

Indice

	Pagina
1 Nozioni di Base	2
1.1 Relazioni Binarie	2
1.2 Grafi	3
1.3 Insiemi	5
1.3.1 Cenni di teoria degli insiemi	5
1.3.2 Rappresentazione di insiemi tramite grafi	6
1.4 Bisimulazione	7
1.4.1 Definizione e risultati generali	8
1.4.2 Bisimulazione massima	10
1.4.3 Interpretazione insiemistica della bisimulazione	11
1.5 RSCP	14
1.5.1 Definizione e risultati generali	14
Bibliografia	16

1 Nozioni di Base

1.1 Relazioni Binarie

Riportiamo la definizione di *relazione binaria* su uno o due insiemi, che sarà utile per definire formalmente il concetto di *grafo*, fondamentale all'interno di questo elaborato.

Definizione 1.1. Chiameremo *relazione binaria* su A, B qualsiasi sottoinsieme del prodotto cartesiano $A \times B$.

Chiameremo *relazione binaria* su A qualsiasi sottoinsieme del prodotto cartesiano $A \times A$.

Diremo che u, v sono *in relazione* rispetto a R se $(u, v) \in R$. In questo caso potremo usare la notazione uRv .

Alcune relazioni binarie mostrano proprietà fondamentali, che presentiamo nella definizione seguente:

Definizione 1.2. Sia R una relazione binaria su A . Diremo che R è

- *riflessiva* se $\forall x \in A, xRx$;
- *simmetrica* se $xRy \implies yRx$ ($x, y \in A$)
- *transitiva* se $(xRy \wedge yRz) \implies xRz$ ($x, y, z \in A$)

Esempio 1.1. La relazione \leq sui naturali è riflessiva e transitiva, ma non simmetrica.

La relazione $=$ ($a = b \iff$ “ a, b sono lo stesso numero”) sui naturali è simmetrica, riflessiva e transitiva.

Definizione 1.3. Una relazione binaria riflessiva, simmetrica e transitiva si dice *relazione di equivalenza*.

Una relazione di equivalenza divide un insieme in *classi di equivalenza* all'interno delle quali tutte le coppie di elementi sono in relazione.

A partire da una relazione binaria R possiamo costruire relazioni binarie che contengono R , e che mostrano una o più delle proprietà presentate sopra, le cosiddette *chiusure*. Ogni chiusura è costituita dall'unione della relazione iniziale e di un insieme di coppie costruito con un criterio differente a seconda del tipo di chiusura, che fornisce la proprietà desiderata.

Definizione 1.4. Sia R una relazione binaria su A . Definiamo le seguenti chiusure:

- *riflessiva*: $R_r = R \cup \{(x, x) \mid \forall x \in A\}$
- *simmetrica*: $R_s = R \cup \{(y, x) \mid \forall x, y : (x, y) \in R\}$
- *transitiva*: $R_t = R \cup \{(x, z) \mid \forall x, z : \exists y : (x, y) \in R \wedge (y, z) \in R\}$

Esempio 1.2. La chiusura riflessiva della relazione $<$ (minore stretto) è la relazione \leq .

Nel seguito sarà fondamentale la definizione seguente:

Definizione 1.5. Sia R una relazione binaria su $A \times B$. Chiameremo *relazione inversa* di R (indicata con $R^{-1}(\cdot)$) la funzione su B definita come $R^{-1}(y) = \{x \in A : xRy\}$. Inoltre denoteremo con $R(\cdot)$ la funzione su A definita come $R(x) = \{y \in B : xRy\}$.

Useremo frequentemente la notazione $R^{-1}(X)$ o $R(X)$ dove X è un insieme. In questo caso l'insieme risultante è dato dall'unione di tutti gli insiemi $R(x)$ o $R^{-1}(x)$, per ogni $x \in X$.

Introduciamo la seguente notazione:

Definizione 1.6. Sia A un insieme. Denoteremo con $|A|$ la *cardinalità* di A , cioè il numero dei suoi elementi.

Analogamente, data una relazione binaria R , denoteremo con $|R|$ il numero delle coppie messe in relazione da R .

1.2 Grafi

Con queste premesse possiamo definire un *grafo* come segue:

Definizione 1.7. Sia V un insieme finito non vuoto. Sia \rightarrow una relazione binaria su V .

Chiameremo *grafo diretto* o *grafo orientato* la coppia $G = (V, \rightarrow)$.

In questo caso

- V è l'insieme dei *nodi* o *vertici*;
- \rightarrow è una relazione binaria che mette in relazione alcuni dei nodi di G

Esempio 1.3. Il grafo di Figura 1 è descritto dalla coppia

- $V = \{a, b, c, d, e\}$
- $\rightarrow = \{(a, b), (a, d), (b, c), (d, c), (c, e), (d, e)\}$

Nel seguito utilizzeremo ampiamente la seguente terminologia:

Definizione 1.8. Sia $G = (V, \rightarrow)$ un grafo. Diremo che un nodo $u \in V$ è una *foglia* se $\nexists v \in V : u \rightarrow v$. Diremo che u è *parente* di v e che v è *figlio* di u se $u \rightarrow v$. Chiaramente ogni nodo può avere più parenti e più figli.

Un grafo è quindi un insieme di elementi (i *nodi*) accoppiato con un insieme di relazioni tra questi elementi (gli *archi* o *rami*).

È naturale associare questo concetto all'idea di percorso: ogni grafo è definito da un insieme di nodi ed un insieme di *cammini* che consentono di spostarsi da un nodo ad un altro.

La seguente definizione sorge in modo spontaneo da questo punto di vista:

Definizione 1.9. Sia $G = (V, \rightarrow)$ un grafo. Siano $u, v \in V$. Diremo che v è *raggiungibile da u* , o in alternativa *esiste un cammino da u a v* , o ancora $u \rightarrow_t v$ (la t in pedice sta per “*transitivo*”), se $\exists x_n \subset V$ (una sequenza finita di nodi) di lunghezza $K : x_K = v, x_0 = u, x_n \rightarrow x_{n+1}$.

L'esistenza di un cammino tra nodi fornisce un criterio immediato per partizionare un grafo in gruppi di nodi. Diamo innanzitutto la seguente definizione:

Definizione 1.10. Diremo che un grafo (V, \rightarrow) è *fortemente connesso* se $\forall v_1, v_2 \in V, v_1 \rightarrow_t v_2$.

In generale un grafo non è fortemente connesso. Tuttavia possiamo individuare facilmente i sottografi massimali fortemente connessi.

Definizione 1.11. Le *componenti fortemente connesse* (*strongly connected components, SCC*) di un grafo (V, \rightarrow) sono le classi di equivalenza della relazione \rightarrow_t [2, Appendice B].

In altre parole, i nodi contenuti in una stessa componente fortemente connessa sono mutuamente raggiungibili.

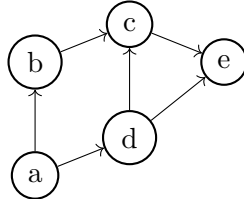


Figura 1: Rappresentazione grafica di un grafo

Esempio 1.4. Nel grafo di Figura 2 le SCC sono evidenziate con colori diversi: $\{a, b, c, d\}, \{e\}$.

Dato un grafo, possiamo definire il partizionamento dei nodi in SCC come segue:

Definizione 1.12. Sia $G = (V, \rightarrow)$ un grafo. Definiamo il grafo $G^{SCC} = (V^{SCC}, \rightarrow^{SCC})$ delle componenti fortemente connesse:

- $V^{SCC} = \{C : \text{"}C \text{ è una classe di equivalenza per } \rightarrow_t \text{ su } V\}$
- $\rightarrow^{SCC} = \{(A, B) \in V^{SCC} \times V^{SCC} : A \neq B, \exists m \in A, n \in B : m \rightarrow n\}$

Riportiamo la seguente proprietà immediata:

Proposizione 1.1. Sia G^{SCC} il grafo delle SCC di un grafo G generico. Allora G^{SCC} è aciclico.

Dimostrazione. Suppongo per assurdo che in G^{SCC} esista un ciclo. Allora tutti i nodi di V^{SCC} facenti parte del ciclo sono mutuamente raggiungibili (percorrendo il ciclo). Quindi tutti i nodi fanno parte della stessa SCC, ma questo è assurdo. \square

Esempio 1.5. La Figura 3.a rappresenta un grafo generico, la Figura 3.b rappresenta il suo grafo delle componenti fortemente connesse associato.

Dato un grafo generico possiamo determinare la partizione in SCC sfruttando un algoritmo avente complessità lineare $\Theta(|V| + |\rightarrow|)$ [6]. L'algoritmo non verrà trattato in questo elaborato.

1.3 Insiemi

1.3.1 Cenni di teoria degli insiemi

In generale supporremo validi gli assiomi su cui si fonda la teoria degli insiemi ZFC, ad eccezione dell'Assioma di Fondazione. Formuliamo innanzitutto i seguenti Assiomi:

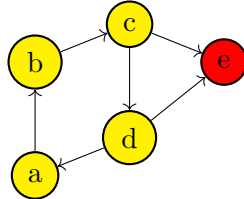


Figura 2: SCC di un grafo

Assioma 1.1 (di estensionalità). Due insiemi sono uguali \iff contengono gli stessi elementi.

Assioma 1.2 (di fondazione). Ogni insieme non vuoto contiene un elemento disgiunto dall'insieme stesso.

Del primo faremo un uso esplicito nel seguito. Il secondo è evidentemente non valido nell'ambito trattato in questo elaborato. Introduciamo la seguente definizione:

Definizione 1.13. Diremo che un insieme è *ben-fondato* se non contiene se stesso. Altrimenti diremo che è *non-ben-fondato*.

Esempio 1.6. L'insieme $\Omega = \{\Omega\}$ è non-ben-fondato. L'insieme $A = \{1, 2, 3\}$ è ben-fondato.

Una formulazione equivalente dell'Assioma 1.2 è la seguente: “ $\forall A$ la relazione \in è ben-fondata su A ” [5, Chapter III.4], che implica la non-esistenza, in ZFC, di insiemi non-ben-fondati.

Rinunciando all'Assioma 1.2 si ottiene un sistema di assiomi che ammette l'esistenza di insiemi non-ben-fondati, tuttavia questo sistema non è più sufficiente a descrivere in modo esaustivo l'aritmetica per mezzo di operazioni su insiemi. Per ovviare a questa mancanza si introduce l'Assioma AFA, che verrà presentato e discusso nel seguito.

1.3.2 Rappresentazione di insiemi tramite grafi

In alcuni casi risulta conveniente fornire un'interpretazione insiemistica della nozione di grafo vista sopra. Introduciamo innanzitutto una nozione fondamentale:

Definizione 1.14. Sia $G = (V, \rightarrow)$ un grafo orientato. Sia $u \in V : \forall v \in V, u \rightarrow_t v$, cioè ogni nodo di G è raggiungibile da u . Allora la terna (V, \rightarrow, u) si dice *accessible pointed graph*, o *APG*.

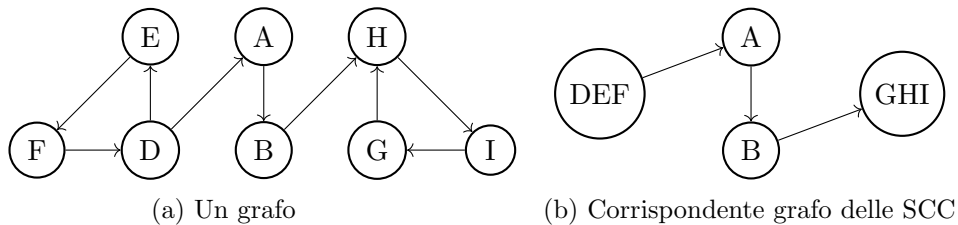


Figura 3: Un grafo ed il corrispondente grafo delle SCC

Per rappresentare un insieme tramite un grafo è necessario definire un processo denominato *decorazione*:

Definizione 1.15. Chiameremo *decorazione* di un APG l'assegnazione di un insieme ad ogni suo nodo.

Associando la relazione di raggiungibilità \rightarrow alla relazione di appartenenza \in disponiamo di tutto il necessario per la rappresentazione di insiemi:

Definizione 1.16. Chiameremo *immagine* (o *picture* in [1]) di un insieme A la coppia composta da un APG (G, v) e da una decorazione, in cui a v è associato A .

Vale la seguente proposizione, dimostrata in [1]:

Proposizione 1.2. *Ad un APG aciclico è possibile associare un'unica decorazione.*

Questo risultato non è stato dimostrato nel caso di un APG contenente almeno un ciclo. Per questo motivo viene presentato il seguente assioma, di cui si è discusso poco sopra:

Assioma 1.3 (AFA, Anti-Foundation-Axiom). Ogni APG possiede un'unica decorazione.

L'assioma AFA ha un'ovvia conseguenza:

Corollario 1.1. *Ogni APG è immagine di un unico insieme.*

Esempio 1.7. In Figura 4 sono rappresentati alcuni insiemi sotto forma di APG.

1.4 Bisimulazione

In questa sezione introdurremo la definizione di bisimulazione ed alcune proprietà immediate. In seguito esamineremo la relazione tra la teoria degli insiemi e la bisimulazione.



Figura 4: Rappresentazione di insiemi tramite grafi

1.4.1 Definizione e risultati generali

Definizione 1.17. Siano $G_1 = (V_1, \rightarrow_1), G_2 = (V_2, \rightarrow_2)$ due grafi. Diremo che una relazione binaria $R : V_1 \times V_2$ è una *bisimulazione* su G_1, G_2 se $\forall a \in V_1, b \in V_2$ valgono congiuntamente le seguenti proprietà:

- $aRb, a \rightarrow_1 a' \implies \exists b' \in V_2 : (a'Rb' \wedge b \rightarrow_2 b')$
- $aRb, b \rightarrow_2 b' \implies \exists a' \in V_1 : (a'Rb' \wedge a \rightarrow_1 a')$

Possiamo definire in modo analogo una bisimulazione su un unico grafo G , ponendo $G_1 = G_2 = G$.

Definiamo un'importante caratteristica di una coppia qualsiasi di grafi, che verrà sfruttata ampiamente nel seguito

Definizione 1.18. Siano $G_1 = (V_1, \rightarrow_1), G_2 = (V_2, \rightarrow_2)$ due grafi. Diremo che sono *bisimili* se $\exists R : V_1 \times V_2 : R$ è una bisimulazione su G_1, G_2 . Diremo che due APG $(G_1, v_1), (G_2, v_2)$ sono *bisimili* se G_1, G_2 sono bisimili e vale $v_1 R v_2$ per almeno una bisimulazione su G_1, G_2 .

Osservazione 1.1. Una bisimulazione può non essere riflessiva, simmetrica, nè transitiva.

Esempio 1.8. La relazione $aRb \iff "a, b \text{ sono lo stesso nodo}"$ su un grafo qualsiasi è una bisimulazione riflessiva, simmetrica e transitiva.

La relazione $R = \emptyset$ è una bisimulazione su un grafo qualsiasi, ma non è riflessiva.

La relazione $R = \{(a, a), (b, b), (c, c), (d, d), (a, b), (b, c), (c, d)\}$ sul grafo $G = (\{a, b, c, d\}, \{(a, b), (b, c), (c, d), (d, d)\})$ è una bisimulazione, ed è solamente riflessiva.

Dalla definizione di bisimulazione possiamo dedurre una proprietà interessante di una qualsiasi sua chiusura:

Teorema 1.1. *Sia R una bisimulazione sul grafo G . La sua chiusura riflessiva, simmetrica o transitiva è ancora una bisimulazione su G .*

Dimostrazione. Consideriamo separatamente le tre relazioni R_r, R_s, R_t , rispettivamente la chiusura riflessiva, simmetrica e transitiva:

- R_r Per definizione $R \subset R_r$, quindi è sufficiente dimostrare che R_r è una bisimulazione quando gli argomenti $u, v \in V$ non sono distinti.
Sia $u \in V$. Chiaramente per definizione di R_r si ha uR_ru . Se $\exists u' \in V : u \rightarrow u'$ allora (sempre per definizione di R_r) si ha $u'R_ru'$.

R_s Per definizione $R \subset R_s$, quindi è sufficiente dimostrare che R_s è una bisimulazione quando per gli argomenti $u, v \in V$ si ha uRv ma non vRu .

Sia $(u, v) \in V \times V$. Allora

$$uR_s v \implies uRv \vee vRu$$

Suppongo ad esempio che vRu .

$$\begin{aligned} &\implies \forall v' \in V : (v \rightarrow v') \exists u' \in V : (u \rightarrow u' \wedge v'Ru') \\ &\implies u'R_s v' \end{aligned}$$

e

$$\begin{aligned} &\implies \forall u' \in V : (u \rightarrow u') \exists v' \in V : (v \rightarrow v' \wedge v'Ru') \\ &\implies u'R_s v' \end{aligned}$$

cioè sono dimostrate le due condizioni caratteristiche della bisimulazione.

La dimostrazione è analoga se uRv .

R_t Per definizione $b \subset R_t$, quindi è sufficiente dimostrare che R_t è una bisimulazione quando per gli argomenti $u, v, z \in V$ si ha uRv, vRz ma non uRz .

Sia $(u, v, z) \in V \times V \times V$ con questa proprietà. Allora $\forall u' \in V : u \rightarrow u' \implies \exists v' \in V : v \rightarrow v' \wedge u'Rv'$. Inoltre $\exists z' : z \rightarrow z' \wedge v'Rz'$.

Riordinando si ha $u'Rv', v'Rz'$. Allora per definizione di b_t , $u'R_t z'$.

In modo speculare si ottiene la seconda condizione caratteristica della bisimulazione.

□

Da questa proposizione si deduce il seguente corollario, che risulta dall'applicazione iterativa delle tre chiusure viste in precedenza:

Corollario 1.2. *Ad ogni bisimulazione R è possibile associare una bisimulazione $\tilde{R} : R \subset \tilde{R} \wedge \tilde{R}$ è una relazione di equivalenza.*

Concludiamo la sezione relativa ai risultati generali sulla bisimulazione con la seguente proposizione, che sarà utile nel seguito:

Proposizione 1.3. *Siano R_1, R_2 due bisimulazioni su G_1, G_2 . Allora $R = R_1 \cup R_2$ è ancora una bisimulazione.*

Dimostrazione. Siano $u, v : uRv$. Sia $u' : u \rightarrow u'$. Allora deve essere $uR_1 v \vee uR_2 v$. Ma quindi $\exists v' : (v \rightarrow v' \wedge u'R_{1|2} v')$. □

1.4.2 Bisimulazione massima

Definiamo ora il concetto di *bisimulazione massima*, che gioca un ruolo chiave nella risoluzione dei problemi considerati in questo elaborato:

Definizione 1.19. Diremo che una bisimulazione R_M su G_1, G_2 è *massima* se $\forall R : "R \text{ è una bisimulazione su } G_1, G_2"$ si ha $uRv \implies uR_Mv$.

Naturalmente la bisimulazione massima dipende dai due grafi presi in esame. Possiamo dedurre alcune caratteristiche in modo molto semplice:

Proposizione 1.4. *Valgono le seguenti proprietà:*

1. *La bisimulazione massima su due grafi G_1, G_2 è unica*
2. *La bisimulazione massima è una relazione di equivalenza*

Dimostrazione. Le proprietà seguono banalmente dal Corollario 1.2 e dall'Osservazione 1.3.

1. Suppongo per assurdo che esistano due bisimulazioni massime R_{M_1}, R_{M_2} . La loro unione è ancora una bisimulazione, che è "più massima" delle supposte bisimulazioni massime.
2. Se per assurdo la bisimulazione massima non fosse una relazione di equivalenza, potremmo considerare la sua chiusura riflessiva, simmetrica e transitiva, che sarebbe "più massima" ed anche una relazione di equivalenza.

□

Naturalmente il concetto di *bisimulazione massima* può essere definito anche su unico grafo G . Questo caso si rivelerà di grande interesse nel seguito. Per ora dimostriamo il seguente risultato:

Teorema 1.2. *Sia G un grafo (finito). Allora $\exists R_M$ la bisimulazione massima su G .*

Dimostrazione. Può esistere solamente un numero finito di relazioni binarie su G , e questo numero fornisce un limite superiore al numero massimo di bisimulazioni su G . Allora possiamo considerare l'unione di questo numero finito di bisimulazioni, che sarà chiaramente la bisimulazione massima. □

1.4.3 Interpretazione insiemistica della bisimulazione

Mostriamo ora una conseguenza diretta dell'Assioma di Estensionalità e di AFA:

Teorema 1.3. *Due APG rappresentano lo stesso insieme \iff sono bisimili.*

Dimostrazione. Dimostriamo separatamente le due implicazioni:

(\implies) Osserviamo innanzitutto che la relazione binaria \equiv su V_A, V_B definita come segue:

$a \equiv b \iff$ "le decorazioni di A, B associano ad a, b lo stesso insieme"

è una bisimulazione sui grafi $G_A = (V_A, \rightarrow_A), G_B = (V_B, \rightarrow_B)$.

Chiaramente se $a \equiv b, a \rightarrow a'$ si ha:

- $a' \in a$, associando ad a, a' gli insiemi che rappresentano secondo la decorazione (l'unica) considerata;
- a, b rappresentano lo stesso insieme.

Quindi per l'Assioma di Estensionalità $\exists b' \in b : b' = a'$, cioè $b \rightarrow b'$ e $a' \equiv b'$. Si procede specularmente per la seconda condizione caratteristica della bisimulazione.

La relazione \equiv è una bisimulazione sugli APG A, B quando si assume per ipotesi che A, B rappresentino lo stesso insieme.

(\impliedby) Sia R una bisimulazione su A, B . Consideriamo la decorazione d_A (l'unica) di A . Vogliamo estrapolarne una decorazione per B . Dalla possibilità di operare questo procedimento, dall'Assioma di Estensionalità e da AFA potremo dedurre l'uguaglianza degli insiemi rappresentati.

Definiamo l'applicazione d , che ad ogni nodo di B associa un insieme:

$$d(v) = d_A(u), \text{ con } u \text{ nodo di } A : uRv$$

Osservazione. Per ogni nodo v di B deve esistere almeno un nodo u di $A : uRv$ perchè si suppone che i due APG siano bisimili.

Dimostriamo che d è una decorazione di B . In altre parole vogliamo dimostrare che per ogni nodo v di B l'insieme $d(v)$ contiene tutti e soli gli insiemi $d(v') \forall v' \text{ nodo di } B : v \rightarrow v'$.

- Supponiamo per assurdo che tra i figli di v “manchi” il nodo corrispondente ad un elemento $X \in d(v)$. Poichè la decorazione di A è ben definita, il nodo u di $A : uRv$ deve avere un figlio corrispondente a X , che chiameremo u' . Ma $uRb, u \rightarrow u' \implies \exists v' : v \rightarrow v', u'Rv'$. Cioè $d(v') = X$. Dunque il nodo mancante è stato identificato.
- Supponiamo per assurdo che tra i figli di v ci sia un nodo “in più”, ovvero un nodo $v' : v \rightarrow v', d(v') = Y$ con $Y \notin d(v)$. Ma allora, sempre considerando u il nodo di $A : uRv$, dovrebbe esistere un $u' : u \rightarrow u', u'Rv'$, cioè $d_A(u') = d(v') = Y$. Ma allora $Y \in d_A(u) = d(v)$, deduzione che è chiaramente in contrasto con l'ipotesi.

E' possibile che ci siano due nodi di a_1, a_2 di A ed un nodo b di B per cui vale $a_1Rb \wedge a_2Rb$. In questo caso la decorazione definita è ambigua. Per questo motivo correggiamo la formulazione di d come segue:

$$d(v) = X, \text{ con } X \text{ l'insieme associato al nodo } u \text{ di } A : uRv, \\ \text{con } u \text{ preso casualmente tra i nodi di } A \text{ bisimili a } v.$$

Per AFA esiste un'unica decorazione di b , quindi si deve avere, alternativamente, per ogni nodo v di B :

- $\exists! u$ nodo di $A : uRv$;
- $\forall u$ nodo di $A : uRv$, la decorazione di A associa a tutti gli u lo stesso insieme.

Dunque l'ambiguità è risolta.

□

Tenendo conto di quanto affermato nella sezione 1.3.2, il Teorema 1.3 dimostra che la bisimulazione può sostituire la relazione di uguaglianza tra insiemi quando questi sono rappresentati con APG [3].

Dopo questa considerazione possiamo dare la seguente definizione:

Definizione 1.20. Sia R una bisimulazione su G che sia anche una relazione di equivalenza. Definiamo un nuovo grafo $G_R = (VR, \rightarrow_R)$ come in [4], che chiameremo *contrazione rispetto alla bisimulazione R di G* :

- $VR = \{A = \{m \in V : \forall n \in A, mRn\}\}$
- $[m]_R \rightarrow_R [n]_R \iff \exists c \in [n]_R : m \rightarrow c$

Si definisce *classe del nodo* a rispetto alla bisimulazione R , con la notazione $[a]_R$, il nodo di VR in cui viene inserito il nodo a .

La Definizione 1.20 è di fondamentale importanza per la seguente osservazione:

Proposizione 1.5. *Sia G un grafo, e sia G_R come nella Definizione 1.20, per una bisimulazione R qualsiasi. Allora G, G_R sono bisimili.*

Dimostrazione. Sia $\equiv \subset V \times VR$ la relazione binaria definita come segue:

$$m \equiv M \iff M = [m]_R$$

Vogliamo dimostrare che tale relazione è una bisimulazione sui grafi G, G_R . Supponiamo che $x \equiv X$, e che $x \rightarrow y$ per qualche $y \in V$. Chiamiamo $Y := [y]_R$. Allora, per la Definizione 1.20, si ha $X \rightarrow Y$. Inoltre vale banalmente $y \equiv Y$.

Per dimostrare la seconda condizione caratteristica della bisimulazione, supponiamo che $x \equiv X$, e che $X \rightarrow Y$ per qualche $Y \in VR$. Sempre per la Definizione 1.20 deve esistere un $y \in Y : (y \equiv Y \wedge x \rightarrow y)$. \square

La Proposizione 1.5 ha una conseguenza ovvia, che risulta evidente per il Teorema 1.3:

Corollario 1.3. *Sia R una bisimulazione che sia anche una relazione di equivalenza. Allora l'APG (G, v) e l'APG $(G_R, [v]_R)$ rappresentano lo stesso insieme.*

Quindi risulta naturale sfruttare le proprietà della bisimulazione per minimizzare la rappresentazione di insiemi, considerando che è sufficiente una bisimulazione sulla rappresentazione iniziale per ottenere una rappresentazione equivalente. Definiamo una relazione d'ordine sulle rappresentazioni:

Definizione 1.21. Diremo che la rappresentazione (G_a, v_a) di un insieme è *minore* della rappresentazione equivalente (G_b, v_b) se $|Va| < |Vb|$. Diremo che una rappresentazione è *minima* se non esiste una rappresentazione equivalente minore.

Osservazione 1.2. La *contrazione per bisimulazione* è una rappresentazione minore (o eventualmente uguale) di quella iniziale.

Concludiamo la sezione con il seguente risultato, che stabilisce in modo univoco la bisimulazione prescelta per minimizzare la rappresentazione di un dato insieme:

Teorema 1.4. *Sia (G, v) un APG rappresentante un insieme. Sia R_M la bisimulazione massima su (G, v) . Allora la contrazione per bisimulazione indotta da R_M su (G, v) fornisce la rappresentazione minima dell'insieme.*

Dimostrazione. Suppongo per assurdo che esista una bisimulazione R_V su (G, v) che fornisce una contrazione avente un numero di nodi strettamente inferiore alla contrazione indotta da R_M . Ma questo implica che esistono almeno due nodi di G che sono in relazione secondo R_V e non secondo R_M . Chiaramente questa deduzione è in contrasto con il fatto che R_M è la bisimulazione massima.

Suppongo per assurdo che, dopo la contrazione indotta da R_M , sia possibile trovare una nuova bisimulazione R_O su $(G_{R_M}, [v]_{R_M})$ che induca una contrazione avente un numero di nodi strettamente inferiore a quello di $(G_{R_M}, [v]_{R_M})$. Chiaramente $R_O \subset VR_M \times VR_M$.

Definisco una nuova bisimulazione $R_{\widetilde{M}} \subset V \times V$ tale che

$$xR_{\widetilde{M}}y \iff (xR_My \vee [x]_{R_M}R_O[y]_{R_M})$$

Per definizione di bisimulazione massima bisogna avere $R_{\widetilde{M}} \subset R_M$, quindi non è possibile che la contrazione indotta da R_O sia una rappresentazione minore di quella indotta da R_M . \square

1.5 RSCP

In questa sezione introduciamo il concetto di *RSCP* o *Relational Stable Coarsest Partition* (partizione “più rozza” stabile rispetto ad una relazione). Evidenzeremo poi il legame tra questo problema e quello della determinazione della bisimulazione massima, che viene sfruttato da molti algoritmi in quanto la formulazione di RSCP lo rende un problema avvicinabile più facilmente dal punto di vista algoritmico di quanto lo sia quello della determinazione della bisimulazione massima.

1.5.1 Definizione e risultati generali

Cominciamo con alcune definizioni su cui struttureremo la formulazione del problema:

Definizione 1.22. Sia A un insieme finito. Sia X un insieme contenente n elementi, per ognuno dei quali si ha $X_i \subseteq A$. Diremo che X è una *partizione* di A se:

$$\bigcup_{i=1}^n X_i = A \quad \wedge \quad X_i \cap X_j = \emptyset \quad \forall i, j \in \{1, \dots, n\} \quad \wedge \quad x_i \neq \emptyset \quad \forall i$$

Inoltre diremo che gli insiemi X_i sono i *blocchi* della partizione X .

Definizione 1.23. Siano X_1, X_2 due partizioni dello stesso insieme. Diremo che X_2 è *più fine* rispetto a X_1 se $\forall x_1 \in X_1, \forall x_2 \in X_2$ si ha $x_2 \subseteq x_1$.

In molti casi è interessante studiare il partizionamento di un insieme quando i blocchi sono sottoposti a particolari restrizioni. Una delle restrizioni più interessanti dal punto di vista pratico è la seguente:

Definizione 1.24. Sia A un insieme, ed R una relazione binaria su A . Sia X una partizione su A . Diremo che X è *stabile* rispetto ad R se $\forall x_i, x_j \in X$ (distinti) vale $x_i \subseteq R^{-1}(x_j) \vee x_i \cap R^{-1}(x_j) = \emptyset$.

In altre parole per ogni coppia di blocchi di una partizione stabile si hanno due alternative:

- Tutti gli elementi del primo blocco sono in relazione con almeno un elemento del secondo blocco;
- Nessuno degli elementi del primo blocco è in relazione con qualche elemento del secondo blocco.

Con queste premesse possiamo definire il problema della determinazione della RCSP:

Definizione 1.25. Sia A un insieme. Sia R una relazione. Diremo che una partizione X è la RCSP di A rispetto ad R se, di tutte le partizioni di A stabili rispetto ad R , è quella contenente il minor numero di blocchi (per questo motivo si dice che è la *più rozza*).

Bibliografia

- [1] Peter Aczel. *Non-well-founded sets*, volume 14 of *CSLI lecture notes series*. CSLI, 1988.
- [2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009.
- [3] Agostino Dovier, Carla Piazza, and Alberto Policriti. A fast bisimulation algorithm. In *International Conference on Computer Aided Verification*, pages 79–90. Springer, 2001.
- [4] Raffaella Gentilini, Carla Piazza, and Alberto Policriti. From bisimulation to simulation: Coarsest partition problems. *Journal of Automated Reasoning*, 31(1):73–103, 2003.
- [5] Kenneth Kunen. *Set theory an introduction to independence proofs*. Elsevier, 2014.
- [6] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.