



MINISTERO DELLO SVILUPPO ECONOMICO
DIREZIONE GENERALE PER LA LOTTA ALLA CONTRAFFAZIONE
UFFICIO ITALIANO BREVETTI E MARCHI

DOMANDA NUMERO	101998900713073
Data Deposito	27/10/1998
Data Pubblicazione	27/04/2000

Sezione	Classe	Sottoclasse	Gruppo	Sottogruppo
G	06	F		

Titolo

MEMORIA PER LA RICERCA DI INFORMAZIONI MEDIANTE ANALISI DI PREFISSI IN PARTICOLARE PER LA REALIZZAZIONE DI TABELLE DI INSTRADAMENTO IN NODI DI RETI DI COMUNICAZIONE AD ALTA VELOCITA', QUALI LA RETE INTERNET.

Descrizione dell'invenzione avente per titolo:

"Memoria per la ricerca di informazioni mediante analisi di prefissi, in particolare per la realizzazione di tabelle di instradamento in nodi di reti di comunicazione ad alta velocità, quali la rete Internet"

a nome CSELT - Centro Studi e Laboratori Telecomunicazioni S.p.A., Via G. Reiss Romoli 274, 10148 Torino, nazionalità italiana.

Inventori: Enrica FILIPPI

Viviana INNOCENTI

Domanda N.

T 0 98A

000009

Depositata il

27 OTT. 1998

CSELT
Centro Studi e Laboratori Telecomunicazioni S.p.A.

==.==.==.==.==

La presente invenzione ha per oggetto una memoria per la ricerca di informazioni mediante analisi di prefissi, destinata in particolare all'impiego per la realizzazione di tabelle di instradamento in reti di comunicazione ad alta velocità, preferibilmente per la rete Internet.

I protocolli di comunicazione per le moderne reti ad alta velocità, quali le reti ATM, la rete Internet ecc., si basano sulla trasmissione di pacchetti informativi contenenti, nell'intestazione, informazioni relative all'indirizzo di destinazione del pacchetto. In un nodo della rete, alla ricezione di un pacchetto, gli organi di controllo dell'instradamento utilizzano l'indirizzo per effettuare una ricerca all'interno di una opportuna tabella o base dati e individuare l'interfaccia di uscita a cui deve essere fornito il pacchetto per l'inoltro a un organo successivo della rete. Attualmente, velocità di trasmissione dell'ordine dei Gbit/s sono comuni e i tempi di instradamento in un nodo devono essere quindi molto brevi per non costituire un ostacolo al regolare svolgimento del traffico. Per fornire un'indicazione quantitativa, si può considerare

che una trasmissione a 1 Gbit/s con pacchetti di circa 1000 bit richiede che il nodo smaltisca circa 1 milione di pacchetti al secondo: tenuto conto dei tempi di commutazione, dei tempi necessari per eventuali elaborazioni dell'intestazione dei pacchetti e di eventuali tempi di guardia tra pacchetti consecutivi, si vede che la ricerca nella tabella di instradamento, che è l'operazione più lunga, deve essere compiuta in un tempo alquanto inferiore a 1 μ s. La situazione ovviamente si complica al crescere della velocità di trasmissione.

Facendo riferimento a titolo di esempio e per semplicità di descrizione alla rete Internet, in essa si sta verificando una crescita esponenziale sia del numero dei calcolatori ospiti collegati alla rete, sia del traffico sia ancora delle esigenze di banda, anche a seguito dello sviluppo continuo di nuove applicazioni che sfruttano la rete Internet. Tutto questo porta come conseguenza un aumento delle dimensioni delle tabelle che rende più lunga l'operazione di ricerca. Ora, è ben noto ai tecnici che, mentre è sempre più facile disporre di quantità di memoria elevate a costi contenuti, la riduzione dei tempi di accesso a memorie di grandi dimensioni comporta considerevoli problemi tecnologici.

Per consentire un più agevole adattamento all'espansione della rete, gli indirizzi Internet possono essere organizzati secondo una scala gerarchica, nel senso che gruppi di indirizzi corrispondenti p. es. a una stessa regione geografica o a uno stesso fornitore condividono i bit più significativi (prefisso) dell'indirizzo. Questi prefissi hanno lunghezza variabile e possono a loro volta costituire la parte più significativa di un altro prefisso (prefissi di prefissi). Maggiori dettagli sulla struttura degli indirizzi Internet possono essere trovati per esempio nei documenti "Das Internet-Protokoll der nächsten Generation" di H. P. Giesiger, Comtec 5, 1998, pagg. 20 - 26, o "IP Next Generation Overview", di R. M. Hinden, datato 14 Maggio 1995 e disponibile al sito

Internet <http://playground.sun.com/pub/ipng/html/INET-IPng-Paper.html>.

Per ridurre la complessità della ricerca, si tiene conto del fatto che le informazioni di instradamento nella rete sono relative soprattutto all'indirizzo del nodo successivo (indirizzo comunemente indicato con il termine di "next hop" o passo successivo) nel cammino che porta alla destinazione del pacchetto che si sta elaborando. Siccome i possibili passi successivi diversi in una tabella di instradamento sono nell'ordine del centinaio, è possibile estrapolare le informazioni su di essi e memorizzarle in un'altra struttura dati. Date le piccole dimensioni, a questa si può accedere con tecniche di accesso diretto tramite un indice, denominato in letteratura "bersaglio" (TARGET), che viene individuato a partire dall'indirizzo del pacchetto ricevuto. In pratica, in un organo di gestione dell'instradamento, utilizzando come chiave di ricerca l'indirizzo di destinazione del pacchetto si accede ad una tabella di inoltro (forwarding) che è un sottoinsieme della tabella completa di instradamento di una rete e in cui ogni riga contiene un prefisso (coppia indirizzo-maschera, dove la maschera indica il numero di bit significativi dell'indirizzo, cioè i bit che costituiscono il prefisso) ed un corrispondente dato (il bersaglio, solitamente un puntatore all'interfaccia di uscita, più altre informazioni). Un prefisso dà origine a una concordanza per la chiave di ricerca (cioè permette di raggiungere un bersaglio) se l'AND bit a bit della chiave con la maschera associata all'indirizzo è uguale all'indirizzo stesso.

La variabilità della lunghezza dei prefissi origina un'ulteriore complicazione per il fatto che la ricerca dell'indice può portare al raggiungimento di più bersagli, in quanto il confronto può dare esito positivo con più prefissi di lunghezza diversa. In queste condizioni il bersaglio effettivo è quello corrispondente al prefisso più lungo, cioè quello più specifico per la destinazione. Questo criterio è noto come "longest prefix

match", cioè concordanza sul prefisso più lungo.

Teoricamente, il metodo più veloce ed efficiente per realizzare una ricerca con concordanza sul prefisso più lungo è quello di memorizzare le voci in una o più memorie CAM (Content Addressable Memory = memoria indirizzabile mediante il contenuto) ternarie. In una CAM ternaria, ciascun bit della memoria può assumere valori: 0, 1 e "indifferente" (don't care). Il confronto tra i bit del registro di confronto ed i bit di memoria posti a livello logico indifferente risulta ovviamente sempre positivo: è pertanto possibile implementare implicitamente le maschere associate ad un dato semplicemente ponendo a don't care gli opportuni bit del dato stesso. L'inconveniente di questo tipo di dispositivi è rappresentato principalmente dal loro prezzo elevato e dalla loro scarsa integrazione.

Per poter usare delle memorie tradizionali a basso costo è allora necessario utilizzare algoritmi di ricerca su basi di dati che siano al contempo veloci e compatti. La velocità di un algoritmo di ricerca è strettamente collegata al numero di accessi in memoria, mentre la compattezza è determinata dalla quantità di memoria necessaria a memorizzare la struttura dati utilizzata. L'obiettivo finale è quello di ridurre il più possibile sia il numero di accessi sia la memoria richiesta per memorizzare le informazioni per la ricerca.

Le implementazioni pratiche degli algoritmi di instradamento adatti a soddisfare il criterio della concordanza sul prefisso più lungo possono essere sia di tipo software che di tipo hardware. In generale, esse devono comunque soddisfare a un certo numero di requisiti comuni, tra i quali:

- velocità di ricerca indipendente per quanto possibile dalle dimensioni della tabella;
- piccola varianza delle prestazioni dell'algoritmo tra il caso peggiore e il caso migliore,

- facilità di aggiornamento incrementale o localizzato della tabella (cioè l'inserzione o la cancellazione di alcune voci non devono richiedere la riscrittura dell'intera tabella o di parti sostanziali della stessa);
- flessibilità, in modo da adattarsi facilmente a futuri sviluppi tecnologici e a variazioni nell'organizzazione degli indirizzi,
- organizzazione della memoria per quanto possibile regolare, per evitare sprechi.

Sono già state proposte numerose soluzioni per il problema dell'instradamento nelle reti Internet in modo da tener conto nella necessità di realizzare la concordanza del prefisso più lungo.

Un primo gruppo di soluzioni si basa su un algoritmo noto come "PATRICIA", che è un algoritmo per la memorizzazione, indicizzazione e ritrovamento di informazioni in archivi di grandi dimensioni, in particolare cataloghi di biblioteche. I principi di questo algoritmo sono descritti nell'articolo "PATRICIA - Practical Algorithm To Retrieve Information Coded In Alphanumeric", di D. R. Morrison, Journal of the Association for Computing Machinery, Vol. 15. N. 4., Ottobre 1968, pagg. 514 e segg. e la sua applicazione alla soluzione dei problemi di instradamento nella rete Internet è ampiamente documentata nella letteratura tecnica relativa. L'algoritmo Patricia è sostanzialmente un algoritmo ad albero dei prefissi costruito sui bit dell'indirizzo, in cui ogni bit corrisponde a un nodo. I bit dell'indirizzo vengono esaminati uno alla volta. Per evitare di esaminare nodi che corrispondono a diramazioni non esistenti, ogni bit è associato a un'indicazione del bit successivo da esaminare. A fine ricerca si effettua un controllo sull'esistenza effettiva del prefisso. Poiché ogni nodo corrisponde a uno o più accessi alla memoria, e tenuto conto che l'accesso alla memoria è molto più lento delle elaborazioni sui dati, l'algoritmo consente effettivamente una riduzione dei tempi complessivi di ritrovamento dell'indirizzo.

Lo svantaggio essenziale delle soluzioni basate sull'algoritmo PATRICIA è che una implementazione pratica va tarata sul caso peggiore (in pratica, prevedere il passaggio per tutti i nodi dell'albero), mentre nella rete Internet odierna si è visto che statisticamente si devono analizzare una ventina di nodi prima di raggiungere il bersaglio (indirizzo cercato). Pertanto, queste soluzioni sono intrinsecamente abbastanza lente e inoltre presentano una forte varianza tra il caso peggiore e il caso tipico.

Per accelerare il processo di ricerca, sono anche stati proposti algoritmi in cui a ogni passo di ricerca lungo l'albero vengono presi in considerazione più bit dell'indirizzo invece di uno solo, ed è stato descritto, p. es. nell'articolo "Putting Routing Tables in Silicon", di T. Pei e C. Zukowski, IEEE Network Magazine, Gennaio 1992, pagg. 42 e segg., come questi algoritmi possano essere implementati in hardware mediante le cosiddette "memorie trie". Il problema principale che si presenta in questo caso è che le memorie trie convenzionali possono essere utilizzate solo se non vi sono concordanze multiple.

Pertanto, molte proposte di algoritmi di ricerca in tabelle di instradamento per le reti Internet, destinati a ridurre la quantità di memoria necessaria per la tabella o comunque a velocizzare la ricerca, si basano su una ristrutturazione dei prefissi in modo da eliminare le concordanze multiple: in questo modo gli algoritmi stessi sarebbero teoricamente implementabili con l'uso di memorie trie.

Esempi di ristrutturazione dei prefissi sono descritti p. es. da M. Degermark e altri nella memoria "Small Forwarding Tables for Fast Routing Lookups" presentata alla conferenza ACM SIGCOMM '97, Cannes (Francia), 16-18 Settembre 1997, e da V. Srinivasan e G. Varghese nella memoria "Faster IP Lookups using Controlled Prefix Expansion" presentata alla conferenza ACM SIGMETRICS '98.

La soluzione proposta da Degermark et al. porta ad avere un albero a tre livelli (che coprono rispettivamente 16, 8 e 8 bit dell'indirizzo) in cui ogni livello è memorizzato in una struttura particolarmente compatta, resa possibile da una opportuna ristrutturazione dei prefissi la quale fa sì che ogni prefisso nell'albero individui un intervallo di indirizzi. Questa soluzione riduce drasticamente la quantità di memoria richiesta e i tempi di accesso alla memoria stessa per la ricerca rispetto alla soluzione basata su PATRICIA (circa 400 ns per il caso peggiore che prevede 12 accessi); tuttavia la riorganizzazione dei prefissi rende impossibile l'aggiornamento incrementale, ma richiede ad ogni aggiornamento la riscrittura completa della memoria: l'aggiornamento è quindi molto lento e può creare problemi al normale esercizio della rete.

La soluzione proposta da V. Srinivasan e G. Varghese riorganizza i prefissi in modo tale che siano possibili solo un numero limitato di lunghezze degli stessi (eventualmente variabili), ed espande i prefissi più brevi sostituendoli con l'insieme dei prefissi aventi il valore di lunghezza prefissato più vicino. La tabella di instradamento risulta trasformata in un certo numero di sottotabelle ciascuna delle quali memorizza dei puntatori alla tabella successiva e la cui lunghezza è ottimizzata in modo da ridurre, a parità di numero di accessi, la memoria necessaria per quello specifico insieme di prefissi. Con questa disposizione si ha in generale una riduzione della quantità di memoria utilizzata, anche se le tabelle non risultano in generale con lunghezza prevedibile e quindi non si ha uno sfruttamento ottimale della memoria fisica. Inoltre, la struttura dati risultante è ottimizzata per uno specifico insieme di prefissi e quindi è poco generale.

Secondo la presente invenzione si fornisce una struttura che garantisce uno sfruttamento ottimale della memoria fisica e che consente un aggiornamento

incrementale della tabella di instradamento (o in generale dell'insieme di informazioni).

La memoria secondo l'invenzione comprende:

un primo elemento di memoria che memorizza un insieme di voci di informazione associate ognuna ad una maschera indicativa del numero di caratteri significativi di un rispettivo prefisso, e a un bersaglio, che costituisce un dato eventualmente utilizzabile per l'accesso a un ulteriore insieme di informazioni, e

organi di controllo per l'esecuzione delle operazioni di ricerca di una specifica informazione in memoria e per l'aggiornamento della memoria, detti organi di controllo operando mediante il confronto tra porzioni successive, di lunghezza prefissata, di una stringa di caratteri ricevuta in ingresso e corrispondenti porzioni dei prefissi memorizzati, detti prefissi potendo avere lunghezza variabile e non multipla della lunghezza di dette porzioni;

ed è caratterizzata dal fatto che, per la realizzazione di un criterio di ricerca basato sulla concordanza del prefisso più lungo:

ciascuna cella è suddivisa in un campo informativo che fornisce o un indirizzo di riga successiva per la continuazione di una ricerca o un'informazione relativa a un bersaglio raggiunto, e una coppia di indicatori che specificano il contenuto del campo informativo,

è previsto un vettore ausiliario, che ha tante celle quante sono le righe della memoria e che, quando gli indicatori di detta cella del primo elemento di memoria segnalano il raggiungimento di un bersaglio in concomitanza con la necessità di proseguire la ricerca in una riga successiva, è atto a memorizzare l'informazione di bersaglio nella cella associata a detta riga successiva, ogni cella di detto vettore ausiliario comprendendo un campo informativo e una coppia di indicatori identici a

quelli delle celle di detto primo elemento di memoria.

L'invenzione riguarda anche un procedimento per la gestione di detta memoria.

A maggior chiarimento si fa riferimento ai disegni allegati, in cui:

- la fig. 1 è uno schema di principio di un nodo di instradamento nella rete Internet,
- la fig. 2 è uno schema dell'organizzazione della memoria secondo l'invenzione;
- le figure 3A - 3c sono rappresentazioni dell'organizzazione dei dati in una cella della memoria,
- le figure 4 - 5 sono diagrammi delle operazioni di ricerca all'interno della memoria di Fig. 2;
- la fig. 6 è un esempio numerico di ricerca,
- le figure 7 - 15 sono diagrammi di flusso delle operazioni di inserimento e cancellazione dati nella memoria secondo l'invenzione.

Con riferimento alla fig. 1, un nodo di instradamento (router) di una rete quale la rete Internet può essere schematizzato da un insieme di interfacce di ingresso IF_a...IF e di interfacce di uscita IF_b...IF_y, ognuna connessa da un lato a linee indicate nel complesso con a, x, b, y, e dall'altro a un organo di commutazione SW, associato a un dispositivo di controllo o processore di rete PR. In generale le interfacce di ingresso comprendono memorie temporanee per i dati in arrivo (non rappresentate in figura) e organi di trattamento delle informazioni destinati a gestire la ricerca di instradamento nel nodo, cioè l'individuazione dell'interfaccia di uscita a cui va inoltrato un pacchetto in arrivo a un'interfaccia d'entrata. Il processore PR dovrà invece compiere tutte le altre elaborazioni richieste per la gestione delle comunicazioni e la gestione del nodo. In altri nodi la ricerca dell'instradamento può però essere assegnata a unità di trattamento specifiche. Gli organi di ricerca dell'instradamento sono stati indicati nel loro complesso con FE. Per poter svolgere tale ricerca, gli

organi FE dovranno contenere una versione locale della tabella di instradamento, relativa ai prefissi delle destinazioni raggiungibili dal nodo stesso. Come accennato nell'introduzione, tali tabelle possono essere strutturate su due livelli (e quindi far uso di due dispositivi di memoria): nel primo livello si individua a partire dall'indirizzo del pacchetto un indice o bersaglio, che è quello che corrisponde al prefisso più lungo compreso in tabella per quell'indirizzo di destinazione, mentre nel secondo con tale indice si accede direttamente all'informazione richiesta per l'inoltro del pacchetto. I due dispositivi sono rappresentati per l'interfaccia IFx, dove sono indicati rispettivamente con M1, M2. Con CT sono indicati i dispositivi di elaborazione e controllo che permettono la ricerca in tabella e l'aggiornamento della tabella.

L'invenzione riguarda una realizzazione della memoria M1 sotto forma di una cosiddetta memoria trie multibit, estesa per realizzare la concordanza sul prefisso più lungo.

Nella figura 2 è rappresentata la struttura logica della memoria oggetto dell'invenzione, supponendo a titolo di esempio che a ogni passo di ricerca si consideri un massimo di k bit del prefisso. Come si vede, la memoria trie è costituita, in modo convenzionale, da una matrice con $L = 2^k$ colonne, ciascuna corrispondente a una delle possibili combinazioni k a k dei bit dell'indirizzo, e un numero R di righe che dipende dal numero e dalla distribuzione degli indirizzi e dal valore di k . Ciascuna cella della memoria trie può essere vuota (cioè corrispondere a una combinazione non prevista in nessuno degli indirizzi memorizzati) oppure contenere l'indirizzo di una riga successiva da raggiungere e/o l'indicazione relativa al fatto che si è raggiunto un bersaglio.

Nel caso in cui la ricerca debba basarsi sulla concordanza del prefisso più lungo, le due indicazioni di bersaglio raggiunto o di passaggio a una riga successiva

non sono mutuamente esclusive e di questo va tenuto conto nell'organizzazione dei dati. Per questo scopo sarebbe ovviamente possibile adottare una struttura in cui le celle comprendono un campo per l'indirizzo della riga successiva e un campo per l'indicazione di bersaglio raggiunto, ma tale soluzione rappresenterebbe un inutile spreco di memoria in quanto la presenza simultanea delle due indicazioni è un'eccezione e non la regola, cosicché per la quasi totalità delle celle uno dei campi sarebbe vuoto. Non è conveniente neppure la soluzione descritta in letteratura, che prevede di ristrutturare o espandere i prefissi in modo da eliminare le concordanze multiple e portare i bersagli al livello finale dell'albero, in quanto una tale soluzione richiede dimensioni di memoria molto maggiori soprattutto se il numero di voci nella tabella è grande e rende molto difficile l'aggiornamento incrementale.

Per poter soddisfare alle esigenze dette sopra, ogni cella della memoria secondo l'invenzione è formata da un campo informativo unico e da una coppia di indicatori (GO e TARGET) la cui combinazione definisce il significato del contenuto del campo informativo. Il campo informativo può comprendere per esempio 22 bit (nel caso attuale di indirizzi Internet su 32 bit), cosicché ogni cella contiene complessivamente 3 byte.

Il primo indicatore, GO, permette di stabilire se la ricerca deve o no proseguire, mentre il secondo indicatore, TARGET, indica l'esistenza o meno di un bersaglio valido. Se GO ha valore 1, il campo di 22 bit contiene l'indicazione della prossima riga da considerare nella ricerca di un indirizzo, sia nella fase di instradamento di un pacchetto che in quelle di inserimento e eliminazione di voci della tabella (campo NEXT_ROW). Se GO ha valore 0, il campo di 22 bit non contiene informazione valida se l'indicatore TARGET ha anch'esso valore 0, altrimenti è interpretabile come l'indicazione di un bersaglio valido da fornire in uscita come risultato della ricerca. In

questo caso il campo di 22 bit è composto dai seguenti sottocampi:

VALUE: 9 bit; codifica l'identità effettiva del bersaglio;

MASK: 4 bit che codificano la maschera associata al bersaglio memorizzato nel sottocampo VALUE. Se k è il passo di ricerca, il valore di MASK può variare da 0 a $k-1$. Un valore inferiore a $k-1$ indica un prefisso la cui lunghezza non è multipla di k .

T_BAK: 9 bit di "bersaglio proprio" (o di supporto): con questo termine si intende un bersaglio passibile di ricopertura, cioè un prefisso di lunghezza non multipla del passo di ricerca che può venire nascosto da un insieme di prefissi di lunghezza maggiore. L'informazione sull'esistenza di questo bersaglio più corto va conservata per impedire che, dopo la cancellazione di un prefisso dalla riga corrispondente della memoria, una ricerca dia esito non corretto, come si spiegherà meglio in seguito. La cella in cui va memorizzata l'informazione T_BAK è determinata con una semplice relazione matematica illustrata più avanti. Nel caso in cui non esista alcun prefisso ricopribile, a T_BAK si assegna un valore prestabilito NULL: p. es., supponendo che la tabella possa dare accesso a un massimo di 512 (cioè 2^9) bersagli, il valore NULL può essere 511, cioè 2^9-1 . Si fa notare che, per memorizzare l'informazione T_BAK, si sfrutta il fatto che i campi VALUE, MASK di un bersaglio hanno lunghezza complessiva inferiore alla lunghezza del campo informativo della cella, e quindi si sfrutta una parte della cella che altrimenti resterebbe inutilizzata. La scelta di dimensioni dei campi VALUE, MASK che consentano ciò non implica perdita di generalità per la struttura.

Il caso $GO = 1$ e $TARGET = 1$ è quello che corrisponde al raggiungimento di un bersaglio e alla simultanea necessità di continuare la ricerca. Poiché, come detto, una

cella non deve contenere contemporaneamente l'indicazione di una riga da visitare e il bersaglio più lungo associato al livello raggiunto, è necessario posticipare la seconda informazione. L'informazione sul bersaglio è comune a tutte le celle della riga indicata dal campo NEXT_ROW, quindi è possibile memorizzarla "idealmente" in testa alla riga stessa. Per mantenere un numero di colonne che sia una potenza di 2, secondo l'invenzione si associa alla matrice M1 un vettore ausiliario di R celle, indicato con AUX, in cui ogni cella è associata a una riga della matrice M1 e ha la stessa struttura delle celle di M1. Tale vettore contiene per ogni riga le informazione di bersaglio che questa ha ereditato da una cella del livello precedente per effetto della necessità di memorizzare in tale cella un valore di prossima riga da visitare. Nel seguito, si userà la notazione AUX[NEXT_ROW(cella corrente)] per indicare che l'informazione di bersaglio trovata in corrispondenza di una cella va memorizzata, nel vettore ausiliario, nella riga che deve essere raggiunta partendo dalla cella stessa.

Per tutte le altre combinazioni degli indicatori GO e TARGET la cella AUX[NEXT_ROW(cella corrente)] è vuota. In conclusione, l'interpretazione dei due campi è:

GO	TARGET	Campo di 22 bit
1	0	Prossima riga da visitare;
1	1	Prossima riga da visitare; AUX[NEXT_ROW(cella corrente)] contiene un bersaglio valido per l'indirizzo
0	1	Bersaglio valido più specifico per l'indirizzo nel sottocampo VALUE (ricerca terminata)
0	0	Cella vuota (ricerca terminata)

Ritornando all'informazione T_BAK, la posizione nella memoria in cui essa va memorizzata si basa sulla considerazione che, per un passo di ricerca k, il numero di

prefissi ricopribili $\sum_{j=1}^{k-1} 2^j - 2$. Quindi è possibile memorizzare i prefissi ricopribili di una

riga nella riga stessa sfruttando il fatto che il numero di celle nella riga è proprio 2^k . I prefissi di lunghezza multipla di k non possono essere ricoperti da altri prefissi più lunghi essendo essi stessi i più lunghi memorizzabili nella riga. Esiste infatti una funzione biunivoca che associa ad ogni coppia "porzione di indirizzo di k bit, maschera della porzione inferiore a k " una cella della riga. Indicando con "index" la porzione di indirizzo e con MASK (minore di k) la maschera corrispondente alla porzione, la cella in cui il bersaglio ricopribile va memorizzato è individuata all'interno della riga dalla relazione:

$$\text{offset} = \text{index} - 1 + 2(k - \text{MASK} - 1)$$

L'utilizzo dell'informazione offset risulterà dalla descrizione dei procedimenti di inserimento e cancellazione.

Si descriveranno ora le operazioni di ricerca, inserimento e cancellazione nella memoria oggetto dell'invenzione, facendo anche riferimento ai diagrammi di flusso allegati. In questi si è adattata la seguente terminologia abbreviata:

- dest_add, dest_route = indirizzo o rispettivamente uscita di destinazione;
- prow = puntatore alla riga da raggiungere (riga corrente);
- ptarget = puntatore a un possibile bersaglio (utilizzato per tenere traccia fino alla fine della ricerca di un bersaglio potenzialmente ricopribile individuato nel corso della ricerca stessa);
- t_info = bersaglio da fornire in uscita;
- cell.go, cell.target, cell.value, cell.row, cell.mask, cell.t_bak = valore degli indicatori/campi omonimi per una cella (per semplicità di rappresentazione grafica e notazione, nei diagrammi una cella è trattata come se comprendesse tanti campi

quante sono le informazioni che deve fornire);

- maskbit = maschera della porzione di indirizzo esaminata;

1) Ricerca (Figure 4 - 6)

Per la ricerca, la memoria riceve come informazione l'indirizzo di destinazione cercato. Si entra sempre nella riga 0 della memoria e si avanza di una riga a ogni gruppo esaminato di k bit (index) della destinazione: l'analisi parte dai k bit più significativi e si procede a gruppi di k bit fino a quando il numero di bit di maschera ancora da esaminare non diventa inferiore o uguale al passo k: la cella corretta all'interno della riga corrente è quella che corrisponde al valore del gruppo in esame.

Ad ogni riga, se l'indicatore GO è a 1, la riga successiva da visitare è indicata nel campo NEXT_ROW della cella corrente. Prima di accedere alla riga successiva però si controlla l'indicatore TARGET. Se è ad 1 significa che la riga successiva ha ereditato il bersaglio dalla cella corrente in fase di inserimento, come si vedrà meglio in seguito. Tale bersaglio è quello più specifico finora trovato e di conseguenza deve essere memorizzato. L'uso del puntatore ptarget permette tale memorizzazione senza bisogno di un effettivo accesso alla memoria per scrivere un'informazione che potrebbe diventare obsoleta nel corso della ricerca.

Se l'indicatore GO è a 0, la ricerca ha termine. Per determinare l'informazione da emettere in uscita occorre tuttavia controllare l'indicatore TARGET. Se è ad 1, la cella corrente contiene il bersaglio più specifico per l'indirizzo in ingresso, altrimenti il bersaglio è già stato determinato durante la ricerca ed è memorizzato nella cella di AUX indicata da ptarget. Si noti comunque che in questo secondo caso il valore del bersaglio nella cella di AUX potrebbe essere NULL, cioè la ricerca è stata infruttuosa in quanto non esiste in memoria un prefisso valido per la destinazione cercata.

Il procedimento di ricerca è stato schematizzato in forma vettorizzata nella fig. 5

ed è illustrato anche dagli esempi numerici di Fig. 6, dove per semplicità si è considerato un indirizzo di destinazione di 8 bit e un passo $k = 2$. In questa figura G e T indicano gli indicatori GO e TARGET.

Si suppone di aver memorizzato i seguenti indirizzi, con relative maschere e bersagli:

indirizzo	maschera	bersaglio
00100000	4	A
00100111	8	B
01101000	5	C
10000000	1	D

e di dover cercare un instradamento per due messaggi i cui indirizzi sono rispettivamente 01101100 e 00100101.

Per il primo messaggio il percorso di ricerca è:

- riga 0, cella 01 --> riga successiva = 4 (GO=1, TARGET=0)
- riga 4, cella 10 --> riga successiva = 5 (GO=1, TARGET=0)
- riga 5, cella 11: in questa cella GO=0 e TARGET=1; la ricerca deve terminare anche se resta la coppia di bit 00 da esaminare: ciò significa che comunque in memoria non esistono prefissi più specifici; si è trovato un bersaglio (C, relativo al terzo dei prefissi memorizzati) e questo viene emesso in uscita; inoltre, poiché la maschera del prefisso con cui si è avuta la concordanza è 5, il prefisso (01101) copre sia la cella 10 che la cella 11 e nella prima si memorizza il valore del bersaglio anche nel campo T_BAK. E' immediato vedere che la cella 10 soddisfa alla relazione indicata in quanto precede per la memorizzazione del bersaglio in T_BAK.

Per il secondo messaggio il percorso di ricerca è

- riga 0, cella 00 --> riga successiva = 1
- riga 1, cella 10 --> a questo punto la ricerca non è conclusa (GO=1), però si ha anche target = 1, perché a questo livello si ha la concordanza con il prefisso del primo indirizzo memorizzato di bersaglio A (il secondo indirizzo memorizzato, anche se ha gli stessi bit iniziali del primo, ha maschera 8 e quindi la concordanza con esso può essere riconosciuta solo dopo l'esame di tutti e 8 i bit); pertanto si deve utilizzare il vettore AUX e nella riga di questo indicata dal campo NEXT_ROW della cella 10 di riga 1 (riga 2) si memorizzano maschera e valore del bersaglio trovato;
- riga 2, cella 01 --> riga successiva = 3
- riga 3, cella 01 --> la cella non contiene alcun dato e quindi il bersaglio da utilizzare è quello di AUX.

2) Inserimento (figure 7 - 10)

In caso di inserimento, l'informazione da caricare in memoria comprende un indirizzo, la maschera e il bersaglio (uscita associata all'indirizzo). Si procede come per la ricerca, partendo dalla riga iniziale (riga 0) della memoria e avanzando ogni volta di una riga (subroutine I_SEARCH_ROW, Figure 7, 8) fino a quando il numero di bit della maschera ancora da esaminare (maskbit) non diventa inferiore o uguale al passo k. La riga in cui si raggiunge questa condizione rappresenta la riga in cui va inserita la nuova voce (subroutine I_MODIFY_ROW, figure 7, 9)

Considerando dapprima la ricerca (fig. 8), le operazioni da compiere in una riga dipendono dal valore dell'indicatore GO. Se GO = 1, la riga esiste già e si deve semplicemente passare alla riga successiva indicata nel campo NEXT_ROW senza compiere alcuna operazione particolare. Se GO=0 la riga è da creare. La riga che si va a occupare è p. es. la prima (HEAD) di una lista di righe vuote (row_free_list) che

viene di conseguenza aggiornata (UPDATE(row_free_list)). Il modo con cui si può organizzare una lista di righe vuote associata a una memoria è ben noto al tecnico e non fa parte della presente invenzione, cosicché non è necessaria alcuna descrizione. Inoltre, se TARGET = 1, il bersaglio contenuto nella cella viene copiato nella memoria ausiliaria, AUX, all'indirizzo corrispondente alla nuova riga. In ogni caso, l'indicatore GO viene posto a 1 per indicare che ora la cella corrente contiene l'indicazione della prossima riga da visitare.

Passando ora alle operazioni da compiere nella riga di destinazione (Fig. 9) queste dipendono dal fatto che il numero di bit da esaminare sia effettivamente uguale a k o sia minore di k . Nel primo caso infatti la cella interessata dall'inserimento è unica, mentre nel secondo caso l'operazione riguarda un gruppo di celle.

Considerando il caso maskbit = k , il bersaglio non è passibile di ricopertura e verrà sicuramente memorizzato in una cella. La cella da utilizzare dipende dal valore dell'indicatore GO e sarà la cella della riga corrente della memoria se GO = 0 oppure la cella del vettore AUX indirizzata dal contenuto nella cella corrente se GO = 1. In entrambi i casi, nella cella si memorizzano il bersaglio e la maschera e, se GO=0, si mette a 1 l'indicatore target della cella del vettore AUX. Fatto ciò, si mette a 1 l'indicatore TARGET della cella della riga corrente.

Se il numero di bit di maschera è inferiore a k , il bersaglio dell'indirizzo in ingresso riguarda tutto l'intervallo di celle nella riga corrente dalla colonna "index" alla colonna indicata dal valore OFFSET ("sup" nel diagramma) - cioè tutto l'intervallo di celle che possono essere raggiunte nei k passi precedenti lungo l'albero. Per ciascuna di queste celle si compiono le operazioni della subroutine I_MODIFY_CELL (fig. 10). Anche queste operazioni sono guidate dai valori di GO e TARGET:

- se GO è 1, la cella contiene già un puntatore di riga e il bersaglio va memorizzato

- in AUX se l'indicatore TARGET è 0, oppure se TARGET è 1 ma la maschera memorizzata nella cella AUX è inferiore oppure uguale alla maschera in ingresso;
- se GO è 0 la cella non contiene puntatori di riga; se TARGET è 1, il bersaglio in ingresso viene memorizzato nella cella della memoria trie se la maschera ivi memorizzata è inferiore oppure uguale alla maschera in ingresso,
- se la cella è vuota (GO = 0 e TARGET = 0) il bersaglio viene memorizzato ponendo contemporaneamente l'indicatore TARGET a 1 e il campo T_BAK al valore NULL.

Raggiunta l'ultima cella (Fig. 9, uscita "no" da $i < \text{sup}$) si controlla ancora il relativo indicatore GO memorizzando il bersaglio nel campo T_BAK della cella della riga corrente o del vettore ausiliario a seconda che GO sia 0 o 1.

Si noti che l'operazione di modifica righe è stata descritta nell'ipotesi che la memoria sia inizializzata a 0; in caso contrario, potrebbero essere necessari controlli sul valore di T_BAK.

3) Eliminazione di una voce (figure 11 - 15)

Per l'eliminazione di una voce occorre percorrere la memoria fino ad arrivare alla riga contenente le informazioni cercate. La riga finale è indicata, come per la ricerca o l'inserimento, dal fatto che il numero di bit da esaminare è minore o uguale a k. L'eliminazione consiste nella cancellazione dalla cella o da un insieme di celle dei valori MASK, VALUE del bersaglio, sostituendoli eventualmente con i valori di un bersaglio meno specifico ricoperto da quello che si vuole cancellare.

Per compiere correttamente queste operazioni è necessario tener traccia delle righe attraversate e a questo scopo si utilizzano un vettore "step" che ha tante celle quante sono le righe della memoria e memorizza l'indirizzo di una riga attraversata (prow) e un contatore "actual" che conta le righe attraversate.

Il processo generale di cancellazione è rappresentato nella fig. 11. Ogni volta che si deve esaminare una nuova riga, si memorizza il relativo indirizzo nella opportuna posizione del vettore "step". A ogni cella raggiunta si controlla anche l'indicatore GO: se questo è 0, vuol dire che l'operazione di cancellazione riguarda un indirizzo di destinazione che in realtà non è memorizzato e ciò provoca ovviamente la fine delle operazioni. Raggiunta la riga di destinazione, si cerca se esiste un bersaglio sostitutivo nella stessa riga (subroutine D_SEARCH_T_BAK), si modifica la riga dove è inserito il prefisso utilizzando l'eventuale bersaglio sostitutivo trovato (subroutine D_MODIFY_ROW) e si effettua un controllo a ritroso sulle righe diventate vuote (subroutine D_BACKTRACE) per eliminarle.

Nella subroutine D_SEARCH_T_BAK (fig. 12), si esaminano le celle che individuano un prefisso della voce da cancellare che, per costruzione, potrebbe essere memorizzato nella riga in questione, cioè si esaminano le celle che potrebbero avere in T_BAK un possibile prefisso più corto. Questa ricerca richiede l'esame delle celle che corrispondono allo spezzone di albero di ricerca che ha condotto dalla riga precedente a quella corrente. La ricerca procede fino a quando si trova un bersaglio valido (nella cella della memoria o in quella del vettore ausiliario, a seconda del valore di GO) o fino a quando non si sono esaminate tutte le celle senza trovare il bersaglio. Se il campo T_BAK della cella in questione non ha il valore NULL si è effettivamente trovato un bersaglio sostitutivo, che viene associato alla rispettiva maschera. Il ritrovamento del bersaglio è segnalato mettendo a 1 un indicatore "found".

Per la modifica della riga (subroutine D_MODIFY_ROW, Fig. 13), come nel caso di inserimento si deve controllare se il gruppo di bit ha esattamente k bit o meno di k bit, per distinguere se la maschera identifica una sola cella o un gruppo di celle. Nel primo caso, occorre ancora distinguere se l'informazione va cercata nel vettore

AUX o nella cella della memoria vera e propria a seconda del valore di GO. Se si era trovato un bersaglio sostitutivo con la subroutine precedente (indicatore "found" a 1), lo si sostituisce al bersaglio da cancellare, mettendo, e l'operazione è terminata. Se non si era trovato alcun bersaglio sostitutivo, effettuata la cancellazione occorrerà ancora verificare che la riga non sia diventata vuota.

Nel caso di $\text{maskbit} < k$, l'eventuale voce da cancellare è memorizzata nel campo T_BAK della cella di indirizzo pari a OFFSET della memoria vera e propria o della corrispondente cella del vettore AUX. Si controllano (subroutine D_MODIFY_CELL) tutte le celle corrispondenti all'intervallo "sup". Se si trova un bersaglio da sostituire si modificano di conseguenza i campi value e MASK della cella; se non si è trovato alcun bersaglio si mettono entrambi gli indicatori GO e TARGET a 0 per indicare che la cella è vuota e si pone inoltre a NULL il campo T_BAK.

Si noti che nel corso della routine D_MODIFY_ROW si controlla anche che la richiesta non riguardi una voce inesistente (ciò che è indicato, per $\text{maskbit} = k$, da TARGET = 0 o da MASK $\neq k-1$ o value $\neq t_info$, e, per $\text{maskbit} < k$, dalla mancanza del valore del bersaglio nel campo T_BAK): in questi casi ovviamente la ricerca ha immediatamente termine.

Infine, per l'eliminazione delle righe vuote (subroutine D_BACKTRACE), si esaminano a ritroso tutte le righe esaminate nella ricerca e, per ognuna di esse, si controllano gli indicatori GO, TARGET di tutte le celle. Quando per tutte le celle di una riga hanno i due indicatori a 0, la riga è vuota e occorre allora risalire alla cella "padre" da cui si era passati alla riga vuota. A questo scopo si utilizzano il contatore ACTUAL e il vettore STEP. Se per questa cella "padre" l'indicatore TARGET è 1, occorre trasferire in essa i dati della cella corrispondente di AUX. Fatto ciò, l'identità della riga

svuotata viene passata in coda alla lista delle righe vuote ($proW = TAIL(row_free_list)$, $UPDATE\ row_free_list$) e si passa alla riga successiva, cioè contenente la cella "padre". Arrivati a una riga in cui almeno una cella non è vuota, le operazioni terminano.

Si ritiene utile fare anche alcune considerazioni sulla realizzazione hardware della memoria testé descritta. L'architettura è costituita essenzialmente da una macchina a stati (facente parte del controllore CT di Fig. 1) che comanda la sequenza di accessi alla struttura dati memorizzata in una memoria esterna (M1 in Fig. 1), che vantaggiosamente è una memoria RAM dinamica. L'architettura complessiva opera come una memoria CAM di tipo ternario. L'inizializzazione e l'aggiornamento della memoria possono invece essere controllati sia da un'altra macchina a stati sia da un microprocessore convenzionale, anch'esso incorporato per semplicità nel controllore CT. L'architettura è estremamente flessibile, in quanto può ricevere una stringa di lunghezza arbitraria, mentre l'uscita è un indice (o identificativo compresso) il cui significato è lasciato libero e dipende dal contesto applicativo e che, nel caso considerato dalla rete Internet, è solitamente un puntatore alla tabella M2 contenente le informazioni di salto successivo o $next_hop$ (indirizzi di livello 2) necessarie all'inoltro dei pacchetti con l'indirizzo destinazione fornito come identificativo di ingresso. La struttura dati esterna è composta da un numero programmabile di righe, ciascuna contenente 2^L celle tutte uguali allocate in modo contiguo, più la cella ausiliaria contenente le informazioni di bersaglio comuni a tutte le celle della riga. Per permettere di ricavare l'indirizzo delle celle per affiancamento di un indirizzo di riga più un indirizzo di cella, le celle ausiliarie di bersaglio di riga sono memorizzate separatamente - nella stessa struttura dati - per esempio dopo l'ultima riga contenente i dati veri e propri. Le dimensioni della memoria dipendono dalla

distribuzione dei prefissi e dal passo di ricerca. A titolo indicativo, una tabella di inoltro esistente con circa 40.000 voci richiede circa 14.000 righe (corrispondenti a circa 0,7 Mbyte di memoria) se $k = 4$ e circa 3.000 righe (per un totale di circa 2,4 Mbyte di memoria) se $k = 8$. Le dimensioni sono quindi perfettamente dominabili.

Da quanto detto sopra risultano chiari i vantaggi offerti dall'invenzione:

- l'adozione dei due indicatori per caratterizzare il contenuto della cella e l'uso di un vettore ausiliario per i casi eccezionali in cui si debba memorizzare un bersaglio trovato prima della fine della ricerca rendono la memoria molto compatta, ed evitano lo spreco insito in una eventuale duplicazione dei campi informativi;
- tenuto conto che il vettore ausiliario, nella realizzazione hardware, costituirà semplicemente una stringa di celle aggiuntiva alla memoria trie vera e propria, lo stesso dispositivo può essere utilizzato anche per il caso di impiego di maschere di lunghezza fissa, per le quali non si pone il problema della concordanza sul prefisso più lungo: l'unica differenza è che in questo caso il vettore ausiliario non sarà mai indirizzato; in questo caso la struttura lavora come una normale CAM binaria, sprecando un unico bit per cella;
- grazie al campo T_BAK, l'esistenza di possibili bersagli con prefisso più corto è gestita direttamente nelle operazioni di inserimento e cancellazione e queste richiedono unicamente l'esame di un numero limitato e predefinito di celle ($k-2$, se k è il passo di ricerca),
- grazie all'uso di un indicatore di fine ricerca esplicito, la struttura non è vincolata a una particolare lunghezza degli indirizzi o delle maschere, e quindi sarà utilizzabile, per quanto riguarda l'applicazione preferita, anche quando si utilizzeranno indirizzi di 128 bit;
- infine, il meccanismo di inserimento e cancellazione è particolarmente semplice e

locale, cioè non richiede la riscrittura della memoria completa o delle sue parti..

E' evidente che quanto descritto è dato unicamente a titolo di esempio non limitativo, e che varianti e modifiche sono possibili senza uscite dal campo di protezione dell'invenzione. In particolare, anche se si è fatto riferimento alla rete Internet, l'invenzione è applicabile a qualsiasi protocollo di comunicazione in cui l'instradamento dei pacchetti è basato sulla concordanza di prefissi di lunghezza fissa o variabile (ad esempio, identificazione di connessioni ATM o di indirizzi MAC). In generale l'invenzione è applicabile non solo nei nodi di sistemi di comunicazione, ma anche in tutti i casi in cui si debba compiere una ricerca di informazioni a partire da un prefisso (per esempio, ricerche su basi dati, anche se in queste in generale non vi sono le esigenze di velocità richieste dalle reti di comunicazione).

CSELT
Centro Studi e Laboratori Telecomunicazioni S.p.A.

RIVENDICAZIONI

1. Memoria per l'implementazione di un procedimento di ricerca di informazioni basato sull'analisi di prefissi costituenti la parte più significativa delle singole voci di informazione, comprendente:

un primo elemento di memoria (M1) che memorizza un insieme di dette voci di informazione associate ognuna ad un'informazione di maschera indicativa del numero di caratteri significativi del rispettivo prefisso, e a un'informazione di bersaglio, che costituisce un dato eventualmente utilizzabile per l'accesso a un ulteriore insieme di informazioni, dette voci d'informazione essendo memorizzate in una rispettive celle di memoria organizzate in righe e colonne, e

organi di controllo (CT) per l'esecuzione delle operazioni di ricerca di una specifica informazione in memoria e per l'aggiornamento della memoria, detti organi di controllo operando mediante il confronto tra porzioni successive, di lunghezza prefissata, di una stringa di caratteri ricevuta in ingresso e corrispondenti porzioni dei prefissi memorizzati, detti prefissi potendo avere lunghezza variabile e non multipla della lunghezza di dette porzioni

caratterizzata dal fatto che, per la realizzazione di un criterio di ricerca basato sulla concordanza del prefisso più lungo:

ciascuna cella comprende un campo informativo che fornisce o un indirizzo di riga successiva per la continuazione di una ricerca o un'informazione relativa a un bersaglio raggiunto, e una coppia di indicatori (GO, TARGET) che specificano il contenuto del campo informativo,

è previsto un vettore ausiliario (AUX), che ha tante celle quante sono le righe della memoria e che, quando gli indicatori di detta cella del primo elemento di memoria segnalano il raggiungimento di un bersaglio in concomitanza con la

necessità di proseguire la ricerca in una riga successiva, è atto a memorizzare l'informazione di bersaglio nella cella associata a detta riga successiva, ogni cella di detto vettore ausiliario comprendendo un campo informativo e una coppia di indicatori identici a quelli delle celle di detto primo elemento di memoria (M1).

2. Memoria secondo la riv. 1, caratterizzata dal fatto che il campo informativo di dette celle dell'elemento di memoria e di dette posizioni del vettore ausiliario è atto a memorizzare, per l'informazione di bersaglio raggiunto, l'identità del bersaglio e un valore indicativo della maschera associata a detto bersaglio.

3. Memoria secondo la riv. 2, caratterizzata dal fatto che, quando l'informazione di bersaglio raggiunto è comune a una pluralità di celle, per una cella di detta pluralità detto campo informativo è atto anche a memorizzare un'informazione ausiliaria relativa a un prefisso più corto ricopribile dal prefisso a cui è associato il bersaglio raggiunto.

4. Memoria secondo la riv. 3, caratterizzata dal fatto che detta cella è posizionata, all'interno della riga, a un indirizzo che è funzione biunivoca della porzione di stringa di ingresso esaminata, di una maschera associata a tale porzione e delle dimensioni della porzione.

5. Memoria secondo una qualsiasi delle rivendicazioni precedenti, caratterizzata dal fatto che tale memoria implementa una tabella di instradamento in un nodo di una rete di comunicazione ad alta velocità e detti prefissi sono indirizzi di destinazione o parti di indirizzi di destinazione associati a informazioni trasmesse in detta rete.

6. Memoria secondo la riv. 5, caratterizzata dal fatto che detta rete è la rete Internet.

7. Procedimento di gestione di una memoria per l'implementazione di ricerche

di informazioni basate sull'analisi di prefissi che costituiscono una parte più significativa della singola informazione, dette informazioni essendo memorizzate unitamente ad un'informazione di maschera indicativa del numero di caratteri significativi del rispettivo prefisso e a un'informazione di bersaglio, utilizzata come puntatore a un ulteriore insieme di informazioni, e detta gestione della memoria richiedendo il confronto tra porzioni successive di una stringa di caratteri ricevuta e corrispondenti porzioni dei prefissi memorizzati, detti prefissi potendo avere lunghezza variabile e non multipla della lunghezza di dette porzioni

caratterizzato dal fatto che, per la realizzazione di un criterio di ricerca basato sulla concordanza del prefisso più lungo:

a ciascuna cella dell'elemento di memoria si associa una coppia di indicatori i quali specificano in base al loro valore logico se un'operazione di ricerca termina in una riga a cui la cella appartiene o deve proseguire in una riga successiva e, in questo secondo caso, se nella riga interessata è stata riscontrata la concordanza con un possibile prefisso e si è quindi raggiunto un bersaglio,

a ciascuna riga della memoria si associa una cella di un vettore ausiliario destinato a memorizzare informazioni di bersagli raggiunti in corrispondenza di una riga che non è la riga a cui la ricerca deve terminare,

si memorizza anche, in un una cella scelta all'interno di un insieme di celle corrispondenti a prefissi di lunghezza multipla della lunghezza di dette porzioni, l'informazione di bersaglio relativa a un prefisso più corto, non multiplo della lunghezza di dette porzioni, che può venire ricoperto da detti prefissi di lunghezza multipla, detta cella essendo una cella della memoria o del vettore ausiliario a seconda se trattasi della cella terminale di una ricerca o di una cella intermedia nel corso della ricerca.

8. Procedimento secondo la riv. 7, caratterizzato dal fatto che, per la cancellazione di un'informazione, al raggiungimento della riga contenente tale informazione:

a) si cerca, all'interno di detto insieme di celle della memoria o nel vettore ausiliario, un eventuale bersaglio sostitutivo, rappresentato da un bersaglio associato a un prefisso più corto;

b) se tale bersaglio sostitutivo esiste, viene memorizzato al posto di quello da cancellare.

9. Procedimento secondo la riv. 8, caratterizzato dal fatto che si memorizza temporaneamente l'identità di tutte le righe attraversate nel corso della ricerca dell'informazione da cancellare, in caso di cancellazione di un bersaglio senza sostituzione, si effettua un controllo a ritroso sulle righe diventate vuote per effetto della cancellazione di un'informazione e se ne memorizza l'identità in una lista di righe vuote associate alla memoria.

P.P. CSELT - CENTRO STUDI E LABORATORI TELECOMUNICAZIONI S.p.A.



(dr. Franco CABELLA - Albo N. 69)



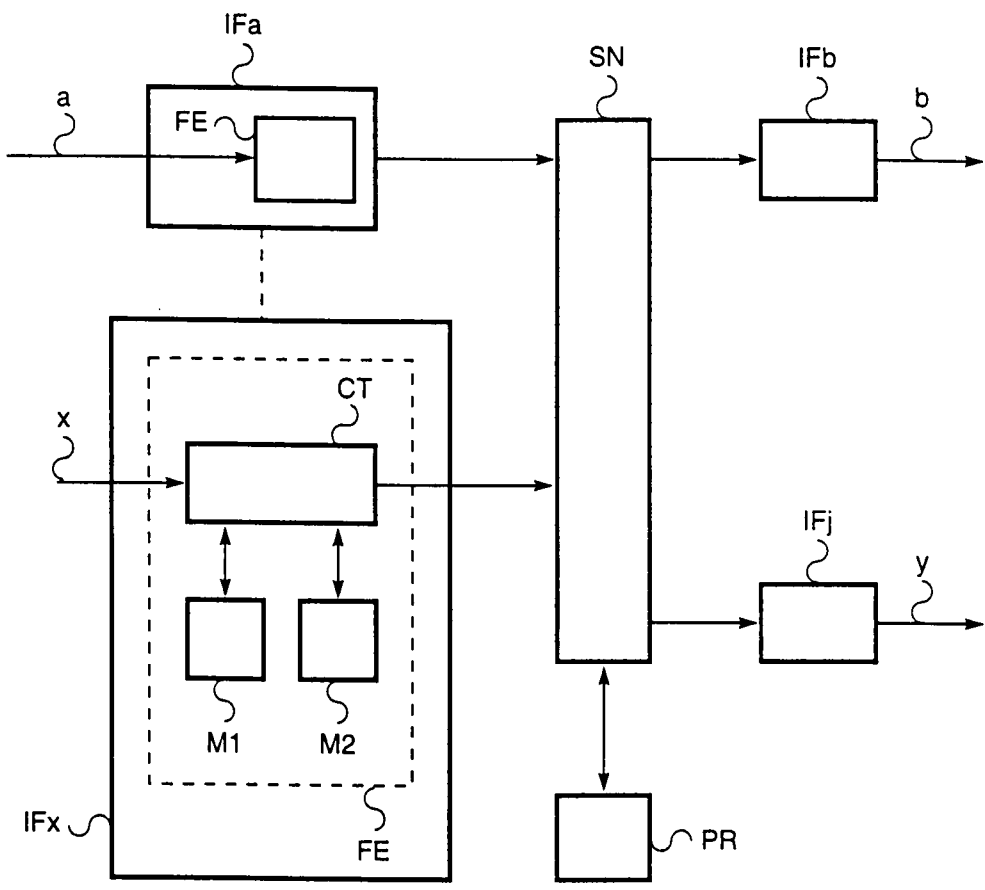


Fig. 1

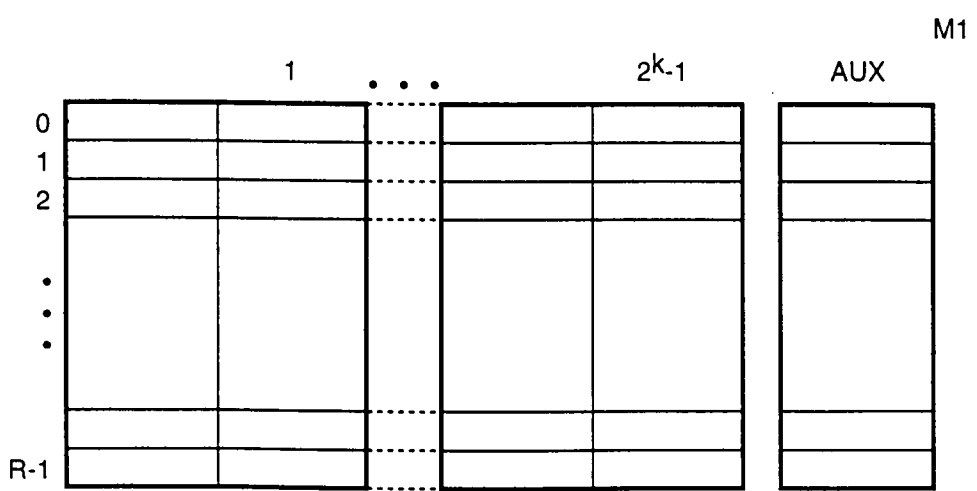


Fig. 2



GO	TARGET	DATO
----	--------	------

Fig. 3A

1	TARGET	NEXTROW
---	--------	---------

Fig. 3B

10	1	MASK	VALUE	TBAK
----	---	------	-------	------

Fig. 3C

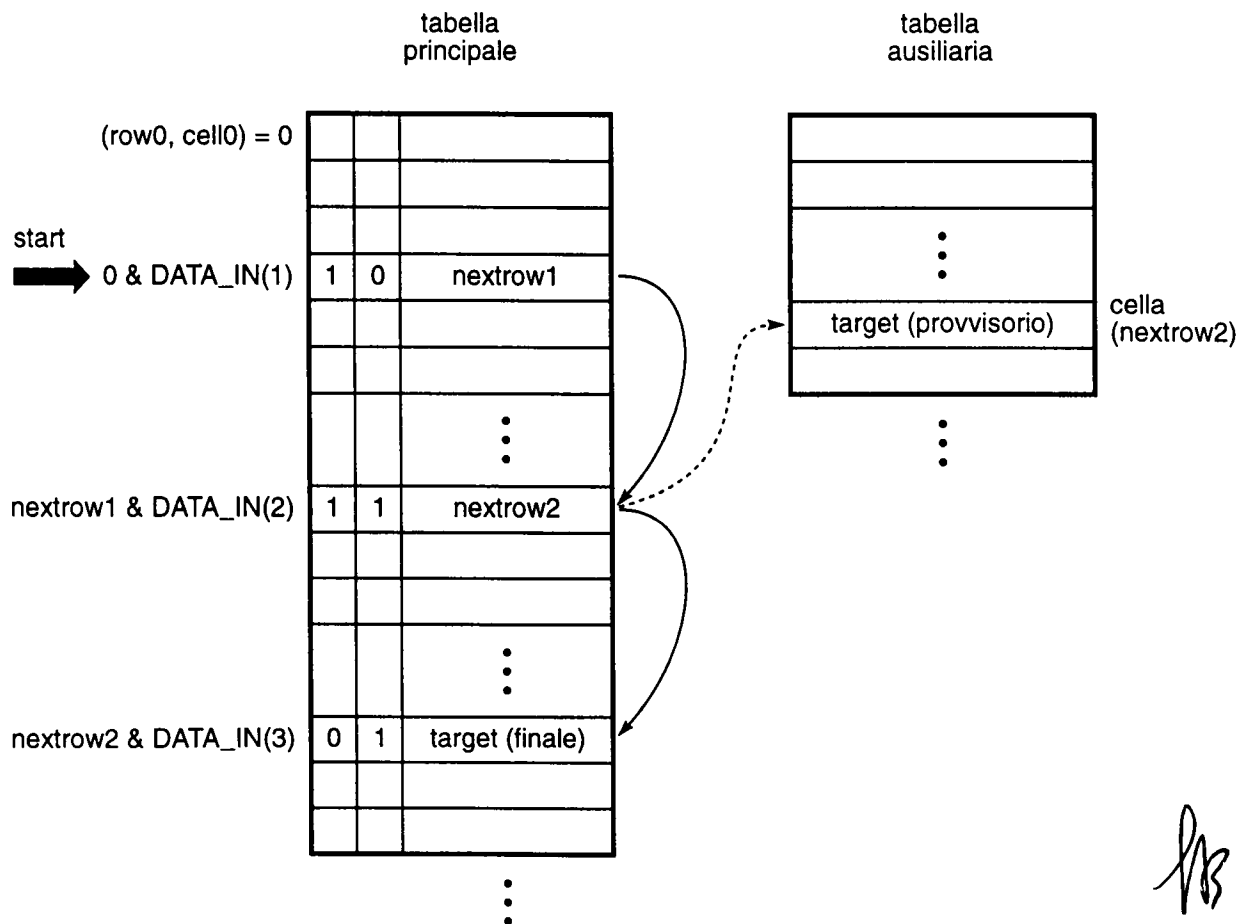


Fig. 5

P.P. CSELT - CENTRO STUDI E LABORATORI
TELECOMUNICAZIONI S.p.A.

Franco Cabella
(dr. Franco CABELLA)

	cella 00		01		10		11		AUX		
riga 0	1	0	1	0	4	0	1	(5,D,D)	0	1	(5,D,-)
1						1	1	2			
2			1	0	3				0	1	(4,A,-)
3									0	1	(8,B,-)
4						1	0	5			
5						0	1	(5,C,C)	0	1	(5,D,D)
6											
7											

G T DATI

Fig. 6

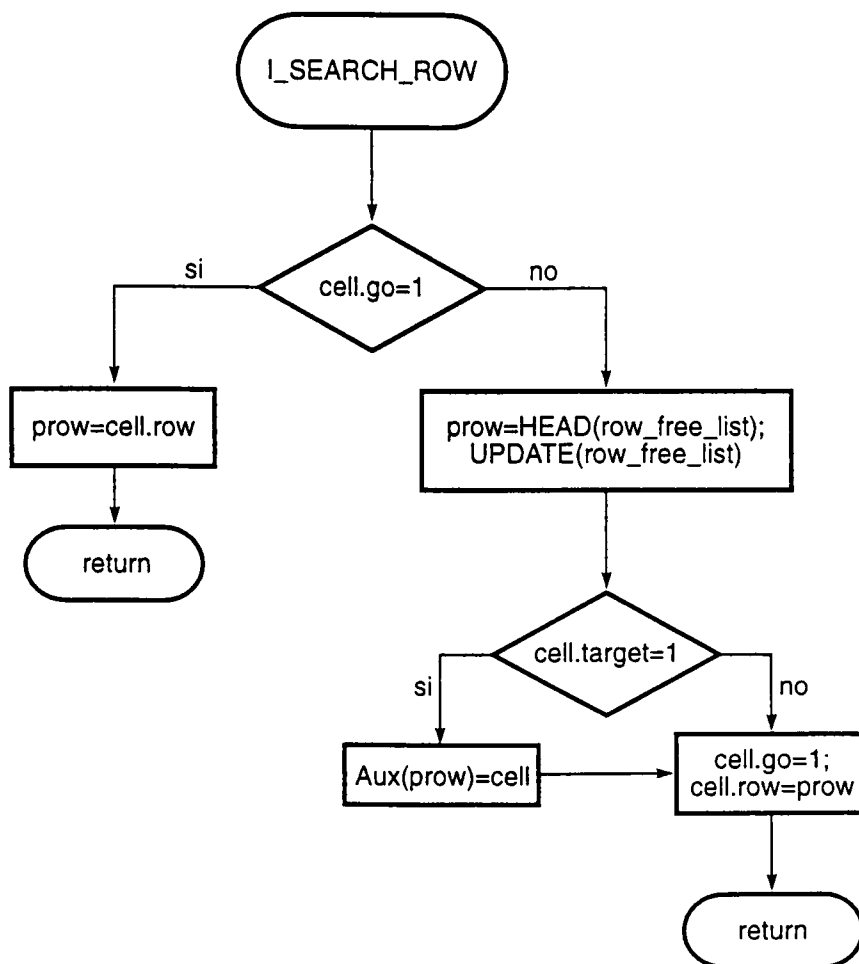


Fig. 8

P.P. CSELT - CENTRO STUDI E LABORATORI
TELECOMUNICAZIONI S.p.A.

Franco Cabella
(dr. Franco CABELLA)

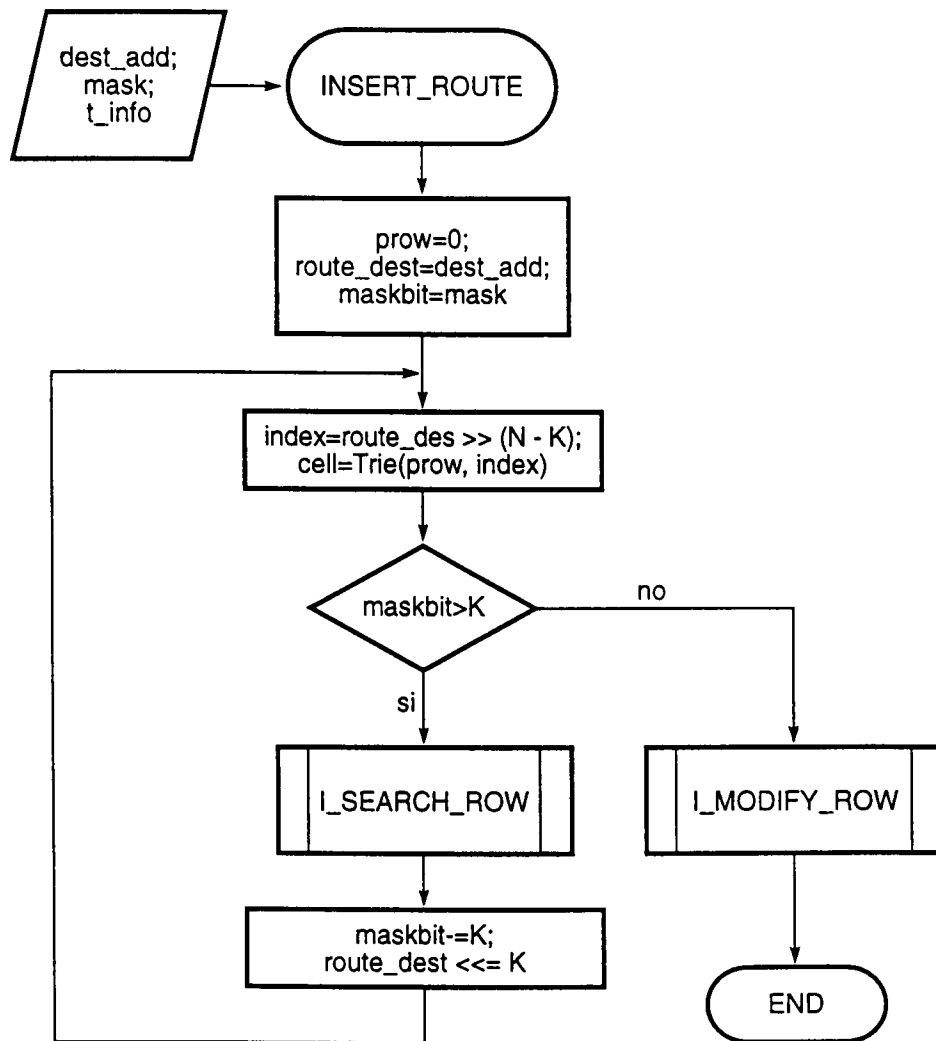


Fig. 7

P.P. CSELT - CENTRO STUDI E LABORATORI
TELECOMUNICAZIONI S.p.A.

Franco Cabella
(dr. Franco CABELLA)

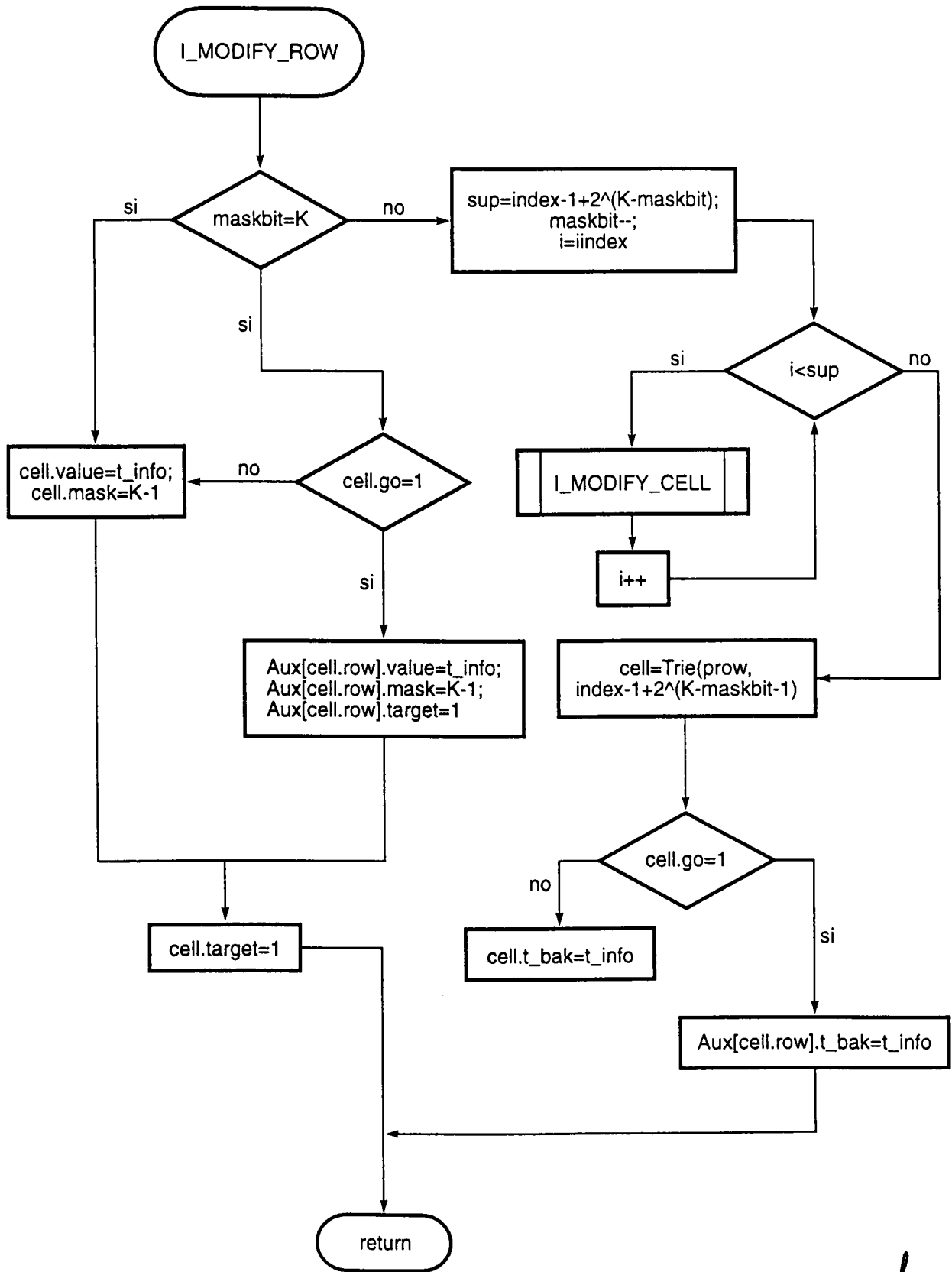


Fig. 9

P.P. CSELT - CENTRO STUDI E LABORATORI
TELECOMUNICAZIONI S.p.A.

Franco Cabella
(dr. Franco CABELLA)

hs

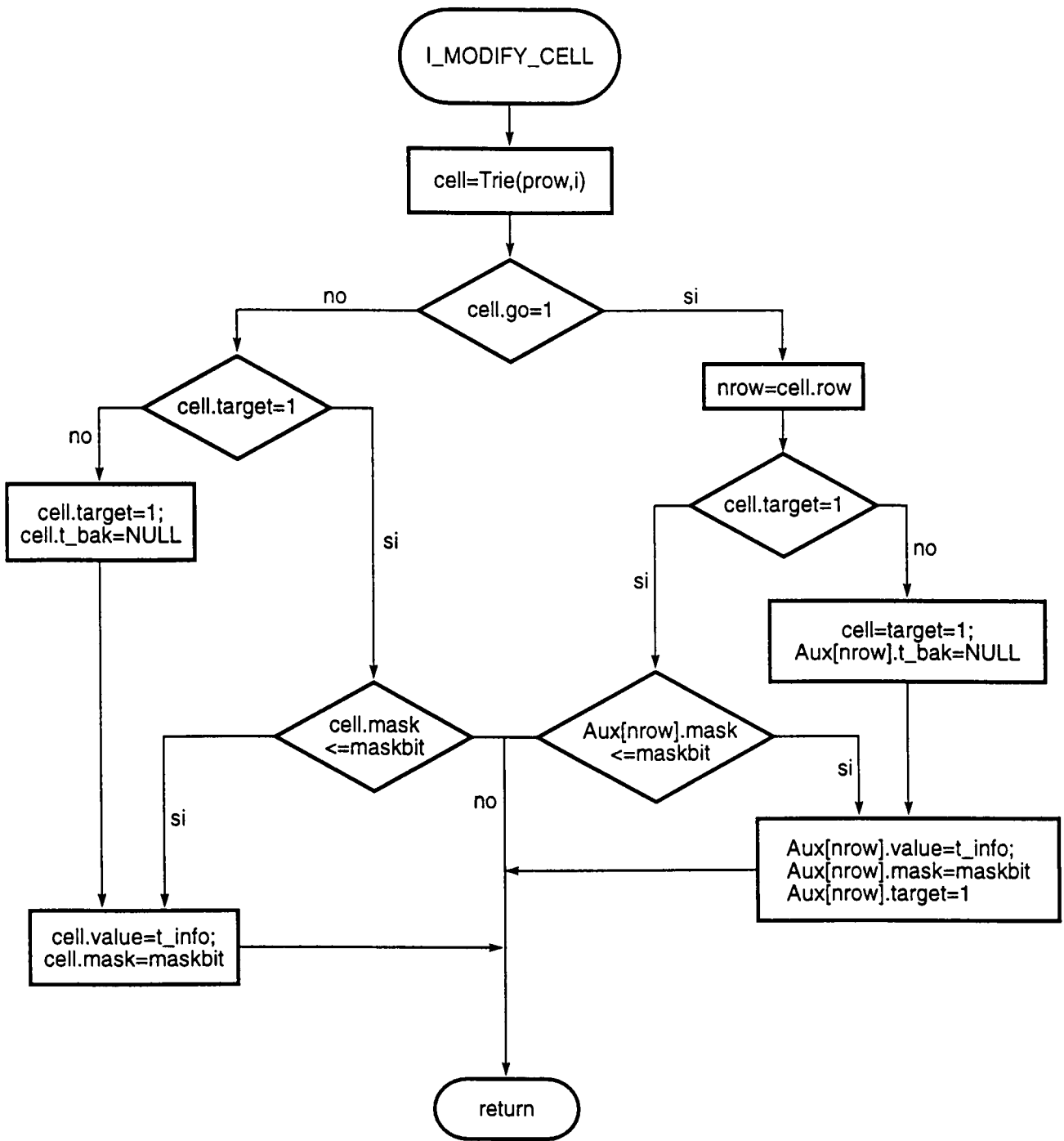
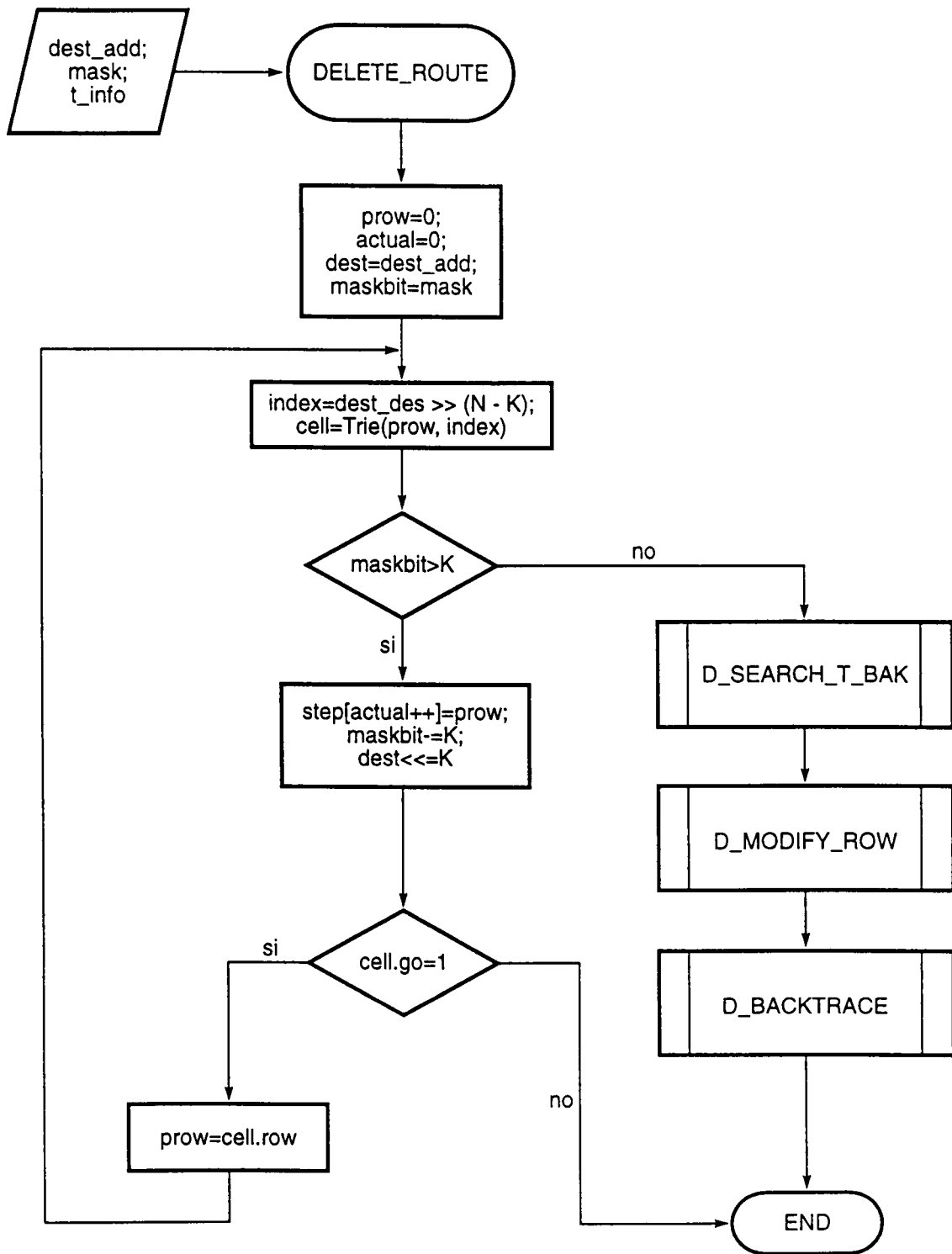


Fig. 10



lps

Fig. 11

P.P. CSELT - CENTRO STUDI E LABORATORI
TELECOMUNICAZIONI S.p.A.

Franco Cabella
(dr. Franco CABELLA)

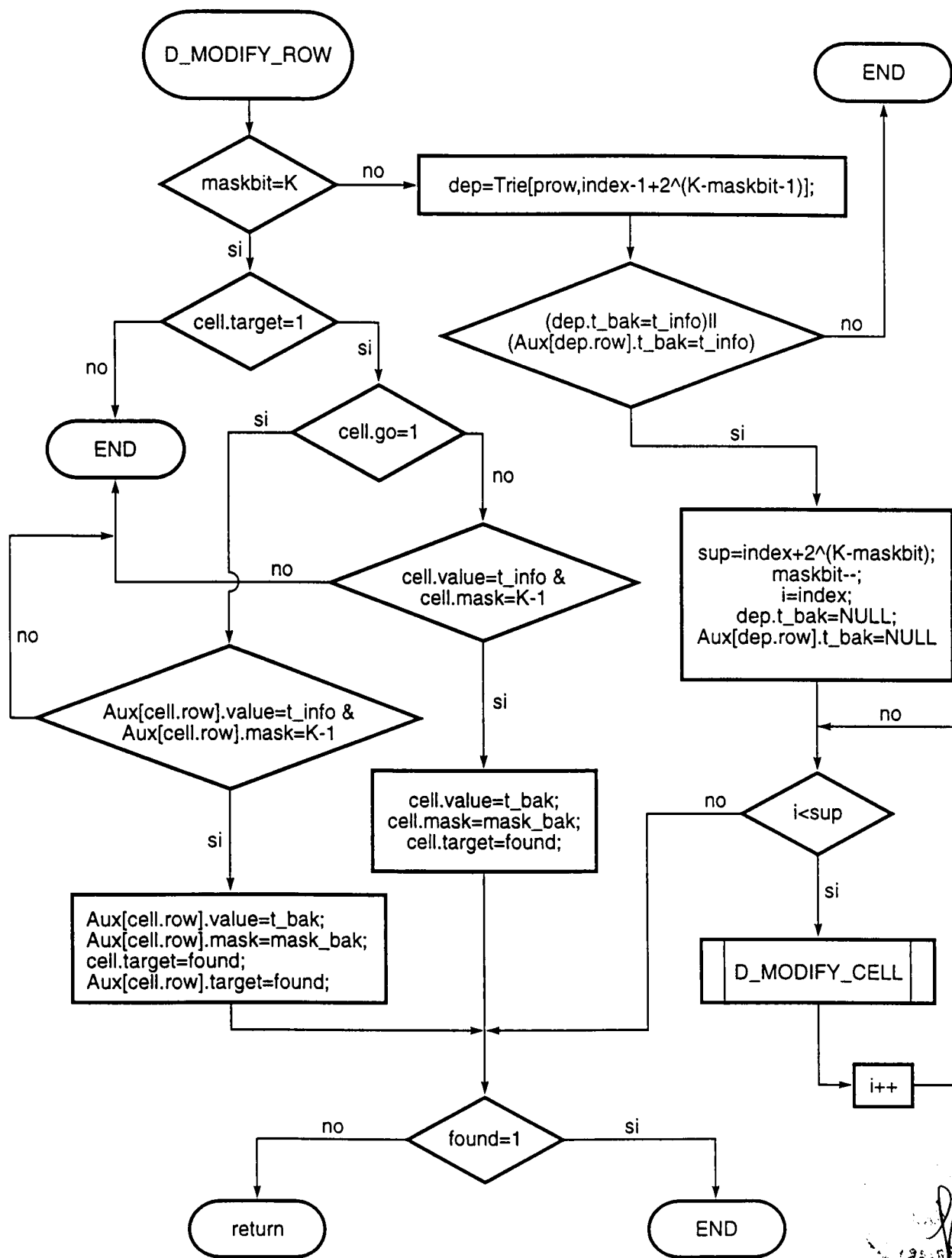


Fig. 12

P.P. CSELT - CENTRO STUDI E LABORATORI
TELECOMUNICAZIONI S.p.A.

Franco Cabella
(dr. Franco CABELLA)

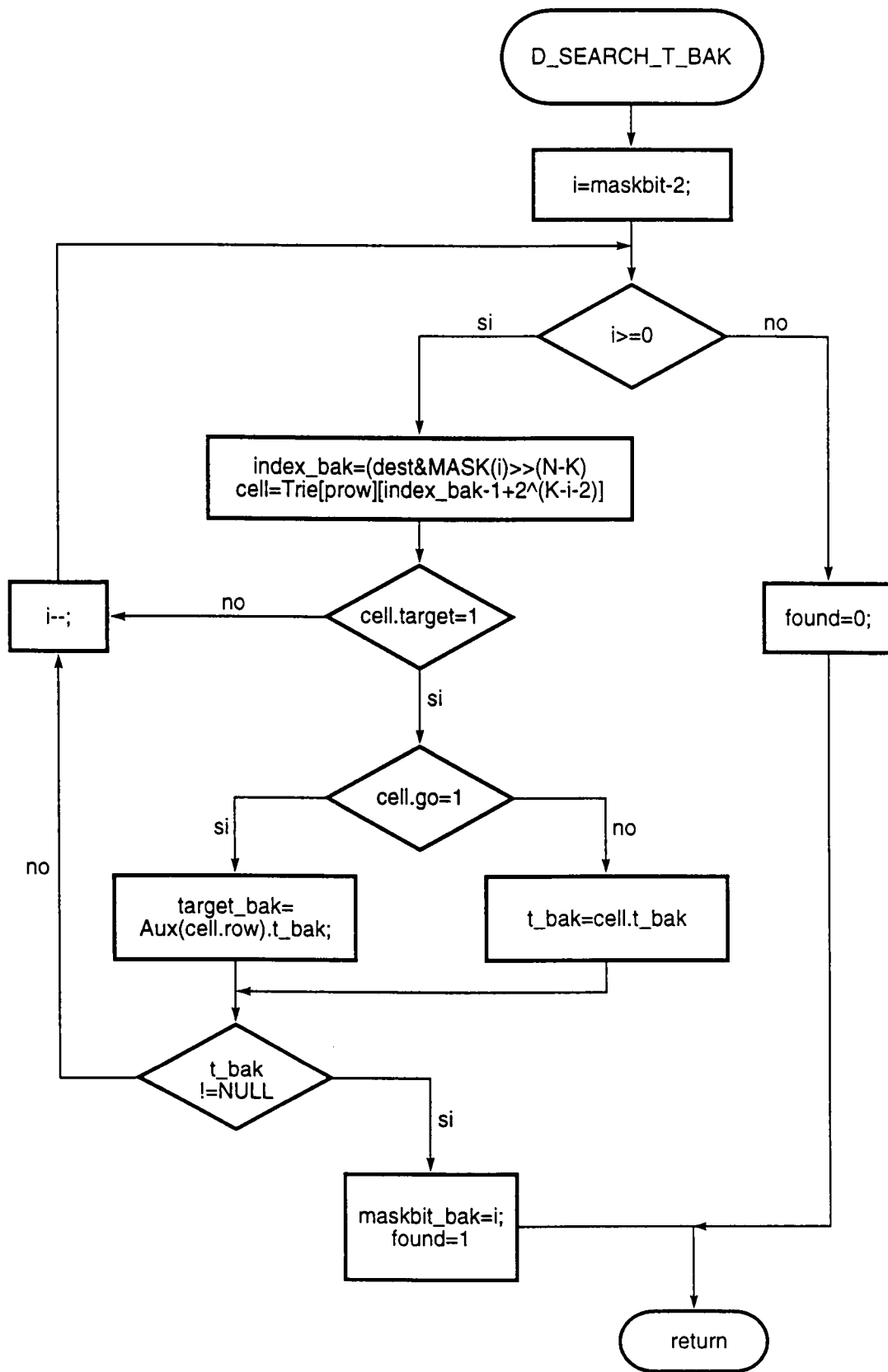


Fig. 13

P.P. CSELT - CENTRO STUDI E LABORATORI
TELECOMUNICAZIONI S.p.A.

Francisco Cabella
(dr. Franco CABELLA)

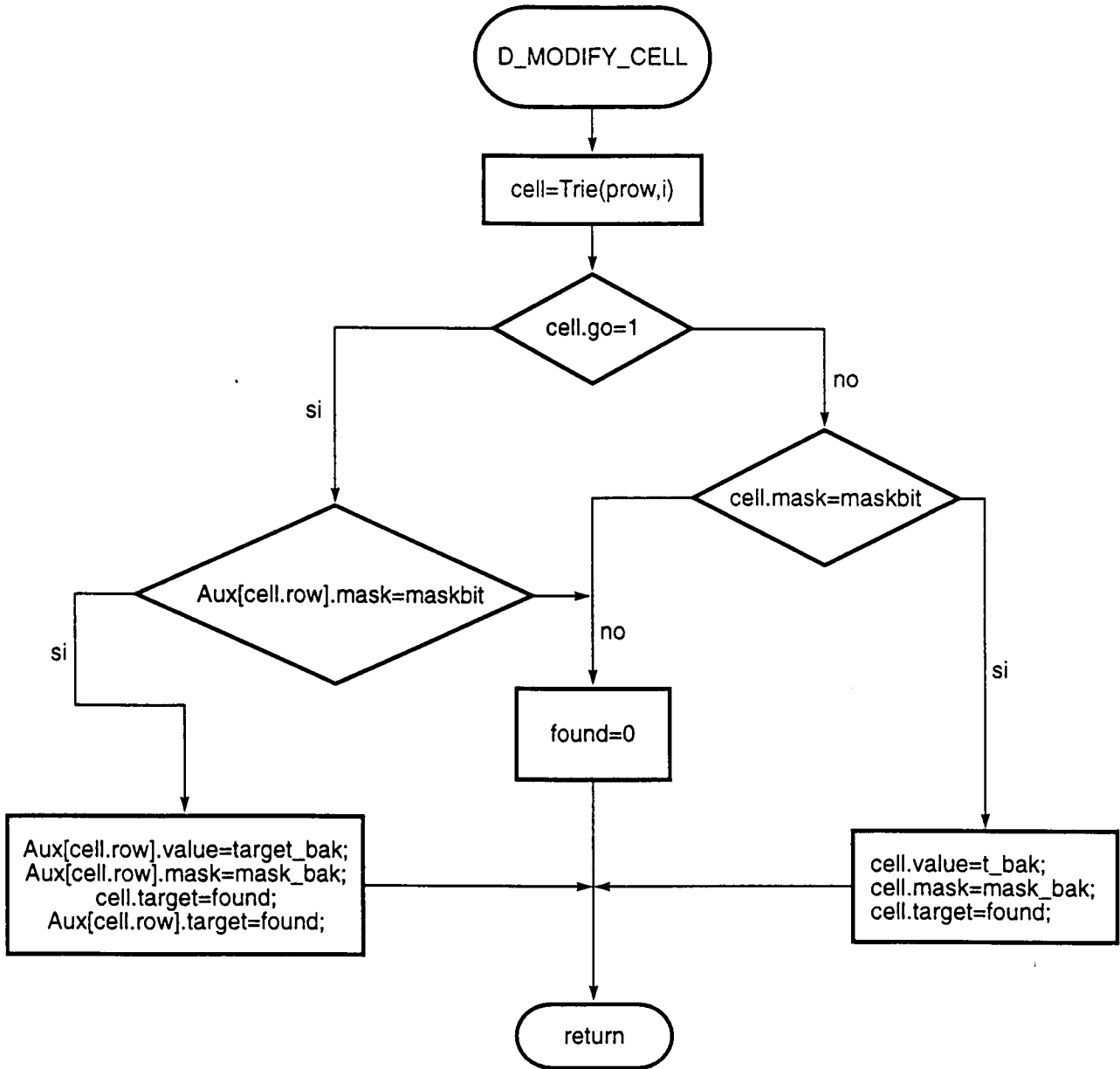


Fig. 14

P.P. CSELT - CENTRO STUDI E LABORATORI
TELECOMUNICAZIONI S.p.A.

Franco Cabella
(dr. Franco CABELLA)

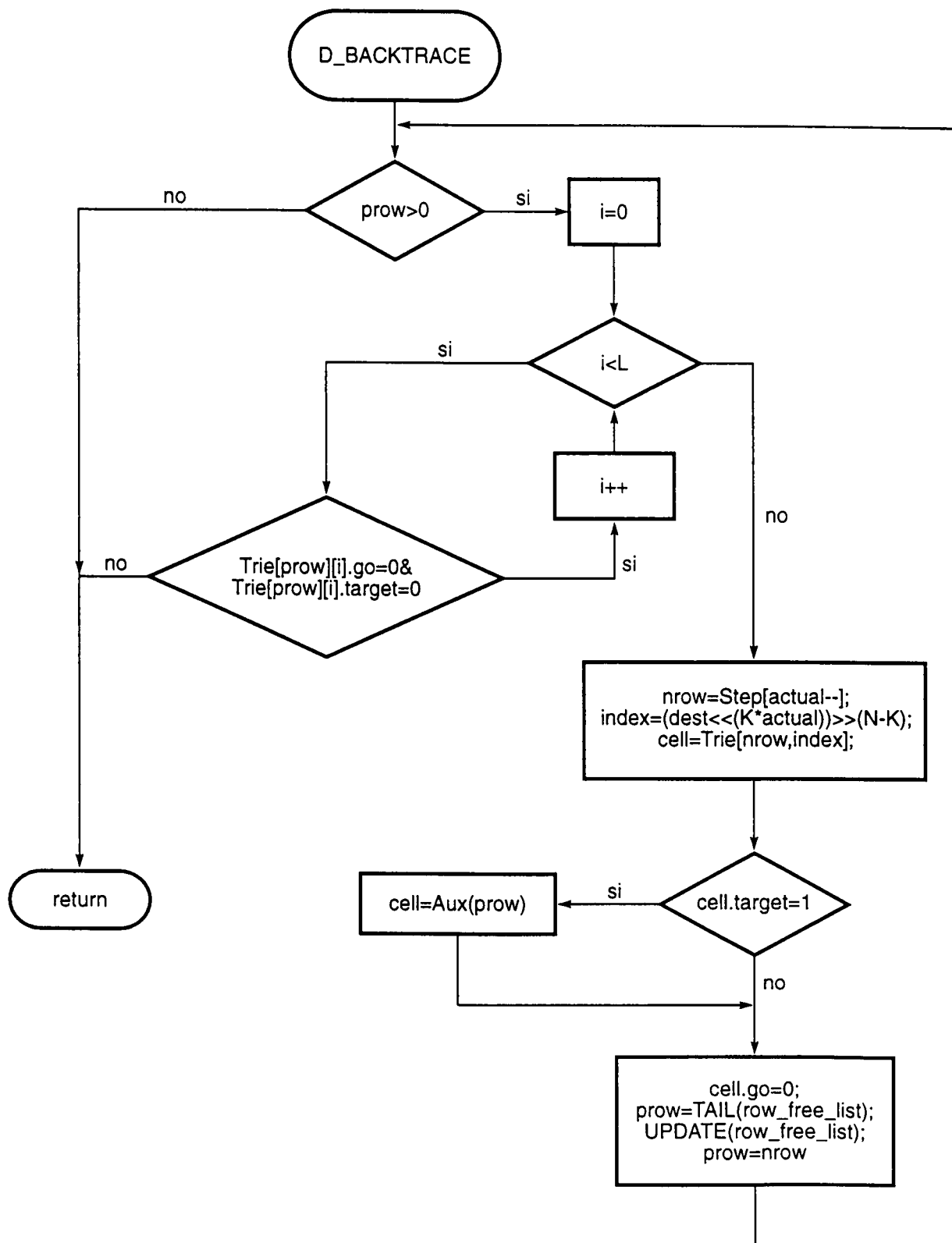


Fig. 15