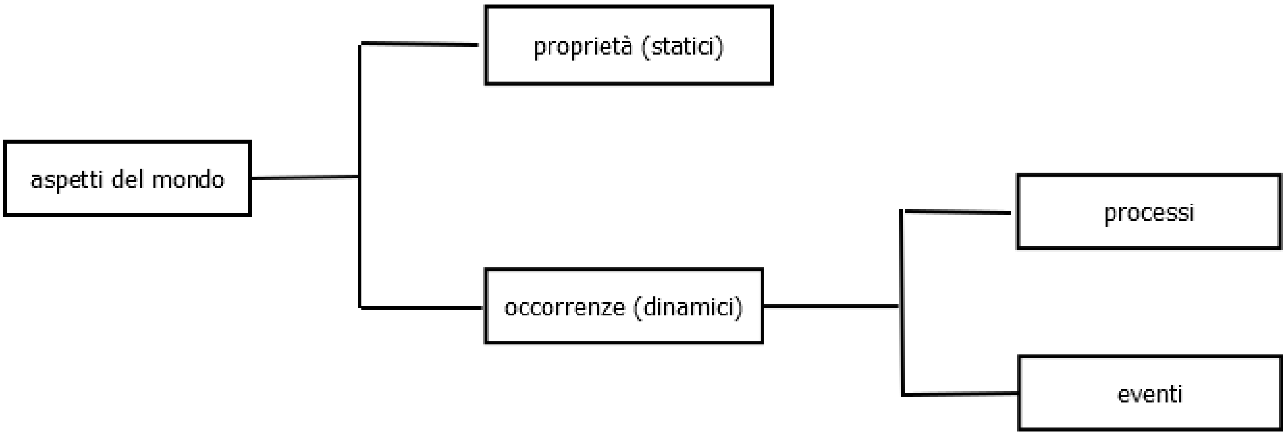
3.3.2 La logica temporale di Allen

La logica temporale adottata da Allen, è basata sulla nozione primitiva di intervalli. Allen adotta questo approccio, sulla base di due considerazioni, la prima è che il mondo ha aspetti statici, le proprietà, e dinamici, le occorrenze: questi aspetti identificano il tempo, in altre parole il tempo è percepibile solo attraverso proprietà e occorrenze ([A] pag. 125-126).

La seconda considerazione è che se si postula il tempo formato da punti istantanei, per evitare inconsistenze, si introducono artificiosità, dovendo considerare gli intervalli semiaperti (vedasi la discussione al par. 3.2 circa gli assiomi A13 e A14).

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

Ne segue che un punto temporale per Allen è un "intervallo molto piccolo" ([A] pag. 128), che non contiene, né si sovrappone ad altri intervalli. Le occorrenze sono suddivise in due sottoclassi: processi ed eventi, lo schema seguente sintetizza quanto espresso:



Queste distinzioni riflettono il modo di essere di fatti e di atti, rispetto agli intervalli temporali con cui sono in relazione: le proprietà "occupano" tutto un insieme di intervalli, cioè se un intervallo I è nell'insieme, allora anche gli intervalli J totalmente contenuti in I, lo sono. Gli eventi occorrono in intervalli tali che in un loro subintervallo l'evento non è più riconoscibile, ad esempio: accendere la luce. Infine i processi non "occupano" tutto l' insieme di intervalli in cui si svolgono, ad esempio: sto passeggiando.

Fra gli intervalli temporali ci sono delle relazioni binarie mutuamente esclusive:

```
X BEFORE Y                    XXX YYY
X EQUAL  Y                        XXX
                                  YYY
X MEET Y                      XXXYYY
X OVERLAPS Y                   XXX
                                YYY
X DURING Y                      XXX
                              YYYYYYY
X STARTS Y                     XXX
                               YYYYY
X FINISH Y                       XXX
                               YYYYY
```

Alcuni assiomi definiscono il comportamento di queste relazioni, fra i più significativi:

```
Overlaps(i1,i2) => ∃(i)During(i,i1) & During(i,i2)
Before(i1,i2) & Before(i2,i3) => Before (i1,i3)
Meets(i1,i2) => ¬∃(i)Before(i1,i) & Before(i,i2)
```

Allen introduce il predicato `Point(i)` per significare che `i` è un intervallo che, né si sovrappone, né contiene altri intervalli, ciò è espresso con gli assiomi:

```
Point(i) => ¬Overlap(i,i1) & ¬Overlap(i1,1)
Point(i1) & Point(i2) => ¬Meet(i1,i2)
```

Il predicato `HOLDS(p,i)` indica che la proprietà `p` è valida nell' intervallo `i`; poiché `p` è un enunciato, e come tale può non essere elementare, Allen introduce delle relazioni, ad esempio ([T] pag.87):

```
HOLDS(not(p),i)      => ¬HOLDS(p,i)
HOLDS(and(p,q),i)    ≡ HOLDS(p,i) & HOLDS(q,i)
HOLDS(or(p,q),i)     ≡ ∀(i1)(point(i1) & During(i1,i) =>
                       HOLDS(p,i1) v HOLDS(q,i1)
HOLDS(all(x,p),i)    ≡ ∀(x)HOLDS(p,i)
HOLDS(exist(x,p),i)  ≡ ∀(i1)(point(i1) & During(i1,i) =>
                        ∃(x)HOLDS(p,i1)
```

`HOLDS` è un predicato modale dal comportamento simile ad **L** per quanto riguarda il connettivo `And`. Allen non giustifica questa assiomatizzazione, ma in seguito ([A] pag. 129 e seg.) ne modifica la definizione, cercando di derivarla assiomaticamente introducendo un predicato `IN(t1,t2)`. `IN` è definito, a sua volta, a partire dalle relazioni primitive `DURING`, `START` e `FINISH`, ed asserisce la totale inclusione di un intervallo in un altro, per cui, ad esempio, si deriva che:

```
HOLDS(or(p,q),T) ≡
∀(t)IN(t,T) => ∃(s)IN(s,t) & (HOLDS(p,s) v HOLDS(q,s))
```

`OCCURR(e,t)` è un predicato che mette in relazione un evento **U** con un intervallo temporale **t**; per esso sussiste l'assioma ([A] pag. 133):

```
OCCURR(e,t) & IN(t',t) => ¬OCCURR(e,t')
```

Si possono definire funzioni per mettere in relazione fra di loro gli eventi, una di queste è: `CHANGE_POS(`*`oggetto`*`,`*`partenza`*`,`*`arrivo`*`)`.

`CHANGE_POS` genera quegli eventi che consistono nel muoversi dell' oggetto dalla posizione di partenza alla posizione di arrivo; la condizione necessaria per la classe di eventi involventi lo spostamento di un oggetto è:

```
OCCURR(CHANGE_POS(oggetto,partenza,arrivo),i) =>
        ∃(i1) ∃(i2)(MEETS(i1,i) & MEETS(i,i2) &
        HOLDS(at(oggetto,partenza),i1)&HOLDS(at(oggetto,arrivo),i2))
```

`at` è un predicato binario non introdotto né in [A], né in [T] ma il cui significato è intuitivo[7].

L' altra sottoclasse delle occorrenze, i processi è formalizzata dal predicato `OCCURRING(process,time)`, con l' assioma:

```
OCCURRING(p,i) => ∃(i1)IN(i1,i) & OCCURRING(p,i1)
```

vale inoltre:

```
OCCURR(e,i) => OCCURRING(e,i)
```

ma non il viceversa.

Allen definisce le azioni come occorrenze causate da Agenti, introducendo funzioni e predicati su di esse.

Anche in Allen vi sono parecchi assiomi, alcuni sono ridondanti, allo scopo di facilitare il calcolo; non è però chiaro quale modello si adatti al calcolo da lui sviluppato.

3.4 La formalizzazione del concetto di evento

Il punto cruciale nella logica temporale, in particolare nelle applicazioni di Intelligenza Artificiale, riguarda i mutui rapporti fra evento e tempo: Mc Dermott definisce gli eventi in termini di istanti; Allen assume come primitivi sia gli eventi che gli intervalli temporali; i primi occorrendo sopra gli intervalli di tempo. Entrambi gli autori, comunque, non chiariscono le relazioni fra i due concetti.

Si noti che i due approcci non sono esclusivi: su di un substrato di punti, gli intervalli si possono considerare insiemi di punti che soddisfano certe imposizioni, in genere di convessità, d'altro lato i punti risultano un passaggio al limite di periodi sempre più piccoli.

Per quanto riguarda le ricerche filosofiche, linguistiche o nel campo della IA, l'evento sembra predominare sull'istante, tuttavia non è ancora emerso un paradigma unico, non tanto per difficoltà intrinseche, quanto per le diverse possibili formalizzazioni della precedenza fra periodi e per la scelta delle relazioni fra di essi da assumersi come primitive.

L' evento è quindi una entità primitiva temporale, in alternativa ai "punti senza durata" che sono alla base delle formalizzazioni del paragrafo 3.2.

L' esposizione che segue è dovuta sostanzialmente a Wiener (1914), con contributi più recenti da Kamp (1978, 1980) e Van Benthem (1983); essa utilizza le due relazioni binarie di precedenza e sovrapposizione temporale fra eventi(**P** e **O**), ed adotta i seguenti assiomi:

E1) $e_1 \mathbf{P} e_2 \Rightarrow \neg(e_2 \mathbf{O} e_1)$

E2) $e_1 \mathbf{P} e_2 \;\&\; e_2 \mathbf{P} e_3 \Rightarrow e_1 \mathbf{P} e_3$

---

[7] Si è seguita la notazione presente in ([A] pag. 133); nell'esposizione fatta in [T], basata su un lavoro precedente (1981), `CHANGE_POS` è un predicato con quattro argomenti: oggetto, punto di partenza, punto di arrivo ed evento.

$E3)\ e_1 \mathbf{O} e_2 \Rightarrow e_2 \mathbf{O} e_1$
$E4)\ e_1 \mathbf{O} e_1$
$E5)\ e_1 \mathbf{P} e_2 \Rightarrow \neg(e_1 \mathbf{O} e_2)$
$E6)\ e_1 \mathbf{P} e_2\ \&\ e_2 \mathbf{O} e_3\ \&\ e_3 \mathbf{P} e_4 \Rightarrow e_1 \mathbf{P} e_4$
$E7)\ e_1 \mathbf{P} e_2 \lor e_1 \mathbf{O} e_2 \lor e_2 \mathbf{P} e_1$

L' ultimo assioma è discutibile, in quanto non è sempre possibile individuare quale delle alternative sia vera; ciò ha dato origine (Kamp 1980) ad una teoria più elaborata.

Intuitivamente gli istanti sono il massimo sottoinsieme di coppie di eventi che si sovrappongono: sia **E** una struttura di eventi ed **i** un istante, sottoinsieme di **E**:

$\forall(e_1, e_2\ \varepsilon\ i)\ (e_1\ \mathbf{O}\ e_2)$
$\forall(e_1\ \varepsilon\ E - i)\ \exists(e_2\ \varepsilon\ i)(\neg e_1\ \mathbf{O}\ e_2)$

La precedenza temporale di istanti nella struttura **E** è definita da:

$i_1 <_E i_2 =_{df} \exists\ (e_1\ \varepsilon\ i_1)\ \exists\ (e_2\ \varepsilon\ i_2)(e_1\ \mathbf{P}\ e_2)$

Si dimostra che $T_E = \langle I(E), <_E \rangle$ possiede un ordine lineare stretto.

Ulteriori imposizioni sulle relazioni temporali si riflettono su relazioni fra eventi, ad esempio la densità richiede la verità del seguente assioma (peraltro poco intuitivo):

$\forall(e_1, e_2)((e_1<e_2) \Rightarrow \exists(e_3)\exists(e_4)((e_3<e_2)\&(e_1<e_4)\&(e_3 \mathbf{O} e_4)$

La teoria degli eventi e degli istanti fornisce una visione unificata dei due concetti, chiarendone in modo soddisfacente le mutue relazioni.

3.5 Logica temporale e programmi

La logica temporale è stata applicata per specificare e verificare programmi (Manna e Pnueli 1981 in [T] pag. 91 e seg.); il modello del tempo adottato è lineare e discreto, con gli operatori temporali **UNTIL** e **NEXT** oltre ad **F** e **G**, e funzioni numeriche, fra le altre `Succ(t)` che fornisce l'istante successivo.

La semantica di **UNTIL** e **NEXT** è espressa da:

$$h(t,\text{NEXT } A) = 1 \text{ sse } h(Succ(t),A) = 1$$

$$h(t,A \text{ UNTIL } B) = 1 \text{ sse } (\exists\ (t' > t)(h(t',B) = 1 \text{ e } \forall(t'')(t \leq t'' \leq t')(h(t'',A) = 1))$$

Mediante questa formalizzazione si possono esprimere di invarianza dei programmi, quali la correttezza parziale o la mancanza di deadlock, proprietà riguardanti l'eventualità di certe computazioni, come la correttezza totale e proprietà di precedenza.

Un programma concorrente P è un insieme di sottoprogrammi P_1, P_2, ... P_m che procedono in parallelo; ognuno di essi è rappresentabile da un grafo di transizioni di stato, i nodi di questo sono istruzioni del programma (locazioni), cui sono associati condizioni o predicati di uscita $E(y_i)$ dipendenti dalle variabili locali **Y** di programma.

Le principali variabili interessate, sono:

- variabili globali $\mathbf{U} = \langle x_1, \ldots x_t \rangle$, o variabili di ingresso,

- variabili locali di programma, $\mathbf{Y} = \langle y_1, \ldots y_n \rangle$,

- variabili locali, che individuano, ad ogni istante, la locazione dei vari sottoprogrammi: $\mathbf{P} = \langle p_1, \ldots p_m \rangle$.

Una sequenza di esecuzione di un programma è una sequenza, eventualmente infinita, di stati s_0, s_1, ..., in cui ogni s_i è formato dal vettore delle locazioni e dal vettore delle variabili interne al tempo i.

Con $F(P,A)$ si indica l' insieme di tutti i calcoli del programma P al variare dei dati di ingresso, con $A(\mathbf{X})$ il predicato di ingresso, con $B(\mathbf{X},\mathbf{Y})$ il predicato di uscita e con **at** il predicato che asserisce dove il programma si trova.

Correttezza

Si definisce un programma parzialmente corretto se essendo vere le condizioni sulle variabili di ingresso, e terminando il programma, allora sono vere le condizioni di uscita.

Un programma è totalmente corretto se essendo vere le condizioni di ingresso, il programma termina e sono vere le condizioni di uscita.

Nella formalizzazione di Manna e Pnueli, ciò si esprime rispettivamente con:

$$F(P,A) \models A(X) \Rightarrow \mathbf{F}(\mathbf{at}l_e \Rightarrow B(X,\mathbf{Y}))$$
$$F(P,A) \models A(X) \Rightarrow \mathbf{F}(\mathbf{at}l_e \ \&\ B(X,\mathbf{Y}))$$

l_e è il vettore delle locazioni finali dei sottoprogrammi.

Mancanza di deadlock

Un programma è in deadlock se nessun suo processo è abilitato all'esecuzione; in altri termini se ogni sottoprogramma P_i è in una locazione in cui le condizioni di uscita sono false.

Quindi il deadlock è evitato se per tutti i processi in locazione di attesa, in almeno uno di essi le condizioni di uscita sono vere:

$$F(P,A) \models A(\mathbf{X}) \Rightarrow G((\mathbf{at}l_1 \ \&\ \ldots\ \mathbf{at}l_m) \Rightarrow (E_1(\mathbf{Y}) \lor \ldots, E_m(\mathbf{Y}))$$

Di recente (1983) Halpern, Manna e Moskowsky ([T] pag. 96), hanno sviluppato una logica temporale basata su intervalli, per la specifica e la verifica di componenti hardware; essi affermano, inoltre, che questa, opportunamente integrata, diventa un linguaggio di programmazione.

Il linguaggio impiega una estensione del calcolo dei predicati con uguaglianza, in esso sono introdotti l' operatore temporale `NEXT`, e l' operatore binario *;* di sequenzializzazione; il modello temporale è lineare e discreto, le sequenze finite di stati sono dette intervalli.

La semantica di `NEXT` e `di ;` è:

$$h(s_0, \ldots, s_n, \text{NEXT } A) = 1 \text{ sse } n / 1 \text{ e } h(s_1, \ldots, s_n,A) = 1$$

$h(s_0, \ldots, s_n, A ; B) = 1$

sse

$\exists(i)(0 \leq i \leq n)(h(s_0, \ldots, s_i, A) = 1 \text{ e } (h(s_i, \ldots, s_n, B) = 1$

3.6 Trattazione modale di Dopo e Prima

Alcuni semplici aspetti temporali possono essere descritti modalmente in termini di Necessità e Possibilità, in questo paragrafo ciò verrà fatto per gli operatori temporali diadici Dopo e Prima, e, ove non specificato il sistema modale adottato sarà **T**.

Si indicherà con **D**(p,q) l'enunciato "*l'evento p è vero dopo che è stato vero l'evento q*"; un tentativo di formalizzazione vero-funzionale di Dopo porta ad una relazione del tipo:

$D(a,b) \equiv a$

che non soddisfa al requisito intuitivo:

TD1) $\neg(D(a,b) \,\&\, D(b,a))$

pur soddisfacendo al requisito della transitività:

TD2) $D(a,b) \,\&\, D(b,c) \Rightarrow D(a,c)$

Una espressione in termini di modalità che soddisfa gli enunciati precedenti è:

$D(a,b) =_{df} \mathbf{L}b \,\&\, (\neg\mathbf{M}a \vee a \,\&\, \neg\mathbf{L}a)$ 3.6.1

Questa definizione afferma l'esistenza di almeno un mondo in cui b è avvenuto ed a non lo è ancora, escludendo qualsiasi forma di contemporaneità. Quest' ultimo concetto può essere descritto in forma stretta o in forma ampia rispettivamente da:

$C_{\&}(a,b) =_{df} \mathbf{L}(a\&b)$ ([8]

$C_{\equiv}(a,b) =_{df} \mathbf{L}(a\equiv b)$

La definizione 3.6.1 di Dopo, prescinde da ipotesi sulla continuità del tempo, ma si limita ad osservazioni di tipo discreto, di fatto un "adesso" ed un "in precedenza". Ciò è modellizzabile con soli due Mondi: siano M = {a, b} e M' = {α, b'}, e il mondo M' sia anteriore al mondo M: la tabella 3.6.1 riporta i valori di verità che soddisfano D(a,b), come espresso dalla definizione 3.6.1, relativamente a due soli mondi.

a	a'	b	b'
F	F	V	V
V	F	V	V

tab. 3.6.1

Con questa definizione l'operatore **and** assume, quando applicato a dei Dopo, la connotazione di operatore di concorrenza, ad esempio: **D**(a,b) & **D**(b,c) è falso (vedasi la dimostrazione di TD2), perché nessuna delle operazioni indicate, può essere svolta contemporaneamente, mentre in D(a,b) & D(a,c) b e c possono avvenire contemporaneamente.

La regola per **and** è quindi la seguente: l' operatore è sempre verofunzionale tranne quando i costituenti sono dei Dopo in cui coincidono il primo costituente dell'uno con il secondo costituente dell'altro.

3.6.1 Alcuni teoremi.

La tabella 3.6.2 evidenzia quali congiunzioni fra modalità e modalità negate non sono mai verificate, essa è utile nelle successive dimostrazioni.

	$\neg\alpha$	$\neg\mathbf{M}\alpha$	$\neg\mathbf{L}\alpha$
α	N	N	-
$\mathbf{M}\alpha$	-	N	-
$\mathbf{L}\alpha$	N	N	N

tab. 3.6.2

[8] questo asserisce la presenza continua di due eventi.

Dimostrazione di ¬**D**(a,b) & **D**(b,a) (TD1).

```
D(a,b) & D(b,a) =df Lb & (¬Ma v a&¬La) & La & (¬Mb v b&¬Lb) =
Lb & La & ¬Ma & ¬Mb            v   ;   La & ¬Ma
Lb & La & ¬Ma & b & ¬Lb        v   ;     "
Lb & La & a & ¬La & ¬Mb        v   ;   La & ¬La
Lb & La & a & ¬La & b & ¬Lb        ;     "
```

I singoli elementi della disgiunzione non sono verificati (a destra sono indicate le congiunzioni che non verificano), ne segue che la negazione è un teorema. ■

Dimostrazione di D(a,b) & D(b,c) => D(a,c) (TD2).

```
Lb & (¬Ma v a&¬La) & Lc & (¬Mb v  b&¬Lb)
          => Lc & (¬Ma v a&¬La)
Lb & Lc & ¬Ma & ¬Mb            v   ;   Lb & ¬Mb
Lb & Lc & ¬Ma & b & ¬Lb        v   ;   Lb & ¬Lb
Lb & Lc & a & ¬La & ¬Mb        v   ;   Lb & ¬Mb
Lb & Lc & a & ¬La & b & ¬Lb        ;   Lb & ¬Lb
          =>
((c & ¬Ma) v (Lc & a & ¬La))
```

Poiché il primo membro è sempre "Falso", l' implicazione è sempre "Vera". ■

Altri teoremi sono:

```
¬D(a,a)                            TD3
D(a,b) & D(a,c) ≡ D(a,b & c) )     TD4
D(a,b) & D(b,c) => D(a,D(b,c))     TD5
(a & D(a,b)) => b                  TD6
D(a,b) => (a => b)                 TD7
b & D(a,b) ≡ D(a,b)                TD8
```

Dimostrazione di ¬**D**(a,a) (TD3):

```
D(a,a) =df La & (¬Ma v a&¬La)
```

Sviluppando si ottengono due disgiunti che non sono mai verificati, quindi lo è sempre la loro negazione.

Dimostrazione di D(a,b) & D(a,c) ≡ D(a,b & c) (TD4):

I membro

```
D(a,b)&D(a,c) =df Lb & (¬Ma v a&¬La) & Lc & (¬Ma v a&¬La)
L(b & c) & (¬Ma v a&¬La)
```

II membro

```
L(b & c) & (¬Ma v a&¬La)
```

■

Dimostrazione di D(a,b) & D(b,c) => D(a,D(b,c)) (TD5):

Analoga a quella di TD2.

Dimostrazione (a & D(a,b) => b (TD6):

```
   ((a & Lb & (¬Ma v a&¬La)) => b
1) ((a & Lb & (¬Ma v a&¬La)) => Lb      PC15
2) Lb => b                              A5
3) ((a & Lb & (¬Ma v a&¬La)) => b      1,2 X Sill.
```

■

La dimostrazione di **D**(a,b) => (a => b) (TD7), si ottiene per esportazione da TD6.

Dimostrazione di (b & **D**(a,b)) ≡ **D**(a,b) (TD8):

```
((b & Lb & (¬Ma v a&¬La)) ≡ (Lb & (¬Ma v a&¬La)
```

È sufficiente dimostrare che b & **L**b ≡ **L**b:

```
1) Lb => b                       A5
2) b => (Lb => (Lb & b))         PC19 (aggiunzione)
3) Lb => (Lb => (Lb & b))        1,2 X sillog.
4) (Lb & Lb) => (Lb & b)         importazione
5) Lb => (Lb & b)
6) (Lb & b) => Lb                PC15
7) (Lb & b) ≡ Lb                 5,6
```

■

Si noti che il Teorema vale anche nella forma più generale:

α & ß & D(a,ß & τ) ≡ α & D(a, ß & τ)

Con α ß τ espressioni qualsiasi.

3.6.2 La modalità Prima

La modalità Prima è definibile dalla modalità Dopo:

D(a,b) $=_{df}$ **P**(b,a)

L' introduzione della modalità Prima, è dovuta al fatto che essa è più intuitiva nel passaggio da una lista di precedenze ad una formula operativa, cioè ad un programma, piano di lavoro, o quanto di per sé gli enunciati possono significare.

I Teoremi precedenti, sono ovviamente ancora validi, in particolare:

```
P(a,b) & P(b,c) => P(P(a,b),c)        TP5
¬P(a,a)                               TP3
P(a,c) & P(b,c) ≡ P(a & b,c)          TP4
a & P(a,b) ≡ P(a,b)                   TP8
α & ß & P(α & τ,b) ≡ ß & P(α & τ,b)   TP8'
```

Ad esempio trasformando in funzione di Dopo TP4, si ottiene:

D(c,a) & **D**(c,b) ≡ **D**(c, a & b)

Che altro non è che TD4.

3.6.3 Applicazione della modalità Prima

Sia data una lista di precedenze:

$P(a_1,b_1)$, $P(a_2,b_2)$ ...,

da esse è possibile ricavare un enunciato "eseguibile", tale che valga:

$P(a_1,b_1)$ & $P(a_2,b_2)$ & ... => $P(\alpha_1,ß_1)$ & $P(\alpha_2,ß_2)$ & ...

dove $ß_i$ è una azione elementare e α_i è o una azione elementare o una espressione P(_,_); ed inoltre le espressioni in And, di pari livello, possono essere svolte contemporaneamente.

La procedura si sviluppa in pochi passi, nel primo passo si applicano TP4 e TP8, ottenendo un enunciato equivalente.

Successivamente si applica TP5 ed eventualmente TP8'.

Siano a, b, ..., a_i azioni elementari, e α, ß, ... azioni elementari o composte; P(a,c) & P(b,c) ≡ P(a & b,c) si traduce nella seguente regola:

RP4) $$\frac{P(\alpha,\mathbf{a_i}) \quad P(ß,\mathbf{a_i})}{P(\alpha \;\&\; ß,\mathbf{a_i})}$$

Per TP8' α & ß & P(α & ç,b) ≡ ß & P(α & ç,b) la regola è:

RP8) $$\frac{\boldsymbol{\alpha} \;\&\; ß \;\&\; P(\boldsymbol{\alpha} \;\&\; \tau,b)}{ß \;\&\; P(\boldsymbol{\alpha} \;\&\; \tau,b)}$$

E da TP5, P(a,b) & P(b,c) => P(P(a,b),c) ha origine la regola:

RP5) $$\frac{P(\alpha,\mathbf{a_i}) \quad P(ß \;\&\; \mathbf{a_i} \;\&\; \tau,a_j)}{P(ß \;\&\; P(\alpha,a_i) \;\&\; \tau,a_j)}$$

L' algoritmo descritto trasforma un **And** di attività che può essere falso (9, in una formula che è falsa solo se

---

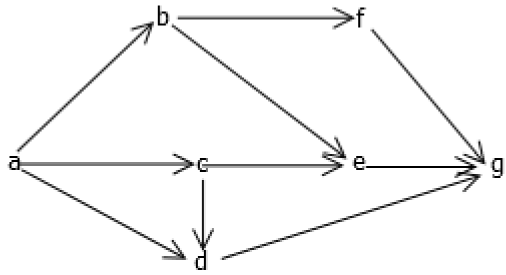
[9] nel senso espresso al paragrafo 1.3 Modalità epistemiche.

ci sono dei cicli, infatti le regole RP4 ed RP8, derivando da equivalenze, non variano il valore di verità (in particolare se le componenti di RP4 sono vere, l' and è vero).
In quanto a RP5:

```
P(a,b) & P(b,c) => P(P(a,b),c)
P(a,b) => (P(b,c) => P(P(a,b),c))      esportazione
```

Se sia `P(a,b)` che `P(b,c)` sono veri, applicando due volte il Modus Ponens, ne segue che è vero `P(P(a,b),c)`

Esempio:



La lista delle precedenze è:

```
P(a,b) P(a,c) P(a,d)
P(b,e) P(b,f)
P(c,e) P(c,d)
P(d,g)
P(e,g)
P(f,g)
```

L' applicazione delle regole alla lista delle precedenze (eseguita con un programma), genera la seguente formula:

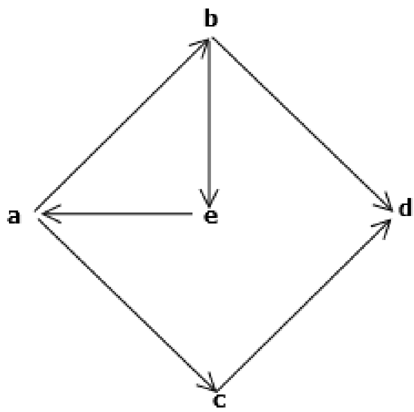
```
P(P(P(a,c),d) & P(P(a,b) & P(a,c),e) & P(P(a,b),f),g)
```

Concorrenza

Tutte le attività in And fra di loro, allo stesso livello possono essere svolte concorrentemente.

La presenza di cicli

La presenza di cicli è individuata dalla presenza di azioni del tipo $P(\alpha \ \& \ \mathbf{a_i} \ \& \ \beta, \mathbf{a_i})$, che, in forza di TP3, rendono la formula mai verificata, esempio:



P(a,b) **P**(a,c)
P(b,d) **P**(b,e)
P(c,d)
P(e,a)

L' applicazione del procedimento descritto porta a:

P(**P**(**a,**b) & **P**(**a,**c)**,** d) & **P**(**P**(**P**(a,b)**,**e)**,a**)

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** (10 (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

4.3 Conoscenza delle Macchine di Turing

$K = \{q_0, q_1, ... \}$ un insieme di stati, non contenente lo stato hlt,
$\mathbf{\Sigma}$ = un alfabeto contenente il simbolo spazio denotato con # ma non i simboli L ed R,
$s \in K$ = uno stato iniziale,
Φ = una funzione da $K \times \mathbf{\Sigma}$ a $(K\ \mathbf{U}\ \text{hlt}) \times (\mathbf{\Sigma}\ \mathbf{U}\ \{L, R\})$

inoltre si indicherà con δ_i l' i-esima posizione del nastro della MdT.

Sostanzialmente la macchina, in un certo stato q_i con in lettura il simbolo $\mathbf{\Sigma}_i$, passa nello stato q_j e, o sposta la testina di lettura/scrittura di una posizione (a destra a sinistra), o rimane ferma scrivendo il simbolo $\mathbf{\Sigma}_j$.

In ogni istante la MdT "conosce" il contenuto della casella attuale.

L' insieme è è correlato all' **insieme degli assiomi** che governano la conoscenza della MdT, se la MdT è nello stato q_i, sarà vero il predicato q_i [11] asserente che la MdT è nello stato q_i; gli assiomi saranno:

$(q_i, \mathbf{\Sigma}_i, q_j, \mathbf{\Sigma}_j) \dashrightarrow q_i\ \&\ \mathbf{K\Sigma}_i \rightarrow q_j\ \&\ \mathbf{K\Sigma}_j$ 4.2.1

$\mathbf{\Sigma}_j$ può essere sia un simbolo dell' alfabeto $\mathbf{\Sigma}$, che il comando di spostare la testina di lettura-scrittura.

Ad esempio sia:

$K = \{ q_0, q_1\}$
$\mathbf{\Sigma} = \{a, \#\}$
$s = \{ q_0\}$

e Φ, con i relativi assiomi, sia:

```
(q0, a, q1, #)    --> q0 & Ka => q1 & K#
(q0, #, hlt, #)   --> q0 & K# => hlt & K#
(q1, a, q0, a)    --> q1 & Ka => q0 & Ka
(q1, #, q0, R)    --> q1 & K# => q0 & K(R)
```

La MdT in oggetto "cancella" le **a** presenti sul nastro.

Se il nastro contiene **aa#**, la sequenza di operazioni è:

```
q0 & Ka -> q1 & K#        cancellazione della prima a
q1 & K# -> q0 & K(R)      spostamento
           q0 & Ka         a destra
```

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

[11] l' uso dello stesso simbolo è una semplificazione che non dovrebbe generare confusione.

```
q0 & Ka -> q1 & K#        cancellazione della seconda a
q1 & K# -> q0 & K(R)      spostamento
            q0 & Ka        a destra
q0 & K# -> hlt & K#        fine
```

In ogni istante è vero un solo stato e la MdT conosce un simbolo, e l' istante successivo è vero un nuovo stato ed è conosciuto un altro simbolo.

L' operatore `->` non può essere l' implicazione (materiale), in quanto la formula sarebbe falsa prima di eseguire l' istruzione e vera dopo; inoltre i due lati di

essa appartengono ad istanti diversi, occorre quindi includere questo fatto nel formalismo (ad esempio tramite modalità temporali).

Un altro aspetto poco soddisfacente di questa interpretazione è che, ad un dato istante, la MdT conosce solamente il simbolo attuale e non conosce nè i dati di input, né il risultato, né il programma stesso. Ci si può limitare a considerare la conoscenza dei dati di input ed output, in quanto, tramite una MdT universale, il programma fa parte dei dati di input.

(in fase di costruzione)

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

├	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