Logica modale e logica MSO

0.1. La logica modale LM.

Definizione 1. *Indichiamo con LM l'insieme delle formule della logica modale sull'insieme di atomi At (un insieme finito fissato), definito induttivamente come segue:*

- *Se* $P \in At$ *si ha* $P \in LM$*;*
- *Se* $\phi \in LM$ *allora* $\neg\phi \in LM$*;*
- *Se* $\phi, \psi \in LM$ *allora* $\phi \wedge \psi \in LM$*;*
- *Se* $\phi \in LM$ *allora* $\Box\phi \in LM$*;*
- *Se* $\phi \in LM$ *allora* $\Diamond\phi \in LM$*.*

Per definire la sematica del linguaggio modale utilizziamo i *modelli di Kripke*, definiti formalmente come segue:

Definizione 2. *Dato un linguaggio* $\mathcal{L} = \{=, R\} \cup \{P : P \in At\}$*, dove* R *è relazionale binario e ogni* $P \in At$ *è identificato con un simbolo relazionale unario, un* modello di Kripke *è un' interpretazione* $M = (D^M, R^M, (P)^M_{P \in At})$ *di* $\mathcal{L}$*. Ad un modello di Kripke associamo in modo naturale una funzione* $L : D^M \to \mathcal{P}(At)$ *data da*

$$L(m) = \{P \in At | m \in P^M\}.$$

Un modello puntato *è una coppia* (M, m) *dove* $m \in D^M$*. Abusando un po' della notazione, spesso indicheremo un modello di Kripke con una tupla* $M = (W, R, L)$ *dove* W *è il dominio e usiamo il simbolo* R *al posto di* R^M*.*

Il dominio $D^M = W$ rappresenta l'insieme dei possibili "stati" o"mondi". Utilizziamo i modelli puntati per evidenziare lo stato corrente, rappresentato appunto da m.
I modelli di Kripke vengono spesso raffigrati tramite dei grafi orientati etichettati in cui D^M è l'insieme dei vertici e R^M è l'insieme degli archi.

Tramite i modelli puntati, possiamo definire la nozione di verità delle formule di LM.

Definizione 3. *Se* $M = (W, R, L)$*, definiamo* $(M, m) \models \phi$ *per induzione sulla complessità di* ϕ*:*

- $(M, m) \models P \Leftrightarrow P \in L(m)$*;*
- $(M, m) \models \neg\phi \Leftrightarrow (M, m) \not\models \phi$*;*
- $(M, m) \models \phi \wedge \psi \Leftrightarrow (M, m) \models \phi$ *e* $(M, m) \models \psi$*;*
- $(M, m) \models \Box\phi \Leftrightarrow$ *per ogni* m' *tale che* $(m, m') \in R^M$ *si ha* $(M, m') \models \phi$*.*

L'operatore $\Box$ è l'operatore di *necessitazione*, e indica appunto che in qualunque stato si evolva lo stato corrente, varrà comunque la proprietà ϕ. L'operatore derivato $\Diamond\phi \equiv \neg\Box\neg\phi$ è l'operatore di *possibilità*. Si osservi che anche $\vee$ è un operatore derivato ($\phi \vee \psi \equiv \neg(\neg\phi \wedge \neg\psi)$). Tramite queste equivalenze si può osservare che ogni formula è esprimibile in *negation normal form*, ovvero facendo uso anche di $\vee$ e $\Diamond$ in modo che gli operatori di negazione siano anteposti solo direttamente alle formule dell'insieme $\{P | P \in At\}$.

Ad esempio, se ϕ è $\neg\Box(P \wedge \Box Q)$, è equivalente a $\Diamond\neg(P \wedge \Box Q) \equiv \Diamond(\neg P \vee \neg\Box Q) \equiv \Diamond(\neg P \vee \Diamond\neg Q)$, e quest'ultima è la sua negation normal form.

È possibile definire una "traduzione" naturale di LM in $FO(x)$, ovvero una funzione $ST : LM \to FO(x)$ tale che $(M, m) \models \phi \Leftrightarrow M \models ST(\phi)(m)$, nel modo seguente:

Definizione 4.
- $ST(P) = P(x)$ *per ogni* $P \in At$*;*
- $ST(\neg\phi) = \neg ST(\phi)$*;*
- $ST(\phi \wedge \psi) = ST(\phi) \wedge ST(\psi)$*;*
- $ST(\Box\phi) = \forall y(R(x, y) \to ST(\phi)\{x/y\})$*, dove* y *è libera per la sostituzione a* x *in* $ST(\phi)$*.*

D'ora in poi diremo che una formula F di $FO(x)$ è *logicamente equivalente* a una di LM se esiste $\phi \in LM$ tale che $ST(\phi)$ è equivalente a F nella logica al prim'ordine.
In informatica i modelli di Kripke sono utilizzati per rappresentare i sistemi informatici (ad esempio i programmi) e la loro evoluzione nel tempo. Per stabilire quando due modellizzazioni rappresentano lo stesso sistema, si usa la nozione di bisimulazione:

Definizione 5. *Una relazione* $B \subseteq D^M \times D^N$ *è una* bismulazione *se per ogni* $(m, n) \in B$ *vale:*
- *Se* $P \in At$ *vale* $P \in L(m) \Leftrightarrow P \in L(n)$*;*
- *Se* $mR^M m'$ *allora esiste* $n' \in D^N$ *tale che* $nR^N n'$ *e* $(m', n') \in B$ *(condizione "forth");*
- *Se* $nR^N n'$ *allora esiste* $m' \in D^M$ *tale che* $mR^M m'$ *e* $(m', n') \in B$ *(condizione "back").*

Inoltre, due modelli puntati (M, w) *e* (N, v) *sono* bisimili *se esiste una bisimulazione* B *tra* M *e* N *tale che* $(w, v) \in B$*.*

È facile verificare che l'essere bisimili è una relazione di equivalenza. È possibile dimostrare il seguente teorema:

Teorema 6 (Van Benthem)**.** *Ogni formula di* $FO(x)$ *è invariante per bisimulazione se e solo se è equivalente a una formula di* LM*.*

Dimostriamo per il momento solo la necessità, ovvero che se (M, m) e (N, n) sono bisimili allora verificano le stesse formule di LM.

Dimostrazione. Dimostriamo l'asserto per induzione sulla complessità di ϕ.
- Se $\phi = P$ per qualche a appartenente a At, allora la tesi vale per la prima condizione che definisce la bisimulazione;
- Se $(M, m) \models \neg\phi$, allora $(M, m) \not\models \phi$. Segue $(N, n) \not\models \phi$ per ipotesi induttiva, dunque $(N, n) \models \neg\phi$ (il ruolo di M e N è simmetrico, essendo la bisimulazione una relazione simmetrica);
- Se $(M, m) \models \phi \wedge \psi$, allora $(M, m) \models \phi$ e $(M, m) \models \psi$. Segue $(N, n) \models \phi$ e $(N, n) \models \psi$ per ipotesi induttiva, dunque $(N, n) \models \phi \wedge \psi$;
- Se $(M, m) \models \Box\phi$, supponiamo per assurdo che $(N, n) \not\models \Box\phi$, e sia n' tale che $nR^N n'$ e $(N, n') \not\models \phi$. Dalla condizione "back" otteniamo che esiste m' tale che $mR^M m'$ e $(m', n') \in B$. Allora, per ipotesi induttiva si ha $(M, m') \not\models \phi$, contro l'ipotesi. Dunque $(N, n) \models \Box\phi$.

□

0.2. **Esercizi.**

(1) Determinare se esiste una bisimulazione tra le seguenti coppie di modelli di Kripke:
- $D^M = D^N = \{0,1,2\}$, $R^M = \{(0,1),(0,2),(2,2)\}$, $R^N = \{(0,1),(1,2)\}$, $p^M = p^N = \{0\}$, $q^M = q^N = \{1,2\}$, $w = v = 0$.
- M, w come sopra, $D^N = \mathbb{N} \cup \{*\}$, $R^N = \{(n, n+1) : n \in \mathbb{N}\} \cup \{(0,*)\}$, $p^N = \{0\}$, $q^N = \mathbb{N} \setminus \{0\} \cup \{*\}$ $v = 0$.
- $D^M = \{0,1\}$, $R^M = \{(0,1),(1,0\}$, $p^M = \{0\}$, $q^M = \{1\}$, $w = 0$, $D^N = \{0\}$, $R^N = \{(0,0)\}$, $p^N = \{0\}$, $q^N = \{0\}$, $v = 0$.
- Come nell'esercizio precedente ma ora tutti gli stati verificano p e non q
- $D^M = \mathbb{N}$, $R^M = \{(0,1),(0,2),(2,3),(0,4),(4,5),(5,6),(0,7),(7,8),(8,9),(9,10),(0,11),.$ $w = 0$, M come N ma riflessivo.

(2) Dati due modelli di Kripke M ed N e un numero naturale n, una n-bisimulazione è data dall'unione $B_0 \cup \ldots \cup B_n$ di relazioni $B_i \subseteq M \times N$ tale che per ogni coppia $(w,v) \in B_i$:
- per ogni lettera proposizionale P si ha $w \in P^M \Leftrightarrow v \in P^N$;
- se $i > 0$ e $wR^M w'$ esiste v' con $vR^N v'$ e $(w', v') \in B_{i-1}$;
- se $i > 0$ e $vR^M v'$ esiste w' con $wR^N w'$ e $(w', v') \in B_{i-1}$;

Dimostrare che una bisimulazione è una n-bisimulazione, per ogni n. Trovare una n-bisimulazione che non sia anche una bisimulazione.

(3) Data una formula modale, il suo grado modale è definito induttivamente da:
$gr(P) = 0$, $gr(\neg F) = gr(F)$, $gr(F \wedge G) = gr(F \vee G) = gr(F \to G) = \max\{gr(F), gr(G)\}$, $gr(\Box F) = gr(\Diamond F) = gr(F) + 1$.
Dimostrare che una formula modale F di grado n è invariante per n-bisimulazione (vedi esercizio precedente per la definizione di n-bisimulazione; l'invarianza significa che, se esiste una n-bisimulazione $B_0 \cup \ldots B_n$ fra M e N e $(w,v) \in B$, si ha $(M,w) \models F \Leftrightarrow (N,v) \models F$).
Viceversa, dimostrare che, per ogni n, la relazione
$B_0 \cup \ldots \cup B_n$ con $B_i = \{(w,v) : (M,w)$ e (N,v) verificano le stesse formule di grado $i\}$ è una n-bisimulazione (suggerimento: a meno di equivalenza logica, esistono solo un numero finito di formule di grado i in un dato linguaggio).

(4) Dati due modelli di Kripke (M,w), (N,v) dimostrare che esiste una n-bisimulazione $B = B_0 \cup \ldots \cup B_n$ con $(w,v) \in B_n$ se e solo se (M,w) e (N,v) verificano le stesse formule di grando n.

(5) Dimostrare o confutare la seguente affermazione: se (M,w) e (N,v) sono n-bisimili per ogni n, allora sono bisimili.

(6) Dimostrare che, restringendosi alla classe dei modelli di Kripke finiti, se (M,w) e (N,v) sono n-bisimili per ogni n allora sono bisimili.

0.3. **Operatori monotoni.** Vediamo adesso un modo di rappresentare le formule modali come *operatori*. Diamo innanzitutto alcuni concetti preliminari.

Definizione 7. *Sia U un insieme (finito). Un* operatore monotono *su U è una funzione $F : \mathcal{P}(U) \to \mathcal{P}(U)$ tale che per ogni X, Y tali che*

$$X \subseteq Y \subseteq U \to F(X) \subseteq F(Y).$$

Un punto fisso *di F è un sottoinsieme X di U tale che $F(X) = X$. Un* minimo punto fisso *è un punto fisso che è contenuto in ogni altro punto fisso (dualmente è definito il massimo).*

Teorema 8 (Tarski-Knaster)**.** *Ogni operatore monotono F su un insieme finito U ha minimo e massimo punto fisso.*

Dimostrazione. Si consideri la successione definta da

- $F^0 = \emptyset$;
- $F^{n+1} = F(F^n)$ per ogni $n > 0$.

Si ha, per la monotonia di F, che $F^n \subseteq F^{n+1}$ per ogni $n \in \mathbb{N}$. Essendo U finito, la successione non può crescere strettamente, dunque esiste un indice i tale che $F^i = F^{i+1}$ (dunque F_i è un punto fisso) e che sia minimo con tale proprietà. Si ottiene $F^i = F^{i+1} = F^{i+2} = \ldots$ (e dunque $F^i = \bigcup_{j \in \mathbb{N}} F^j$). Osserviamo che F^i è il minimo punto fisso, infatti se $B = F(B)$ essendo $\emptyset \subseteq B$ segue $F(\emptyset) \subseteq F(B) \subseteq B$, e per induzione $F^n \subseteq B$ per ogni n, in particolare per $n = i$. □

Le dimostrazioni analoghe per il massimo punto fisso si ottengono facilmente con opportune modifiche (cambiando i versi delle inclusioni, sostituendo le intersezioni di famiglie di insiemi con le unioni, e $\emptyset$ con U).

Diamo adesso un'altra caratterizzazione del minimo punto fisso.

Teorema 9. *Il minimo punto fisso LFP_F dell'operatore monotono F è dato da*

$$LFP_F = \bigcap \{A \subseteq U | F(A) \subseteq A\}.$$

Dimostrazione. Sia $S = \bigcap \{A \subseteq U | F(A) \subseteq A\}$. Vogliamo mostrare che S è il minimo punto fisso LFP_F di F. Poiché LFP_F è un punto fisso di F, si ha $F(LFP_F) = LFP_F$ e quindi LFP_F è uno degli insiemi che con la loro intersezione formano l'insieme S. Ne segue che ha $S \subseteq LFP_F$. Per dimostrare l'incusione opposta, basta utilizzare il Teorema precedente e dimostrare per induzione su n che $F^n \subseteq S$ per ogni n. Per questo, è sufficiente far vedere che $F^n \subseteq A$, per ogni A tale che $F(A) \subseteq A$. La base dell'induzione, $\emptyset \subseteq A$ è banalmente verificata; se $F^n \subseteq A$ allora $F^{n+1} \subseteq F(A) \subseteq A$. □

Osserviamo che da questa dimostrazione segue che

$$\bigcap \{A | F(A) = A\} = \bigcap \{F(A) \subseteq A\}.$$

0.4. **Formule Modali e Operatori Monotoni.** Vediamo adesso un modo di estendere la logica modale che fa uso dei concetti appena visti sugli operatori monotoni. Introduciamo nel linguaggio $\mathcal{L}$ di partenza le variabili proposizionali libere (che indicheremo usualmente con $X, Y, \ldots$). Ogni modello $M = (W, R, L)$ di Kripke in un linguaggio con atomi At può essere esteso ad un modello per il linguaggio con atomi $At' = At \cup \{X\}$ scegliendo un sottoinsieme A del dominio W come valore per X: denotando questo nuovo modello con $M, [X/A] = (W, R, L')$ si ha $L'(P) = L(P)$, per ogni $P \in At$ e $L'(X) = A$. Ogni formula modale ϕ nel linguaggio At' può allora essere interpretata sul modello M come l' operatore

$$||\phi||^M : Pow(W) \to Pow(W)$$

che manda il sottoinsieme A nell'insieme dei punti di $M, [X/A]$ che rendono vera ϕ:

$$||\phi||^M(A) = \{w \in W : M, [X/A], w \models \phi\}$$

(nel seguito, a volte ometteremo l'apice M in $||\phi||^M$, scrivendo semplicemente $||\phi||$).

In particolare, se ϕ non ha variabili proposizionali libere, $||\phi||$ è un operatore costante. Si può dimostrare che se nella negation normal form di una data formula ϕ le variabili libere non sono negate (ovvero, non vi sono espressioni del tipo $\neg X$) allora l'operatore $||\phi||^M$ è monotono.

In modo analogo per interpretare una formula modale $\phi(X_1, \ldots, X_n)$ con n variabili libere, avremo bisogno di n sottoinsiemi del dominio. Una tale formula darà quindi luogo ad un operatore F_ϕ dall'insieme delle n-ple di sottoinsiemi di M ai sottoinsiemi di M definito da

$$||\phi||^M(A_1, \ldots, A_n) = \{w \in M : M, [X_1/A_1, \ldots, X_n/A_n, w \models \phi\}.$$

1. Formule di CTL come punti fissi di operatori

Data la formula $\phi = P \vee EX(Y)$ di CTL ed un modello di Kripke $M = (W, R, L)$ del linguaggio $L = \{P\}$, consideriamo l'operatore corrispondente, $||\phi||_M : Pow(W) \to Pow(W)$, definito da

$$||\phi||(A) = \{v \in W : M, [Y|A], v \models \phi\}.$$

L'operatore $||\phi||$ è monotono, quindi ha un massimo punto fisso $LFP^M_\phi \subseteq W$ e un minimo punto $LFP^M_\phi \subseteq W$.

Dalla teoria sui massimi e minimi punti fissi sappiamo che, se $n = |W|$, allora

$$LFP^M_\phi = ||\phi||^n(\emptyset), \quad GFP^M_\phi = ||\phi||^n(W).$$

In particolare,

$$LFP^M_\phi = ||P|| \vee ||EX(P)|| \vee \ldots ||EX(EX \ldots (EX(P)) \ldots)||,$$

da cui si riconosce che $LFP^M_\phi = ||EF(P)||$.

Anche gli altri operatori di CTL hanno un'interpretazione come massimi o minimi punti fissi di operatori. Per ottenere la semantica di $EG(P)$, ad esempio, basta considerare il massimo punto fisso dell'operatore $\phi := P \wedge EX(Y)$: si ha $LFP^M_\phi = ||\phi||^n(W)$ e quindi

$$LFP_\phi^M = P \wedge ||EX(P)|| \wedge \ldots ||EX(EX \ldots EX(P) \ldots)||,$$

da cui si riconosce facilmente che $||EG(P)|| = LFP_\phi^M$.

Per quanto riguarda gli altri operatori di CTL, guardare gli esercizi.

2. Esercizi

(1) Qual è il massimo punto fisso dell'operatore $||\phi||_M$ se $\phi = P \vee EX(Y)$?
(2) Dimostrare che l'operatore $E(PUQ)$ di CTL si può ottenere come minimo punto fisso dell'operatore $\phi := Q \vee (P \wedge EX(Y))$.
(3) Dimostrare che l'operatore $AF(P)$ di CTL si può ottenere come minimo punto fisso dell'operatore $\phi := P \vee (\Diamond(\top) \wedge AX(Y))$.
(4) Dimostrare che l'operatore $AG(P)$ di CTL si può ottenere come massimo punto fisso dell'operatore $\phi := P \wedge AX(Y)$.
(5) Sia $F : Pow(S) \times Pow(S) \to Pow(S)$ una funzione tale che, se $A \subseteq B \subseteq S$ e $C \subseteq S$ allora

$$F(A, C) \subseteq F(B, C)$$

. Per ogni fissato $C \subseteq S$, sia $LFP_F(-, C)$ il minimo punto fisso (omassimo punto fisso) dell'operatore monotono $F(-C) : Pow(S) \to Pow(S)$. Dimostrare che l'operatore $G : Pow(S) \to Pow(S)$ definito da $G(C) = LFP_F(-, C)$ è monotono. Notare come questo esercizio permetta di giustificare l'iterazione di punti fissi.

3. μ-Calcolo

Definiamo adesso una logica che fa uso dei concetti di minimo e massimo punto fisso, il μ-calcolo.

Definizione 10. *Sia $PROP$ un insieme di variabili proposizionali. Indichiamo con L_μ l'insieme delle formule del μ-calcolo, definito come segue:*

- *Se P è allora $P, \neg P \in L_\mu$;*
- *Se X è una variabile proposizionale allora $X \in L_\mu$;*
- *Se $\phi \in L_\mu$ allora $\mu X.\phi$ e $\nu X.\phi \in L_\mu$;*
- *Se $\phi, \psi \in L_\mu$ allora $\phi \wedge \psi, \phi \vee \psi, \Box\phi, \Diamond\phi \in L_\mu$.*

La semantica del μ-calcolo si ottiene estendendo la semantica della logica modale sui modelli di Kripke. Gli operatori μ e ν sono gli operatori dei minimo e massimo punto fisso, interpretati nel seguente modo:

$$M, [X_1/A_1, \ldots, X_k/A_k], w \models \mu X.\phi(X, X_1, \ldots, X_k) \Leftrightarrow w \in LFP(F_\phi^M(X, A_1, \ldots, A_n)),$$

$$M, [X_1/A_1, \ldots, X_k/A_k], w \models \nu X.\phi(X, X_1, \ldots, X_k) \Leftrightarrow w \in GFP(F_\phi^M(X, A_1, \ldots, A_n))$$

dove LFP e GFP sono rispettivamente il minimo e il massimo punto fisso dell'operatore.

Nel seguito spesso identificheremo la sintassi e la semantica delle approssimazioni dei punti fissi:

Se ϕ è una formula, indicheremo con $\phi(\bot)$ sia la formula che l'insieme $||\phi||^M(\emptyset)$ e cosí per le altre approssimazioni.

Esempio. Sia $\phi = \mu X.\Diamond(X \vee p)$. Allora si ha
$\phi^0 := \phi(\bot) = \Diamond(\bot \vee p) = \Diamond p$
$\phi^1 := \phi(\phi(\bot)) = \phi(\Diamond p) = \Diamond\Diamond p \vee \Diamond p$
$\vdots$
$\phi^{n+1} = \Diamond^n p \vee \Diamond^{n-1} p \vee \cdots \vee \Diamond p$.
Allora $(M, w) \models \mu X.\Diamond(X \vee p) \Leftrightarrow$ esiste un cammino di lunghezza maggiore o uguale a 1 che porta a uno stato in cui vale P.

3.1. **Esercizi.**

(1) Dimostrare che in un dato modello di Kripke M, l'insieme

$A = \{v \in M :$ da v è possibile raggiungere in un numero finito di passi uno stato in cui vale $P\}$

è il minimo punto fisso dell'operatore $\Diamond(X) \vee P$.

(2) Sia $\top = \neg\bot$; verificare che, dato un modello di Kripke M finito, l'insieme

$$A = \{w \in M : \forall n((M, w) \models \Diamond^n(\top))\}$$

è il massimo punto fisso dell'operatore $\Diamond(X)$.

(3) Dimostra che la formula

$$F = \nu Y.\mu X.(\Diamond(X) \vee ((P) \wedge \Diamond(Y)))$$

è vera in uno stato v di un modello M se e solo se da v parte un cammino che incontra infinite volte P.

3.2. **Giochi modali.** Anche nella logica modale possiamo definire un gioco tra due giocatori, che chiameremo Falsifier e Verifier. Lo scopo sarà quello di rispondere alla domanda "$(M, w) \models \phi$?" per una data ϕ modale e un dato modello di Kripke M.
Definiamo il gioco in tal modo: le posizioni del gioco saranno coppie del tipo (ψ, v) con ψ sottoformula di ϕ (nella sua scrittura in negation normal form) e $v \in M$. In particolare, la posizione iniziale sarà proprio (ϕ, w). Le mosse effettuabili (da entrambi i giocatori) da una data posizione (ψ, v) sono le seguenti:

- Se $\psi = \Box\psi'$ oppure $\Diamond\psi'$ allora si possono effettuare mosse del tipo (ψ', v') con $vR^M v'$;
- Se $\psi = \psi_1 \wedge \psi_2$ oppure $\psi_1 \vee \psi_2$ allora si può effettuare come mossa (ψ_1, v) o (ψ_2, v);
- Se $\psi = Q$ è un letterale il gioco è finito, e Verifier vince se e solo se $v \in Q^M$.

A differenza di quanto visto riguardo ai giochi di Ehrenfeucht, Falsifier e Verifier non giocheranno facendo una mossa ciascuno, bens il giocatore a cui tocca muovere viene determinato dalle seguenti regole:

- Se ψ è del tipo $\Box\psi'$ oppure è una congiunzione, muove Falsifier;
- Se ψ è del tipo $\Diamond\psi'$ oppure è una disgiunzione, muove Verifier.

Se uno dei due giocatori, al suo turno, non può effettuare alcuna mossa, allora ha perso (ciò può capitare solo in presenza di una formula del tipo $\Box\psi'$ o $\Diamond\psi'$). È facile osservare (facendo induzione sulla complessità della formula ϕ) che con queste regole Verifier ha una strategia vincente se e solo se $(M, w) \models \phi$.

Il gioco può essere esteso a formule del μ-calcolo prive di variabili libere. In questo caso il gioco fra Verifier e Falsifier corrisponde ad un gioco noto anche in ambito economico come *gioco di parità*. Iniziamo col capire come funziona il gioco nel caso in cui ogni variabile è quantificata una volta sola. Dunque sono possibili anche posizioni del tipo (X, v), $(\mu X.\psi(X), v)$ e $(\nu X.\psi(X), v)$. Da queste tre posizioni si passa in automatico (nel senso che la mossa non viene effettuata da uno dei due giocatori) alla posizione $(\psi(X), v)$ (nel caso di (X, v), la ψ è tale che $\mu X.\psi(X)$ oppure $\nu X.\psi(X)$ è una sottoformula della formula originaria ϕ; per le condizioni poste all'inizio, tale ψ esiste ed è unica). È evidente che in questo modo le partite possono diventare infinite, come ad esempio nel caso di una formula $\phi = \mu X \Box(X)$ in cui una tipica partita attraverserà le posizioni seguenti:

$$(\mu X \Box(X), v_1), (\Box(X), v), (X, v_2), (\Box(X), v_2), \ldots, (X, v_n), (\Box(X), v_n), \ldots$$

dove $v_n R v_{n+1}$ per ogni n. Ci serve dunque una condizione per la vittoria nel caso infinito. Questa condizione dipende da come è fatta la formula ϕ. Se ad esempio $\phi := \mu X.\psi(X)$ con ψ modale, stabiliamo che Falsifier vince tutte le partite infinite. Si ha:

Teorema 11. *Sia (M, w) un modello finito e $\psi(X)$ una formula modale con una variabile libera X. Le seguenti condizioni sono equivalenti:*

(1) $(M, w) \models \mu X \psi(X)$;
(2) Verifier ha una strategia vincente nel μ-gioco di $\mu X \psi(X)$ su (M, w).

Dimostrazione. Se $M, w \models \mu X.\psi(X)$, allora (per la caratterizzazione del minimo punto fisso degli operatori vista nel Teorema 8) esiste $n \geq 1$ tale che $M, w \models \psi^n(\bot)$. Quindi, per quanto già sappiamo sui giochi modali, Verifier ha una strategia vincente nel gioco modale della formula $\psi^n(\bot) = \psi[X/\psi^{n1}(\bot)]$ su (M, w). Consideriamo allora la seguente strategia di Verifier nel μ-gioco di $\mu X.\psi$ su (M, w): dopo la prima mossa "automatica" del gioco μ: $(\mu X.\psi(X), w) \to (\psi(X), w)$ Verifier "copia" la strategia modale che ha nel gioco di $\psi[X/\psi^{n1}(\bot)]$ su (M, w) fino a quando il gioco modale non arriva ad una posizione (l, v), dove l è un letterale di At, oppure ad una posizione del tipo $(\psi^{n1}(\bot), v)$, dove la sottoformula $\psi^{n1}(\bot)$ corrisponde all'occorrenza che sostituisce la variabile X in $\psi(X)$; in quest'ultimo caso, nel μ gioco Verifier non ha a sua disposizione tale mossa e la sostituisce con la mossa (X, v), che verrà automaticamente "rigenerata" in $(\psi(X), v)$. Nel primo caso, la μ partita termina con una vittoria di Verifier, perché lo stesso accade nella corrispondente partita modale; nel secondo caso, sappiamo che $(M, v) \models \psi^{n1}(\bot)$, perché la strategia seguita da Verifier nel caso modale è vincente. Ne segue che, necessariamente, si deve avere $n > 1$ e quindi $\psi^{n1}(\bot) = \psi[X/\psi^{n2}(\bot)]$; Verifier può allora ricominciare a "copiare" la strategia modale nel μ gioco, fino a che il gioco modale non raggiunge una posizione (l, v), dove l è un letterale di At, oppure una posizione di tipo $(\psi^{n2}(\bot), v)$, dove la sottoformula $\psi^{n2}(\bot)$ corrisponde all'occorrenza che sostituisce la variabile X in $\psi(X)$; in quest'ultimo caso si deve avere necessariamente $n > 2$ e quindi $\psi^{n2}(\bot) = \psi[X/\psi^{n3}(\bot)]$. Continuando in questo modo, ci accorgiamo che la variabile X viene rigenerata nel μ gioco sempre

in corrispondenza ad un passaggio, nel gioco modale, da una approssimazione $\psi^k(\bot)$ ad una nuova approssimazione $\psi^{k1}(\bot)$, partendo da $k = n$; ne segue che X non potrà essere rigenerata per più di n volte, perché questo corrisponderebbe nel gioco modale a raggiungere una posizione di tipo $(\bot, v)$ vincente per Falsifier. Ne segue che, se Verifier segue la strategia sopra descritta, il gioco μ avrà solo partite finite e queste saranno vinte da Verifier. Viceversa, supponiamo che Verifier abbia una strategia vincente σ nel μ gioco di $\mu X.\psi(X)$ su (M, w). Questo significa che nessuna partita giocata da Verifier seguendo tale strategia può essere infinita. Dunque l'albero delle possibili posizioni raggiunte dalla posizione iniziale $(\mu X.\psi(X), w)$, in tutte le possibili partite che vedono Verifier giocare con la strategia σ, non ha rami infiniti. D'altra parte, ogni nodo ha un numero finito di successori (essendo minore o uguale alla cardinalità del modello). Dunque l'albero ha un numero finito di nodi, in particolare ha altezza finita. Sia n il numero massimo di nodi del tipo (X, v) in un ramo dell'albero (n è chiaramente limitato dall'altezza), ovvero il numero di volte che la variabile X può essere "rigenerata". Allora, dimenticando la prima mossa e le mosse di rigenerazione, Verifier può "copiare" σ anche nel gioco modale di $\psi^n(\bot)$ su (M, w), facendo "il contrario" di quanto visto nella prima parte della dimostrazione. Allora Verifier ha una strategia vincente nel gioco modale di $\psi^n(\bot)$ su (M, m), ovvero $(M, m) \models \psi^n(\bot)$ per quanto visto sui giochi modali. Allora $(M, m) \models \mu X.\psi(X)$. □

Esempio. Sia $M = \{0\}$, $Q^M = \emptyset$ e $R^M = \{(0, 0)\}$. Allora la formula $\mu X.\Diamond(X \vee Q)$ è falsa (allo stato 0). A meno che Verifier faccia a un certo punto la mossa $(Q, 0)$ (che lo farebbe perdere), la partita che segue è:

$$(\Diamond(X \vee Q), 0) \longmapsto (X \vee Q, 0) \longmapsto (X, 0) \longmapsto (\Diamond(X \vee Q), 0) \longmapsto (X \vee Q, 0) \longmapsto \cdots$$

ed è vinta da Falsifier (essendo infinita).

Dualmente, per fomule del tipo $\nu X\psi(X)$ con ψ modale il gioco viene definito in modo che le partite infinite siano sempre vinte da Verifier.

Le condizioni di vittoria per Verifier nel caso generale sono un poco più complicate. Ne diamo solo un accenno. Considerando una qualsiasi formula ϕ del μ calcolo priva di variabili libere, si riscrive la formula in modo che ogni variabile venga quantificata una sola volta. In questo caso diremo che una variabile X è una μ-variabile, se esiste una sottoformula di ϕ del tipo $\mu X\psi$; X è invece una ν-variabile se esiste una sottoformula di ϕ del tipo $\nu X\psi$.

Se non vi sono variabili innestate (in cui ad esempio una variabile X compare nella sottoformula $\nu Y\psi(X, Y)$ di un'altra variabile) nel corso di una partita vincente per Verifier, una μ variabile può essere riciclata (passando da una posizione (X, v) ad una di tipo $(\psi(X), v)$) solo un numero finito di volte, mentre una ν variabile può essere riciclata infinite volte. Se vi sono variabili ineestate, le condizioni si complicano. Consideriamo ad esempio la formula seguente:

$$\nu Y.\mu X.(\Diamond(X) \vee ((P) \wedge \Diamond(Y)))$$

che è vera in uno stato v di un modello M se e solo se da v parte un cammino che incontra infinite volte P.

In questo caso la condizione di vittoria per Verifier prevede che la X possa venire rigenerata anche un numero infinito di volte, ma solo per via di infinite rigenerazioni della variabile Y. Quindi Verifier vince tutte le partite infinite in cui, se X viene rigenerata infinite volte, anche Y viene rigenerata infinite volte.

Scambiando l'ordine dei quantificatori si ottiene la formula

$$\mu X.\nu Y.(\Diamond(X) \vee ((P) \wedge \Diamond(Y)))$$

che è vera in uno stato v di un modello M se e solo se da v parte un cammino sul quale da un certo punto in poi vale sempre P.

In questo caso la condizione di vittoria per Verifier prevede che la X possa venire rigenerata solo un numero finito di volte, mentre la Y pu'o subire infinite rigenerazioni.

Nel caso di una formula del tipo

$$\sigma X_1 \sigma X_2 \sigma X_3 \dots \sigma X_n \psi,$$

con $\sigma \in \{\nu, \mu\}$, la condizione di vittoria per Verifier prevede che nelle partite infinite la variabile di indice massimo fra le variabili rigenerate infinite volte sia una ν variabile.

3.3. **La logica MSO.**

Definizione 12. *La* logica al second'ordine *(SO) è una logica che estende quella al prim'ordine (FO) con l'introduzione di* variabili relazionali*, in particolare assumiamo che ci siano infinite variabili relazionali k-arie per ogni k. Le formule di SO sono definite nel seguente modo:*

- *Se $t_1, \dots, t_k$ sono termini si FO, X è una variabile relazionale k-aria e R è un simbolo relazionale k-ario di FO, allora $X(t_1, \dots, t_n)$ e $R(t_1, \dots, t_n) \in SO$;*
- *Se $\phi_1, \phi_2 \in SO$ e x è una variabile individuale allora $\phi_1 \wedge \phi_2, \phi_1 \vee \phi_2, \neg\phi_1, \exists x\phi, \forall x\phi \in SO$;*
- *Se $\phi \in SO$ allora $\forall X\phi, \exists X\phi \in SO$ per ogni X variabile relazionale.*

Definizione 13. *La logica MSO (o logica al second'ordine monadica) è la restrizione di SO in cui tutte le variabili relazionali hanno arietà 1.*

Le variabili relazionali di arietà 1 vanno interpretate come sottonisiemi del dominio dell'interpretazione, e in particolare definiamo $M \models \exists X\phi(X)$ se e solo se esiste un sottoinsieme $A \subseteq D^M$ tale che $M \models \phi(A)$.
La logica MSO è strettamente più espressiva di FO. Infatti, ad esempio, in FO non possiamo esprimere la proprietà di fondatezza per quanto visto all'inizio del corso. Invece, in MSO posso dire che una relazione binaria R è infondata se e solo se

$$\exists X(\exists x X(x) \wedge \forall y(X(y) \to \exists z(X(z) \wedge R(y, z)))).$$

Infatti, se R è infondata allora l'insieme X cercato è infatti l'insieme costituito dagli elementi della successione che testimonia l'infondatezza (cioè $X = \{d_i : i \in \mathbb{N}\}$ con $R(d_i, d_{i+1})$ per ogni i).
Viceversa, se esiste un tale insieme allora R è infondata, infatti un cammino che lo testimonia è definito scegliendo x come primo elemento, e ad ogni passo successivo scelgo come d_{n+1} un elemento dell'insieme in relazione con d_n.

3.3.1. *Traduzione delle formule del μ-calcolo.* La traduzione ST delle formule modali in $FO(x)$ può essere estesa a formule del μ-calcolo in $MSO(x)$. Ricordando infatti il Teorema 9 si ottiene che un elemento $v \in M$ appartiene al minimo punto fisso di un operatore fissato se e solo se appartiene ad ogni suo "pre-punto fisso", mentre appartiene al massimo punto fisso se e solo se appartiene ad almeno un "post-punto fisso". Dunque possiamo definire:

- $ST(\mu X.\phi) = \forall X(\forall y(ST(\phi)(y) \to X(y)) \to X(x))$;
- $ST(\nu X.\phi) = \exists X(\forall y(X(y) \to ST(\phi)(y)) \wedge X(x))$.

In analogia al teorema di Van Benthem, vale anche il seguente risultato:

Teorema 14 (Janin-Walukiewicz)**.** *Una formula $F \in MSO(x)$ è invariante per bisimulazione se e solo se $ST(F)$ è equivalente a una formula di L_μ.*

L'idea per la dimostrazione della necessità per le formule del tipo $\mu X.\phi$ con ϕ modale nel caso dei modelli finiti è la seguente: se M è un modello di Kripke con $w \in M$ tale che $(M, w) \models \mu X.\phi$, allora $(M, w) \models \phi^n(\bot)$ per qualche n. Ma $\phi^n(\bot)$ è una formula modale, dunque se (N, v) è bisimile a (M, w) allora $(N, v) \models \phi^n(\bot)$, dunque $(N, v) \models \mu X.\phi$.

3.4. **Esercizi.**

(1) Dimostrare che, per ogni k, la classe dei grafi k-colorabili è esprimibile con una formula ϕ_k di MSO (un grafo si dice k-colorabile se è possibile assegnare un colore scelto fra k colori possibili ad ogni vertice, in modo che vertici adiacenti abbiano colore diverso).
Dimostrare che la classe dei grafi hamiltoniani è esprimibile in SO sui grafi finiti (suggerimento: un grafo è Hamiltoniano se posso ordinare linearmente i suoi vertici in modo che se due vertici sono uno il successore dell'altro in quest'ordine, allora sono adiacenti nel grafo, e l'ultimo elemento dell'ordine è adiacente al primo).
(2) Esprimere con un enunciato della logica MSO nel linguaggio dei grafi etichettati la proprietà "Dallo stato corrente è possibile raggiungere in un numero finito di passi uno stato in cui vale P".
(3) Tradurre la formula $\nu X.\Diamond(X \wedge P)$ in una formula di $MSO(x)$. Qual è il significato di questa formula?