



UNIVERSITA' DEGLI STUDI DI PADOVA
FACOLTA' DI SCIENZE STATISTICHE
CORSO DI LAUREA SPECIALISTICA IN SCIENZE STATISTICHE,
ECONOMICHE, FINANZIARIE E AZIENDALI

TESI DI LAUREA

ALGORITMI D'ORDINAMENTO SU ALBERI
PER IL CALCOLO DEL RISCHIO OPERATIVO:
ASPETTI TEORICI, APPLICAZIONI E COMPLESSITÀ COMPUTAZIONALE.

RELATORE: CH.MO PROF. MICHELE BONOLLO

LAUREANDO: GILBERTO CRISANTI

ANNO ACCADEMICO 2004/2005

Ai miei genitori

RINGRAZIAMENTI

Ringrazio i miei genitori per avermi permesso, con i loro sacrifici, di arrivare fino a questo punto, vorrei che sapessero che non me ne dimenticherò.

Ringrazio il professor Bonollo per aver avuto fiducia in me ed avermi permesso di fare quest'esperienza in Engineering, per me molto importante.

Un grazie va naturalmente a Maite, per avermi sempre sostenuto ed aver creduto in me in ogni momento.

Un grazie alle colleghe dell'ufficio per avermi dato allegria anche nei momenti in cui gli impegni mi schiacciavano. Un grazie a Ruggero per avermi lasciato concludere la tesi senza oberarmi di lavoro.

Un grazie a mia sorella Elisabetta, mio cognato Andrea e la piccola Emma per avermi reso zio.

E un grazie va ovviamente agli amici di Riese Pio X: Merigo, John, Enea, Gomez, Alessio, Ghana, Bordi e Ivano, per essere così come sono.

INDICE

<i>INTRODUZIONE</i>	5
<i>CAPITOLO 1</i>	9
BASILEA 2 E I RISCHI OPERATIVI	9
1.1 IL RISCHIO	9
1.2 LE TIPOLOGIE DI RISCHIO	10
1.2.1 I Rischi di Mercato	11
1.2.2 I rischi di credito.....	12
1.2.3. I Rischi Operativi	14
1.3 IL COMITATO DI BASILEA	15
1.3.1 Il Primo Accordo di Basilea	16
1.3.2 Il Nuovo Accordo: Basilea2	17
1.4 I RISCHI OPERATIVI IN BASILEA 2	21
1.4.1 “Basic Indicator Approach” o Metodo dell’Indicatore Semplice	24
1.4.2 “Standardised Approach” o Metodo Standard	25
1.4.3 “Advanced Measurement Approaches” o Metodi Avanzati di Misurazione (AMA).....	27
1.5 COME NASCE IL PROGETTO SUI RISCHI OPERATIVI	28
1.5.1 Engineering Ingegneria Informatica.....	29
1.5.2 Progetto per la Gestione Integrata dei Rischi Operativi.....	30
1.5.3 Come Vengono Trattati i Rischi Operativi.....	31
1.6 GLI ALBERI DI GUASTO E LA FAULT TREE ANALYSIS (FTA).....	32
<i>CAPITOLO 2</i>	35
ORDINAMENTO E ALBERI	35
2.1 CORRISPONDENZE E RELAZIONI.....	36

2.2 LE RELAZIONI IN UN INSIEME	37
2.2.1 Definizione: Proprietà Riflessiva	38
2.2.2 Definizione: Proprietà simmetrica	38
2.2.3 Definizione: Proprietà antisimmetrica	39
2.2.4 Proprietà transitiva.	40
2.3 RELAZIONI	41
2.3.1: Relazioni d'Equivalenza	42
2.3.2. Relazioni d'ordine	44
2.4 TIPI DI RELAZIONI D'ORDINE	45
2.4.1 Definizione: <i>ordine parziale</i>	46
2.4.2 Definizione: <i>ordine totale</i>	46
2.4.3 Definizione: <i>preordine</i>	47
2.5 ORDINAMENTO NEGLI ALBERI.....	48
2.6 I GRAFI	48
2.7 GLI ALBERI.....	52
2.7.1 Terminologia	54
2.7.2 Alberi Binari	54
2.8 VISITE AD UN ALBERO.....	56
2.8.1 Visite in profondità'	56
2.8.2 Visita in Ampiezza.....	57
2.8.3 Vosota in preordine o ordine anticipato	57
2.8.4 Visita in ordine simmetrico o inordine	59
2.8.5 Visita in postordine	60
2.9 IL NOSTRO ALGORITMO	61
CAPITOLO 3.....	63
SVILUPPO DELL'ALGORITMO	63
3.1 COME NASCE LA NECESSITA' DI UN ALGORITMO CHE GENERI L'INDENTATURA.....	63
3.2 COME VIENE PENSATA UNA SOLUZIONE	64
3.3 L'ALGORITMO	66
3.3.1 Le due Fasi	67

1. ordinamento.....	67
2. INBBBDINNENTATURA	67
3.4 COME è STRUTTURATO L'ALGORITMO	68
3.4.1 Primo Modulo.....	69
3.4.2 La procedura di assegnazione dei livelli	72
3.4.3 Processo di Inversione livelli	77
3.4.4 Conclusione Processo di Assegnazione Livelli.....	78
3.4.5 Secondo Modulo:La creazione dell'indentatura.....	80
3.4.6 Tabella con il tracciato e i livelli senza duplicati	81
3.4.7 Generazione dell'indentatura	82
3.5 TABELLE E QUERY CHE INTERAGISCONO CON L'ALGORITMO	88
3.6 COME VIENE CREATA L' INDENTATURA.....	92
3.7 ANALISI DELLA COMPLESSITA'	95
3.7.1 Analisi Teorica	97
3.7.2 Analisi Empirica:Il tempo di calcolo	101
1. Confronto fra i rapporti	102
2. Analisi Grafica	103
3.7.3 Regressione	105
CONCLUSIONE.....	107
ALLEGATI	111
IL CODICE COMPLETODELL'ALGORITMO	114

INTRODUZIONE

In questo lavoro tratteremo un'applicazione che è stata sviluppata all'interno di un progetto per la gestione dei rischi operativi. E' giusto prima di tutto inquadrare questi rischi, ecco la definizione che ne da il Comitato di Basilea:

“Il rischio di perdite derivanti da processi, comportamenti del personale o sistemi interni inadeguati o non andati a buon fine, oppure derivanti da eventi esterni”¹.

Il Comitato di Basilea è un organo fondato alla fine del 1974 dai governatori delle banche centrali dei paesi del G10. Non legifera, formula proposte e linee guida, tuttavia le prime proposte , risalenti al 1988, sono diventate norma vincolante in oltre 100 paesi.

L'importanza di gestire questo tipo di rischio non era stata presa in considerazione nel primo Accordo di Basilea del 1988, ma dal 1998 il Comitato ha cominciato a sollevare la questione fino a che nel 2001 ha reso pubblico un documento nel quale il *rischio operativo* si affianca ai *rischi di mercato* e di *credito* per la determinazione del capitale minimo che le banche devono detenere a fronte del rischio.

In coerenza con l'idea di abbandonare un approccio rigido e indifferenziato all'adeguatezza patrimoniale, per il rischio operativo il Comitato propone un trattamento articolato in tre metodi in ordine crescente di complessità e sensibilità al rischio: *indicatore semplice*, *metodo standard* e *misurazione interna*.

Le banche sono esortate a evolvere verso gli approcci più complessi in coerenza con lo sviluppo di sistemi e pratiche per la misurazione del rischio operativo più sofisticati.

¹ *The New Basel Capital Accord*

Per studiare i rischi operativi il metodo che la società Engineering ha deciso di utilizzare è il metodo dell'analisi degli alberi di guasto, o Fault Tree Analysis.

Si tratta di una struttura gerarchica che mostra le relazioni tra un *evento potenziale* che influisce sulle performance del sistema, e le *ragioni* o le *cause sottostanti* di quell'evento.

Il nostro lavoro nasce da un progetto reale, svolto presso la società Engineering Ingegneria Informatica, per la gestione dei rischi operativi di Banca Imi. Nel momento in cui è cominciato il nostro rapporto erano necessarie delle evoluzioni sul software già presente, ma ancora in elaborazione, di gestione del rischio operativo. Le evoluzioni durante i vari mesi di stage presso l'azienda sono state le più svariate, ma in questo scritto approfondiremo una di queste.

Il tema portante di questo lavoro è l'ordinamento di un albero di guasto, ovvero la creazione di un algoritmo che, dato uno o più alberi riesca a riordinarne le componenti in base a un criterio assegnato di esplorazione degli alberi. In tale modo restituire una struttura di facile lettura, simile alla navigazione fra le cartelle di ogni computer.

Nel capitolo 1 parleremo del contesto all'interno del quale si pone questa tesi, verranno presentati quindi quali sono i rischi che le banche devono quantificare: *rischi di mercato*, *rischi di credito*, *rischi operativi*.

In seguito verrà descritto brevemente cos'è il Comitato di Basilea per la Vigilanza Bancaria e il Nuovo Accordo di Basilea.

Fatto questo inquadramento generale approfondiremo i *Rischi Operativi* e gli indicatori proposti nell'Accordo.

Dopodichè introdurremo il progetto vero e proprio di gestione del rischio operativo, con una breve descrizione della Fault Tree Analysis.

Nel capitolo 2 vedremo gli aspetti teorici che stanno dietro l'ordinamento di un albero. Partiremo dalle proprietà delle relazioni

d'ordine, i concetti di relazione d'ordine e relazione d'equivalenza. Poi passeremo agli alberi, spiegando cosa sono dal punto di vista teorico e a che famiglia appartengono. Questo capitolo servirà anche come introduzione al vocabolario usato in seguito nel terzo capitolo

Nel terzo capitolo verrà presentato l'algoritmo. Parleremo di come nasce l'esigenza di un algoritmo che svolga queste funzioni, vedremo come è stato affrontato il problema, come è stato suddiviso e risolto. Dopo un'analisi più approfondita delle sue funzionalità. Passeremo all'analisi della complessità che sarà divisa in due parti: Teorica ed Empirica. Scopriremo, come cresce il tempo di calcolo al crescere delle dimensioni dell'albero e quindi vedremo se il lavoro fatto è spendibile effettivamente per l'uso che se ne deve fare nelle applicazioni bancarie del calcolo del rischio operativo.

Naturalmente, essendo questo un algoritmo di ordinamento su alberi, il suo utilizzo non è limitato alla funzione per cui è nato, ma può essere usato anche in ambiti diversi.

Alcune applicazioni in sede aziendale, infatti, hanno già dimostrato la sua validità in applicazioni diverse dai Rischi Operativi.

CAPITOLO 1

BASILEA 2 E I RISCHI OPERATIVI

1.1 IL RISCHIO

Parlare di rischio nel 2004 è semplice e complesso allo stesso tempo. E' semplice perché la percezione del rischio è aumentata di recente e la cultura corrente favorisce il dibattito su questi argomenti: basti pensare che dalla caduta delle torri di Manhattan, il mondo si è abituato a vivere in un clima di tensione, nel quale sembra normale correre il rischio di un attentato, o sembra normale avere paura di montare in treno o in aereo l'undici di ogni mese.

D'altra parte lo studio del rischio è complesso perché ci si è accorti che non si può disporre di comportamenti consolidati e di standard accettati, per cui è necessaria una certa misura di sperimentazione con costi e ritorni incerti, complicando molto le cose.

Date le forti pressioni alle quali le banche sono sottoposte: mercati finanziari sempre più complessi e dinamici e la crescente concorrenza, le banche devono dotarsi di strumenti di controllo e analisi flessibili e accurati che consentano loro di raccogliere e monitorare le informazioni relative a rischi, ricavi e costi, controllando al contempo la propria esposizione al rischio. E' per questo che, alla luce di eventi che hanno scosso il mondo bancario, si è sentita l'esigenza di un ente internazionale che potesse garantire vigilanza e trasparenza delle attività bancarie.

Il nuovo millennio si è aperto, per banche e istituzioni finanziarie in genere, all'insegna di una novità epocale, ormai nota con un semplice marchio: Basilea2. Tale marchio sintetizza efficacemente la proposta di revisione dell'Accordo corrente (esiste una Basilea1) sull'Adeguatezza Patrimoniale delle banche, predisposta dal Comitato di Basilea per la Vigilanza Bancaria.

1.2 LE TIPOLOGIE DI RISCHIO

Partiamo innanzitutto dal definire quali sono le tipologie di rischio principali trattate nell'Accordo di Basilea. Sono tre:

- **Rischio di mercato:** rischio che dipende dai fattori che influiscono sull'andamento generale del mercato e che non può, quindi, essere eliminato o ridotto mediante la diversificazione del portafoglio.
- **Rischio di credito:** ossia rischi in cui incorre il titolare di un'attività finanziaria per l'eventuale incapacità parziale o totale della controparte ad assolvere l'impegno assunto. Pensiamo a Parmalat o alla Cirio, gruppi di chiara fama che non sono riusciti ad adempiere i loro doveri verso gli investitori che avevano comprato le loro obbligazioni.
- **Rischi operativi:** sono gli ultimi entrati nell'Accordo di Basilea2, sono rischi derivanti da errori manuali, da malfunzionamento dei sistemi informativi o da malfunzionamento di operazioni connesse allo svolgimento di attività bancaria. Dai più banali errori manuali che provocano acquisto di azioni invece che vendita, fino ai molteplici

virus che possono intasare un server e impedire il normale svolgersi delle attività di lavoro.

Andiamo adesso a vedere ogni singolo rischio un po' più nel dettaglio.

1.2.1 I Rischi di Mercato

Il **rischio di mercato** è il *rischio in cui incorre chi ha investito in strumenti finanziari a seguito di variazioni dell'andamento generale del mercato.*

Il rischio di mercato è andato assumendo negli ultimi tempi una rilevanza crescente per tutti coloro che stabilmente operano nei mercati finanziari; ciò a causa del processo di titolarizzazione ed innovazione, che ha interessato la produzione di strumenti finanziari, e la crescente integrazione tra i mercati finanziari internazionali.

Da ciò nasce l'esigenza di analizzare, in modo più rigoroso e continuo, la rischiosità delle posizioni assunte. Tale esigenza ha indotto i principali istituti bancari e finanziari a sviluppare **modelli statistico-matematici** per la misurazione ed il controllo del rischio di mercato. La principale risposta è stata data con l'elaborazione dei modelli **Value at Risk**, che è una grandezza di natura statistica per la quantificazione del rischio di mercato di un portafoglio; tutti questi modelli hanno in comune il fatto che si propongono di misurare la **massima perdita potenziale** che un portafoglio potrebbe subire, in termini di valore di mercato,

- data una *certa variazione dei parametri* di mercato,
- per un *dato intervallo di confidenza statistica* e
- nell'ambito di un determinato *orizzonte temporale*.

La distinzione tra i diversi metodi riguarda solo il metodo di calcolo utilizzato, mentre le **caratteristiche comuni** sono principalmente:

- forniscono una visione globale del rischio finanziario utilizzando una unità di misura comune (la valuta);
- tengono conto delle correlazioni esistenti tra le varie componenti del portafoglio nonché di quelle tra queste ultime e le variabili del mercato. Identificano e misurano, pertanto, i benefici derivanti dalla diversificazione per la riduzione del rischio complessivo del portafoglio;
- il rischio di mercato è rappresentato dalla possibile perdita del valore del portafoglio della banca a causa dell'incertezza e dell'imprevedibilità dei valori futuri che le principali variabili finanziarie possono assumere. Le perdite future non sono prevedibili con certezza, ma è possibile stimare le perdite attese specificando la distribuzione di probabilità ad esse associata.

Il VaR ha permesso di superare i limiti delle metodologie statiche tradizionali ed il suo successo si deve principalmente alla possibilità di misurare il rischio globale della banca tramite un'unica e sintetica unità di misura.

1.2.2 I rischi di credito

Si fa riferimento alla *probabilità di insolvenza di controparti debentrici della banca in relazione ad adempimenti contrattuali definiti*: mutui, prestiti, fidi, ecc.

L'investitore/banca si attende infatti, a fronte dell'attività di impiego e di investimento (il prestito):

- il ritorno del capitale investito;
- gli interessi che ne rappresentato il frutto;

Si parla di rischio di credito con riferimento all'eventualità che tale aspettativa della banca risulti disattesa, parzialmente o totalmente.

Il rischio di credito può essere diversificato, ma è difficilmente ipotizzabile una sua completa eliminazione a causa della sua componente identificabile come *rischio sistemico*.

Il primo passo per una gestione del rischio di credito consiste nell'**applicazione coerente di valutazioni e rating**² a tutte le opportunità di investimento esistenti.

Al fine di incrementare il grado di sofisticazione della gestione del rischio creditizio, sarebbe opportuno associare ai *rating* ed alle probabilità di perdita attesa le relative *volatilità*, replicando anche nel credito la **metodologia del VaR** di cui si è detto in precedenza.

Il VaR è anche uno degli strumenti alla base del CreditRisk+, un modello, utilizzato ormai da più di dieci anni, creato e proposto dal Credit Suisse First Boston. Tale modello si è mostrato particolarmente adatto allo studio dei rischi di credito soprattutto all'interno del nostro paese. Il suo successo è legato alla compattezza del modello, che necessita di poche variabili.

² Rating: valutazione della solvibilità di una società, del suo patrimonio nonché della capacità di quest'ultima di far fronte agli impegni finanziari. È la risultanza finale di una valutazione che contiene molteplici informazioni. Standard & Poor's, una delle agenzie di rating più rinomate, si avvale di lettere maiuscole per la valutazione: AAA è il voto migliore mentre D è il peggiore.

1.2.3. I Rischi Operativi

I **rischi operativi** sono associati alla “*probabilità di subire perdite a causa di inefficienza nei controlli procedurali interni, di errori umani, di frodi o di carenze di tipo tecnologico quali, ad esempio, guasti o inefficienze dei sistemi informativi interni*”.

Quindi abbiamo per esempio:

- **rischio di execution**: si riferisce a situazioni dove guasti nei sistemi interni o errori procedurali hanno come effetto mancate transazioni o ritardi che possono trasformarsi in elevati costi per la banca;
- **rischio di frode**: si riferisce a situazioni in cui soggetti interni o esterni alla banca adottino comportamenti illeciti quali, ad esempio, manipolazione di informazioni, occultamento di posizioni in essere o di perdite ecc..
- **rischio tecnologico**: si riferisce alla necessità di proteggere i sistemi interni da accessi non autorizzati e problemi analoghi .
- **model risk**: si riferisce al potenziale pericolo che modelli di pricing, di valutazione, di misurazione del rischio, ecc., risultino inadatti alle tipologie di prodotto per le quali i modelli stessi vengono utilizzati.

Nel seguito vedremo quali problemi si riscontrano nell’analisi del rischio operativo e quali metodi sono stati utilizzati per risolverli.

Passiamo ora a presentare il Comitato di Basilea.

1.3 Il Comitato di Basilea

Il Comitato di Basilea, che opera in seno alla Banca dei Regolamenti Internazionali con sede, appunto, a Basilea, fu creato nel 1974 dalle banche centrali dei paesi appartenenti al G10, a seguito di un avvenimento rimasto per molto tempo nella memoria del mercato:

il fallimento della tedesca Bankhaus Herstatt.

Questo fatto determinò sin dall'inizio la caratterizzazione dei lavori del Comitato, interessato al funzionamento e alla stabilità del sistema finanziario globale. Il Comitato è formato dai soggetti rappresentanti le banche centrali e le autorità che hanno responsabilità sulla regolamentazione di vigilanza per il business bancario. Il Comitato non legifera, formula proposte e linee guida, tuttavia le prime proposte, risalenti al 1988, sono diventate normativa vincolante in oltre 100 paesi. Da qui l'importanza del Comitato, i cui lavori intendono perseguire due fondamentali obiettivi:

- 1) estendere la regolamentazione a tutte le istituzioni bancarie
- 2) rendere sempre più efficace la regolamentazione di vigilanza bancaria.

Un importante obiettivo del Comitato è stato colmare le lacune presenti nella copertura della vigilanza internazionale nel rispetto di due principi base:

1. nessuna impresa bancaria estera dovrebbe evitare la vigilanza;
2. la vigilanza dovrebbe essere adeguata.

Nel 1988 viene proposto un sistema di misurazione dell'adeguatezza patrimoniale, noto come Accordo di Basilea (o Basilea1). Successivamente, nel giugno del 1999 il Comitato propose una nuova struttura per l'Accordo, che portò, nell'aprile 2003, alla pubblicazione di un documento finale di consultazione. Quest'ultimo documento introduce la struttura del nuovo accordo di adeguatezza patrimoniale

(anche noto come *Basilea 2*), che dovrà essere formalmente pubblicato entro la fine del 2006.

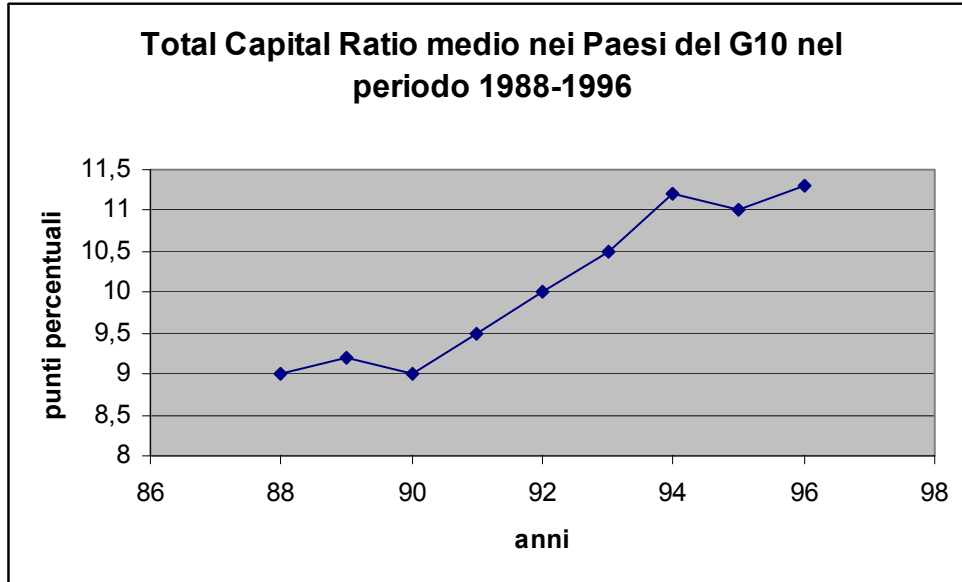
1.3.1 IL PRIMO ACCORDO DI BASILEA

Il sistema dei requisiti patrimoniali è di semplice formulazione. Ogni attività posta in essere da un'impresa finanziaria comporta l'assunzione di un certo grado di rischio. Il rischio deve essere quantificato e supportato da capitale.

La norma impone di determinare il cosiddetto "attivo ponderato" che si ottiene moltiplicando il valore dell'attività (il prestito) per un coefficiente normativamente stabilito. Da questo si ricorre ad una percentuale fissa, l'8% del valore ponderato che è il rischio.

Benché strettamente collegato al rischio di credito, il coefficiente minimo dell'8% fra patrimonio di vigilanza e attivo ponderato in base al rischio veniva ritenuto sufficiente ad offrire una copertura implicita anche nei confronti di altri rischi bancari: di mercato, operativo, di liquidità. Un effetto è stato ottenuto sicuramente, così come si vede dal grafico1: l'aumento del livello complessivo di patrimonializzazione del sistema bancario.

Grafico1: Aumento del grado di patrimonializzazione del sistema bancario dopo l'Accordo del 1988 sul Capitale



1.3.2 IL NUOVO ACCORDO: BASILEA2

Il comitato di Basilea ha deciso di effettuare, nel 1999, un secondo giro di consultazioni per una proposta più dettagliata sulla

regolamentazione dell'adeguatezza patrimoniale che nella versione finale sostituirà l'accordo di Basilea1.

Il nuovo accordo di Basilea sui requisiti patrimoniali presenta un'estensione e una complessità maggiori rispetto all'accordo del 1988. Questo a testimonianza degli sforzi per elaborare una regolamentazione sensibile al rischio, che contempla una gamma di nuove opzioni per la misurazione sia del rischio di credito, sia del rischio operativo.

Qui di seguito poniamo la premessa che lo stesso comitato ha fatto alla nuova versione dei requisiti patrimoniali³:

“Il Comitato ritiene che il miglioramento dello schema di adeguatezza patrimoniale possa recare importanti benefici di ordine generale lungo due direttrici principali. In primo luogo, elaborando una regolamentazione del capitale che ricomprende non solo i requisiti patrimoniali minimi, ma anche il controllo prudenziale e la disciplina di mercato. In secondo luogo, accrescendo significativamente la sensibilità al rischio dei coefficienti patrimoniali minimi”

“Il nuovo schema di adeguatezza patrimoniale è volto a conferire maggiore rilevanza alla gestione del rischio da parte delle banche. [...]”

Il Comitato individua in maniera esplicita nelle tecniche di *Risk Management* e nella loro efficace applicazione all'interno dell'organizzazione aziendale, il punto di contatto tra rispetto delle esigenze di stabilità del sistema e opportunità di efficienza degli intermediari finanziari e bancari.

Un postulato fondamentale afferma che i requisiti della sicurezza e della solidità non possono essere conseguiti esclusivamente attraverso i requisiti patrimoniali minimi.

Con riferimento a questa osservazione, infatti, il nuovo Accordo consiste di tre parti, denominati Pilastri, che si rafforzano reciprocamente:

³ Punti 2 e 3 del documento “Presentazione del Nuovo Accordo di Basilea sui requisiti patrimoniali”, rilasciato dal Comitato di Basilea per la vigilanza bancaria di Aprile 2003

1. *requisiti patrimoniali minimi;*
2. *controllo prudenziale dell'adeguatezza patrimoniale;*
3. *disciplina di mercato*

Primo Pilastro: Requisiti Patrimoniali Minimi. Per quanto riguarda il *Primo Pilastro*, nella nuova formulazione dell'Accordo, le regole che definiscono il patrimonio a fini di vigilanza restano invariate, ovvero non viene modificato il coefficiente minimo richiesto dell'8%, viene però ampliata il numero di categorie di rischio da calcolare. Si deve infatti determinare l'impatto dei rischi operativo, oltre a quelli di credito e di mercato:

$$\frac{\text{Patrimonio di vigilanza}}{(\text{rischio di mercato})+(\text{rischio di credito})+(\text{rischio operativo})} > 8\%$$

I cambiamenti intervengono in ciò che attiene alla definizione di attività ponderate per il rischio, ovvero nelle metodologie impiegate per misurare i rischi in cui incorrono le banche. I nuovi metodi per il calcolo delle attività ponderate sono volti a migliorare la valutazione della rischiosità da parte delle istituzioni bancarie e, pertanto, a rendere più significativi i coefficienti patrimoniali che da quella derivano.

Nella definizione di attività ponderate l'Accordo attuale copre in maniera esplicita due sole tipologie di rischio: il rischio di credito e il rischio di mercato.

Con il nuovo Accordo invece vi saranno: **mutamenti sostanziali al trattamento del rischio di credito** previsto dall'Accordo attuale e **l'introduzione di un esplicito trattamento del rischio operativo**.

Il Comitato, sia per il rischio di credito che per quello operativo prevede tre metodi con crescente sensibilità al rischio proprio per consentire a banche e autorità di vigilanza di scegliere quello o quelli

ritenuti più appropriati allo stadio di sviluppo dell'operatività bancaria e dell'infrastruttura di mercato.

Rischio di credito	Rischio operativo
1) <i>Metodo standard</i>	1) <i>Metodo dell'indicazione semplice</i> (“basic indicator approach”)
2) <i>Metodo IRB di base</i>	2) <i>Metodo standard</i>
3) <i>Metodo IRB avanzato</i>	3) <i>Metodi avanzati di misurazione</i> (“Advanced Measurement Approach” – AMA)

Le banche sono esortate a evolvere verso gli approcci più complessi in coerenza con lo sviluppo di sistemi e pratiche per la misurazione del rischio operativo più sofisticati.

Ci si attende che le banche con operatività internazionale e quelle con una significativa esposizione al rischio operativo o di credito non usino gli approcci di livello più basso, bensì approcci più avanzati. Tuttavia alle banche non sarà consentito utilizzare approcci più semplici una volta che siano state giudicate idonee all'uso di approcci avanzati.

Secondo Pilastro: Controllo Prudenziale. Il comitato considera il controllo prudenziale un complemento essenziale dei requisiti patrimoniali minimi e della disciplina di mercato. Il secondo pilastro della nuova regolamentazione mira ad assicurare che in ogni banca operino adeguati procedimenti interni per valutare l'adeguatezza del proprio patrimonio sulla base di una misurazione accurata dei rischi cui essa è esposta. Le autorità di vigilanza saranno chiamate ad accertare se le banche valutano in modo corretto la propria adeguatezza patrimoniale

in rapporto ai rischi, tenendo conto delle correlazioni esistenti tra le diverse tipologie di rischio.

Terzo Pilastro: Disciplina di Mercato. Scopo del *Terzo* Pilastro è quello di integrare i requisiti patrimoniali minimi stabiliti nel primo pilastro e il processo di controllo prudenziale affrontato dal secondo.

Il Comitato si è adoperato per incoraggiare la disciplina di mercato mediante l'elaborazione di una serie di obblighi di trasparenza che consentano agli operatori di valutare le informazioni cruciali sul profilo di rischio e sui livelli di capitalizzazione di una banca, in modo da assicurare un'adeguata informazione al pubblico sui livelli patrimoniali, sull'esposizione al rischio sulla loro gestione.

1.4 I RISCHI OPERATIVI IN BASILEA 2

“Il rischio operativo è definibile come il rischio di perdite derivanti da disfunzioni a livello di procedure, di personale e sistemi interni, oppure

da eventi esogeni. Tale definizione include il rischio giuridico, ma non quelli strategico e di reputazione”⁴

Così il Comitato di Basilea definisce quella categoria di rischi divenuta oggetto di studio negli ultimi anni. Ma da cosa nasce l'esigenza di controllo e valutazione di questa classe di rischi?

L'esistenza di rischi derivanti da errori manuali, dal cattivo funzionamento dei sistemi informativi o dell'esecuzione delle operazioni connesse allo svolgimento dell'attività bancaria è nota da tempo. L'attenzione è tuttavia accresciuta nel periodo più recente, sia perché tale categoria di rischio ha acquisito un contenuto più ampio, sia perché il contesto esterno e le caratteristiche stesse dell'attività bancaria si sono modificati nel tempo.

In altri termini, fenomeni come

- la crescente globalizzazione dei mercati,
 - la più elevata competitività,
 - lo sviluppo delle tecnologie,
 - l'estrema varietà dell'offerta,
 - l'inevitabile maggiore articolazione degli assetti organizzativi
- hanno straordinariamente aggravato l'esposizione delle banche verso perdite originate da malfunzionamenti o da carenze di uno o più aspetti delle procedure aziendali. Non si tratta tanto di nuove tipologie di rischio quanto di una diversa modalità di manifestazione di rischi già tradizionalmente noti agli intermediari.

Il Comitato di Basilea iniziò nel 1998 a sollevare il tema dei rischi operativi e infatti nel gennaio del 2001, con il nuovo Accordo di Basilea, la regolamentazione del capitale minimo che le banche devono detenere a fronte del rischio include anche i rischi operativi.

⁴ Nuovo Accordo di Basilea sui Requisiti Patrimoniali, documento a fini di consultazione. Aprile 2003.

A proposito di tali rischi si è dibattuto molto su una loro precisa definizione, il che ha accentuato le problematiche connesse con una corretta misurazione e un attento controllo degli stessi.

Per quanto riguarda le metodologie di misurazione, diversamente da quanto avvenuto in passato per i rischi di mercato e di credito, i rischi operativi risultano difficilmente riportabili a uno schema teorico-statistico prontamente utilizzabile. La loro misurazione, infatti, è fortemente condizionata dalla mancanza di dati sulle perdite subite.

A livello internazionale è partita da alcuni anni un'attività di implementazione di database delle perdite operative. Anche a livello nazionale l' **ABI** ha sviluppato una propria metodologia di valutazione dei rischi operativi che viene indicata con l'acronimo *A.P.R.O.* (*Approccio Rischio Operativo*). Ad ogni modo, nel corso degli anni si sono venute affermando varie famiglie di approcci metodologici tra loro differenti. Lo stesso Comitato di Basilea ha proposto una gamma di opzioni per il trattamento prudenziale per il rischio operativo e più precisamente si prevede un trattamento articolato in **tre metodologie con livelli crescenti di complessità**:

- **Indicatore semplice** (“basic indicator approach”);
- **Metodo standard**;
- **Metodi avanzati di misurazione i.** (“Advanced Measurement Approach” – AMA)

Le banche sono incoraggiate a procedere lungo lo spettro delle metodologie disponibili man mano che elaborano più sofisticati sistemi e prassi di misurazione del rischio operativo. I requisiti per l'ammissione ai metodi standard e avanzati sono:

- Le banche attive a livello internazionale e quelle con significative esposizioni al rischio operativo possono utilizzare il metodo più appropriato al loro *grado di rischio*. Le autorità di vigilanza

esamineranno i coefficienti patrimoniali ottenuti mediante l'approccio scelto dalla banca per valutarne la credibilità e la bontà; in caso di scarsa credibilità, le autorità valuteranno le opportune iniziative da intraprendere nell'ottica del secondo pilastro. Ad una banca, inoltre, verrà consentito di utilizzare un approccio per alcuni settori ed un altro approccio differente per altri settori.

- Solo con l'autorizzazione dell'organo di vigilanza, le banche possono decidere, nel caso stessero utilizzando un approccio più complesso, di tornare da un approccio più semplice.
- L'autorità di vigilanza può richiedere alla banca, ove lo ritenesse necessario, di passare, per una o più operazioni, da un metodo più complesso ad uno più semplice, finché questa non ripristini le condizioni regolamentari per il ritorno ad un metodo più avanzato.

1.4.1 “Basic Indicator Approach” o Metodo dell'Indicatore Semplice

Tra le tre metodologie di calcolo del requisito patrimoniale a fronte del rischio operativo proposte nel nuovo Accordo, questa risulta sicuramente la più semplice. Viene richiesto alle banche di detenere una dotazione di capitale pari ad una percentuale fissa del reddito lordo medio annuo riferito ai tre esercizi precedenti. L'onere per la banca può essere espresso come segue:

$$K_{BIA} = GI \cdot \alpha$$

dove:

- K_{BIA} = requisito patrimoniale definito nel “basic indicator approach”.
- GI = reddito lordo medio annuo riferito ai tre esercizi precedenti; tale reddito è definito come (reddito netto da interessi + reddito netto non da interessi): in questo modo tale reddito dovrebbe essere al lordo di ogni accantonamento, esclude i profitti o le perdite realizzate sulla vendita di titoli del “banking book”, ed infine esclude le partite

straordinarie, gli errori di omissioni, nonché i provenienti derivanti da assicurazioni.

- $\alpha = 15\%$ (percentuale stabilita dal Comitato).

Per ciò che concerne il calcolo del patrimonio di vigilanza, il nuovo Accordo non stabilisce criteri specifici per l'impiego di questo indicatore.

1.4.2 “Standardised Approach” o Metodo Standard

Questo approccio suddivide l'attività della banca in *otto linee operative*, all'interno di ciascuna delle quali il reddito lordo costituisce un indicatore della probabile scala dimensionale dell'esposizione al rischio operativo.

Classificazione delle linee operative		
Livello 1	Livello 2	Gruppi di attività
Corporate Finance	Corporate finance	Fusioni e acquisizioni (M&A), sottoscrizioni a fermo, privatizzazioni, cartolarizzazione, ricerca, obbligazioni (debito pubblico, alto rendimento), azioni, sindacazione di presidi, offerte pubbliche iniziali, collocamenti privati sul mercato secondario
	Finanza governo/enti locali	
	Merchant Banking	
	Servizi di consulenza	
Negoziazioni e vendite	Vendite	Reddito fisso, azioni, valute, merci, credito, finanziamento, compravendita titoli in proprio, prestiti e PcT, intermediazione, debito, prime brokerage
	Market-making	
	Attività in proprio	
	Tesoreria	
Retail Banking	Retail Banking	Prestiti e depositi, servizi bancari, gestioni fiduciarie e immobiliari
	Private Banking	Prestiti e depositi, servizi bancari, gestioni fiduciarie e immobiliari, consulenza agli investimenti
	Carte di credito	Personalì, commerciali e aziendali, marchi commerciali e retail
Commercial Banking	Commercial Banking	Project finance, mutui immobiliari, credito all'esportazione, credito al

		commercio, factoring, leasing, prestiti, garanzie, titoli cambiari
Pagamenti e regolamenti	Clientela esterna	Pagamenti e incassi, trasferimento fondi, compensazione e regolamento
Gestioni fiduciarie	Custodia	Conti/depositi presso terzi, ricevute di deposito, prestito titoli (clientela), atti societari
	Rappresentanza	Mandati di emissione a pagamento
	Persone giuridiche	
Asset management	Gestione discrezionale fondi	Gestione cumulativa, separata, al dettaglio, istituzionale, fondi chiusi e aperti, "private equity"
	Gestione non discrezionale fondi	Gestione cumulativa, separata, al dettaglio, fondi chiusi e aperti
Intermediazioni e al dettaglio	Intermediazione al dettaglio	Esecuzione degli ordini, "full service" (consulenza ed assistenza agli investimenti, ecc.)

Il requisito patrimoniale per ciascuna linea operativa è calcolato moltiplicando il reddito lordo per un fattore β attribuito a quella linea; tale fattore rappresenta il rapporto tra le perdite per rischi operativi storicamente riscontrate e il livello aggregato di reddito lordo per ogni singola linea. Il coefficiente patrimoniale totale è calcolato come semplice sommatoria dei coefficienti di ciascuna linea. Esso può essere espresso come:

$$K_{STA} = \sum_{i=1}^8 (GI_i \cdot \beta_i)$$

dove:

- K_{STA} = coefficiente patrimoniale nel quadro del metodo standard;
- GI_i = livello medio annuo del reddito lordo dei tre esercizi precedenti (definito come per il "basic indicator approach") per ciascuna delle otto linee operative;
- β_i = percentuale fissa, stabilita dal Comitato, che rapporta il livello di capitale richiesto a quello del reddito lordo per ciascuna delle otto linee. I valori di β sono riportati di seguito.

Linea operativa	Fattori β
Corporate Finance (β_1)	18%
Negoziazione e vendite (β_2)	18%
Retail Banking (β_3)	12%
Commercial Banking (β_4)	15%
Pagamenti e regolamenti (β_5)	18%
Gestioni fiduciarie (β_6)	15%
Asset management (β_7)	12%
Intermediazione al dettaglio (β_8)	12%

1.4.3 “Advanced Measurement Approaches” o Metodi Avanzati di Misurazione (AMA)

Si tratta di un modello interno della banca per la misurazione del rischio operativo, soggetto all’approvazione dell’organo di vigilanza. Per essere ammessa all’uso dei metodi standard o avanzati, una banca deve almeno assicurare che:

- il consiglio di amministrazione e l’alta direzione, a seconda dei casi, siano attivamente coinvolti nella supervisione del sistema di gestione del rischio operativo;
- si dispone di un sistema di gestione del rischio che sia concettualmente solido e applicato nella sua interezza;

- si dedichi sufficienti risorse all'applicazione del metodo prescelto nelle principali linee operative, nonché alle aree di controllo e di audit.

Presupposto dell'AMA è il ricorso ad una base di dati, riferita sia alle perdite rilevate sia alla loro frequenza. Il periodo di osservazione dei dati interni di perdita deve essere di almeno 5 anni; se la banca adotta un AMA per la prima volta il periodo di osservazione può essere ridotto a tre anni.

Le attività della banca sono classificate, come per il metodo standard, per *business lines o linee operative*; si è definito, inoltre, un ampio insieme di tipologie di *eventi di perdita* da applicare a tutte le aree operative.

Alla luce della continua evoluzione degli approcci analitici al rischio operativo, il Comitato non fornisce specificazioni riguardo al metodo delle ipotesi distributive impegnato per generare le misure del rischio operativo a fini di calcolo dei requisiti patrimoniali. Tuttavia, una banca deve essere in grado di dimostrare che l'approccio seguito sia in grado di cogliere potenziali gravi eventi di perdita nella coda della distribuzione (*tail loss event*). Quale che sia il metodo usato, essa deve provare che la sua misura del rischio soddisfa uno standard di validità comparabile a quello offerto dal metodo IRB per il rischio di credito.

1.5 COME NASCE IL PROGETTO SUI RISCHI OPERATIVI

Come abbiamo detto precedentemente l'utilizzo delle metodologie AMA presuppone un periodo di osservazione dei dati interni dei rischi operativi di almeno 5 anni; il periodo di osservazione può essere ridotto a tre anni se la banca adotta un AMA per la prima volta. Inoltre tale banca deve disporre di un sistema indipendente di gestione del rischio operativo tale che:

- Codifichi le politiche e le procedure, a livello di intera azienda, riguardanti la gestione ed il controllo del rischio operativo;
- Progetti ed attui metodologie aziendali di valutazione del rischio operativo e dei connessi sistemi di segnalazione;
- Elabori strategie volte ad individuare, stimare, monitorare e controllare il rischio operativo.

Proprio per lo sviluppo di questi punti è iniziata una collaborazione tra **Banca IMI** (il Cliente) ed **Engineering Ingegneria Informatica** (il Fornitore), con lo scopo di realizzare un sistema integrato per l'analisi e la gestione qualitativa-quantitativa dei rischi operativi, in risposta ai requisiti di Basilea 2.

1.5.1 Engineering Ingegneria Informatica

Engineering Ingegneria Informatica è una società appartenente al Gruppo Engineering, leader sul mercato nazionale nella *system e business integration* e nei servizi di *application management* alle imprese, Engineering interviene sullo sviluppo di nuove logiche di business integrando le tecnologie tradizionali e le nuove frontiere di Internet con le singole strutture aziendali.

Core business del Gruppo è la realizzazione di progetti e di architetture innovative ad elevato contenuto tecnologico, ma sono i

servizi di *application management* la componente più dinamica delle attività: tali servizi rappresentano più del 30% del fatturato globale.

1.5.2 Progetto per la Gestione Integrata dei Rischi Operativi

Il Progetto che Engineering Ingegneria Informatica S.p.A realizzerà per conto di Banca IMI consiste nella realizzazione di un sistema integrato per la gestione qualitativa-quantitativa dei rischi operativi, con le seguenti finalità:

- Garantire un livello di dettaglio adeguato per la rilevazione qualitativa dei rischi che consenta un'analisi completa ed esaustiva degli stessi.
- Creazione di una biblioteca elettronica dei rischi operativi. Si è utilizzato ARIS, un tool di modellazione e analisi organizzativa prodotto dalla casa tedesca IDS-Sheer;
- Garantire rapidità nel reperimento delle informazioni e completezza delle stesse. Tali informazioni saranno strutturate in ARIS in base al modello quantitativo previsto dagli strumenti di calcolo e di analisi;
- Rendere facile ed integrata la gestione dei dati in termini di concentrazione e classificazione delle informazioni ed in termini di riutilizzo da parte degli strumenti di analisi quantitativa;
- Creazione del Data Base delle perdite aziendali ed integrazione con standard e strumenti messi a disposizione dalla banca stessa;
- Implementazione e gestione di strumenti per la misura gestionale e regolamentare del Rischio Operativo.

1.5.3 COME VENGONO TRATTATI I RISCHI OPERATIVI

Abbiamo visto che il Comitato ha suggerito delle linee guida da seguire per il monitoraggio di questi rischi, ma ha lasciato ampia libertà di movimento alle banche sulle tecniche e sulle metodologie da usare.

In questo modo ogni società bancaria ha potuto cercare un modo il più ottimale e valido possibile per effettuare le sue stime ed i suoi studi.

Il problema si è rivelato molto complesso per questo tipo di rischio rispetto agli altri, visto che, a differenza dei rischi di credito o di mercato, per i rischi operativi:

- non è così facile capire che struttura vanno ad intaccare e con che peso;
- non ci sono molti dati storici su cui effettuare le stime dei parametri.

Quello di cui si ha bisogno è un modello che sappia cogliere tutti gli aspetti caratteristici dei rischi operativi, ovvero che sappia cogliere:

- le diverse attività che compongono i processi bancari;
- come i processi e le attività bancarie sono tra loro collegate;
- le diverse classi di rischio operativo;
- relazioni tra categorie di rischio e rischi;
- relazioni tra attività e output;
- relazioni tra attività e rischi;

L'utilizzo degli *Alberi di Guasto* e della *Fault Tree Analysis* si è rivelato essere il metodo più efficace per cogliere queste aspetti.

1.6 GLI ALBERI DI GUASTO E LA FAULT TREE ANALYSIS (FTA)

Come abbiamo detto, uno dei metodo che si è dimostrato più adatto per lo studio dei rischi è la *Fault Tree Analysis* (FTA), un metodo di analisi che permette di

- effettuare delle **predizioni**: studiando gli stati delle componenti di un sistema arriva ad ottenere informazioni sullo stato dell'intero sistema.
- effettuare un'**analisi diagnostica**: partendo da un analisi "generale" e complessiva dello stato dell'intero sistema, arriva ad individuarne gli stati sui suoi componenti

Una FTA è condotta per scoprire tutte le cause sottostanti di un particolare fallimento/guasto, in modo da poter identificare ed eliminare, se possibile, quelle cause che ne stanno alla radice. E' uno strumento che utilizza grafici e statistiche per analizzare un evento e permette di predire come, e quanto spesso, esso fallirà.

I motivi principali per cui si utilizza la Fault tree Analysis sono che:

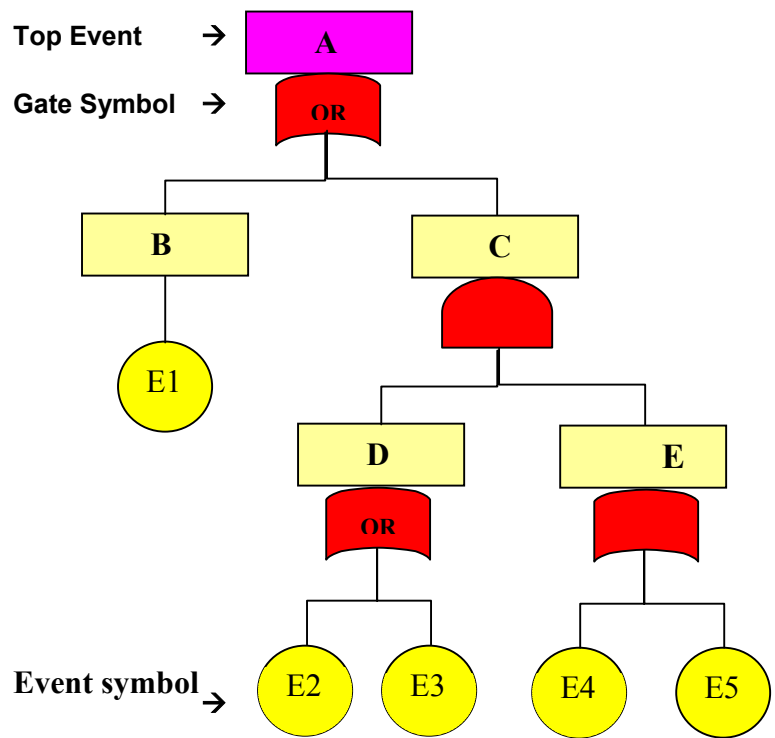
- È molto facile e immediata da capire;
- È un modo effettivo e concreto per rappresentare i problemi sottostanti ad un determinato sistema;
- Aiuta ad organizzare le possibili cause di un problema in un sistema;

La FTA effettua il suo studio con l'utilizzo dell'**Albero di Guasto** (*Fault Tree* o FT) che consiste in un grafo, ottenuto con un approccio *top-down*, e che rappresenta un sistema: si tratta di una struttura gerarchica che mostra le relazioni tra un *evento potenziale* che influisce sulle performance del sistema, e le *ragioni* o le *cause sottostanti* di quell'evento. Più semplicemente un FT illustra lo stato del sistema (**Top Event**) in base agli stati (funzionamento o guasto) delle componenti del

sistema (*Basic Event*). Le connessioni tra questi elementi sono rappresentate tramite simboli logici detti *gate*, dove l'output di un *gate* è determinato dagli input che a lui arrivano.

Di seguito abbiamo un esempio di albero do Guasto.

ESEMPIO DI UN ALBERO DI GUASTO



CAPITOLO 2

ORDINAMENTO E ALBERI

Il capitolo precedente ci è servito per inquadrare l'ambito all'interno del quale opera l'algoritmo da noi sviluppato. Questo secondo capitolo serve invece come introduzione teorica di quei concetti più strettamente legati all'algoritmo che è alla base del nostro lavoro. Quindi, poiché il progetto che è stato sviluppato consiste in un algoritmo che ordina un qualsiasi albero di guasto dato il nome del nodo e del padre, è necessario fare una breve introduzione teorica ai concetti di:

- **relazioni d'ordine**
- **relazioni d'equivalenza**
- **Alberi e ordinamento**

- **Visite in profondità**
 - **Visite in preordine**
 - **Visite in ordine simmetrico**
 - **Visite in post ordine**
- **Visita in ampiezza**

2.1 CORRISPONDENZE E RELAZIONI

Definizione 1.1: Il concetto di **corrispondenza**:

Dati A, B due insiemi. Si dice corrispondenza tra A e B un qualsiasi sottoinsieme R del prodotto cartesiano $A \times B$. Quindi elementi di due insiemi corrispondono quando è data una regola che stabilisce con certezza quando essi sono associati.

Per esempio: Si chiama corrispondenza fra l'insieme A e l'insieme B la terna

$$\Gamma = (G, A, B)$$

Dove G è un grafico. Si dice che G è il grafico di Γ , A l'insieme di partenza e B l'insieme d'arrivo.

Esempio:

A = l'insieme degli attori italiani

B = l'insieme delle attrici americane

R = "x ha avuto una relazione con z"

G è contenuto in $A \times B$ e (G, A, B) è una corrispondenza fra A e B .

2.2 LE RELAZIONI IN UN INSIEME

Se una corrispondenza ha come insieme di partenza e d'arrivo lo stesso insieme A , si chiama *Relazione binaria nell'insieme A* o semplicemente relazione in A e si scrive:

$$\Pi = (R, A, A) = (R, A)$$

Dati due elementi $x, y \in A$, si dice che x ed y sono in *relazione* fra loro secondo Π se $(x, y) \in R$.

Se $(x, y) \notin R$ allora x ed y non sono in relazione fra loro secondo Π

Una relazione fra x e y può scrivere in diversi ma equivalenti modi:

$(x, y) \in R$, xRy , xry , x è in relazione con y

In sostanza una relazione in un insieme è data quando c'è una regola che stabilisce con certezza quali fra gli elementi dell'insieme vanno considerati nella relazione data e quali no.

Abbiamo visto quindi che le relazioni sono un caso particolare delle corrispondenze nelle quali l'insieme di partenza e d'arrivo è lo stesso.

Dato per buono il concetto di relazione passiamo ora a presentare brevemente i vari tipi di relazioni esistenti.

Definiamo quindi delle variabili che useremo nelle definizioni dei diversi tipi di relazioni.

Siano A un insieme, $\Pi = (R, A)$ una relazione in A e $x, y, z \in A$.

2.2.1 Definizione: Proprietà Riflessiva

Quando $\forall x \in A$ si ha che $(x,x) \in R$

Esempio 1: fra persone, la relazione “x è indistinguibile da y” è riflessiva.

Esempio 2: Immaginiamo una relazione R è definita nell'insieme $A = \{2, 3, 4, 5, 6, 8\}$ nel seguente modo

$$R = \{ (x, y) \mid x + y \text{ è pari} \}$$

La relazione è l'insieme formato dalle seguenti coppie:

$$R = \{ (2,2), (2,4), (2,6), (2,8), (3,3), (3,5), (4,2), (4,4), (4,6), (4,8), (5,3), (5,5), (6,2), (6,4), (6,6), (6,8), (8,2), (8,4), (8,6), (8,8) \}$$

E' riflessiva perché ogni elemento è in relazione con sé stesso.

Nella relazione sono infatti presenti le coppie (2,2), (3,3), (4,4), (5,5), (6,6), (8,8)

2.2.2 Definizione: Proprietà simmetrica

Quando $\forall x, y \in A : (x,y) \in R \Rightarrow (y,x) \in R$

Esempio 1: Prendiamo un esempio fra persone: la relazione « x è partner di y » è simmetrica.

Esempio 2: La relazione R è definita nell'insieme $A = \{2,3,4,5,6,8\}$ nel seguente modo

$$R = \{ (x, y) \mid x + y \text{ è pari} \}$$

La relazione è l'insieme formato dalle seguenti coppie:

$$R = \{(2,2), (2,4), (2,6), (2,8), (3,3), (3,5), (4,2), (4,4), (4,6), (4,8), (5,3), (5,5), (6,2), (6,4), (6,6), (6,8), (8,2), (8,4), (8,6), (8,8)\}$$

La relazione è simmetrica in conseguenza della proprietà commutativa dell'addizione. Infatti

$$\text{se } x+y \text{ è pari} \Rightarrow y+x \text{ è pari}$$

Controesempio 1: La relazione R è definita nell'insieme $A = \{2,3,4,5,6,8\}$ nel seguente modo

$$R = \{(x,y) \mid x \text{ è divisore di } y\}$$

La relazione è l'insieme formato dalle seguenti coppie:

$$R = \{(2,2), (2,4), (2,6), (2,8), (3,3), (3,6), (4,4), (4,8), (5,5), (6,6), (8,8)\}$$

La relazione è riflessiva perché ciascun elemento è in relazione con sé stesso.

Non è simmetrica: perché non è vero che per ogni $x, y \in A$ se x è *divisore di* $y \Rightarrow$ non è y *divisore di* x

2.2.3 Definizione: Proprietà antisimmetrica

Si dice relazione antisimmetrica quando

$$\forall x, y \in A \text{ tale che } (x,y) \in R \text{ e } (y,x) \in R \Rightarrow (x = y)$$

ovvero, se $\forall x, y \in A$ tale che x è in relazione con y e y è in relazione con x allora $x = y$

Esempio: Immaginiamo una relazione R definita in un insieme $A = \{2,3,4,5,6,8\}$ nel seguente modo:

$$R = \{(x,y) \mid x \text{ è divisore di } y\}$$

In questo caso la relazione sarà formata dalle seguenti coppie:

$$R = \{(2,2), (2,4), (2,6), (2,8), (3,3), (3,6), (4,4), (4,8), (5,5), (6,6), (8,8)\}$$

E' antisimmetrica infatti presa una qualunque coppia (x,y) della relazione, la sua simmetrica (y,x) non appartiene alla relazione, tranne i casi nei quali $x=y$

Controesempio:

Nell'insieme degli alunni di una classe considera la seguente relazione:

$$R = \{(x,y) \mid x \text{ è innamorato di } y\}$$

Questa relazione non è simmetrica infatti vi saranno senz'altro persone innamorate e non corrisposte (x è in relazione con y e non (y è in relazione con x)). Ma non è neppure antisimmetrica infatti non si può dire che se vi sono degli innamorati corrisposti (xry e yrx), allora $x = y$, quindi sono fra la stessa persona(!).

2.2.4 Proprietà transitiva.

Una relazione definita in un insieme A si dice transitiva se, presi comunque tre elementi x, y, z e tali che sia xry e $y rz$, allora è anche $x rz$.

$$\forall x,y,z \in A : ((x,y) \in R) \text{ e } ((y,z) \in R) \Rightarrow ((x, z) \in R)$$

Esempio 1: fra persone, “ x è superiore gerarchico di y ” è transitiva.

Esempio 2 La relazione R è definita nell'insieme $A=\{2,3,4,5,6,8\}$ nel seguente modo

$$R=\{(x,y) \mid x < y \}$$

La relazione è l'insieme formato dalle seguenti coppie:

$$R=\{(2,3),(2,4),(2,5),(2,6),(2,8),(3,4),(3,5),(3,6),(3,8),(4,5),(4,6),(4,8), \\ (5,6),(5,8),(6,8)\}$$

La relazione è transitiva come si verifica facilmente: per ogni terna dell'insieme se $a < b$ e $b < c$ allora $a < c$ per le proprietà della disuguaglianza.

Esempio 3: "x è parallela ad y" nell'insieme delle rette del piano

Esempio 4: "x è della stessa classe y" nell'insieme degli alunni della tua scuola

Controesempio 1: "x è perpendicolare a di y" nell'insieme delle rette del piano: infatti non è riflessiva, né transitiva

Controesempio 2: "x è figlio di y" nell'insieme delle persone: infatti non è riflessiva, né simmetrica né transitiva

2.3 RELAZIONI

Dopo aver compreso che cosa sia una relazione e quali siano le sue principali proprietà, possiamo proseguire il nostro percorso con la definizione di **relazioni di equivalenza** e **relazioni d'ordine**, per spiegare le quali avremo bisogno delle proprietà enunciate nei paragrafo

i precedenti.

2.3.1: RELAZIONE D'EQUIVALENZA

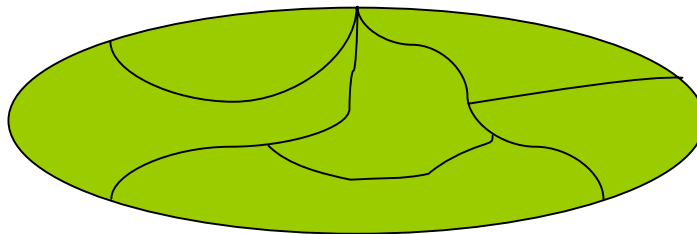
Definizione:

Una relazione in A si dice *relazione d'equivalenza* oppure *un'equivalenza*, se è riflessiva, simmetrica e transitiva.

Esempio1:

Consideriamo l'insieme delle piante di un certo ambiente. I botanici hanno stabilito dei criteri che ci consentono di affermare se due piante sono della stessa specie. Con il predicato a due posti "...è della stessa specie di..." e considerando i criteri, riusciamo ad effettuare una divisione (partizione) dell'insieme P delle piante di quell'ambiente in vari sottoinsiemi, ciascuno con le piante di una stessa specie.

P



Esempio2:

Consideriamo ora l'insieme N dei numeri naturali e formiamo in esso una partizione in questo modo:

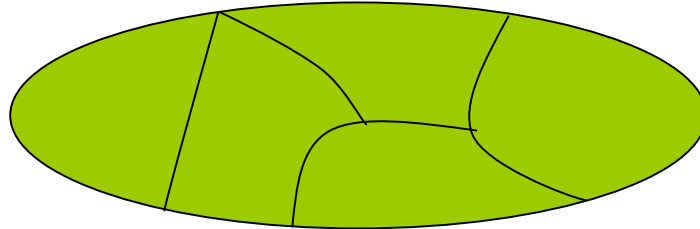
$$I_1 = \{0, 5, 10, 15, 20, \dots\}$$

$$I_2 = \{1, 6, 11, 16, 21, \dots\}$$

$$I_3 = \{2, 7, 12, 17, 22, \dots\}$$

$$I_4 = \{3, 8, 13, 18, 23, \dots\}$$

$$I_5 = \{4, 9, 14, 19, 24, \dots\}$$



Quale criterio abbiamo seguito per eseguire la partizione?

Problemi di questo tipo sono importanti perché da una situazione, in questo caso una partizione dell'insieme dei numeri naturali \mathbb{N} , ci consentono di trovare il **criterio che l'ha determinata**.

In quest' esempio, nel primo insieme abbiamo messo i numeri che divisi per 5 danno resto 0, nel secondo quelli che divisi per 5 danno resto 1, nel terzo quelli che danno resto 2, nel quarto quelli che danno resto 3, nel quinto quelli che danno resto 4.

Questi sono due esempi di relazioni di equivalenza, nel cui predicato a due posti o relazione in genere compare la parola "stesso" ("stessa specie", "stesso resto").

Queste relazioni godono della proprietà transitiva, infatti se lo 0 è nella stessa classe del 10 e il 10 è nella stessa classe del 20, possiamo dire che lo 0 è nella stessa classe del 20.

Esse godono della proprietà simmetrica, infatti se lo 0 è nella stessa classe del 10 allora il 10 è nella stessa classe dello 0.

Infine godono della proprietà riflessiva perché ogni elemento è nella propria classe essendo in relazione con se stesso.

2.3.2.RELAZIONE D'ORDINE

Le relazioni d'ordine sono relazioni binarie che ci consentono di "ordinare" gli elementi di un certo insieme.

Ad esempio se si vogliono mettere in fila un insieme di oggetti in base alla loro lunghezza oppure delle persone in base alla loro statura, la relazione è $R_1 = \{x, y \mid x \text{ è meno lungo di } y\}$ oppure " x è meno alto di y " avremo un **ordine crescente**.

Se invece scegliamo la relazione $R_2 = \{x, y \mid x \text{ è più lungo di } y\}$ oppure " x è più alto di y " avremo un **ordine decrescente**.

Consideriamo l'insieme A degli oggetti e la relazione R_1 . In pratica confrontiamo gli oggetti a due a due in base alla caratteristica "lunghezza", più precisamente il confronto viene effettuato non sugli oggetti in sé ma sulle loro misure.

Una relazione R affinché possa definirsi relazione d'ordine occorre che ammetta le proprietà transitiva, antisimmetrica e riflessiva, quindi non è una relazione d'equivalenza perché manca la proprietà simmetrica, anzi **le relazioni d'ordine sono legate strettamente da una asimmetria** (si può andare dal più piccolo al più grande o viceversa).

Definizione: relazione d'ordine

Una relazione in A si dice **relazione d'ordine**, oppure un ordine, se è riflessiva, antisimmetrica e transitiva

Esempio: Sia $A = \mathbb{R}$.

Dati due qualsiasi elementi di \mathbb{R} , la relazione " x è minore o uguale di y " è una relazione d'ordine in \mathbb{R} .

$$R = \{(x, y) \mid x \text{ è minore o uguale di } y\}$$

E' riflessiva perché qualunque numero è minore o uguale a se stesso.

E' antisimmetrica perché se $x \leq y$ e $y \leq x$ allora $x=y$

E' transitiva perché se $x \leq y$ e $y \leq z$ allora $x \leq z$

Controesempio: In Z la relazione la relazione “successore” nRm se $m = n+1$ non è una relazione d'ordine perché non soddisfa la proprietà transitiva.

Infatti se 0 è il successore di -1 ed 1 è il successore di 0 , non è vero che 1 sia il successore di -1 .

In conclusione, poiché una relazione non può essere simmetrica e antisimmetrica insieme, una relazione non può essere un'equivalenza ed un'ordine allo stesso tempo (salvo $x = y$).

2.4 TIPI DI RELAZIONI D'ORDINE

Finora abbiamo visto quindi qual è la differenza tra corrispondenze e relazioni. Abbiamo compreso che le relazioni sono un sottoinsieme delle corrispondenze nelle quali l'insieme di partenza e d'arrivo è lo stesso, quello che in tutti gli esempi abbiamo chiamato A .

In seguito abbiamo visto come le relazioni possono essere a loro volta relazioni d'equivalenza e d'ordine, in particolare le relazioni d'equivalenza soddisfano le proprietà riflessiva, simmetrica e transitiva, mentre le relazioni d'ordine soddisfano le proprietà riflessiva, antisimmetrica e transitiva. Risulta chiaro quindi che una relazione d'ordine non può essere anche un'equivalenza in quanto non può soddisfare allo stesso tempo le proprietà simmetrica e anti simmetrica.

A questo punto facciamo un'ulteriore distinzione all'interno delle relazioni d'ordine, infatti vi possono essere relazioni di

- Ordine Parziale
- Ordine Totale

2.4.1 Definizione: *ordine parziale*

Una relazione R in A si dice di ***ordine parziale*** se soddisfa le proprietà Riflessiva, Antisimmetrica e Transitiva. Quindi una relazione d'ordine senza nessuna altra restrizione è una relazione d'ordine parziale

Esempio:

La relazione “divide” in \mathbb{N} è una relazione d'ordine, ma è parziale. I numeri 2 e 3, ad esempio, non sono confrontabili tra loro in quanto né 2 divide 3, né 3 divide 2 in \mathbb{N} .

Controesempio:

La relazione “divide” in \mathbb{Z} non è una relazione d'ordine perché non soddisfa la proprietà antisimmetrica: 2 e -2 sono divisori l'un l'altro in \mathbb{Z} ma non sono uguali.

2.4.2 Definizione: *ordine totale*

Una relazione d'ordine R in A si dice **ordine totale** se due elementi qualsiasi $x, y \in A$ sono sempre confrontabili, ossia vale sempre (almeno) una tra $x R y$ e $y R x$.

Una relazione d'ordine non totale si dice ordine parziale

Proprietà importante:

Gli insiemi numerici $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ con le rispettive relazioni d'ordine " \leq " sono totalmente ordinati.

2.4.3 Definizione: *preordine*

Una relazione in A si dice *relazione di preordine*, oppure un *preordine*, se è transitiva.

Esempio:

Sia $A = \mathbb{R}$

Dati due qualsiasi elementi di \mathbb{R} , la relazione "x è strettamente minore di y" è una relazione di preordine in \mathbb{R} .

Questo tipo di relazione non è esattamente una relazione d'ordine in quanto non soddisfa due delle tre proprietà necessarie: antisimmetrica e riflessiva. Tuttavia la presentiamo per completezza tra le relazioni di ordinamento degli insiemi.

2.5 ORDINAMENTO NEGLI ALBERI

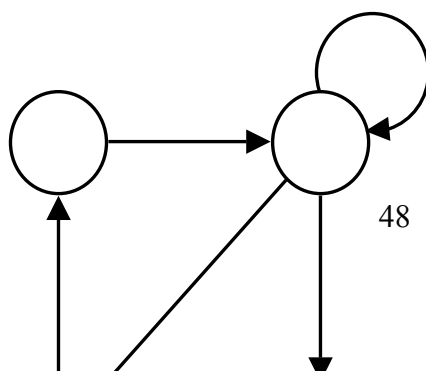
Finora abbiamo presentato i concetti di relazioni di ordine, di preordine, d'ordine parziale e totale. In questo paragrafo andremo più a fondo nei concetti più strettamente legati al nostro lavoro. Andremo a vedere brevemente che cosa sono gli alberi e come i concetti di ordinamento appena visti si adattano al nostro scopo, quello di ordinare un albero di guasto. Visiteremo modi esistenti di ordinare un albero e infine presenteremo il modo da noi utilizzato nel progetto per la gestione dei rischi operativi.

Per presentare il concetto di alberi dovremo prima introdurre il concetto di grafi, una grande famiglia all'interno della quale vi sono anche gli alberi.

2.6 I GRAFI

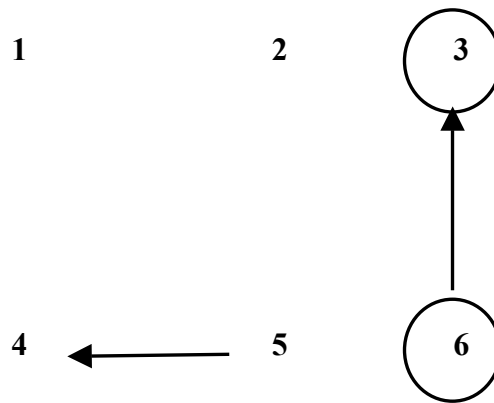
2.6.1 Definizione: Grafo Orientato

- Un *grafo orientato* (o diretto) G è una coppia (V,E) dove V è un insieme finito detto dei *vertici* e E è una relazione binaria su V che forma l'insieme degli *archi*. Gli archi sono delle coppie ordinate di vertici.



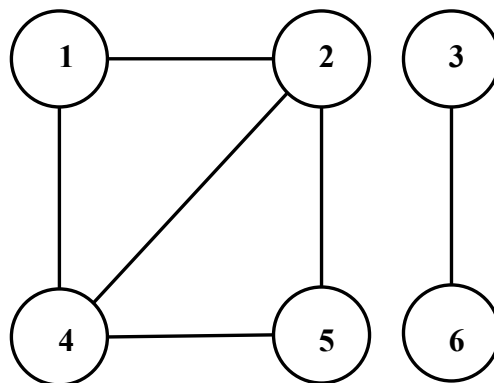
$$V = \{1, 2, 3, 4, 5, 6\}$$

$$E = \{(1, 2), (2, 2), (2, 4), (2, 5), (4, 1), (4, 5), (5, 4), (6, 3)\}$$



2.6.2 Definizione: Grafo non orientato

- Un grafo non orientato è un grafo in cui gli archi sono coppie non ordinate di vertici, cioè un arco fra i vertici u,v è un insieme di due elementi $\{u,v\}$ piuttosto che una coppia (u,v) .
- Tuttavia si indica l'arco sempre con notazione (u,v)



- Sia (u,v) un arco di un grafo orientato, si dice che:
 - l'arco esce dal vertice u
 - l'arco entra nel vertice v
- un arco (u,v) di un un grafo non orientato si dice che è incidente sui vertici v e u
- si dice che v è adiacente a u
 - in un grafo non orientato la relazione di adiacenza è simmetrica
 - in un grafo orientato v è adiacente a u , ma non è vero il viceversa, e si indica con la notazione $u \rightarrow v$

2.6.3 Definizione: Cammino

Un cammino in un grafo (non orientato) o catena è una sequenza di nodi

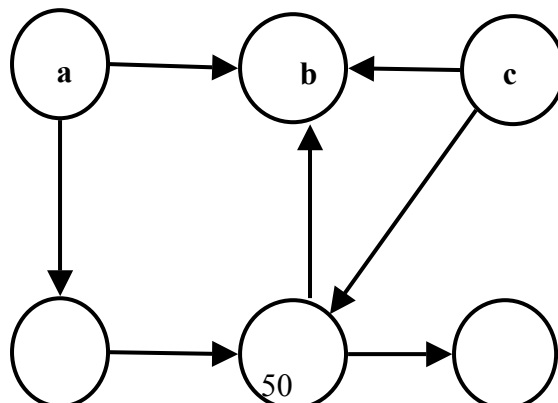
$$v_1, v_2, \dots, v_k \in V$$

tale che $(v_i, v_{i+1}) \in E \quad \forall i=1, \dots, k-1$

In un grafo orientato un **cammino** è tale che

$(v_i, v_{i+1}) \in E$ oppure $(v_{i+1}, v_i) \in E \quad \forall i=1, \dots, k-1$

Esempio

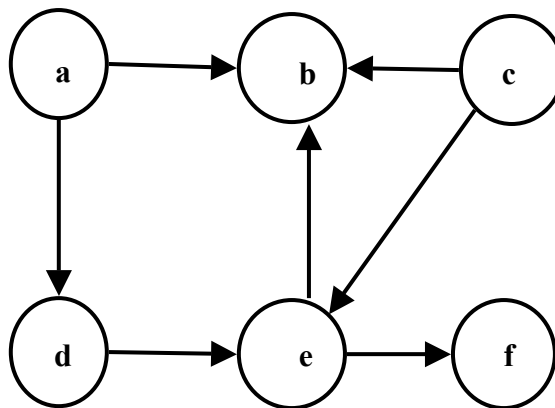


d e f

a,b,e,f è un cammino (non orientato o catena)

2.6.4 Definizione:Ciclo

Un ciclo è un cammino in cui $v_k = v_1$



a,b,c,e,d,a è un ciclo.

Un grafo (orientato o no) è **connesso** se tra ogni coppia di nodi esiste almeno un cammino.

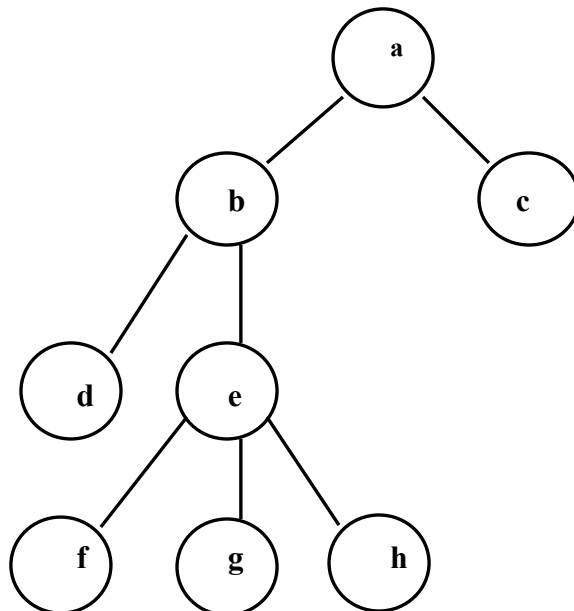
Un grafo si dice **aciclico** se non contiene cicli

2.7 GLI ALBERI

Dopo aver compreso cosa sono i cicli, passiamo agli alberi che sono un tipo particolare di cicli.

Sia T un grafo con V nodi ed E archi. Questo grafo è un albero se ha le seguenti caratteristiche:

- E' connesso ed ha $V-1$ archi
- E' privo di cicli
- Tra ogni coppia di nodi in T esiste uno e un solo cammino
- T è connesso, ma eliminando un arco qualsiasi diviene sconnesso
- T è aciclico, ma connettendo con un arco due nodi qualsiasi, si crea un ciclo.



Questo è un albero ed è connesso, aciclico, con 8 nodi e 7 archi.

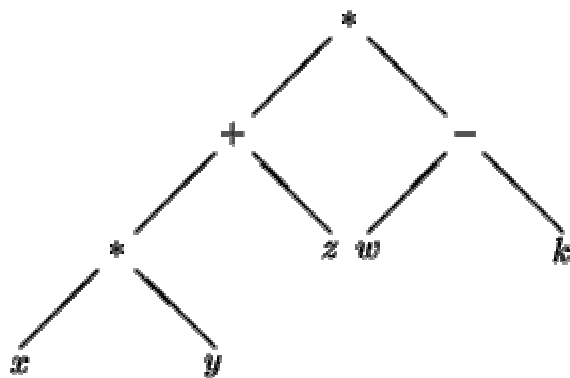
Le strutture ad albero hanno svariate applicazioni.

Ad esempio, con una struttura ad albero è possibile rappresentare organizzazioni gerarchiche, come l'indice di un libro. Un libro infatti è diviso in capitoli, ciascuno dei quali è diviso in paragrafi, che a loro volta possono essere suddivisi in sottoparagrafi. Possiamo rappresentare ciascuna di queste entità con un nodo. La radice dell'albero rappresenta il capitolo 1. Dato un nodo, rappresentiamo alla sua destra i nodi successivi dello stesso livello, e alla sua sinistra i nodi di livello inferiore. Pertanto, il figlio destro della radice corrisponderà al capitolo 2, il figlio sinistro della radice al paragrafo 1.1, il figlio destro del figlio sinistro della radice al paragrafo 1.2, e così via.

Le espressioni aritmetiche hanno una naturale rappresentazione come alberi binari. Ad esempio, l'espressione

$$(x*y+z)*(w-k)$$

può essere rappresentata dal seguente albero:



2.7.1 TERMINOLOGIA

Introduciamo ora parte della terminologia relativa agli alberi:

- a) **livello** di un nodo x : e' la distanza dalla radice al nodo x (distanza intesa come numero di archi che collegano la radice al nodo in questione x).Quindi:
 - La radice ha livello 0.
 - se un nodo di A , ha livello k , tutti i suoi figli hanno livello $k+1$
- b) **profondita'** o altezza dell'albero: e' il massimo livello dei nodi di un albero
- c) **foglie**: sono i nodi terminali dell'albero (nodi senza figli)
- d) **relazione fra i nodi**: padre, figlio, fratello...tipo le relazioni parentali.
- e) **Ramo**: è il percorso dalla radice alla foglia
- f) **Altezza**: è data dal numero dei nodi del ramo più lungo

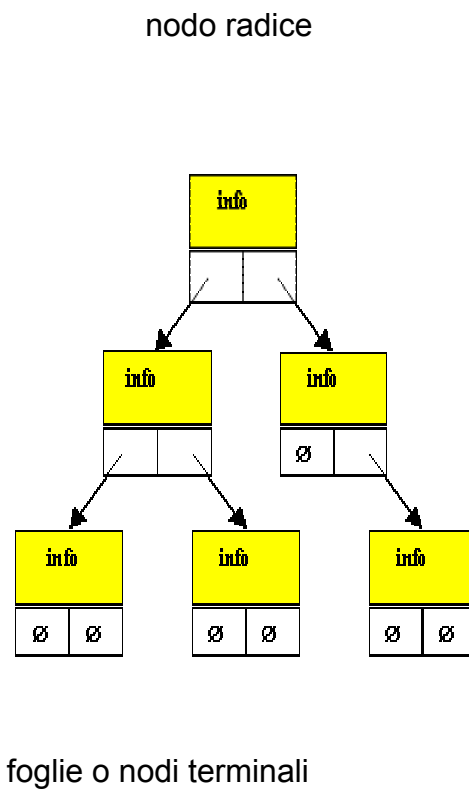
2.7.2 ALBERI BINARI

Esiste un insieme di alberi detti alberi binari, i quali sono una struttura di dati astratta, in cui ogni nodo ha al massimo due figli. Un albero binario si dice completo se ogni suo nodo, che non sia una foglia, ha un numero di figli pari a 2 e tutte le foglie appartengono allo stesso livello.

Possiamo dedurre che un albero binario completo avente profondità k , ha un numero di foglie pari a: 2^k .

L'albero della figura 1 e' un albero binario ma non e' completo.

Esempio di albero binario: Fig.1



Data questa infarinatura al concetto di albero, abbiamo compreso quali siano le principali parti che lo compongono e il loro nome. Più avanti,

nella spiegazione dell'algoritmo, parleremo spesso dei concetti appena illustrati di foglie, radice e livelli. In particolare la radice verrà quasi sempre chiamata con il sinonimo Top Gate, in quanto nelle applicazioni bancarie all'interno delle quali si pone l'algoritmo in questione, questo era il nome usato più spesso.

I concetti di cui invece andremo a parlare adesso sono quelli più vicini al lavoro svolto. Andremo a vedere infatti in quali modi si può visitare un albero e infine quale sia la maniera utilizzata da noi per visitare l'albero di guasto che è servito per studiare i rischi operativi della banca.

2.8 VISITE AD UN ALBERO

Un albero può essere visitato in diversi modi:

- Visita in profondità (depth-first search, a scandaglio):DFS
 - Vengono visitati i rami uno dopo l'altro
 - Ci sono tre varianti:
 - Visita in preordine
 - Visita in ordine simmetrico
 - Visita in postordine
- Visita in ampiezza (breadth-first search, a ventaglio):BFS
 - A livelli, partendo dalla radice

2.8.1 VISITE IN PROFONDITA'

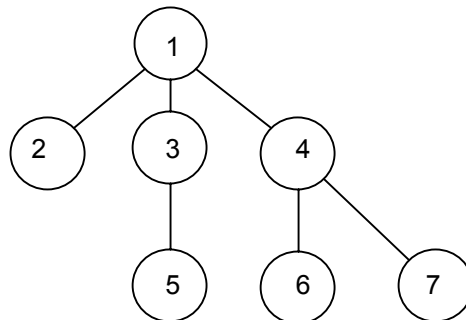
1) Visita in preordine o ordine anticipato: Si visita prima la radice r , poi il sottoalbero sinistro T_{sx} e infine il sottoalbero destro T_{dx} .

2) Visita in ordine simmetrico o inordine: Si visita prima sottoalbero sinistro T_{sx} , poi la radice r e infine il sottoalbero destro T_{dx} .

3) visita in postordine o ordine posticipato: Si visita prima il sottoalbero sinistro T_{sx} , poi il sottoalbero destro T_{dx} e infine la radice r .

2.8.2 VISITA IN AMPIEZZA

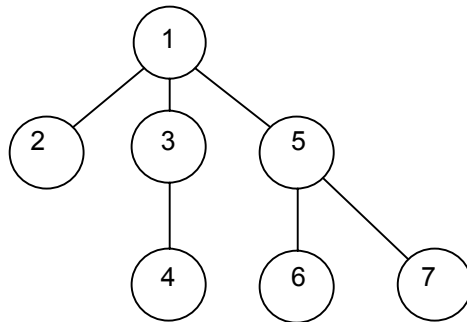
Questo è un tipo di visita a livelli, partendo dalla radice



2.8.3 VISITA IN PREORDINE o ORDINE ANTICIPATO

Andiamo ora a capire più nel dettaglio come può funzionare un algoritmo che effettua una visita ad un albero in preordine.

Immaginiamo di avere un albero di questo tipo:



Facciamo un esempio in modo da chiarire come può ragionare un algoritmo che visiti l'albero in preordine:

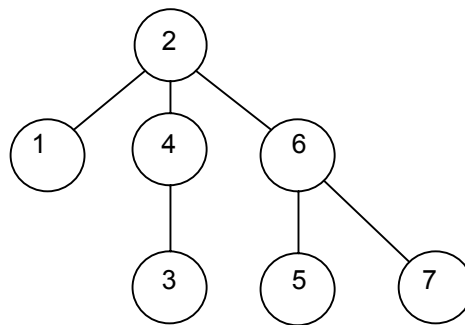
- Partenza dalla radice, stampa della radice (1)
- Spostamento in basso verso il primo figlio più a sinistra(2),
- Ricerca di ulteriori figli del nodo 2, non ha figli.
- Spostamento a destra verso il fratello più a destra del nodo 2.(3)
- Ricerca di figli del nodo 3 partendo da sinistra (4)
- Trovato nodo 4
- Ricerca dei figli del nodo 4. Non ha figli.
- Ricerca fratello più a destra del nodo 4. Non ha fratelli.
- Ritorno al nodo 3.
- Spostamento verso il fratello più a destra del nodo 3 (5).
- Trovato nodo 5.
- Ricerca di figli del nodo 5 partendo da sinistra
- Trovato nodo 6
- Ricerca dei figli del nodo 6. Non ha figli.
- Ricerca fratello più a destra del nodo 6.
- Trovato nodo 7
- Ricerca dei figli del nodo 7. Non ha figli
- Ricerca fratello più a destra del nodo 7. Non ha fratelli.
- Ritorno al nodo 5.

- Ricerca fratello più a destra del nodo 5. Non ha fratelli
- Fine.

2.8.4 VISITA IN ORDINE SIMMETRICO O INORDINE

Facciamo ora lo stesso esempio che abbiamo visto prima però con questo secondo metodo di visita ad un albero: la visita in ordine simmetrico.

Usiamo lo stesso albero di prima, in cui abbiamo cambiato i numeri dei nodi per rendere più chiaro quale sarà il nuovo ordinamento.

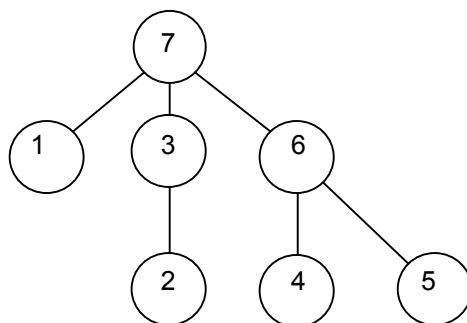


- Partenza dalla radice(2)
- Spostamento verso il figlio più a sinistra di 2. (1)
- Spostamento verso il figlio più a sinistra.Non ha figli.
- Stampa 1
- Torna a 2. stampa 2
- Spostamento verso il fratello più a destra di 1 (4)

- Spostamento verso il figlio più a sinistra di 4 (3)
- Spostamento verso il figlio più a sinistra di 3. Non ha figli.
- Stampa 3.
- Torna a 4. Stampa 4
- Spostamento verso il fratello più a destra di 4 (6)
- Spostamento verso il figlio più a sinistra di 6 (5)
- Spostamento verso il figlio più a sinistra di 5. Non ha figli.
- Stampa 5
- Torna a 6. Stampa 6
- Spostamento verso il fratello più a destra di 6 (7).
- Spostamento verso il figlio più a sinistra di 7. Non ha figli
- Stampa 7

2.8.5 VISITA IN POSTORDINE

Infine vediamo come verrebbe riordinato un albero attraverso una visita in postordine.



- Partenza dalla radice (7)
- Spostamento verso il figlio più a sinistra di 7.(1)
- Spostamento verso il figlio più a sinistra di 1. Non ha figli
- Stampa 1
- Spostamento verso il fratello più a destra di 1. (3)
- Spostamento verso il figlio più a sinistra di 3.(2)
- Spostamento verso il figlio più a sinistra di 2. Non ha figli
- Stampa 2
- Ritorna a 3. Stampa 3.
- Spostamento verso il fratello più a destra di 3. (6)
- Spostamento verso il figlio più a sinistra di 6.(4)
- Spostamento verso il figlio più a sinistra di 4. Non ha figli
- Stampa 4
- Spostamento verso il fratello più a destra di 4. (5)
- Spostamento verso il figlio più a sinistra di 5. Non ha figli
- Stampa 5
- Ritorna a 6. Stampa 6.
- Ritorna a 7. Stampa 7.

2.9 IL NOSTRO ALGORITMO

Dopo questa breve, ma speriamo esauriente illustrazione delle principali parti costituenti un albero ed aver visto i diversi modi per

visitarlo, ci sembra opportuno anticipare che il tipo di visita che verrà usato per navigare l'albero nell'algoritmo da noi sviluppato è una **visita in profondità** ed esattamente la **visita in preordine**. Quindi riassumendo, quella visita per la quale si parte dalla radice, si visita prima il ramo sinistro e poi si scorre l'albero passando ai rami via via più a destra e sempre partendo dalla radice verso le foglie.

CAPITOLO 3

SVILUPPO DELL'ALGORITMO

3.1 COME NASCE LA NECESSITA' DI UN ALGORITMO CHE GENERI L'INDENTATURA

Come abbiamo detto precedentemente la struttura dei rischi operativi risulta efficacemente rappresentata dagli alberi di guasto.

Lavorando con OpRisk per alcuni mesi era chiaro che la rappresentazione ad Albero ci era stata di grande aiuto per capire in modo semplice quali fossero le relazioni e le gerarchie che uniscono i nodi tra loro. Senza una tale rappresentazione, l'ingresso in questo mondo per una persona che non abbia dimestichezza con le strutture e i termini qui impiegati, richiederebbe grandissimi sforzi.

Ciononostante per rendere possibili le elaborazioni necessarie per il calcolo del Rischio Operativo non era possibile lavorare con gli alberi e basta, ma è stato necessario trasformare questi alberi in tabelle. Da queste tabelle partiva infatti ogni analisi volta al calcolo del rischio operativo. Nonostante questa trasformazione, la relazione padre-figlio veniva comunque salvaguardata anche nella tabella, infatti ogni nodo o evento contiene sempre al suo interno il nome del padre a cui si riferisce. Ciò

che veniva perso è la possibilità di visualizzare facilmente le relazioni esistenti.

3.2 COME VIENE PENSATA UNA SOLUZIONE

In pratica gli eventi di rischio e i nodi che costituiscono l'albero e quindi tutte le relazioni esistenti, apparivano come una massa di dati difficile da capire. Per fare un esempio si vedano le due tabelle di seguito.

Tab 1

fname	ftparent
AUTORIZZARE il rilascio del pagamento in PA41	Regolamento (derivati OTC CP e TF - Strutturati - Spot/Forward)
Generazione messaggi di pagamento con errori formali	AUTORIZZARE il rilascio del pagamento in PA41
Non corretta identificazione del beneficiario dell incasso	EFFETTUARE il matching manuale degli incassi con i relativi avvisi
EFFETTUARE il matching manuale degli incassi con i relativi avvisi	Regolamento (derivati OTC CP e TF - Strutturati - Spot/Forward)
Regolamento (derivati OTC CP e TF - Strutturati - Spot/Forward)	DERIVATI SU CAMBI OTC CP

Tab 2

fname	ftparent	livello
Regolamento (derivati OTC CP e TF - Strutturati - Spot/Forward)	DERIVATI SU CAMBI OTC CP	1
EFFETTUARE il matching manuale degli incassi con i relativi avvisi	Regolamento (derivati OTC CP e TF - Strutturati - Spot/Forward)	2
Non corretta identificazione del beneficiario dell incasso	EFFETTUARE il matching manuale degli incassi con i relativi avvisi	3

AUTORIZZARE il rilascio del pagamento in PA41	Regolamento (derivati OTC CP e TF - Strutturati - Spot/Forward)	2
Generazione messaggi di pagamento con errori formali	AUTORIZZARE il rilascio del pagamento in PA41	3

Nella prima tabella si può avere un esempio di com'era la visualizzazione dei dati prima dell'utilizzo di quest'algoritmo. Si può notare come l'unico modo per risalire alla parentela degli eventi/nodi è quello di leggere il campo ftparent di un evento e poi cercare ciò che si è letto negli altri record nel campo fname. Per esempio per cercare il padre di "Autorizzare il rilascio..." dobbiamo andare a leggere il campo ftparent che pone: "Regolamento(derivati...)". A questo punto bisogna andare a cercare tra tutti i record quale ha nel campo fname proprio questo "Regolamento(derivati...)", questo sarà il padre. Immaginiamo che i dati non siano 5 ma siano centinaia, e che i livelli, quindi le relazioni padre-figlio siano decine, non una sola. A questo punto ci sarà chiaro come la rappresentazione della seconda tabella ci faciliti le cose.

Con la nuova rappresentazione, per vedere chi è il padre basta guardare la tabella e con un colpo d'occhio si capiscono tutte le relazioni. Questa visualizzazione ricorda un po' la gestione delle risorse di un qualunque computer. Quindi abbiamo una cartella, per esempio, che si chiama "Regolamento..." che contiene due cartelle: "Effettuare il matching..." e "Autorizzare il rilascio...", a sua volta ognuna di queste cartelle contiene una sottocartella, rispettivamente "Non corretta identificazione..." e "Generazione messaggi...". Anche se la numerosità della tabella cresce la lettura delle cartelle e delle sottocartelle, che in questo caso saranno nodi e eventi, rimane semplice.

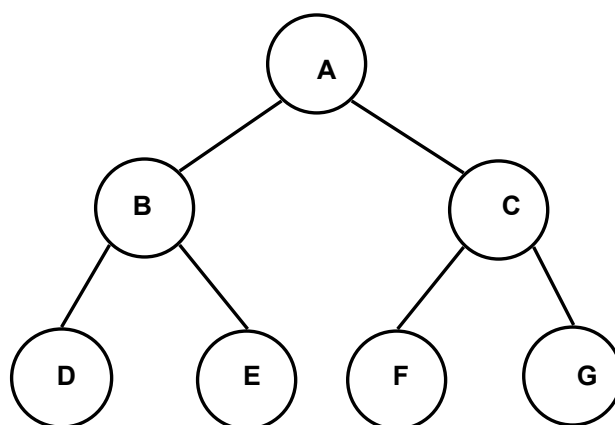
Lo scopo di questo lavoro è quindi trovare un algoritmo matematico che trasformi qualunque albero nella rappresentazione tabellare che

abbiamo visto sopra. Le tabelle da cui si parte naturalmente contengono per ogni evento o nodo il nome del padre a cui si riferiscono, ciò che fa l'algoritmo è assegnare un livello ad ogni nodo/evento in base alla propria distanza dal Top Gate e in seguito ricreare la rappresentazione ad albero mediante in dentatura.

3.3 L'ALGORITMO

Passiamo ora all'analisi del funzionamento dell'algoritmo. Partiremo descrivendo brevemente, mediante semplici esempi e grafici, come viene impostato il problema, quali obiettivi si pone il nostro studio, per poi passare alla struttura vera e propria dell'algoritmo.

Immaginiamo di dover studiare un albero di guasto di questo tipo:



3.3.1 LE DUE FASI

1. **Ordinamento:** La fase in cui l'albero viene scomposto in modo da ottenere l'ordinamento simile a quello della Gestione Risorse.
2. **Indentatura:** La fase in cui gli elementi di livello superiore acquisiscono un margine via via maggiore.

1. ORDINAMENTO

Un buon modo per rappresentare quest'albero per i nostri scopi è mediante la **visita in preordine**. Questo tipo di visita di un albero consiste nel partire dal nodo principale(A), detto radice, per poi scendere al primo non sulla sinistra(B). Da qui scendere ancora verso il primo nodo sulla sinistra fino a giungere alla foglia(D). A questo punto si passa al primo fratello della foglia sulla destra(E), se questo ha figli si passa ai figli sempre da sinistra verso destra, altrimenti si passa ad un altro fratello. Quando finiscono i fratelli si sale al nodo superiore e si passa al fratello più a destra del nodo in cui ci troviamo(C). Da qui si opera nello stesso modo cercando i figli prima a sinistra fino alle foglie(F) e poi a destra(G).

Il risultato per l'albero di esempio che abbiamo sopra è: **A,B,D,E,C,F,G**.

2. INDENTATURA

Il nostro studio usa la visita di un albero in preordine, ma non si limita a ciò, nel passaggio successivo infatti si opera quel processo di indentatura che rende la tabella così ordinata molto facile da essere letta e simile alla navigazione fra le cartelle di ogni computer.

Il risultato a cui si vuole arrivare è il seguente:

A		
	B	
		D
		E
	C	
		F
		G

Questa visualizzazione permette quindi di non perdere di vista la relazione padre-figlio, e inoltre, essendo una tabella, ci rende possibili tutte le operazioni necessarie al fine dei calcoli che devono essere fatti.

3.4 COME è STRUTTURATO L'ALGORITMO

L'algoritmo è diviso in due parti o moduli,

- la prima parte **asigna ad ogni record il suo livello corrispondente**, ovvero assegna al Top Gate il livello 0, ai suoi figli il livello 1, ai figli dei figli il livello 2 e così via.
- La seconda parte invece riordina i dati in modo da dare come output la struttura **ordinata e indentata** che è lo scopo di questo lavoro.

DOPO IL PRIMO MODULO

	Livello
A	0
B	1
C	1
D	2
E	2
F	2
G	2



DOPO IL SECONDO MODULO

	Livello
A	0
B	1
D	2
E	2
C	1
F	2
G	2

3.4.1 IL PRIMO MODULO

INTESTAZIONE:

```
Option Compare Database
Option Explicit
Public time_ini, time_fin, time_medio, time_parz, time_tot
```

Vengono definite delle variabili pubbliche che serviranno per registrare il tempo di calcolo.

Time_ini registra l'inizio della procedura

Time_medio registra il momento in cui il sistema ha terminato di eseguire il primo modulo

Time_fin registra la fine dell'esecuzione del programma.

Time_parz= time_medio-time_ini

Time_tot=time_fin-time_ini

INIZIALIZZAZIONE DELLE VARIABILI:

```
Sub trova_livelli()  
Dim dbs As Database  
Dim rst_BE As Recordset, rst_ALL As Recordset, rst_Query As  
Recordset  
Dim sql_BE As String, sql_ALL As String, sql_Query As String  
Dim i As Integer  
Dim i1 As Integer, i2 As Integer, i3 As Integer, i4 As Integer, i5 As  
Integer  
Dim i6 As Integer, i7 As Integer, i8 As Integer, i9 As Integer, i10 As  
Integer  
Dim j As Long  
Dim N As Integer  
Dim Livello As Integer  
Dim ftparent As String, ftParent1 As String, ftParent2 As String,  
ftParent3 As String  
Dim ftParent4 As String, ftParent5 As String, ftParent6 As String,  
ftParent7 As String  
Dim rst_1 As Recordset  
Dim sql_1 As String  
Dim n_Livelli As Integer
```

Questa procedura serve per assegnare ad ogni nodo o evento il proprio livello. Per livello si intende la distanza dal Top Gate misurata in archi che separano l'evento/nodo dal Top Gate.

Facciamo partire il timer che misura il tempo di calcolo del processo

```
time_ini = Timer
```

Diamo al ftparent del TopGate il Valore "", questo serve nel caso venga importato un albero parziale, per non permettere a questo problema di bloccare il sistema. Generalmente il TopGate ha sempre ftparent vuoto, ma nel caso in cui l'utente stia facendo delle prove con pezzi di tracciato, allora queste due query assicurano il buon funzionamento dell'algoritmo.

```
DoCmd.OpenQuery "G_Q_riga_FT_tracciato_selez"  
DoCmd.OpenQuery "G_Q_riga_FT_tracciato_agg"
```

Assegnamo alla variabile sql_BE l'elenco di tutti i basic event presenti nel tracciato

```
Set dbs = CurrentDb  
sql_BE = "SELECT fname,ftp,parent,ftype FROM G_Q_trova_BE "  
Set rst_BE = dbs.OpenRecordset(sql_BE)
```

Attribuiamo alla variabile sql_ALL tutto il tracciato caricato in Fault_Tree_Tracciato che può essere quello importato da Aris o Da Relex

```
sql_ALL = "SELECT fname,ftp,parent,ftype FROM  
FAULT_TREE_TRACCIATO "  
Set rst_ALL = dbs.OpenRecordset(sql_ALL)
```

Assegnamo a N la numerosità totale del tracciato di cui sarà fatta l'indentatura

```
rst_ALL.MoveLast  
N = rst_ALL.RecordCount
```

Diamo Alla Matrice una dimensione che dipende dalla numerosità del tracciato importato

```
ReDim OpVar_ini(N * 5, 3) As Variant
```

Inizializzo variabili "Livello" e "i" a zero.

```
Livello = 0  
i = 0
```

Inizializziamo tabella che conterrà il tracciato con i livelli corretti (TopGate=0, poi a crescere)

```
db.Execute "DELETE * FROM G_TAB_OpVar_ini_Livelli_ok"
```

3.4.2 LA PROCEDURA DI ASSEGNAZIONE DEI LIVELLI

La procedura è divisa al suo interno in due parti: nella prima si cerca di ricostruire l'albero al contrario, ovvero si parte da ogni foglia e si ricostruisce l'albero andando verso il TopGate. In pratica è come se

cercassimo di ricostruire un albero genealogico partendo dal basso e cercando prima il padre, poi il nonno e così via verso gli avi più remoti.

Ovviamente si termina la ricerca nel momento in cui il campo `ftparent` (padre) è vuoto, il che significa, nella ricostruzione di un albero genealogico, che non ci sono fonti sufficienti per scoprire che sia il padre di quel dato Avo. Nel nostro caso significa che siamo giunti al Top Gate. In conclusione questa procedura assegna livello=0 al Basic Event, livello=1 al padre di questo Basic Event, livello=2 al padre del padre e così via fino a giungere al Top Gate.

rst_BE.MoveFirst

Si parte da ogni Foglia

Do While Not rst_BE.EOF

Creiamo una matrice temporanea `OpVar_ini` che conterrà tutti gli elementi ordinati dalle foglie al topgate. I campi che vengono presi in considerazione per il riordinamento sono:

`ftname= rst_BE.Fields(0)`: Il nome del Basic Event o Nodo

`ftparent= rst_BE.Fields(1)`: Il padre di quel Basic Event o nodo

`fttype= rst_BE.Fields(2)`: 7 Se si tratta di Basic Event, 2 se si tratta di nodo

OpVar_ini(i, 0) = rst_BE.Fields(0)

OpVar_ini(i, 1) = rst_BE.Fields(1)

OpVar_ini(i, 2) = rst_BE.Fields(2)

OpVar_ini(i, 3) = Livello

La tabella G_TAB_OpVaR_ini contiene i singoli percorsi dalle foglie al topGate con i livelli. (Tabella inizializzata a ogni ciclo)

```
dbms.Execute "DELETE * FROM G_TAB_OpVar_ini"
```

Vengono accodati alla tabella G_TAB_OpVar_ini i valori della foglia

```
dbms.Execute "insert into G_TAB_OpVar_ini  
(fname,ftp,parent,ftype,livello) VALUES"  
          & "(" & CStr(OpVar_ini(i, 0)) & "," &  
CStr(rst_BE.Fields(1).Value) & "," &  
          & "" & CStr(rst_BE.Fields(2).Value) & "," & CStr(Livello) &  
          & ");"  
i=i+1
```

Assegnamo alla variabile “ftp,parent” il campo ftp,parent del Basic Event

```
ftp,parent = rst_BE.Fields(1)
```

Ci posizioniamo all’inizio di “rst_all”, variabile stringa che contiene tutti i records del tracciato

```
rst_ALL.MoveFirst
```

In questo punto si avvia la procedura che cerca in tutto il tracciato di ricostruire un ramo partendo dalla foglia per arrivare al TopGate.

La procedura continua finché non trova un padre che ha il campo “ftp,parent” vuoto

Do While Not ftparent = ""

Se l'algoritmo trova fra tutti i record del tracciato un record che ha il campo(0),quello destinato al nome del nodo,uguale alla variabile "ftparent" assegnata precedentemente al padre del Basic Event, allora l'algoritmo procede, altrimenti continua la ricerca in tutti i record del tracciato

If ftparent = rst_ALL.Fields(0) Then

Aumenta la variabile livello di 1

Livello = Livello + 1

Assegna alla matrice OpVar_ini i valori del nodo appena trovato che è il padre dell'evento /nodo trovato precedentemente.

OpVar_ini(i, 0) = rst_ALL.Fields(0)

OpVar_ini(i, 1) = rst_ALL.Fields(1)

OpVar_ini(i, 2) = rst_ALL.Fields(2)

OpVar_ini(i, 3) = Livello

Questo controllo attribuisce la stringa "TOPGATE" nel campo ftparent del Top Gate. Questo controllo serve per ovviare ad un problema di Access che avrebbe creato degli errori se questo campo fosse rimasto vuoto.

If rst_ALL.Fields(1) = "" Then

OpVar_ini(i, 1) = "TOPGATE"

End If

Viene scritto nella tabella G_TAB_OpVar_ini il nodo appena trovato e il suo relativo livello

```

        dbs.Execute "insert into G_TAB_OpVar_ini
(ftname,ftparent,ftype,livello) VALUES" _
        & "(" & CStr(rst_ALL.Fields(0).Value) & "," &
CStr(OpVar_ini(i, 1)) & "," _
        & "" & CStr(rst_ALL.Fields(2).Value) & "," & CStr(Livello) &
        ");"
        i = i + 1
    
```

Alla variabile “ftparent” viene ora assegnato il valore del campo(1) di rst_ALL, dove risiede il nome dell’ftparent del nodo appena trovato.

```

        ftparent = rst_ALL.Fields(1)
    
```

Torna all’inizio del tracciato e riavvia la ricerca in tutto il tracciato di un record che abbia come ftname il valore che è appena stato alla variabile “ftparent”.

```

        rst_ALL.MoveFirst
    Else
    
```

Se non trova il record che cerca va al successivo

```

        rst_ALL.MoveNext
    End If
Loop
    
```

'-----

3.4.3 PROCESSO DI INVERSIONE LIVELLI

Alla fine di questo ciclo la tabella G_TAB_OpVaR_ini contiene un' intero percorso da una foglia al TopGate con i relativi livelli. I livelli però sono calcolati all'inverso, ovvero il basic Event ha livello 0, mentre il Top Gate ha il livello più alto. Nella realtà invece dovrà essere che il Top Gate ha il livello 0, mentre il Basic Event ha il livello più alto. Questa parte della procedura serve appunto per attribuire ad ogni nodo/evento il proprio livello ufficiale e definitivo.

Attribuiamo alla variabile rst_1 il percorso contenuto nella tabella G_TAB_OpVaR_ini

```
sql_1 = "SELECT ftname,ftparent,ftype,livello FROM
G_TAB_OpVar_ini"
Set rst_1 = dbs.OpenRecordset(sql_1)
```

Attribuiamo alla variabile n_Livelli il valore che la variabile "Livello" assume in questo momento. Questo significa che n_Livelli rappresenta il **numero di livelli presenti nel Ramo** appena registrato nella tabella. Sarà ovviamente a partire da questo valore che si ricostruirà l'albero con i **livelli ufficiali**, ovvero **zero** per il **Top Gate**, **uno per i figli** e così via fino ai basic event.

```
n_Livelli = Livello
```

Vengono scritti nella Tabella "G_TAB_Opvar_ini_Livelli_ok" i records del ramo con i livelli ricalcolati.

```
Do While Not rst_1.EOF
N = n_Livelli - CInt(CStr(rst_1.Fields(3).Value))
```

```

dbms.Execute "Insert into
G_TAB_Opvar_ini_Livelli_ok(ftname,ftparent,ftype,livello) VALUES" _
& "(" & CStr(rst_1.Fields(0).Value) & "," &
CStr(rst_1.Fields(1).Value) & "," _
& "" & CStr(rst_1.Fields(2).Value) & "," &
CStr(n_Livelli - CInt(CStr(rst_1.Fields(3).Value))) & ");"
rst_1.MoveNext
Loop

```

```

i = i + 1

```

Si passa al Basic Event successivo

```

rst_BE.MoveNext

```

Viene inizializzata la variabile Livello

```

Livello = 0
Loop

```

3.4.4 CONCLUSIONE PROCESSO DI ASSEGNAZIONE LIVELLI

A questo punto della procedura nella tabella G_TAB_Opvar_ini_Livelli_ok ci sono tutti i rami dell'albero e ad ogni record è assegnato il suo vero livello misurato come distanza dal Top Gate. Naturalmente, per come è stato ricostruito l'albero, ci sono molti nodi duplicati. Infatti **ogni ramo è stato ricostruito singolarmente dalle**

foglie al Top Gate, quindi il Top Gate è ripetuto nella tabella tante volte quante sono i Basic Event..

Questa duplicazione non è comunque un problema, poiché con un semplice Raggruppamento tutti i **duplicati** verranno **eliminati**.

Viene assegnata alla variabile pubblica `time_medio` il valore del tempo di calcolo fino a questo momento, ovvero fino alla fine del primo modulo

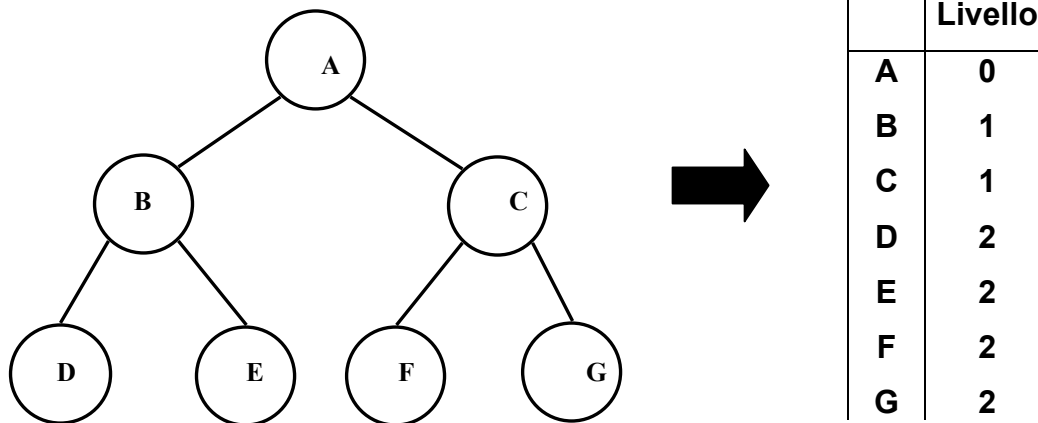
```
time_medio = Timer
```

Chiamata a subroutine che continuerà il lavoro partendo dalla situazione attuale.

```
crea_matrice_opvar  
End Sub
```

Ora viene chiamato un altro modulo del programma il quale continuerà l'elaborazione che ha interessato l'albero fino a questo momento. Ma a che punto siamo dell'elaborazione?

E' interessante sapere in che stato si trova fino a questo momento l'albero che avevamo in partenza. I due grafici di seguito spiegano a che punto ci troviamo. Come si vede ad ogni nodo è stato associato un livello, livello che dipende dalla distanza del nodo dalla radice, ci teniamo a ricordare che la distanza tra due nodi si misura in archi che separano il nodo dalla radice.



3.4.5 SECONDO MODULO: LA CREAZIONE DELL'INDENTATURA

Dichiarazione di Variabili:

```

Sub crea_matrice_opvar()
Dim dbs As Database
Dim rst_BE As Recordset, rst_ALL As Recordset, rst_Query As Recordset
Dim sql_BE As String, sql_ALL As String, sql_Query As String
Dim i As Integer
Dim i1 As Integer, i2 As Integer, i3 As Integer, i4 As Integer, i5 As Integer
Dim i6 As Integer, i7 As Integer, i8 As Integer, i9 As Integer, i10 As Integer
Dim j As Long
Dim N As Integer
Dim N_1 As Integer 'per4 calcolo dimensioni matrici
Dim Livello As Integer
Dim ftParent As String, ftParent1 As String, ftParent2 As String, ftParent3 As String
Dim ftParent4 As String, ftParent5 As String, ftParent6 As String, ftParent7 As String
    
```

```
DoCmd.SetWarnings False
Set dbs = CurrentDb
```

3.4.6 TABELLA CON IL TRACCIATO E I LIVELLI SENZA DUPLICATI

- **L'albero** è interamente **contenuto** nella tabella "G_TAB_Opvar_ini_Livelli_ok". In questa tabella
 - ad ogni Nodo/Evento è associato il proprio livello
 - Ci sono molti records duplicati
- La Query "G_Q_Raggruppa" raggruppa tutti i records duplicati in modo da riottenere il tracciato da cui si era partiti con in più i livelli associati
- I dati contenuti nella Query "G_Q_Raggruppa" vengono presi dalla Query "G_q_OpVar_con_livelli" che toglie la stringa "TOPGATE" al campo ftparent del Top Gate. Ricordiamo che questa scritta era stata aggiunta per motivi tecnici per evitare dei problemi di Access.

A questo punto, quindi, nella Query di Selezione "G_Q_OPVAR_con_livelli" è **presente l'intero tracciato** di cui si vuole fare l'ordinamento con **tutti i livelli** associati.

Viene assegnata alla variabile rst_Query i dati del tracciato contenuti nella query "G_Q_OPVAR_con_livelli"

```
sql_Query = "SELECT fname,ftp parent,ftype,livello FROM
G_Q_OPVAR_con_livelli "
Set rst_Query = dbs.OpenRecordset(sql_Query)
```

Vengono contati quanti sono in tutto i records del tracciato e viene assegnato questo valore a N_1

```
rst_Query.MoveLast  
N_1 = rst_Query.RecordCount
```

Vengono definite delle matrici con delle dimensioni proporzionali al numero di records presenti.

```
ReDim OpVar_ini(N_1 * 4, 3) As Variant, OpVar(N_1 * 4, 3) As  
Variant, OpVar_q(N_1, 3)  
i = 0  
rst_Query.MoveFirst
```

Viene assegnata alla matrice OpVar_q i dati della query contenente il tracciato con i livelli

```
Do While Not rst_Query.EOF  
OpVar_q(i, 0) = rst_Query.Fields(0)  
OpVar_q(i, 1) = rst_Query.Fields(1)  
OpVar_q(i, 2) = rst_Query.Fields(2)  
OpVar_q(i, 3) = rst_Query.Fields(3)  
i = i + 1  
rst_Query.MoveNext  
Loop
```

3.4.7 GENERAZIONE DELL'INDENTATURA

Viene inizializzata la tabella G_OpVar_ordinata che sarà la tabella che conterrà i dati orientati nel modo particolare e specifico di questo studio.

Lo scopo infatti sarà quello di ordinare i dati in modo da avere una struttura di facile lettura dalla quale si possa capire facilmente come è strutturato l'albero pur rimanendo in una struttura di tipo tabulare.

Per raggiungere questo scopo abbiamo realizzato una struttura indentata grazie alla quale è molto semplice capire quali sono i figli di un dato padre proprio perché questi compaiono dopo il padre in una posizione del foglio più spostata a destra, indentata appunto.

Questa struttura è stata realizzata in questo modo:

- Viene **cercato il Top Gate**, ovvero quel record che ha il campo "ftparent" vuoto, in tutto il tracciato.
- Viene assegnato questo record alla tabella G_OpVaR_ordinata
- Poi viene **cercato il figlio**, ovvero chi ha come ftparent ciò che per il TopGate è l'fname
- Viene assegnato questo record alla tabella G_OpVaR_ordinata
- Poi viene **cercato il figlio del figlio** fino a giungere all' estremità del ramo:il Basic Event

Tutte le ricerche vengono fatte verificando la condizione If, ma Ogni condizione If è sottostante ad un ciclo FOR . Questo significa che non viene trovato solo il primo figlio di un dato padre, ma viene sondato il tracciato in modo da ottenere il tracciato ordinato secondo i nostri scopi.

*db.Execute "DELETE * FROM G_OPVAR_ordinata"*

$j = 0$

Viene cercato il TopGate, ovvero chi ha ftparent= ""

```
For i = 0 To N
If OpVar_q(i, 1) = "" Then 'se è il topgate
```

Vengono assegnati i dati del TopGate alla matrice OpVar

```
OpVar(j, 0) = OpVar_q(i, 0)
OpVar(j, 1) = OpVar_q(i, 1)
OpVar(j, 2) = OpVar_q(i, 2)
OpVar(j, 3) = OpVar_q(i, 3)
```

Viene attribuita la stringa "TOPGATE" al campo ftparent del Top Gate per evitare un problema dovuto ai campi vuoti in Access

```
If OpVar(j, 1) = "" Then
    OpVar(j, 1) = "TOPGATE"
End If
```

Vengono assegnati i dati del Top Gate alla Tabella G_OPVAR_ordinata

```
dbms.Execute "insert into G_OPVAR_ordinata
(ID,fname,ftp,parent,ftype,livello) VALUES" _
    & "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," & CStr(OpVar(j, 1)) & "," & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) & ");"
```

Diamo alla variabile ftparent il valore di fname del Top Gate

```
ftp = OpVar(0, 0)
```

```
j = j + 1
If j <= N Then
```


A questo punto siamo al livello 1, cerchiamo tutti I figli del Top Gate, e assegnando i dati alla tabellaG_OPVAR_ordinata

```

For i1 = 0 To N
  If OpVar_q(i1, 1) = ftparent Then
    OpVar(j, 0) = OpVar_q(i1, 0)
    OpVar(j, 1) = OpVar_q(i1, 1)
    OpVar(j, 2) = OpVar_q(i1, 2)
    OpVar(j, 3) = OpVar_q(i1, 3)
    dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES" _
    & "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," & CStr(OpVar(j, 1)) & "," & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) & ");"
    j = j + 1
    ftParent1 = OpVar_q(i1, 0)

```

Da questo punto in poi siamo di fronte ad una struttura iterativa sempre uguale: si cercano i figli, i dati vengono scritti nella tabella e si procede dapprima in profondità nell'albero, poi in laterale passando ai rami vicini. Questo è possibile grazie ai seguenti cicli FOR annidati.

```

For i2 = 0 To N
  If OpVar_q(i2, 1) = ftParent1 Then
    OpVar(j, 0) = OpVar_q(i2, 0)
    OpVar(j, 1) = OpVar_q(i2, 1)
    OpVar(j, 2) = OpVar_q(i2, 2)
    OpVar(j, 3) = OpVar_q(i2, 3)
    dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES" _
    & "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," & CStr(OpVar(j, 1)) & "," & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) & ");"
    j = j + 1
    ftParent2 = OpVar_q(i2, 0)
    For i3 = 0 To N
      If OpVar_q(i3, 1) = ftParent2 Then
        OpVar(j, 0) = OpVar_q(i3, 0)
        OpVar(j, 1) = OpVar_q(i3, 1)
        OpVar(j, 2) = OpVar_q(i3, 2)
        OpVar(j, 3) = OpVar_q(i3, 3)

```

```

        dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES"
        & "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," & CStr(OpVar(j, 1)) & "," & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) & ");"
        j = j + 1
        ftParent3 = OpVar_q(i3, 0)
        For i4 = 0 To N
        If OpVar_q(i4, 1) = ftParent3 Then
            OpVar(j, 0) = OpVar_q(i4, 0)
            OpVar(j, 1) = OpVar_q(i4, 1)
            OpVar(j, 2) = OpVar_q(i4, 2)
            OpVar(j, 3) = OpVar_q(i4, 3)
            dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES"
            & "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," & CStr(OpVar(j, 1)) & "," & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) & ");"
            j = j + 1
            ftParent4 = OpVar_q(i4, 0)
            For i5 = 0 To N
            If OpVar_q(i5, 1) = ftParent4 Then
                OpVar(j, 0) = OpVar_q(i5, 0)
                OpVar(j, 1) = OpVar_q(i5, 1)
                OpVar(j, 2) = OpVar_q(i5, 2)
                OpVar(j, 3) = OpVar_q(i5, 3)
                dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES"
                & "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," & CStr(OpVar(j, 1)) & "," & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) & ");"
                j = j + 1
                ftParent5 = OpVar_q(i5, 0)
                For i6 = 0 To N
                If OpVar_q(i6, 1) = ftParent5 Then
                    OpVar(j, 0) = OpVar_q(i6, 0)
                    OpVar(j, 1) = OpVar_q(i6, 1)
                    OpVar(j, 2) = OpVar_q(i6, 2)
                    OpVar(j, 3) = OpVar_q(i6, 3)
                    dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES"
                    & "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," & CStr(OpVar(j, 1)) & "," & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) & ");"
                    j = j + 1
                    ftParent6 = OpVar_q(i6, 0)
                    For i7 = 0 To N
                    If OpVar_q(i7, 1) = ftParent6 Then
                        OpVar(j, 0) = OpVar_q(i7, 0)
                        OpVar(j, 1) = OpVar_q(i7, 1)
                        OpVar(j, 2) = OpVar_q(i7, 2)
                        OpVar(j, 3) = OpVar_q(i7, 3)
                        dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES"
                        & "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," & CStr(OpVar(j, 1)) & "," & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) & ");"
                        j = j + 1
                    End If
                End For
            End If
        End For
    End For
End For

```



```
End If  
Next
```

A questo punto l'elaborazione è conclusa e i Timer seguenti ci dicono quanto tempo l'elaborazione ha impiegato

```
time_fin = Timer  
time_parz = time_medio - time_ini  
time_tot = time_fin - time_ini
```

```
DoCmd.OpenQuery "G_Q_OPVAR_ordinata_togli_TOPGATE "
```

```
End Sub
```

3.5 TABELLE E QUERY CHE INTERAGISCONO CON L'ALGORITMO

G_Q_riga_FT_tracciato_selez : Query

G_Q_riga_FT_tracciato_agg :Query

Queste due Query hanno il compito di cancellare qualunque cosa vi sia nel campo ftparent del Top Gate. Questa è una funzione di sicurezza, infatti il campo che le due query vanno a cancellare dovrebbe essere sempre vuoto. Ciononostante, svariati test fatti hanno mostrato come

possa capitare che, con l'importazione di tracciati parziali, questo problema possa bloccare l'esecuzione dell' algoritmo che genera l'indentatura.

G_prima_riga_FT_tracciato

Questa tabella viene creata dalla query G_Q_riga_FT_tracciato_selez. Contiene il record del Top Gate e si tratta naturalmente di una tabella di servizio che serve come passaggio intermedio per l'eliminazione dell'ftparent del Top Gate.

G_q_trova_BE

Questa query di selezione prende dalla tabella FAULT_TREE_TRACCIATO, ovvero dalla tabella dove è contenuto il tracciato che si vuole elaborare, i dati necessari e sufficienti per procedere con l'ordinamento e l'indentatura:

ftname = il nome del nodo o dell'evento

ftparent= il nome del padre

fttype =il tipo, ovvero se si tratta di un nodo(2) o di un evento(7).

Di questi dati seleziona solo gli eventi, ovvero prende solo le foglie dell'albero, quelli che hanno fttype=7.

FAULT_TREE_TRACCIATO

La tabella che contiene il tracciato da elaborare

G_TAB_OpVaR_ini

Questa tabella viene popolata e cancellata tante volte quanti sono i Basic Event. Infatti contiene i singoli percorsi da una foglia al Top Gate con i livelli assegnati a partire dal Basic Event. Quindi assegnando 0 al Basic Event, 1 al padre, 2 al padre del padre e così via.

G_TAB_OpVar_ini_Livelli_ok

Questa tabella viene inserito tutto il tracciato con tutti i livelli assegnati nel modo corretto, ovvero invertiti rispetto alla tabella G_TAB_OpVaR_ini. I dati sono in questa tabella duplicati e questo è dovuto al fatto che i rami dell'albero di guasto sono stati ricostruiti singolarmente a partire dai rami.

G_q_Raggruppa

Questa query di selezione prende i dati dalla Tabella G_TAB_OpVaR_ini_Livelli_ok e raggruppa gli elementi duplicati.

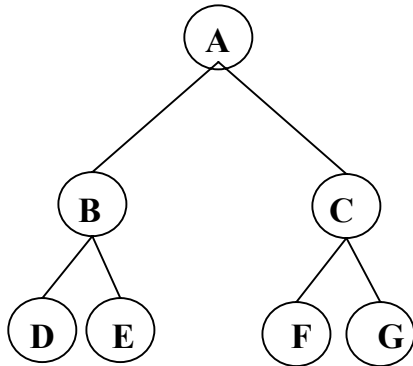
G_q_OpVar_con_livelli

Questa Query di selezione prende i dati da G_q_Raggruppa e svuota il campo fparent del Top Gate che era stato popolato dalla stringa "TOPGATE" per superare un problema che aveva Access nel gestire i campi vuoti.

G_OPVAR_ordinata

Questa tabella contiene tutto il tracciato ordinato secondo un metodo opportuno che vede tutti i figli di un padre in posizione successiva al padre. Graficamente ciò che si trova in questa tabella è quanto segue:

Albero di partenza



G_OpVar_Ordinata

Nome	Padre	Tipo	Livello
A	TopGate	2	0
B	A	2	1
D	B	7	2
E	B	7	2
C	A	2	1
F	C	7	2
G	C	7	2

G_Q_OPVAR_ordinata_togli_TOPGATE

Questa Query di aggiornamento aggiorna la tabella G_OpVar_Ordinata svuotando il campo ftparent del Top Gate. La tabella viene quindi aggiornata a questo:

Nome	Padre	Tipo	Livello
A		2	0
B	A	2	1
D	B	7	2
E	B	7	2
C	A	2	1
F	C	7	2
G	C	7	2

G_Q_Nuovo_Report_Var

Questa Query di selezione svolge due funzioni:


- Unisce i dati ordinati provenienti dalla tabella G_OpVar_Ordinata ai risultati provenienti dal calcolo del Rischio Operativo associando a ciascun nodo o evento il proprio VarR e le perdite attese
- Crea e visualizza l'indentatura usando i dati sui livelli presenti per ogni nodo e forniti dal primo modulo dell'algoritmo.

3.6 COME VIENE CREATA L' INDENTATURA

All'interno della query G_Q_Nuovo_Report_Var viene creata una variabile chiamata "Spazi". Questa Variabile sarà composta da un numero di caselle vuote che dipende dal livello del nodo o evento che viene analizzato. Quindi quando il sistema analizza il Top Gate che ha livello=0, alla variabile "Spazi" saranno attribuiti zero spazi (""). Quando verrà analizzato un nodo sottostante il Top Gate, allora alla variabile "Spazi" saranno attribuiti 3 spazi vuoti (" "). Per un nodo sottostante, quindi un figlio del figlio del Top Gate alla variabile "Spazi" saranno attribuiti 6 spazi vuoti(" ") e così via.

Per esempio ritornando all'esempio di prima la tabella G_OpVar_Ordinata viene trasformata.

Nome	Padre	Tipo	Livello
A		2	0
B	A	2	1
D	B	7	2
E	B	7	2
C	A	2	1
F	C	7	2
G	C	7	2



Nome	Padre	Tipo	Livello
A		2	0
B	A	2	1
D	B	7	2
E	B	7	2
C	A	2	1
F	C	7	2
G	C	7	2

r_Stima_OpVar_1

Questo Report viene aperto dopo il calcolo del VaR Operativo ed è la rappresentazione finale e complessiva di tutto il lavoro. In questo report infatti ci sono tutti i risultati ricavati dalla stima del Rischio Operativo e sarà ovviamente in questo report dove dovrà essere inserito il tracciato ordinato e indentato. E' per questo che il sottoreport di questo report:

Sottoreport G_Q_Nuovo_Report_Var

Prende i dati dalla Query vista sopra G_Q_Nuovo_Report_Var.

Qui di seguito abbiamo un report di esempio che fa capire qual è il risultato finale dell'elaborazione.

Nodo	Parametri				Indicatori di Rischio		
	Simulaz	Num gg	Lavoraz	Conf.	N. guasti	OpVaR	VaR correlato
BANCA IMI Business Lines	500	1	Sì	99,00%	424	€ 135.297,82	€ 191.071,91

Misure di Rischio

Processo/Basic Event	FiParent	P/B-Liv	Procedura	VaR
----------------------	----------	---------	-----------	-----

Investment Banking	BANCA IMI Business Lines	P	1	135297,82
Trading & Sales	Investment Banking	P	2	135297,82
Market Making e Gestione Posizione	Trading & Sales	P	3	135297,82
Negoziare Conto Proprio e MM	Market Making e Gestione Posizione	P	4	135297,82
PRODOTTI STRUTTURATI CP	Negoziare Conto Proprio e MM	P	5	895,16
Spunta conferme e riconciliazione (Prodotti strutturati)	PRODOTTI STRUTTURATI CP	P	6	669,54
CORREGGERE l'operazione in MxG2000	Spunta conferme e riconciliazione (Prodotti strutturati)	P	7	0,00
Errato input manuale nei sistemi: correzione su MxG2000	CORREGGERE l'operazione in MxG2000	B	8	Strutturati 0,00
ARCHIVARE la documentazione relativa all'operazione	Spunta conferme e riconciliazione (Prodotti strutturati)	P	7	1,33
Perdita/smarrimento documenti archiviati	ARCHIVARE la documentazione relativa all'operazione	B	8	Strutturati 1,33
RICEVERE la conferma modificata	Spunta conferme e riconciliazione (Prodotti strutturati)	P	7	14,29
Mancato chasing delle conferme della ctp	RICEVERE la conferma modificata	B	8	Strutturati 14,29
CORREGGERE le istruzioni di regolamento	Spunta conferme e riconciliazione (Prodotti strutturati)	P	7	595,76
Errata trascrizione dati (risultati spunta telefonica)	CORREGGERE le istruzioni di regolamento	B	8	Strutturati 595,76
VERIFICARE la conferma BIMl con la conferma ricevuta dal cliente	Spunta conferme e riconciliazione (Prodotti strutturati)	P	7	15,76
Mancata/non corretta riconciliazione conferme ctp	VERIFICARE la conferma BIMl con la conferma ricevuta dal die	B	8	Strutturati 1,45

G_r_OpVar_ordinata_indentata

Questo Report non serve per il risultato finale del calcolo del VaR Operativo, ma è utile quando siamo interessati solo a vedere il tracciato ordinato e indentato. Infatti in questo Report non rientrano i calcoli e le stime ma solo i risultati dell'algorithm.

Tracciato ordinato e indentato

<i>finame</i>	<i>figarent</i>	<i>type liv</i>
BANCAIMI Business Lines	BANCAIMI Business Lines	P 0
Investment Banking	Investment Banking	P 1
Trading & Sales	Trading & Sales	P 2
Market Making e Gestione Posizione	Trading & Sales	P 3
Negoziazione Conto Proprio e MM	Market Making e Gestione Posizione	P 4
PRODOTTI STRUTTURATI CP	Negoziazione Conto Proprio e MM	P 5
Spunta conferme e riconciliazione (Prodotti strutturati)	PRODOTTI STRUTTURATI CP	P 6
CORREGGERE l'operazione in MxG2000	Spunta conferme e riconciliazione (Prodotti strutturati)	P 7
Errato input manuale nei sistemi: correzione su MxG2000	CORREGGERE l'operazione in MxG2000	B 8
ARCHIVIARE la documentazione relativa all'operazione	Spunta conferme e riconciliazione (Prodotti strutturati)	P 7
Perdita/ smarrimento documenti archiviati	ARCHIVIARE la documentazione relativa all'operazione	B 8
RICEVERE la conferma modificata	Spunta conferme e riconciliazione (Prodotti strutturati)	P 7
Mancato chasing delle conferme della ctp	RICEVERE la conferma modificata	B 8
CORREGGERE le istruzioni di regolamento	Spunta conferme e riconciliazione (Prodotti strutturati)	P 7
Errata trascrizione dati (risultati spunta telefonica)	CORREGGERE le istruzioni di regolamento	B 8
VERIFICARE la conferma BIMi con la conferma ricevuta dal cliente	Spunta conferme e riconciliazione (Prodotti strutturati)	P 7
Mancata/non corretta riconciliazione conferme ctp	VERIFICARE la conferma BIMi con la conferma ricevuta dal cliente	B 8
Gestione distinta in MxG2000 delle componenti delle strutture	VERIFICARE la conferma BIMi con la conferma ricevuta dal cliente	B 8
INVIARE la conferma firmata	Spunta conferme e riconciliazione (Prodotti strutturati)	P 7
Mancato rispetto disp. interne (conferme non firmate) (b)	INVIARE la conferma firmata	B 8
VERIFICARE i flussi di cassa con la ctp	Spunta conferme e riconciliazione (Prodotti strutturati)	P 7
Mancata tracciabilità spunta telefonica	VERIFICARE i flussi di cassa con la ctp	B 8
Mancata comunicazione intra-servizio sulla presenza di pdt strutturati	VERIFICARE i flussi di cassa con la ctp	B 8
Disallineamento tra evidenze BO e strumenti prezzati con FLEX	VERIFICARE i flussi di cassa con la ctp	B 8
Ricezione flussi contabili (MXRATES) 26	PRODOTTI STRUTTURATI CP	P 6

3.7 ANALISI DELLA COMPLESSITA'

L'Analisi della complessità dell'algoritmo serve per vedere come l'aumentare dei dati trattati influenza il tempo di calcolo.

L'analisi che è stata compiuta può essere divisa in due aree:

- **Analisi Teorica**
- **Analisi Empirica**

L'analisi Teorica misura la complessità dell'algoritmo in termini di numero di operazioni che vengono svolte. In pratica abbiamo spezzettato l'algoritmo in tanti piccoli moduli e siamo andati a vedere esattamente quante operazioni vengono svolte in ogni modulo, infine abbiamo rimesso insieme i pezzi e sommato tutte le operazioni che venivano svolte. In questo modo avevamo una fotografia il più esatto possibile di quante fossero effettivamente le operazioni svolte dal computer. Ma non solo questo: l'importanza di questo studio infatti non stava solo nel capire quante operazioni svolgesse il computer in un dato momento, ma capire come queste operazioni crescevano col crescere dell'albero di guasto a cui l'algoritmo era sottoposto.

L'analisi empirica guarda invece, come dice la parola stessa, i fatti. In quest'analisi non prendiamo più in considerazione quanti calcoli deve fare la cpu (la parte che fa i calcoli in un computer), ma andiamo a vedere direttamente quanto tempo ci mette. E' chiaro che questo tipo di analisi sarà tanto più accurata quanti più sono i campioni a disposizione per poter testare il sistema. Nel nostro caso, grazie anche all'aiuto di alcuni colleghi che ci hanno aiutato a creare alberi di diverse dimensioni per procedere con i test, i campioni a disposizione sono ben 9 e spaziano da alcuni molto piccoli di 26 o 47 elementi fino ad un albero molto grande composto di 854 elementi. Avendo un numero più che soddisfacente di campioni a disposizione il prossimo passo è stato naturalmente quello di mettere alla prova l'algoritmo con tutti questi alberi e di misurarne i risultati. Poiché l'algoritmo è diviso in due moduli completamente separati e con funzioni completamente diverse, allora anche l'analisi empirica ha dovuto tener conto di questo separando l'analisi dei due settori. Per questo si avrà l'analisi del primo modulo per tutte le numerosità, l'analisi del secondo modulo anche questo per tutte le

numerosità e infine l'analisi del tempo totale dato dalla somma dei due tempi precedenti. Grazie alla divisione in due parti possiamo essere certi che i risultati dell'analisi siano attendibili con un buon margine, infatti essendo ogni modulo indipendente sarebbe potuto capitare benissimo di avere un modulo con complessità N^2 , ovvero la cui complessità cresce in modo quadratico, e l'altro modulo con complessità N , ovvero lineare, ciò avrebbe creato possibili disagi e incomprensioni in un'analisi in blocco che invece appaiono chiare da un'analisi scissa.

3.7.1 ANALISI TEORICA

In questa analisi siamo andati a spezzettare l'algoritmo in ogni sua procedura per vedere in modo concreto quanti calcoli vengono svolti. Quello che viene fatto in pratica consiste in studiare nel dettaglio che cosa fanno i vari cicli "FOR" e "WHILE" che costituiscono l'algoritmo.

L'analisi, come ormai dovrebbe essere chiaro è divisa in primo e secondo modulo. Qui partiamo con l'analisi del primo modulo:

PRIMO MODULO

Do While Not rst BE.EOF : Questo primo ciclo prende in considerazione tutti i Basic Event che da un'analisi empirica fatta sui dati possono essere quantificati come circa il 45% della totale numerosità del tracciato che chiamerò N da qui in avanti. **Complessità = $0.45 \cdot N$** .

Do While Not ftparent = "": Questo ciclo scorre tutto il tracciato in cerca del padre dell'evento da cui si è partiti nel ciclo precedente e una volta trovato gli assegna il livello e in seguito cerca il padre del nodo appena trovato. Così via finché non raggiunge il TopGate. **Complessità = numero dei livelli**

Il fatto di scorrere il tracciato finché non viene trovato il padre dev'essere considerato come una complessità inferiore in termini di tempo di calcolo, infatti non viene eseguita alcuna operazione durante lo scorrimento e quindi la procedura non ne viene appesantita in modo importante. Potremmo quindi concludere che la

$$\text{Complessità} = \text{numero dei livelli} \times C_1 + C_2 \times N/2$$

Dove $C_2 = o(C_1)^5$, quindi la complessità del ciclo dipende prevalentemente dal numero dei livelli.

Do While Not rst 1.EOF: Questo ciclo ripercorre ogni ramo del tracciato assegnando i livelli in modo definitivo. **Complessità = numero di livelli**

⁵ C_2 è un o piccolo di C_1

La **Complessità totale del primo modulo** è quindi:

$$0.45 \cdot N \cdot (\text{numero dei livelli} \times C_1 + C_2 \times N/2) \cdot \text{numero dei livelli} \\ \approx 0.45 \times C_1 \times \text{numero_dei_livelli}^2 \times N$$

Se chiamiamo quindi $k_1 = 0.45 \times C_1 \times \text{numero_livelli}^2$, allora la complessità del primo modulo è una funzione lineare rispetto alla numerosità.

$$t = k_1 \times N$$

Secondo modulo

Do While Not rst Query.EOF: Questo ciclo crea una matrice con i dati di tutto il tracciato e i livelli che sono stati assegnati nel modulo precedente. **La complessità è quindi = N**

```

For i = 0 To N
  For i1 = 0 To N
    For i2 = 0 To N
      For i3 = 0 To N
        For i4 = 0 To N
          For i5 = 0 To N
            For i6 = 0 To N
              For i7 = 0 To N

```

For i8 = 0 To N

Anche se a prima vista siamo di fronte a 9 cicli annidati ed è quindi da aspettarsi che la complessità sia di N^9 , questa deduzione sarebbe errata. Non siamo infatti di fronte a N^9 operazioni, ma come nei casi precedenti, abbiamo a che fare con cicli che scorrono il tracciato in cerca di un dato che soddisfi la condizione IF. Allora quello che ci deve interessare è sapere quante volte in questi cicli il programma compie effettivamente delle operazioni. Questa quantità è uguale alla numerosità dei dati, quindi N .

Gli altri N^9 cicli di ricerca influenzano il tempo di calcolo nuovamente come un o piccolo. Possiamo quindi esprimere la complessità attraverso questa funzione:

$$\text{Complessità} = N \times C_1 + N^9 \times C_2$$

Dove ancora una volta $C_2 = o(C_1)$.

La complessità totale del secondo modulo è quindi:

$$t = N \times C_1 + N^9 \times C_2 + N \approx N + N$$

Anche nel secondo modulo quindi, analizzato separatamente, la complessità è una funzione lineare della numerosità.

$$\text{Complessità Totale} \approx k_1 \times N + 2 \times N$$

$$\text{Quindi raccogliendo risulta } t \approx (k_1 + 2) \times N$$

Quindi poiché K_1 e 2 sono due costanti per i nostri fini, allora possiamo assumere che $C = K_1 + 2$. Quindi possiamo riassumere che:

$$\text{Complessità Totale} \approx C \times N$$

Quindi possiamo concludere che la Complessità totale dell'algoritmo è del tipo $O(n)$, quindi lineare.

L'analisi dei cicli dell'algoritmo ci dice che la relazione che lega l'aumento della numerosità e l'aumento del tempo di calcolo è lineare.

Proviamo adesso ad analizzare l'algoritmo attraverso il tempo di calcolo per vedere se le conclusioni che abbiamo tratto finora vengono confermate.

3.7.2 ANALISI EMPIRICA, IL TEMPO DI CALCOLO

L'analisi condotta tenendo conto solo del tempo di calcolo ha portato a risultati univoci, soprattutto quando la numerosità dei dati è superiore al centinaio: la complessità cresce in modo lineare.

Il tempo totale di calcolo dell'algoritmo è stato diviso in due tempi parziali, il primo tempo è relativo al primo modulo nel quale viene assegnato ad ogni dato il proprio livello, il secondo tempo è relativo al modulo che crea l'ordinamento e l'indentatura. Ho preferito analizzare i

due moduli separatamente per vedere se vi fosse qualche differenza di comportamento nelle due parti del programma.

Dati	t_1	t_2	tempo_tot
26	0,16	0,05	0,21
47	0,37	0,13	0,51
106	1,62	0,24	1,87
164	2,60	0,37	2,97
207	3,28	0,47	3,76
349	5,50	0,75	6,26
525	8,92	1,16	10,08
652	11,63	1,53	13,17
854	14,56	2,16	16,72

1. CONFRONTO FRA I RAPPORTI

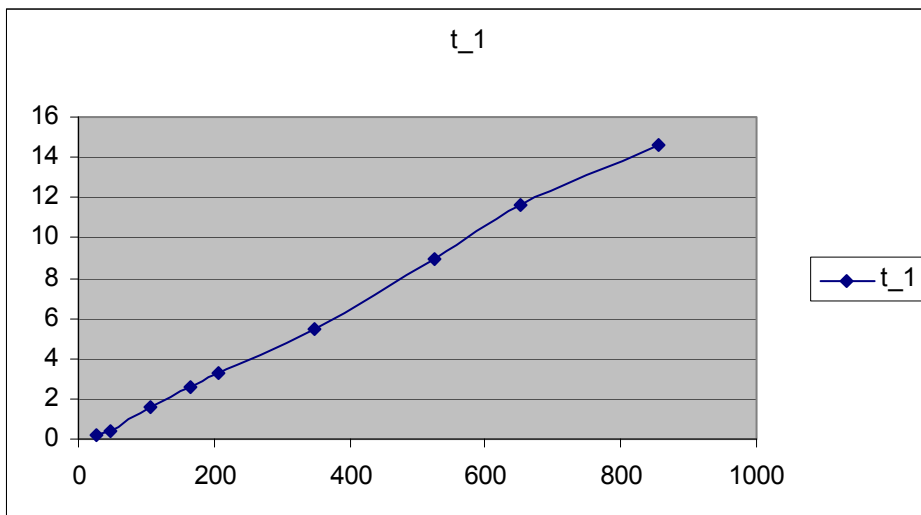
Una prima analisi è stata condotta confrontando i rapporti fra i tempi di calcolo e fra i dati. Si può vedere facilmente che, a parte i primi due rapporti, che usano numerosità basse, i rapporti fra i dati e fra i tempi sono pressoché coincidenti. Questa prima analisi macroscopica tende quindi a farci pensare a un rapporto 1 a 1 fra la crescita della numerosità e la crescita del tempo impiegato per il calcolo. Quindi all'aumentare di una unità i dati ne segue l'aumento di una unità di tempo costante.

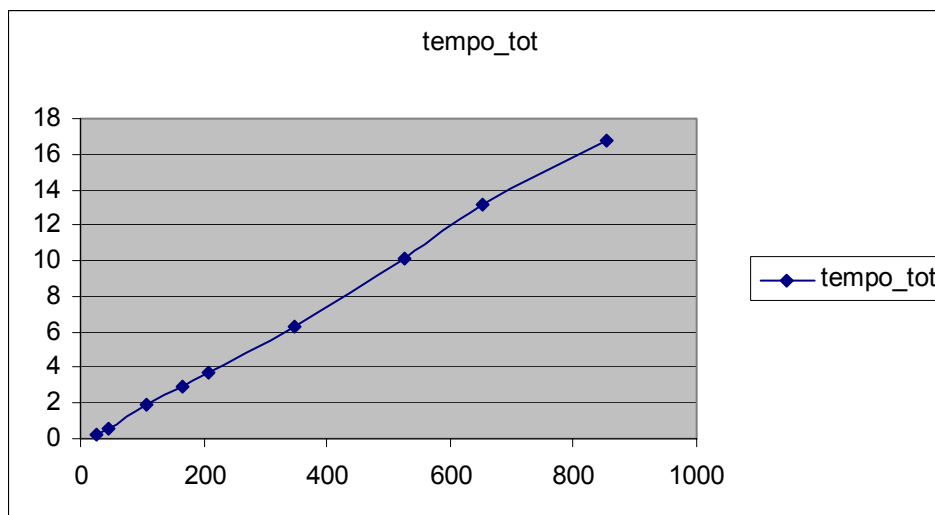
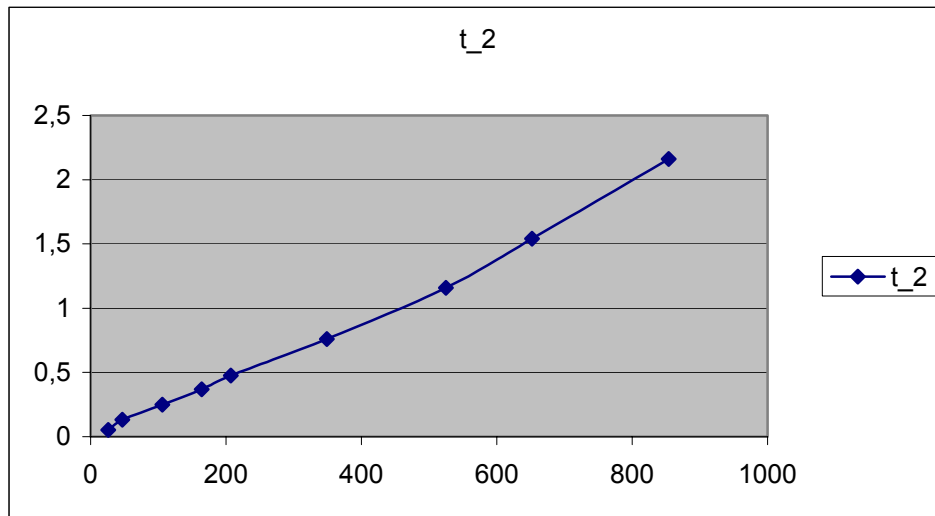
	rapp dati	rapp t_1	rapp t_2	rapp t_tot
854/26	32,84	89,55	41,65	77,96
854/47	18,17	38,42	16,27	32,68
854/106	8,056	8,94	8,69	8,91

854/164	5,20	5,59	5,84	5,62
854/207	4,12	4,43	4,52	4,44
854/349	2,44	2,64	2,85	2,67
854/525	1,62	1,63	1,86	1,65
854/652	1,30	1,25	1,40	1,26

2. ANALISI GRAFICA

L'analisi grafica mostra in modo inequivoco che il **rapporto** fra crescita della numerosità e aumento del tempo di calcolo è **lineare** sia nel primo modulo che nel secondo che nel programma nella sua interezza.





Dopo questa analisi usando 9 campioni con diverse numerosità ogni prova ci fa tendere verso l'accettazione di quella che potremmo definire l'ipotesi nulla di Relazione Lineare.

Un'ulteriore analisi che abbiamo voluto condurre per accertarci di aver colto correttamente la complessità dell'algoritmo è stata l'analisi attraverso regressione.

3.7.3 REGRESSIONE

Proviamo comunque a fare dei test per assicurarci che quanto detto sia effettivamente vero, proviamo a supporre che esista una relazione di tipo quadratico fra la numerosità e il tempo di calcolo. Immaginiamo un modello di questo tipo:

$$t = c_1 \times N + c_2 \times N^2$$

Proviamo a fare una regressione fra i tempi calcolati empiricamente con l'algoritmo e questo modello. Se c_2 risulta essere non significativamente diverso da zero allora potremmo avere un test a favore della tesi di un rapporto lineare e non quadratico tra numerosità e tempo.

I dati su cui faremo la regressione sono i seguenti:

- nella prima colonna abbiamo le numerosità dei tracciati presi ad esame nei test,
- nella seconda abbiamo i quadrati e
- nella terza i tempi di calcolo che sono stati necessari per il processo.

N	N ²	t
26	676	0,16
47	2209	0,37
106	11236	1,62
164	26896	2,60
207	42849	3,28
854	729316	14,56

OUTPUT RIEPILOGO

<i>Statistica della regressione</i>	
R multiplo	0,9994091
R al quadrato	0,9988186
R al quadrato corretto	0,7485233
Errore standard	0,2084687
Osservazioni	6

ANALISI VARIANZA

	<i>gdl</i>	<i>SQ</i>	<i>MQ</i>	<i>F</i>	<i>Significatività</i>
Regressione	2	146,97525	73,487624	1690,9559	2,639E-05
Residuo	4	0,1738369	0,0434592		
Totale	6	147,14908			

	<i>Coefficienti</i>	<i>Errore stand</i>	<i>Stat t</i>	<i>p-value</i>	<i>Inferiore 95%</i>	<i>Superiore 95%</i>
Intercetta	0	#N/D	#N/D	#N/D	#N/D	#N/D
N	0,015	0,0009044	16,635814	7,649E-05	0,0125341	0,0175561
N^2	0,00001	1,115E-06	2,1117468	0,1022888	-7,414E-07	5,452E-06

OUTPUT RESIDUI

<i>Osservazione</i>	<i>Previsto</i>	<i>t</i>	<i>Residui</i>
1	0,3927645	-0,2301645	
2	0,7123222	-0,3334222	
3	1,621245	0,006255	
4	2,5307464	0,0707536	
5	3,2152616	0,0679384	
6	14,566374	-0,005474	

Dalla regressione risulta che il coefficiente di N^2 non è significativamente diverso da zero, infatti ha un p-value=0.1022888 e il suo intervallo di confidenza passa per lo zero.

Possiamo quindi concludere che il modello quadratico non sia un buon modello.

Anche questo test dimostra che il vero modello è quello lineare.

CONCLUSIONE

L'Algoritmo che abbiamo visto, è un algoritmo che permette una visita ad un albero in preordine. In preordine significa che l'albero viene visitato partendo dalla radice, andando prima nel sottoalbero sinistro e poi nei sottoalberi via via più a destra.

Come abbiamo visto l'algoritmo è diviso in due parti, nella prima parte viene assegnato un livello ad ogni nodo e nella seconda si usano le informazioni acquisite per riordinare l'albero e generare l'indentatura.

Abbiamo poi analizzato la complessità, ovvero come cresce il tempo di calcolo al crescere delle dimensioni dell'albero, e abbiamo visto che la Complessità è del tipo $O(n)$, ovvero cresce in modo lineare. Questo risultato è molto incoraggiante perché significa che con l'aumentare del numero dei nodi che compongono l'albero il tempo di calcolo cresce in modo lineare. Quindi anche quando lavoriamo con alberi molto grandi, il suo tempo di calcolo rimane nell'ordine di alcuni secondi. Le prove che abbiamo fatto certificano infatti che l'algoritmo impiega circa 16 secondi per riordinare alberi con più di 800 nodi, sulle macchine che abbiamo avuto a disposizione (normale personal computer). Un risultato assolutamente soddisfacente.

Una cosa forse si potrebbe aggiungere a questo punto. Bisogna notare che per come è stato costruito l'algoritmo, esso può essere slegato completamente dalla struttura in cui è nato, ovvero per il calcolo dei rischi operativi.

Questo significa che l'algoritmo può ordinare anche altri alberi purché gli siano date le informazioni minime di cui ha bisogno, ovvero il nome del nodo o della foglia, il nome del padre e il tipo (se è nodo o foglia). Effettivamente siamo orgogliosi di essere stati già partecipi del

primo uso dell'Algoritmo in un ambito diverso da quello per il quale era stato progettato. Infatti, proprio alcuni giorni prima che scrivessimo queste pagine, l'algoritmo è stato adattato per un altro progetto presso l'azienda Engineering, con lo scopo di ordinare l'albero delle Query di un complesso database.

ALLEGATO 1

Esempio di Albero degli Eventi di Guasto non Ordinato

Name	Parent
BANCA IMI Business Lines	
Investment Banking	BANCA IMI Business Lines
Banking	Investment Banking
Agency Services	Banking
Custodia	Agency Services
Corporate Finance	Investment Banking
Trading & Sales	Investment Banking
Custodia e Amministrazione Titoli	Custodia
Gestione c/correnti accessori	Custodia
AMM.NE TITOLI E GESTIONE CONTO CLIENTELA IN GESTIONE C/CORRENTI ACCESSORI	Custodia e Amministrazione Titoli Gestione c/correnti accessori
Apertura, aggiornamento e chiusura c/c GESBANK	GESTIONE C/CORRENTI ACCESSORI
Regolamento (Custodia e Amministrazione Titoli)	AMM.NE TITOLI E GESTIONE CONTO
Gestione dividendi, cedole e aumenti di capitale (CAT)	AMM.NE TITOLI E GESTIONE CONTO
Market Making e Gestione Posizione	Trading & Sales
Tesoreria	Trading & Sales
Sales	Trading & Sales
Tesoreria e Funding	Tesoreria
Municipal/ Government Finance	Corporate Finance
Merchant Banking	Corporate Finance
Negoziazione Conto Proprio e MM	Market Making e Gestione Posizione
DERIVATI SU CAMBI OTC CP	Negoziazione Conto Proprio e MM
DERIVATI SU TASSI OTC CP	Negoziazione Conto Proprio e MM
OBBLIGAZIONI ITALIA, BOND ILLIQUIDI, IMIDIRECT	Negoziazione Conto Proprio e MM
DERIVATI SU TITOLI E INDICI LISTED ITALIA CP	Negoziazione Conto Proprio e MM
PRODOTTI STRUTTURATI CP	Negoziazione Conto Proprio e MM
TITOLI DI STATO ITALIA CP	Negoziazione Conto Proprio e MM
DERIVATI SU TITOLI E INDICI OTC CP	Negoziazione Conto Proprio e MM
TITOLI DI STATO ESTERO CP	Negoziazione Conto Proprio e MM
TITOLI AZIONARI ITALIA CP	Negoziazione Conto Proprio e MM
TITOLI AZIONARI ESTERO CP	Negoziazione Conto Proprio e MM
EUROBOND, BOND ILLIQUIDI, IMIDIRECT CP	Negoziazione Conto Proprio e MM
COVERED WARRANT E CERTIFICATI CP	Negoziazione Conto Proprio e MM
SPOT/FORWARD CP	Negoziazione Conto Proprio e MM
DERIVATI SU TITOLI E INDICI LISTED ESTERO CP	Negoziazione Conto Proprio e MM
PCT SU FIXED INCOME (23)	Tesoreria e Funding
DEPOSITI INTERBANCARI (22)	Tesoreria e Funding
STOCK LENDING (25)	Tesoreria e Funding
FRA, IRS e FOREX SWAP (21)	Tesoreria e Funding
EQUITY REPO (24)	Tesoreria e Funding
ATTIVARE la procedura informatica per la moviment. c/c e	Regolamento (Custodia e Amministrazione
RICEVERE il messaggio di buon esito del trasferimento	Regolamento (Custodia e Amministrazione
VERIFICARE la capienza dei conti del cliente	Regolamento (Custodia e Amministrazione

AGGIORNARE le condizioni di conto	Apertura, aggiornamento e chiusura c/c
AGGIORNARE il file Excel	Apertura, aggiornamento e chiusura c/c
INSERIRE i dati per l'apertura dei rapporti di conto	Apertura, aggiornamento e chiusura c/c
INSERIRE la data di fine validità	Apertura, aggiornamento e chiusura c/c
Gestione dividendi, cedole e aumenti di capitale (cash Italia)	TITOLI AZIONARI ITALIA CP
Quadratura stanze e contabilizzazione (Cash)	TITOLI AZIONARI ITALIA CP
Quotazione (CW)	COVERED WARRANT E
Quadratura stanze e contabilizzazione (Cash)	COVERED WARRANT E
Regolamento (derivati OTC CP e TF - Strutturati - Spot/Forward)	DEPOSITI INTERBANCARI (22)
Ricezione flussi contabili (MXRATES)	DEPOSITI INTERBANCARI (22)
Regolamento (derivati OTC CP e TF - Strutturati - Spot/Forward)	DERIVATI SU CAMBI OTC CP
Invio conferme (derivati su cambi OTC)	DERIVATI SU CAMBI OTC CP
Contatto (derivati su cambi OTC)	DERIVATI SU CAMBI OTC CP
Esercizio opzioni e gestione scadenze futures (derivati su cambi)	DERIVATI SU CAMBI OTC CP
Input e Validazione (derivati su cambi OTC- Spot/Forward)	DERIVATI SU CAMBI OTC CP
Ricezione flussi contabili (MXRATES)	DERIVATI SU CAMBI OTC CP
Esecuzione (derivati su cambi OTC)	DERIVATI SU CAMBI OTC CP
Spunta conferme e riconciliazione (derivati su cambi OTC)	DERIVATI SU CAMBI OTC CP
Ricezione flussi contabili (MXRATES)	DERIVATI SU TITOLI E INDICI OTC
Rivalutazione fixing differenziali (derivati titoli e indici OTC)	DERIVATI SU TITOLI E INDICI OTC
Esercizio opzioni e gestione scadenze futures (derivati titoli e indici)	DERIVATI SU TITOLI E INDICI OTC
Esecuzione (derivati su titoli e indici OTC)	DERIVATI SU TITOLI E INDICI OTC
Regolamento (derivati OTC CP e TF - Strutturati - Spot/Forward)	DERIVATI SU TITOLI E INDICI OTC
Spunta conferme e riconciliazione (derivati su titoli e indici OTC)	DERIVATI SU TITOLI E INDICI OTC
Invio conferme (derivati su titoli e indici OTC)	DERIVATI SU TITOLI E INDICI OTC
Quadratura stanze e contabilizzazione (Cash)	EUROBOND, BOND ILLIQUIDI,
Spunta conferme e riconciliazione (Prodotti strutturati)	PRODOTTI STRUTTURATI CP
Ricezione flussi contabili (MXRATES)	PRODOTTI STRUTTURATI CP

ALLEGATO 2

Esempio di Albero degli Eventi di Guasto Ordinato		
Name	Parent	Liv
BANCA IMI Business Lines		0
Investment Banking	BANCA IMI Business Lines	1
Trading & Sales	Investment Banking	2
Market Making e Gestione Posizione	Trading & Sales	3
Negoziazione Conto Proprio e MM	Market Making e Gestione Posizione	4
SPOT/FORWARD CP	Negoziazione Conto Proprio e MM	5
Input e Validazione (derivati su cambi OTC- VALIDARE l operazione in MxG2000 288	SPOT/FORWARD CP	6
Mancata validazione dei deals nei sistemi di FO	Input e Validazione (derivati su cambi	7
INSERIRE i dati relativi all operazione nel foglio	VALIDARE l operazione in MxG2000	8
Errato input manuale (parametri operazione) 288	Input e Validazione (derivati su cambi	7
Ricezione flussi contabili (MXRATES) 286	INSERIRE i dati relativi all operazione	8
ATTIVARE la query di investigation in PERRY	SPOT/FORWARD CP	6
Inefficienza procedurale: mancata identificazione	Ricezione flussi contabili (MXRATES)	7
ATTIVARE le procedure informatiche per l	ATTIVARE la query di investigation in	8
Errata trasformazione in msg SWIFT dei file	Ricezione flussi contabili (MXRATES)	7
Regolamento (derivati OTC CP e TF - Strutturati - IMPORT estratti conto da banche agenti e saldi da	ATTIVARE le procedure informatiche	8
Indisponibilità dei sistemi di collegamento verso	SPOT/FORWARD CP	6
EFFETTUARE il matching manuale degli incassi c	Regolamento (derivati OTC CP e TF - IMPORT estratti conto da banche	7
Non corretta identificazione del beneficiario dell	agenti e	8
AUTORIZZARE il rilascio del pagamento in	Regolamento (derivati OTC CP e TF -	7
Generazione messaggi di pagamento con errori	EFFETTUARE il matching manuale	8
ELABORARE la procedura di contabilizzazione	Regolamento (derivati OTC CP e TF -	7
Disallineamento tra sistemi di pagamento e	AUTORIZZARE il rilascio del	8
ATTIVARE procedura per il netting dei flussi 287	Regolamento (derivati OTC CP e TF -	7
Errato inserimento manuale dei dati nei sistemi	ELABORARE la procedura di	8
AGGIORNARE le istruzioni di regolamento 287	Regolamento (derivati OTC CP e TF - Strutturati - Spot/Forward) 287	7
Errato inserimento manuale dei dati nei sistemi	ATTIVARE procedura per il netting	8
PRODOTTI STRUTTURATI CP	Regolamento (derivati OTC CP e TF -	7
Spunta conferme e riconciliazione (Prodotti	AGGIORNARE le istruzioni di	8
CORREGGERE l operazione in MxG2000	Negoziazione Conto Proprio e MM	5
Errato input manuale nei sistemi: correzione su	PRODOTTI STRUTTURATI CP	6
ARCHIVIARE la documentazione relativa all	Spunta conferme e riconciliazione	7
Perdita/ smarrimento documenti archiviati	CORREGGERE l operazione in	8
RICEVERE la conferma modificata	Spunta conferme e riconciliazione	7
Mancato chasing delle conferme della ctp	ARCHIVIARE la documentazione	8
CORREGGERE le istruzioni di regolamento	Spunta conferme e riconciliazione	7
Errata trascrizione dati (risultati spunta	RICEVERE la conferma modificata	8
VERIFICARE la conferma BIMBI con la conferma	Spunta conferme e riconciliazione	7
	CORREGGERE le istruzioni di	8
	Spunta conferme e riconciliazione	7

Mancata/non corretta riconciliazione conferme	VERIFICARE la conferma BIMI	8
Gestione distinta in MxG2000 delle componenti	VERIFICARE la conferma BIMI	8
INVIARE la conferma firmata	Spunta conferme e riconciliazione	7
Mancato rispetto disp. interne (conferme non	INVIARE la conferma firmata	8
VERIFICARE i flussi di cassa con la ctp	Spunta conferme e riconciliazione	7
Mancata tracciabilità spunta telefonica	VERIFICARE i flussi di cassa con	8
Mancata comunicazione infra-servizio sulla	VERIFICARE i flussi di cassa con	8
Disallineamento tra evidenze BO e strumenti	VERIFICARE i flussi di cassa con	8
Ricezione flussi contabili (MXRATES) 276	PRODOTTI STRUTTURATI CP	6
ATTIVARE la query di investigation in PERRY	Ricezione flussi contabili	7
Inefficienza procedurale: mancata identificazione	ATTIVARE la query di	8
ATTIVARE le procedure informatiche per l	Ricezione flussi contabili	7
Errata trasformazione in msg SWIFT dei file	ATTIVARE le procedure	8
Invio conferme (Prodotti strutturati)	PRODOTTI STRUTTURATI CP	6
VERIFICARE la correttezza dei dati tecnici	Invio conferme (Prodotti strutturati)	7
Mancata/non corretta verifica dati conferma di	VERIFICARE la correttezza dei dati	8
AUTORIZZARE l invio della conferma	Invio conferme (Prodotti strutturati)	7
Mancato rispetto disp. interne (conferme non	AUTORIZZARE l invio della	8
Regolamento (derivati OTC CP e TF - Strutturati -	PRODOTTI STRUTTURATI CP	6
IMPORT estratti conto da banche agenti e saldi da	Regolamento (derivati OTC CP e TF	7
Indisponibilità dei sistemi di collegamento verso	IMPORT estratti conto da banche	8
EFFETTUARE il matching manuale degli incassi	Regolamento (derivati OTC CP e TF	7
Non corretta identificazione del beneficiario dell	EFFETTUARE il matching manuale	8
AUTORIZZARE il rilascio del pagamento in	Regolamento (derivati OTC CP e TF	7
Generazione messaggi di pagamento con errori	AUTORIZZARE il rilascio del	8
ELABORARE la procedura di contabilizzazione	Regolamento (derivati OTC CP e TF	7
Disallineamento tra sistemi di pagamento e	ELABORARE la procedura di	8
ATTIVARE procedura per il netting dei flussi 279	Regolamento (derivati OTC CP e TF	7
Errato inserimento manuale dei dati nei sistemi	ATTIVARE procedura per il netting	8
AGGIORNARE le istruzioni di regolamento 279	Regolamento (derivati OTC CP e TF	7
Errato inserimento manuale dei dati nei sistemi	AGGIORNARE le istruzioni di	8
Esecuzione (Prodotti strutturati)	PRODOTTI STRUTTURATI CP	6
ELABORARE la proposta	Esecuzione (Prodotti strutturati)	7
Errata parametrizzazione FLEX	ELABORARE la proposta	8

APPENDICE 3

IL CODICE

COMPLETO DELL'ALGORITMO

Option Compare Database

Option Explicit

Public time_ini, time_fin, time_tot, time_medio, time_parz

Sub trovalivelli()

Dim dbs As Database

Dim rst_BE As Recordset, rst_ALL As Recordset, rst_Query As Recordset

Dim sql_BE As String, sql_ALL As String, sql_Query As String

Dim i As Integer

Dim i1 As Integer, i2 As Integer, i3 As Integer, i4 As Integer, i5 As Integer

Dim i6 As Integer, i7 As Integer, i8 As Integer, i9 As Integer, i10 As Integer

Dim j As Long

Dim N As Integer

Dim Livello As Integer

Dim ftparent As String, ftparent1 As String, ftparent2 As String, ftparent3 As String

Dim ftparent4 As String, ftparent5 As String, ftparent6 As String, ftparent7 As String

'-----

'Per riordinamento livelli '|

Dim rst_1 As Recordset '|

Dim sql_1 As String '|

Dim n_Livelli As Integer '|

'-----

'Questa procedura serve per assegnare ad ogni nodo o evento il proprio livello

DoCmd.SetWarnings False

'calcolo il tempo del processo

time_ini = Timer

'Do al ftparent del TopGate il Valore "", nel caso importo albero parziale

DoCmd.OpenQuery "G_Q_riga_ARIS_FT_tracciato_temp_selez"

DoCmd.OpenQuery "G_Q_riga_ARIS_FT_tracciato_temp_agg"

'Assegno a sql_BE tutti i basic event

Set dbs = CurrentDb

sql_BE = "SELECT fname,ftp,parent,ftype FROM G_Q_trova_BE "

Set rst_BE = dbs.OpenRecordset(sql_BE)

'Assegno a sql_ALL tutto il tracciato presente in fault_tree_tracciato che può essere quello imp da Aris o Da Relex

sql_ALL = "SELECT fname,ftp,parent,ftype FROM

FAULT_TREE_TRACCIATO "

Set rst_ALL = dbs.OpenRecordset(sql_ALL)

```

rst_ALL.MoveLast
N = rst_ALL.RecordCount
'Assegno Alla Matrice una dimensione che dipende dal tracciato
importato
ReDim OpVar_ini(N * 4, 3) As Variant, OpVar(N * 4, 3) As Variant

```

```

Livello = 0
i = 0

```

```

'Inizializzo tabella che conterrà il tracciato con i livelli corretti
(TopGate=0, poi a crescere)
dbs.Execute "DELETE * FROM G_TAB_OpVar_ini_Livelli_ok"

```

```

'La procedura parte dalle foglie per arrivare al TopGate e calcola i livelli
dal basso verso l'alto
'(quindi 0=BE;1,2,3...i nodi fino al TopGate,poi cambierà)
rst_BE.MoveFirst

```

```

'Si parte da ogni Foglia a risalire fino alla radice
Do While Not rst_BE.EOF

```

```

'creo una matrice temporanea OpVar_ini che conterrà tutti gli elementi
ordinati dalle foglie al topgate. questa matrice è indispensabile anche se a
prima vista sembra che possa essere sostituita con rst_BE e rst_ALL.
Perchè non si possono aggiornare questi ultimi facilmente come la
matrice

```

```

OpVar_ini(i, 0) = rst_BE.Fields(0)
OpVar_ini(i, 1) = rst_BE.Fields(1)
OpVar_ini(i, 2) = rst_BE.Fields(2)
OpVar_ini(i, 3) = Livello

```

```

'tabella inizializzata a ogni ciclo. Qsta tabella contiene i singoli
percorsi dalle foglie al topGate con i livelli da invertire
dbs.Execute "DELETE * FROM G_TAB_OpVar_ini"

```

```

'accodo alla tabella G_TAB_OpVar_ini la foglia
dbs.Execute "insert into G_TAB_OpVar_ini
(ftname,ftparent,ftype,livello) VALUES" _
& "(" & CStr(OpVar_ini(i, 0)) & "," &
CStr(rst_BE.Fields(1).Value) & "," _
& "" & CStr(rst_BE.Fields(2).Value) & "," & CStr(Livello) &
");"

```

```

'Assegno a fparent il campo fparent del Basic Event
ftparent = rst_BE.Fields(1)
rst_ALL.MoveFirst

```



```

'cerca in tutto il tracciato di ricostruire un ramo partendo dalla foglia
per arrivare al TopGate
'ciclo:N/2 * N°Livelli (N/2 perchè parte ogni volta dal principio e
quando lo trova torna daccapo)
Do While Not ftparent = ""
    If ftparent = rst_ALL.Fields(0) Then ' chi ha ftparent come ffname
        Livello = Livello + 1

        OpVar_ini(i, 0) = rst_ALL.Fields(0)
        OpVar_ini(i, 1) = rst_ALL.Fields(1)
        OpVar_ini(i, 2) = rst_ALL.Fields(2)
        OpVar_ini(i, 3) = Livello
        If rst_ALL.Fields(1) = "" Then
            OpVar_ini(i, 1) = "TOPGATE"
        End If
        dbs.Execute "insert into G_TAB_OpVar_ini
(ftname,ftp parent,ftype,livello) VALUES" _
        & "(" & CStr(rst_ALL.Fields(0).Value) & "," &
CStr(OpVar_ini(i, 1)) & "," _
        & "" & CStr(rst_ALL.Fields(2).Value) & "," & CStr(Livello) &
        ");"
        i = i + 1
        ftparent = rst_ALL.Fields(1) 'dai a ftparent il valore di ftparent
appena trovato
        rst_ALL.MoveFirst
    Else
        rst_ALL.MoveNext
    End If
Loop

'-----
'Processo di aggiustamento livelli
sql_1 = "SELECT ffname,ftp parent,ftype,livello FROM
G_TAB_OpVar_ini"
Set rst_1 = dbs.OpenRecordset(sql_1)
n_Livelli = Livello
Do While Not rst_1.EOF
'Debug.Print rst_1.Fields(3).Value
N = n_Livelli - CInt(CStr(rst_1.Fields(3).Value))
dbs.Execute "Insert into
G_TAB_Opvar_ini_Livelli_ok(ftname,ftp parent,ftype,livello) VALUES"
-
        & "(" & CStr(rst_1.Fields(0).Value) & "," &
CStr(rst_1.Fields(1).Value) & "," _

```

```

        & "" & CStr(rst_1.Fields(2).Value) & "," & CStr(n_Livelli -
CInt(CStr(rst_1.Fields(3).Value))) & "");"
rst_1.MoveNext
Loop

```

```
'-----
```

```

    i = i + 1
    rst_BE.MoveNext
    Livello = 0
Loop
'Debug.Print i
time_medio = Timer
'Chiamata a subroutine
crea_matrice_opvar

```

```
End Sub
```

```

Sub crea_matrice_opvar()
Dim dbs As Database
Dim rst_BE As Recordset, rst_ALL As Recordset, rst_Query As
Recordset
Dim sql_BE As String, sql_ALL As String, sql_Query As String
Dim i As Integer
Dim i1 As Integer, i2 As Integer, i3 As Integer, i4 As Integer, i5 As
Integer
Dim i6 As Integer, i7 As Integer, i8 As Integer, i9 As Integer, i10 As
Integer
Dim j As Long
Dim N As Integer
Dim N_1 As Integer 'per4 calcolo dimensioni matrici
Dim Livello As Integer
Dim ftparent As String, ftParent1 As String, ftParent2 As String,
ftParent3 As String
Dim ftParent4 As String, ftParent5 As String, ftParent6 As String,
ftParent7 As String
'Dim OpVar_ini(10000, 3) As Variant, OpVar(10000, 3) As Variant,
OpVar_q(10000, 3)

```

```

DoCmd.SetWarnings False
Set dbs = CurrentDb
sql_Query = "SELECT ftname,ftparent,fttype,livello FROM
G_Q_OPVAR_con_livelli " _

```

```

Set rst_Query = dbs.OpenRecordset(sql_Query)
rst_Query.MoveLast

```

```

N_1 = rst_Query.RecordCount
ReDim OpVar_ini(N_1 * 4, 3) As Variant, OpVar(N_1 * 4, 3) As
Variant, OpVar_q(N_1 * 4, 3)

i = 0
rst_Query.MoveFirst
Do While Not rst_Query.EOF
    OpVar_q(i, 0) = rst_Query.Fields(0) 'OpVar_q è la matrice =alla
query q_opvar_conlivelli
    OpVar_q(i, 1) = rst_Query.Fields(1)
    OpVar_q(i, 2) = rst_Query.Fields(2)
    OpVar_q(i, 3) = rst_Query.Fields(3)
    ' Debug.Print OpVar_q(i, 0);
    ' Debug.Print " ";
    ' Debug.Print OpVar_q(i, 3)
    i = i + 1
    rst_Query.MoveNext
Loop
Debug.Print i

dbs.Execute "DELETE * FROM G_OPVAR_ordinata"
N = i 'i in questo punto è all'ultima riga della matrice opvar
j = 0
For i = 0 To N
    If OpVar_q(i, 1) = "" Then 'se è il topgate

        OpVar(j, 0) = OpVar_q(i, 0)
        OpVar(j, 1) = OpVar_q(i, 1)
        OpVar(j, 2) = OpVar_q(i, 2)
        OpVar(j, 3) = OpVar_q(i, 3)
        'Debug.Print OpVar(j, 0) & " " & OpVar(j, 3)
        If OpVar(j, 1) = "" Then
            OpVar(j, 1) = "TOPGATE"
        End If
        dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES"
        & "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," &
CStr(OpVar(j, 1)) & ","
        & "" & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) & ");"
        ftparent = OpVar(0, 0)
        j = j + 1
        If j <= N Then
            For i1 = 0 To N
                If OpVar_q(i1, 1) = ftparent Then
                    OpVar(j, 0) = OpVar_q(i1, 0)
                    OpVar(j, 1) = OpVar_q(i1, 1)
                End If
            Next i1
        End If
    End If
Next i

```

```

OpVar(j, 2) = OpVar_q(i1, 2)
OpVar(j, 3) = OpVar_q(i1, 3)
dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES" _
& "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," &
CStr(OpVar(j, 1)) & "," _
& "" & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) & ");"
j = j + 1
ftParent1 = OpVar_q(i1, 0)
For i2 = 0 To N
If OpVar_q(i2, 1) = ftParent1 Then
OpVar(j, 0) = OpVar_q(i2, 0)
OpVar(j, 1) = OpVar_q(i2, 1)
OpVar(j, 2) = OpVar_q(i2, 2)
OpVar(j, 3) = OpVar_q(i2, 3)
dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES" _
& "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," &
CStr(OpVar(j, 1)) & "," _
& "" & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) & ");"
j = j + 1
ftParent2 = OpVar_q(i2, 0)
For i3 = 0 To N
If OpVar_q(i3, 1) = ftParent2 Then
OpVar(j, 0) = OpVar_q(i3, 0)
OpVar(j, 1) = OpVar_q(i3, 1)
OpVar(j, 2) = OpVar_q(i3, 2)
OpVar(j, 3) = OpVar_q(i3, 3)
dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES" _
& "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," &
CStr(OpVar(j, 1)) & "," _
& "" & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) &
");"
j = j + 1
ftParent3 = OpVar_q(i3, 0)
For i4 = 0 To N
If OpVar_q(i4, 1) = ftParent3 Then
OpVar(j, 0) = OpVar_q(i4, 0)
OpVar(j, 1) = OpVar_q(i4, 1)
OpVar(j, 2) = OpVar_q(i4, 2)
OpVar(j, 3) = OpVar_q(i4, 3)
dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES" _
& "(" & CStr(j) & "," & CStr(OpVar(j, 0)) & "," &
CStr(OpVar(j, 1)) & "," _

```

```

& "" & CStr(OpVar(j, 2)) & ", " & CStr(OpVar(j, 3))
& ");"
    j = j + 1
    ftParent4 = OpVar_q(i4, 0)
    For i5 = 0 To N
    If OpVar_q(i5, 1) = ftParent4 Then
        OpVar(j, 0) = OpVar_q(i5, 0)
        OpVar(j, 1) = OpVar_q(i5, 1)
        OpVar(j, 2) = OpVar_q(i5, 2)
        OpVar(j, 3) = OpVar_q(i5, 3)
        dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,fttype,livello) VALUES"
        & "(" & CStr(j) & ", " & CStr(OpVar(j, 0)) & ", " &
CStr(OpVar(j, 1)) & ", "
        & "" & CStr(OpVar(j, 2)) & ", " & CStr(OpVar(j,
3)) & ");"
    j = j + 1
    ftParent5 = OpVar_q(i5, 0)
    For i6 = 0 To N
    If OpVar_q(i6, 1) = ftParent5 Then
        OpVar(j, 0) = OpVar_q(i6, 0)
        OpVar(j, 1) = OpVar_q(i6, 1)
        OpVar(j, 2) = OpVar_q(i6, 2)
        OpVar(j, 3) = OpVar_q(i6, 3)
        dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,fttype,livello) VALUES"
        & "(" & CStr(j) & ", " & CStr(OpVar(j, 0)) & ", "
& CStr(OpVar(j, 1)) & ", "
        & "" & CStr(OpVar(j, 2)) & ", " & CStr(OpVar(j,
3)) & ");"
    j = j + 1
    ftParent6 = OpVar_q(i6, 0)
    For i7 = 0 To N
    If OpVar_q(i7, 1) = ftParent6 Then
        OpVar(j, 0) = OpVar_q(i7, 0)
        OpVar(j, 1) = OpVar_q(i7, 1)
        OpVar(j, 2) = OpVar_q(i7, 2)
        OpVar(j, 3) = OpVar_q(i7, 3)
        dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,fttype,livello) VALUES"
        & "(" & CStr(j) & ", " & CStr(OpVar(j, 0)) &
", " & CStr(OpVar(j, 1)) & ", "
        & "" & CStr(OpVar(j, 2)) & ", " &
CStr(OpVar(j, 3)) & ");"
    j = j + 1
    ftParent7 = OpVar_q(i7, 0)

```

```

                For i8 = 0 To N
                If OpVar_q(i8, 1) = ftParent7 Then
                    OpVar(j, 0) = OpVar_q(i8, 0)
                    OpVar(j, 1) = OpVar_q(i8, 1)
                    OpVar(j, 2) = OpVar_q(i8, 2)
                    OpVar(j, 3) = OpVar_q(i8, 3)
                    dbs.Execute "insert into G_OPVAR_ordinata
(ID,ftname,ftparent,ftype,livello) VALUES"
                    & "(" & CStr(j) & "," & CStr(OpVar(j, 0))
& "," & CStr(OpVar(j, 1)) & "," & CStr(OpVar(j, 2)) & "," & CStr(OpVar(j, 3)) & ");"
                    j = j + 1
                End If
                Next
            End If
        Next
    End If
Next
End If
Next
End If
Next
End If
Next
End If
Next
End If
Next
End If
Next
End If
Next
time_fin = Timer
time_parz = time_medio - time_ini
time_tot = time_fin - time_ini '(Hour(time_fin - time_ini) / 3600) +
Minute(time_fin - time_ini)/60 +

DoCmd.OpenQuery "G_Q_OPVAR_ordinata_togli_TOPGATE "

'DoCmd.OpenReport "G_r_OpVar_ordinata_indentata", acViewPreview
End Sub

```

BIBLIOGRAFIA

- *INTRODUCTION TO ALGORITHMS, SECOND EDITION.*
Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest,
Clifford Stein.
- *THE ART OF COMPUTER PROGRAMMING. VOLUME
1,FUNDAMENTAL ALGRITHMS, THIRD EDITION.* Donald E.
Knuth.
- *AMMINISTRAZIONE &FINANZA ORO, LE MODERNE
TECNICHE DI RISK MANAGEMENT:IL VAR,* Fernando
Metelli, Daniela De Febis, Paolo Fiorini Morosini, Gianfranco
Tabasso
- *VISUAL BASIC 6, GUIDA COMPLETA.* Greg Perry
- *VISUAL BASIC, LA PROGRAMMAZIONE DEI DATABASE.*
Giuseppe Naccarato
- *NUOVO ACCORDO DI BASILEA SUI REQUISITI
PATRIMONIALI,* Documento ai fini di consultazione. Comitato
di Basilea per la Vigilanza Bancaria.
- *OPERATIONAL RISK, Consultative Document.* Basel
Committee on Banking Supervision.

- BASILEA 2: PROBLEMI E OPPORTUNITÀ PER LE PMI, in Venezia Opportunità, 2003. P. Biffis, G. Levorato. Mezzani Editori.
- DATA STRUCTURES AND ALGORITHMS IN JAVA,second edition. Michael T.Goodrich, Roberto Tomassia

WEBGRAFIA

- www.bis.org (documenti ufficiali del Comitato di Basilea)
- www.bancaitalia.it (versione italiana dei documenti del Comitato di Basilea)
- www.basilea2.com
- www.basilea2.it
- <http://www-math.science.unitn.it/~luminati/>
- <http://homes.dico.unimi.it/~pighizzi>
- <http://www.itg-rondani.it/dida/Matem/ipermonica/insiemi/relazioni/>
- <http://www.dipmat.unipg.it/~bartocci/mat/cap3-2.doc>
- <http://www.matematicamente.it/approfondimenti/>
- <http://digilander.libero.it/capano/lezioni>
- <http://google.it> (per la ricerca di tutti i siti elencati)