METODOLOGIE DI RAPPRESENTAZIONE E DI CONTROLLO PER UN SISTEMA DI COMPRENSIONE DEI TESTI

Francesco Antonacci, Marina Russo, (#)
Paola Velardi (*), Maria Teresa Pazienza (+),

(#) *Centro di Ricerca IBM di Roma,*
(*) *Istituto di Informatica Universita' di Ancona,*
(+) *Dip. Informatica e Sistemistica Universita' di Roma*

Sommario

Un prototipo per la comprensione di testi scritti in linguaggio naturale e' stato sviluppato presso il Centro di Ricerca IBM di Roma. La complessita' e le dimensioni del dominio di analisi trattato (notizie giornalistiche di Economia e Finanza) hanno richiesto un considerevole sforzo iniziale nell'individuazione delle metodologie piu' adatte per la rappresentazione delle strutture dati e la formalizzazione degli algoritmi di controllo. Nel sistema, interamente realizzato in Prolog, sono stati pertanto adottati diversi formalismi di rappresentazione al fine di *descrivere* nella maniera piu' adeguata ciascuna delle fasi in cui si articola l'analisi del testo (morfologia, sintassi e semantica). Questi formalismi (grammatiche formali e grafi concettuali) sono stati integrati da opportune procedure di controllo dei meccanismi di deduzione, al fine di incrementare l'efficienza dell'intero sistema.

Introduzione

Un prototipo per la comprensione di testi scritti in linguaggio naturale e' stato sviluppato presso il Centro di Ricerca IBM di Roma.
Il suo obiettivo e' quello di analizzare brevi testi narrativi (notizie di agenzia su argomenti economico-finanziari), produrre una rappresentazione formale del significato dei testi analizzati, e rispondere a domande in linguaggio naturale sul contenuto di detti testi.
L'intero sistema (programmi e strutture dati) e' stato realizzato in Prolog: la logica del primo ordine ci e' sembrato infatti lo strumento piu' adeguato per lavorare in un ambito, come quello del linguaggio naturale, in cui e' assai piu' semplice *descrivere* i problemi che *prescrivere* una loro possibile soluzione.

Tuttavia, sistemi puramente descrittivi presentano spesso problemi di efficienza, a causa del gran numero di variabili da tenere in considerazione prima di poter prendere una qualsiasi decisione. Parte essenziale del processo di definizione dell'architettura di questo prototipo e' stato pertanto lo studio di opportune strategie per la *rappresentazione* dei dati e il *controllo* del processo di deduzione, al fine di incrementare la efficienza dell'intero sistema.
Nei paragrafi che seguono l'attenzione sara' concentrata su questi due aspetti: il lettore interessato ad approfondire gli argomenti trattati potra' riferirsi ai lavori citati nella bibliografia.

Descrizione generale del sistema

Per determinare il significato di un testo sono necessari diversi tipi di conoscenza: morfologica, sintattica e semantica.
Molti tra gli ultimi tentativi nel campo della comprensione dei testi si sono posti come obiettivo quello di determinare il significato del testo costruendo la struttura semantica direttamente a partire dalla frase in input [14] [27] [48].
Questo approccio viene criticato in maniera assai convincente da Ritchie in [44]. Rimane tuttavia da affrontare il problema se sia meglio effettuare l'analisi semantica durante o dopo la fase di analisi sintattica.
Esempi di integrazione tra sintassi e semantica sono abbastanza frequenti nella letteratura (cfr. [19] [21] [29] [37] [53]). L'uso di vincoli semantici durante la fase di analisi sintattica puo' rivelarsi infatti assai utile per ridurre il numero di interpretazioni ammissibili, anche se in molti casi le ambiguita' presenti in una frase non sono risolvibili prima che questa sia completamente analizzata.

Il prototipo realizzato effettua l'analisi di un testo attraverso tre fasi distinte di analisi: morfologia, sintassi e semantica, con un primo intervento della semantica gia' durante la fase di analisi sintattica, al fine di *verificare* la correttezza delle interpretazioni; la rappresentazione semantica, tuttavia, viene generata solo al termine dell'analisi sintattica.
Ognuna delle tre fasi utilizza differenti modelli di rappresentazione e diverse strutture dati, per meglio tenere conto delle caratteristiche intrinseche di ciascuna:

- la morfologia si serve di una grammatica context-free [25] [26] per il riconoscimento di ogni parola, e di altre grammatiche, sempre di tipo context-free per il riconoscimento di strutture regolari, come i tempi composti dei verbi. Come struttura dati utilizza un lessico morfologico di circa 7000 lemmi "elementari" strutturato in forma relazionale.

- la sintassi utilizza una grammatica ad attributi, che descrive in circa 150 regole un ampio frammento della grammatica dell'Italiano.
- la semantica e' basata sul formalismo dei Grafi Concettuali [49]. Utilizza come struttura dati un dizionario concettuale di circa 1000 definizioni di concetti.

A conoscenza degli autori, non ci risulta esistere un sistema di comprensione della lingua Italiana che, per dimensioni e completezza, sia paragonabile a quello presentato in questo lavoro. Tra i principali sistemi realizzati per la lingua Italiana vanno citati due lavori sviluppati rispettivamente presso l'Istituto di Psicologia del CNR di Roma e i laboratori CSELT di Torino.
Il primo dei due e' un sistema di parsing [50] [51] basato sul concetto di grafo orientato e in cui le informazioni sintattiche sono fornite dalla struttura lessicale. La frase analizzata viene rappresentata semanticamente sotto forma di rete.
Il secondo lavoro e' basato sul formalismo dei grafi concettuali [11]. La conoscenza semantica e' rappresentata mediante *grafi canonici* che descrivono le aspettative semantiche associate ai significati delle parole, nel contesto considerato. I grafi vengono utilizzati per stabilire relazioni semanticamente plausibili fra costituenti sintattici della frase (gruppi nominali, gruppi preposizionali, ecc.).

Analisi morfologica

Scopo di un analizzatore morfologico e' quello di produrre per ogni parola della frase da analizzare la lista delle sue caratteristiche morfologiche (genere, numero, tempo, modo), sintattiche (nome, verbo, avverbio, ...) e semantiche (il lemma di derivazione).
Un analizzatore morfologico puo' fornire inoltre ulteriori informazioni, ad esempio sul *tipo sintattico* dei verbi (servile, fraseologico, ecc.) o sul *tipo semantico* dei nomi (alterato diminutivo, accrescitivo, ecc). Tali informazioni possono rivelarsi particolarmente utili nelle successive fasi dell'analisi.

La formazione delle parole dell'Italiano e' soggetta ad alcuni fenomeni (composizione, affissazione, alterazione, ecc) che rendono potenzialmente infinito l'insieme delle parole della lingua e dimostrano che l'uso di un normale dizionario puo' talvolta rivelarsi insufficiente.

Es:

```
                    -> idealista -> idealistico
          -> ideale -> idealizzare -> idealizzabile
idea                                -> idealizzazione
          -> ideare -> ideatore
```

Tali fenomeni rendono pertanto poco conveniente la diretta elencazione di tutte le parole della lingua (*metodo estensivo*) e dall'altra parte seguono regole ben determinate che rendono possibile la descrizione della struttura di una parola in termini delle sue componenti elementari (*metodo intensivo*).

Il sistema per l'analisi morfologica si fonda pertanto su:

- alcune grammatiche *context-free*, di cui una descrive la struttura della PAROLA in termini delle sue componenti elementari, una serve per il riconoscimento dei numeri, e le altre vengono utilizzate per il riconoscimento di sequenze "fisse" di parole (tempi composti dei verbi, date, ecc.).
- il lessico, che contiene solo i lemmi *elementari* (cioe' privi di affissi), strutturato in forma di tabella di una base di dati relazionale, e l'insieme delle componenti elementari della parola, strutturate come fatti Prolog.

Questa fase di analisi si propone di "riconoscere" ogni parola dell'Italiano "verificando" che essa sia "legalmente costituita" in termini delle sue componenti elementari (*affissi, desinenze, radici, alterazioni ed enclitiche*) e delle regole di composizione che sono state formalizzate mediante l'uso di una grammatica context-free, il cui assioma e' la PAROLA (cfr. Appendice A).
L'analizzatore scandisce la parola da analizzare da sinistra verso destra scomponendola nelle sue parti elementari. Una volta "ricostruita" correttamente la parola (l'analizzatore ha trovato una interpretazione), viene attivato il meccanismo di backtracking del Prolog allo scopo di ottenere tutte le possibili altre analisi della parola.
Per esempio, la parola "comportamentali" viene scomposta in:

prefisso	radice	suffisso	suffisso	desinenza
CON	PORT	AMENT	AL	I

in cui CON e' un prefisso che muta la N finale in M davanti a parole che iniziano per P, M e B; PORT e' la radice del verbo PORTARE; AMENT e' la radice del suffisso deverbale AMENTO; AL e' la radice del suffisso denominale ALE; I e' la desinenza del maschile (o femminile) plurale.

Vi sono tuttavia alcune categorie di parole per le quali un'applicazione diretta e incontrollata della grammatica porterebbe a:

1. effettuare calcoli inutili,
2. produrre risultati erronei,
3. fornire analisi parziali.

1. Alla prima categoria appartengono i lemmi le cui radici non ammettono suffissazioni. E' questo il caso dei pronomi e di tutti quei lemmi che hanno la stessa radice: per il riconoscimento di *port-ament-o*, ad esempio, e' sufficiente farlo derivare dal lemma *port-are*, e non serve ripetere il procedimento con *port-a*, *port-o*, ecc..
2. Alla seconda categoria appartengono i nomi "falsi alterati" . Un *canone*, per esempio, non e' un *grosso cane*. D'altra parte un'applicazione incontrollata della grammatica produrrebbe in questo caso sia CANON-E (analisi corretta) che CAN-ON-E (analisi erronea).
3. Alla terza categoria appartengono tutti i lemmi che possono avere piu' di una interpretazione, di cui almeno una con prefisso e una senza. Ad esempio, *conformi* deve essere analizzato sia come plurale dell'aggettivo *conforme*, sia come voce del verbo *conformare*. Questo verbo, tuttavia, non e' presente nel lessico, perche' viene a sua volta derivato dal prefisso CON e dal verbo FORMARE. Un'applicazione incontrollata della grammatica fornirebbe tuttavia soltanto la prima delle due analisi, in quanto in generale se l'analisi senza prefissi ha avuto successo, l'analizzatore non effettua un successivo controllo con prefissi.

Una adeguata ciassificazione dei dati permette di gestire questi problemi a livello di rappresentazione della struttura dati.

Il lessico morfologico

Considerazioni di efficienza ci hanno indotto ad operare una netta distinzione tra l'insieme dei lemmi e l'insieme delle parti componenti della parola (affissi, alterazioni, desinenze e enclitiche). Il primo e' stato strutturato in forma di tabella di una base dati relazionale, sotto il controllo dell'SQL/DS [1]. Le dimensioni del lessico e la necessita' di tenerlo costantemente aggiornato rendevano infatti opportuno l'utilizzo di un meccanismo efficiente di gestione e di ricerca dei dati, in grado di garantirne l'integrita', quale solo una base dati relazionale poteva offrire.
Il lessico contiene solo lemmi "elementari" (privi di affissi), allo scopo di ridurne le dimensioni. Per ogni lemma vengono riportate le informazioni relative alla sua *radice*, alla sua *classe desinenziale* e alla sua *categoria sintattica* ; un indicatore numerico caratterizza inoltre la radice e serve a guidare l'analizzatore:

IND	SIGNIFICATO	AZIONE
1	caso generale	l'analisi viene effettuata regolarmente
2	il lemma non ammette suffissazioni	l'analisi passa direttamente dalla ricerca della radice a quella della desinenza
3	l'analisi corretta e' solo la prima (quella con la radice piu' lunga)	viene aggiunto un cut per inibire le analisi successive
4	il lemma deve essere analizzato anche come prefissato	il procedimento di analisi viene ripetuto due volte: una volta senza prefissi e una con i prefissi.

Gli indicatori 2, 3, e 4 permettono il controllo rispettivamente delle categorie di parole 1, 2 e 3 viste nel precedente paragrafo.

LEMMA	RADICE	CL_DESIN	CAT_SINT	IND
cane	can	dn_amore	nome.comune	1
cane	cagn	dn_cosa	nome.comune	2
canone	canon	dn_amore	nome.comune	3
conforme	conform	da_gentile	agg.qualif.	4
porta	port	dn_cosa	nome.comune	2
portare	port	dv_1coniug	v.tran.sempl.	1
porto	port	dn_oggetto	nome.comune	2

I dati rimanenti sono insiemi di dimensioni finite e limitate, e sono pertanto stati strutturati direttamente come predicati Prolog.
L'elemento portatore della caratteristica di una parola e' la **desinenza** [17]. Le desinenze sono state strutturate in "classi": una **classe desinenziale** e' l'insieme di tutte e sole le desinenze che sono in comune ad una certa classe di lemmi. Ad esempio, la classe da noi denominata *dv_1coniug* e' l'insieme di tutte le desinenze che possono seguire la radice di un qualunque verbo della prima coniugazione. Per la classificazione dei verbi irregolari, cfr [47]. La struttura delle classi desinenziali e' stata definita per mezzo del predicato DESINENZE, in cui sono elencate le diverse modalita' di flessione relative a ciascuna classe desinenziale e sono specificate le rispettive caratteristiche morfologiche.

DESINENZE(cl_desin,desinenza,car_morf).
DESINENZE(da_gentile,e,fem.sing.).
DESINENZE(da_gentile,e,mas.sing.).
DESINENZE(da_gentile,i,fem.plur.).
DESINENZE(da_gentile,i,mas.plur.).

Gli affissi si dividono in **prefissi** e **suffissi**. I suffissi alterano la categoria sintattica e la caratteristica morfologica di nomi, aggettivi e verbi. Ad esempio dal verbo *portare* col suffisso "amento" si ottiene il nome *port-amento*, e dal nome femminile *porta* col suffisso "ale" si ottiene il nome maschile *port-ale*. Il predicato SUFFISSI reca pertanto l'informazione della classe desinenziale del suffisso e della categoria sintattica di cui esso e' portatore [10].

SUFFISSI(suffisso,radice,cl_desin,cat_sint).
SUFFISSI(ale,al,da_gentile,agg.qualif.).
SUFFISSI(amento,ament,dn_oggetto,nome_comune).

In appendice A sono riportate alcune delle grammatiche utilizzate e le principali classi di parole analizzate dalla morfologia. Per una trattazione piu' completa dell'argomento cfr. [28][45] [46] [47].

Questo approccio presenta il vantaggio di un'analisi estremamente flessibile e, pur avendo richiesto all'inizio un rilevante sforzo nella formalizzazione delle regole, ha notevolmente semplificato la classificazione delle parole italiane, ivi comprese le eccezioni e le irregolarita', rendendo possibile una loro trattazione organica.
L'analizzatore morfologico e' attualmente completo, cioe' consente di analizzare senza errore tutte le parole incontrate nel dominio di applicazione (oltre 100.000 parole differenti).

Analisi sintattica

I risultati dell'analisi morfologica vengono passati all'analizzatore sintattico, il cui compito e' quello di:

- **verificare** che le parole della frase si presentino in sequenze rispondenti alle regole dell'analisi grammaticale (ad esempio la sequenza *articolo nome* e' permessa, mentre *articolo articolo* non lo e');
- **individuare** la struttura superficiale di una frase, cioe il legame esistente fra i vari elementi della frase (ad esempio tra il *verbo* e il *complemento oggetto*).

L'analizzatore si basa su di una *grammatica ad attributi* che descrive in circa 150 regole di produzione un'ampia parte della sintassi dell'Italiano. Le grammatiche ad attributi [25] [26] vengono generalmente usate per specificare la semantica dei linguaggi di programmazione. Una grammatica ad attributi e' una grammatica di tipo context-free, in cui e' possibile associare condizioni semantiche alle regole e attributi ai simboli terminali di queste. Formalmente si puo' definire come una quintupla: **<G,A,V,R,C>** in cui **G** e' una grammatica context-free, **A** e' l'insieme degli attributi associati ad ogni simbolo di G, **V** e' l'insieme dei valori degli attributi, **R** e' l'insieme delle regole di derivazione dei valori degli attributi, **C** e' l'insieme delle condizioni semantiche associate alle regole di produzione di G. Alcune regole della grammatica ad attributi per la sintassi dell'Italiano sono riportate in Appendice B.

Per definire le regole di produzione della grammatica e' stato sviluppato **MEGA** (MEta Analizzatore per Grammatiche ad Attributi) [5]. Questo sistema, interamente sviluppato in Prolog, e' uno strumento flessibile e di facile utilizzo, che permette di scrivere regole di produzione e generare da

queste il parser corrispondente, indipendentemente da ogni linguaggio di programmazione. Tramite MEGA un utente non necessariamente esperto di tecniche di programmazione e' in grado di definire, correggere e aggiornare grammatiche ad attributi e, se queste sono consistenti, generarne i relativi parser. Dopo avere analizzato tutte le regole, MEGA le traduce in predicati Prolog, generando un analizzatore di tipo top-down non deterministico, sequenziale, a discesa ricorsiva. Il parser cosi' ottenuto presenta le stesse caratteristiche di un parser generato a partire da una DCG [41] [3].

Allo scopo di accrescere l'efficienza del sistema, durante il processo di traduzione delle regole, viene calcolato ed associato ad ogni regola il suo *insieme guida*: ossia la lista di tutti i "primi simboli terminali" raggiungibili da quella regola. Gli insiemi guida sono una metodologia comunemente usata nelle grammatiche formali per i linguaggi di programmazione [20]. Con l'introduzione degli insiemi guida, durante l'analisi di una frase, vengono provate solo le regole il cui insieme guida contiene la categoria grammaticale della parola in esame. Si considerino, come esempio, le seguenti regole con i corrispondenti insieme guida:

```
FRASE  -> GR_NOM , GR_VERBALE    { ARTICOLO , NOME }
GR_NOM -> articolo , nome        { ARTICOLO }
GR_NOM -> nome                   { NOME }
```

L'analisi sintattica puo' essere effettuata in due modi differenti:

1. costruendo dapprima tutte le strutture sintatticamente ammissibili, relativamente alla frase da analizzare, per passarle poi ad una successiva fase di analisi semantica;
2. utilizzando, in alcuni determinati punti dell'analisi, un controllo da parte della semantica, che convalidi la costruzione di alcune strutture sintattiche.

Si e' deciso di seguire il secondo approccio, introducendo dei *controlli semantici* in alcune delle regole di produzione che piu' danno luogo ad ambiguita' (principalmente quelle relative ai gruppi relativi e preposizionali), al fine di limitare quanto possibile l'esplosione combinatoria degli alberi di analisi. Questa scelta e' motivata anche da studi di psicolinguistica, i quali asseriscono che nel processo di comprensione l'uomo utilizza *parallelamente* e non soltanto *sequenzialmente* conoscenze semantiche e sintattiche.

L'utilizzo di queste diverse strategie (attributi, insiemi guida e controlli semantici) nelle regole di produzione della grammatica ha consentito di diminuire notevolmente i tempi di elaborazione da un lato (dell'ordine di 1:3) e il numero degli alberi di analisi prodotti dall'altro (dell'ordine di 1:4).

Successive estensioni della grammatica prevedono la copertura di altri costrutti sintattici di interesse, quali l'anafora e l'ellissi. La attuale copertura e' di circa l'80% sul corpus di notizie analizzate (circa 700 scelte in maniera casuale sull'intero corpus di notizie giornalistiche).

I principali costrutti sintattici riconosciuti dalla grammatica sono in Appendice B. Per una trattazione piu' approfondita dell'argomento cfr. [4] [5] [31].

Analisi semantica

Le ambiguita' che rimangono dopo l'esecuzione della fase sintattica, sono risolte per mezzo dell'analisi semantica.
L'analizzatore semantico rappresenta, in modo formale, il significato dei termini e le reciproche relazioni, all'interno di una frase.
Il formalismo utilizzato per rappresentare il linguaggio naturale e' quello dei *grafi concettuali*, introdotti da Sowa: una frase viene percio' rappresentata come una rete di concetti e di relazioni concettuali, le quali evidenziano i nessi semantici tra i concetti. I vantaggi di questo formalismo, in particolare come modellpo di rappresentazione del linguaggio naturale, sono stati ampiamente discussi in [49] ed evidenziati in [52] Fra l'altro, il modello dei grafi concettuali offre strumenti per la trattazione formale di contesti, quantificatori, determinativi.
Nella frase *Il presidente firma un accordo*, ad esempio, esiste una relazione di AGENTE tra i concetti PRESIDENTE e FIRMARE e il grafo corrispondente e':

[PRESIDENTE] <- (AGENTE) <- [FIRMARE] -> (ARGOMENTO) -> [ACCORDO]

I concetti sono rappresentati da predicati Prolog con una variabile, che ne rappresenta il *referente*. I referenti possono essere nomi propri o costanti (rappresentano la specifica istanza del concetto) o delle semplici variabili (rappresentano un'istanza generica). Il simbolo # precedente un referente indica un concetto istanziato (es: "il presidente"); il simbolo * indica un concetto generico ("un presidente"); i simboli { } attorno ad un referente indicano un set ("gli edifici").
I concetti sono stati ordinati in una gerarchia di tipi ed in una di proprieta': nella prima sono presenti entita' (UOMO, CASA) ed azioni (FIRMARE); la seconda include funzionalita' (LUOGO, MESTIERE) o caratteristiche (BELLO, EFFICIENTE).

Anche le relazioni concettuali, il cui dizionario e' stato ricavato da un'analisi dei complementi della lingua italiana [6] [12], sono ordinate in modo gerarchico.[1]
Il significato di ogni termine e' descritto in modo dettagliato, elencando tutti i "possibili" (cioe' "semanticamente permessi") nessi semantici fra il concetto in questione ed altre classi concettuali.
Ogni definizione contiene il *tipo* e la *descrizione* del concetto. Ad esempio nella definizione di ACCORDO si dira' che un ACCORDO ha come PARTECIPANTI delle *entita' umane* (cioe' persone e organizzazioni umane), come SCOPO degli *atti sociali* (decisioni, cambiamenti di situazione, transazioni), ecc. (in media ogni definizione contiene 10-20 nessi semantici). In questo esempio PARTECIPANTI e SCOPO sono relazioni concettuali (descrivono tipi di nessi semantici), *entita' umane* e *atti sociali* rappresentano classi concettuali. I nessi semantici permessi sono descrittivamente delle regole:

```
PARTECIPE(ACCORDO,x) <- COND(ENTITA_UMANA,x).
SCOPO(ACCORDO,y) <- COND(ATTI_SOCIALI,y).
```

in cui si assume vera la prima regola per tutti i concetti x tali che ENTITA_UMANA > x e la seconda per tutti i concetti y tali che ATTI_SOCIALI > y.

L'analizzatore semantico ha il compito di "verificare" la correttezza semantica dei legami sintattici prodotti nella precedente fase di analisi. Questi legami vengono trasmessi all'analizzatore sotto forma di una lista di predicati sintattici che esprimono le relazioni sintatticamente valide tra coppie di termini. Ad esempio, per la frase *Il presidente firma un accordo sul piano di ristrutturazione* l'analisi sintattica produce i seguenti predicati:

```
soggetto(firmare,presidente).
oggetto(firmare,accordo).
fpn(accordo,su,piano).
fpn(piano,di,ristrutturazione).
fpn(accordo,di,ristrutturazione).
```

Notare che in questa frase sono presenti due ambiguita' non risolvibili nelle precedenti fasi. Vi e' un'ambiguita' sintattica: fpn(accordo,di,ristrutturazione) o fpn(piano,di,ristrutturazione). Il primo predicato e' ammissibile in generale dal punto di vista semantico (per cui non viene scartato dal controllo semantico precedente), ma non lo e' nel contesto dell'intera frase in questione, per cui va scartato. Vi e' poi un'ambiguita' di significato, perche'

[1] Sono state individuate circa 50 relazioni concettuali, il cui elenco completo e' stato presentato in [39]

fpn(accordo,su,piano) puo' significare sia "accordo su un pianoforte" che "accordo riguardo un progetto."
L'algoritmo di verifica semantica procede nel seguente modo:

1. viene consultato un lessico che associa ad ogni parola contenuta in ciascun predicato sintattico tanti concetti quanti sono i possibili significati (ad esempio nel caso di PIANO avremo PIANOFORTE e PROGETTO e nel caso di ACCORDO avremo ACCORDO_MUSICALE e ACCORDO_COMMERCIALE);
2. viene consultata una tabella di conversione che associa ad ogni predicato sintattico tutte le relazioni concettuali che possono interpretare il senso del legame sintattico:

```
fpn(x,SU,y) <- PARTIZIONE(y,x). /* es: otto su dieci */
fpn(x,SU,y) <- SUPPORTO(x,y). /* es: penna su tavolo */
fpn(x,SU,y) <- ARGOMENTO(x,y). /* es: accordo su progetto */
```

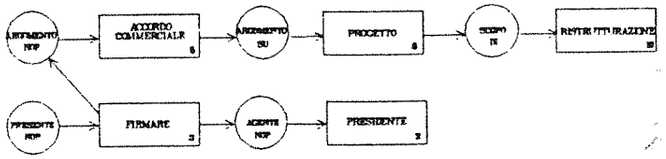
3. vengono consultate le definizioni relative a ciascuna coppia di concetti in questione e viene eseguito un doppio controllo: ossia si verifica che ogni possibile ipotesi semantica (la parte destra delle regole viste sopra) sia verificata per entrambi i concetti. Nell'esempio visto sopra, la prima ipotesi viene scartata e viene presa in considerazione la seconda nel senso di SUPPORTO(ACCORDO_MUSICALE,PIANOFORTE);
4. il passo precedente viene ripetuto per tutti i predicati sintattici, verificando contemporaneamente la consistenza delle singole interpretazioni. Ad esempio verra' scartata l'interpretazione SUPPORTO(ACCORDO_MUSICALE,PIANOFORTE) perche' incompatibile con l'azione di FIRMARE (non si firma un "accordo musicale"), e verra' presa in considerazione l'interpretazione ARGOMENTO(ACCORDO_COMMERCIALE,PROGETTO).

Il risultato sara' un grafo concettuale, che offre una rappresentazione esplicita delle relazioni di significato tra gli elementi della frase. Qualora, come nell'esempio visto in precedenza, vi sono piu' interpretazioni della frase ammissibili, viene adottato un criterio pragmatico di *vicinanza*, per cui tra le due possibili interpretazioni:

[ACCORDO_COMMERCIALE]- > (ARGOMENTO)- > [PROGETTO]- > (SCOPO)- > [RISTRUTTURAZIONE]

[RISTRUTTURAZIONE] < -(SCOPO) < -[ACCORDO_COMMERCIALE]- > (ARGOMENTO)- > [PROGETTO]

viene "preferita" la prima, e il grafo risultante dell'intera frase sara' pertanto:



Il numero di concetti presenti nel dizionario costituisce il collo di bottiglia per tutti i sistemi attualmente realizzati per l'analisi del linguaggio naturale.
Il sistema dispone attualmente di un dizionario concettuale di circa 1000 definizioni di concetti, estratti tra quelli piu' frequenti nel dominio di applicazione. E' in via di realizzazione uno strumento semi-automatico per la definizione di nuovi concetti a partire dal corpus di notizie di agenzia.

I grafi ottenuti dall'analisi delle frasi vengono memorizzati in una base di conoscenza in forma di espressioni della logica del primo ordine.
Un modulo per il trattamento delle frasi interrogative permette di porre domande in linguaggio naturale sul contenuto delle frasi memorizzate nella base di conoscenza. Il testo della domanda viene analizzato secondo la stessa procedura delle frasi affermative. Il grafo concettuale cosi' prodotto viene passato ad un modulo di reperimento delle informazioni, il cui compito e' quello di estrarre dalla base di conoscenza l'informazione o le informazioni richieste (ossia di trovare uno o piu' grafi che "includano" il grafo della domanda). Un modulo di generazione ha poi il compito di produrre una risposta in linguaggio naturale, ripercorrendo in senso inverso le diverse fasi dell'analisi (semantica, sintattica e morfologica) a partire dal grafo concettuale risposta precedentemente trovato [52].
Questo meccanismo di domanda/risposta ha permesso di valutare le reali capacita' di comprensione del sistema. I risultati ottenuti sono soddisfacenti: il sistema e' in grado di rispondere ad un largo spettro di domande del tipo "dove...?", "perche'...?", "che cosa?", "di quanto...?", "come...?", ecc. E' in corso uno studio sui meccanismi di deduzione, al fine di ampliare le capacita' di comprensione del significato delle frasi.

Conclusioni

E' stato descritto un sistema per la comprensione di testi scritti nell'ambito economico-finanziario. L'intero sistema utilizza modelli logici di rappresentazione delle strutture dati e dei programmi (grammatiche formali e grafi concettuali). Allo stato attuale dell'arte il sistema e' in grado di:

- estrarre le caratteristiche morfologiche di ciascuna parola derivabile dal lessico morfologico (copertura del 100% su piu' di 100000 parole differenti),
- ricavare la struttura sintattica di frasi (copertura dell'80% su un corpus di circa 700 notizie giornalistiche),
- ottenere una rappresentazione del significato delle frasi, esplicitando il significato dei concetti e delle relazioni che intercorrono tra essi,
- memorizzare tale rappresentazione in una base di conoscenza in forma di espressioni della logica del primo ordine,
- rispondere a domande in linguaggio naturale sul contenuto della base di conoscenza estraendo da questa le informazioni richieste.

Appendice A. Morfologia

Alcune grammatiche

- Grammatica per il riconoscimento della Parola

  1 PAROLA -> {prefisso}^{n} radice R1
  2 R1 -> {suffisso}^{n}{alterazione}^{n} R2
  3 R1 -> desinenza {suffisso}^{n} {alterazione}^{n} R2
  4 R2 -> desinenza {enclitica}^{n}

- Grammatica per il riconoscimento delle date (es. *lunedi' 20 settembre*)

  1 DATA -> nome.prop_giorno DA1
  2 DATA -> DA1
  3 DATA -> DA2
  4 DA1 -> numero(< 31) nome.prop_mese
  5 DA1 -> numero(< 31) DA2
  6 DA2 -> nome.prop_mese agg.num.

- Grammatica per il riconoscimento dei tempi composti dei verbi (es. *ero stato portato*)

  1 TEMPO_COMPOSTO -> v.tran.aus {avverbio}^{n} v.tran.(part_pass)
  2 TEMPO_COMPOSTO -> v.intr.aus {avverbio}^{n} PP
  3 PP -> v.intr.aus(part_pass) {avverbio}^{n} v.tran.(part_pass)
  4 PP -> v.tran.(part_pass)
  5 PP -> v.intr.(part_pass)

Le principali classi di parole analizzate

- **parole** "semplici" riconosciute dalla grammatica:

```
port-a            -> radice-desinenza
port-ic-in-a      -> radice-suffisso-alterazione-desinenza
com-port-ament-o  -> prefisso-radice-suffisso-desinenza
im-port-ar-lo     -> prefisso-radice-desinenza-enclitica
```

- **nomi composti** riconosciuti applicando due volte la grammatica a ciascuna delle due parole componenti e il cui plurale viene effettuato mediante specifiche regole:

```
pass-a-port-o  -> verbo+nome(masc)  pass-a-port-i
port-a-cener-e -> verbo+nome(fem)   port-a-cener-e
cass-a-fort-e  -> nome+aggettivo    cass-e-fort-i
```

- **tempi composti dei verbi** riconosciuti per mezzo di una grammatica context-free le cui regole sono applicate ogni volta che l'analizzatore trova un participio passato nella frase (se c'e' un avverbio in mezzo viene spostato: "sono spesso portato" - "sono portato spesso")

```
ho portato          -> passato prossimo - attivo
sono portato        -> presente - passivo
sono stato          -> passato prossimo - attivo
sono stato portato -> passato prossimo - passivo
```

- **grado comparativo e superlativo degli aggettivi** riconosciuti per mezzo di una grammatica context-free le cui regole sono applicate ogni volta che l'analizzatore trova gli avverbi **piu'** o **meno** seguiti da un aggettivo qualificativo o da un participio passato.

```
il piu' conforme -> superl_rel di "conforme"
piu' conforme    -> comp_magg  di "conforme"
meno amato       -> comp_min   di "amato"
il migliore      -> superl_rel di "buono"
```

- **espressioni numeriche** riconosciute da due grammatiche context-free, una per i numeri cardinali (che trasforma stringhe in numeri) e una per le espressioni miste.

```
trecentocinquanta -> 350
3 milioni 500000  -> 3500000
```

- **date**: riconosciute da una grammatica context-free (es. "venerdi' 14 aprile" o "25 maggio 1987").

Appendice B. Sintassi

Alcune regole della grammatica

- GR_PREP -> PREPOSIZIONE GR_NOM /* es: del bilancio */

```
Condizioni di Applicabilita':

Calcolo degli Attributi:
GR_PREP.TIPO=0.TIPO, GR_PREP.PREP=0.PROVENIENZA,
GR_PREP.PROVENIENZA=1.PROVENIENZA, GR_PREP.CONG=1.CONG,
GR_PREP.SECPROVEN=1.SECPROVEN, GR_PREP.SPROV=1.SPROV,
GR_PREP.TIPONOME=1.ALTERAZIONE
```

- GR_NOM_UNO -> NOME GR_PREP /* es: analisi del bilancio */

```
Condizioni di Applicabilita':
CHECK_SEMANTIC( FPN,0.PROVENIENZA,1.PREP,1.PROVENIENZA )

Calcolo degli Attributi:
GR_NOM_UNO.GENERE=0.GENERE, GR_NOM_UNO.NUMERO=0.NUMERO,
GR_NOM_UNO.PROVENIENZA=0.PROVENIENZA, GR_NOM_UNO.TIPO=1.TIPO,
GR_NOM_UNO.PREP=1.PREP, GR_NOM_UNO.ARTICOLO:=NO, GR_NOM_UNO.PREPPRES:=SI,
GR_NOM_UNO.ALTERAZIONE=0.ALTERAZIONE, GR_NOM_UNO.ATTR:=NO,
GR_NOM_UNO.CONG:=NO, GR_NOM_UNO.APPOS:=NO,GR_NOM_UNO.APP:=SI,
GR_NOM_UNO.AGGETTIVO:=NO, GR_NOM_UNO.AGG:=NO, GR_NOM_UNO.CONGRIC:=NO,
GR_NOM_UNO.APPCONG:=NO, GR_NOM_UNO.NOME=0.NOME
```


- GR_NOM -> GR_NOM_UNO GR_PREP /* es: analisi del bilancio della Fiat */

```
Condizioni di Applicabilita':
CHECK_SEMANTIC( FPN,0.PROVENIENZA,1.PREP,1.PROVENIENZA ),
0.ATTR:=NO, 0.CONGINIZ:=NO

Calcolo degli Attributi:
GR_NOM.GENERE=0.GENERE, GR_NOM.NUMERO=0.NUMERO ,
GR_NOM.PROVENIENZA=0.PROVENIENZA, GR_NOM.PREP =1.PREP,
GR_NOM.ART=0.ARTICOLO, GR_NOM.PREPPRES:=SI, GR_NOM.SPROV:=NO ,
GR_NOM.APPOS=0.APPOS, GR_NOM.TIPO=0.TIPO, GR_NOM.APP=0.APP,
GR_NOM.AGG=0.AGG, GR_NOM.AGGETTIVO=0.AGGETTIVO,GR_NOM.CONG=0.CONG,
GR_NOM.NOME=0.NOME, GR_NOM.ALTERAZIONE=0.ALTERAZIONE
```

I principali costrutti sintattici analizzati:

- Frasi:

semplici	***il presidente approva il bilancio***
passive	***il contratto e' firmato dalla societa'.***
subordinate	il presidente dice ***che il progetto procede***
relative	il progetto prevede un piano ***che coinvolga i lavoratori***
interrogative	***chi firma un contratto?***

- Gruppi:

nominali	***il presidente*** firma
nominali-incisi	la Perugina ***(gruppo IBP)*** investe
aggettivali	***il primo grande congresso nazionale*** e' aperto
avverbiali	il progetto procede ***velocemente*** e con buoni risultati
preposizionali	l'assemblea ***degli azionisti*** nomina il presidente ***per tre anni***
verbali	l'assemblea ***si riunisce***

- Coordinazione tra gruppi:

nominali	***le aziende ed il governo*** firmano
preposizionali	le borse ***di Milano e di Roma*** sono in rialzo
aggettivali	le strutture ***amministrative, produttive e commerciali*** sono valide
avverbiali	studia ***molto e bene***
verbali	l'assemblea ***si riunisce e nomina***

Appendice C. Semantica

Alcune delle relazioni concettuali utilizzate

AGENTE lega concetti di tipo ANIMATI ANIMALI a concetti di tipo TIPO AZIONE:

[ANIMATO_ANIMALE] <- (AGENTE) <- [TIPO_AZIONE]
es. *il presidente firma*

PARTECIPE lega concetti di tipo ENTITA' UMANA a concetti di tipo RIUNIONE:

[ENTITA_UMANA] <- (PARTECIPE) <- [RIUNIONE]
es. *accordo con i lavoratori*

SCOPO lega concetti di tipo TIPO AZIONE a concetti di tipo TIPO AZIONE:

[TIPO_AZIONE] <- (SCOPO) <- [TIPO_AZIONE]
es. *progetto di ristrutturazione*

ARGOMENTO lega concetti di tipo ATTI DI PENSIERO, ATTI DI COMUNICAZIONE o DOCUMENTO a qualsiasi concetto (tipo TOP):

[TOP] <- (ARGOMENTO) <- [ATTI_DI_PENSIERO]
[TOP] <- (ARGOMENTO) <- [ATTI_DI_COMUNICAZIONE]
[TOP] <- (ARGOMENTO) <- [DOCUMENTO]
es: *pensare a un accordo, elaborazione di un piano, discorso di economia*

SUPPORTO lega concetti aventi la proprieta' di LUOGO a concetti di tipo ENTITA CONCRETE o di tipo MODIFICA DI POSIZIONE:

[LUOGO] <- (SUPPORTO) <- [ENTITA_CONCRETA]
[LUOGO] <- (SUPPORTO) <- [MODIFICA_DI_POSIZIONE]
es: *penna sul tavolo, posare sulla finestra*

Bibliografia

[1] **SQL/Data System**, *Terminal User's Reference*, SH24-5017-2, IBM Corp., 1983.

[2] **VM/Programming in Logic**, *Program Description/Operation Manual*, SH20-6541-0, IBM Corp., 1985.

[3] H.Abramson, **Definite Clause Translation Grammars**, *Proc. of the Int. Symposium on Logic Programming*, IEEE 1984,

[4] F.Antonacci, **Using Semantic Hints in a Syntactic Parser**, *IBM Technical Report*, SCR-0004, maggio 1987.

[5] F.Antonacci, R.Vitale, **Un Meta-Compilatore efficiente per un analizzatore dell'Italiano**, *Atti del Congresso Annuale AICA*, Trento, 30 settembre-2 ottobre, 1987.

[6] S.Battaglia, V.Pernicone, **La Grammatica Italiana**, ed. Loescher, 1968.

[7] J.Bresnan, R.Kaplan, **Lexical-Functional Grammar: a formal system for grammatical representation**, *In J.Bresnan ed. "The Mental Representation of Grammatical Relations"*, MIT Press, Cambridge, pp. 173-281, 1982.

[8] E.Charniak, **A Parser with Something for Everyone**, *in M.King ed. "Parsing Natural Language"*, Academic Press, pp. 117-149, 1983.

[9] K.Church, R.Patil, **Coping with Syntactic Ambiguity or How to Put the Block in the Box on the Table**, *American Journal of Computational Linguistics*, vol. 8, n. 3-4, pp. 139-149, 1982.

[10] W.D'Addio, **Su alcune modalita' di suffissazione in Italiano**, *Atti del II Convegno di Studi della Societa' Linguistica* Italiana, Bulzoni, Roma, 1968.

[11] M. Danieli, F. Ferrara, R. Gemello, C. Rullent, **Integrating Semantics and Flexible Syntax by Exploiting Isomorphism between Grammatical and Semantical Relations**, *Proc. of the 3rd Conference of the European Chapter of the ACL*, Copenhagen, 1987.

[12] M.Dardano, P.Trifone, **La Lingua Italiana**, Zanichelli, 1985.

[13] R.Davis, D.Lenat, **Knowledge-based Systems in Artificial Intelligence**, Mc.Graw Hill, 1982.

[14] G.F.De Jong, **Skimming stories in real time: An experiment in integrated understanding**, *Yale Univ. Dept. of Computer Sc., Technical Report*, n. 158, 1979.

[15] R.Farrow, **LINGUIST-86 Yet another translator writing system based on attribute grammars**, *Proc. of the SIGPLAN 82 Symposium on Compiler Construction*, ACM, pp. 160-171, June, 1982.

[16] H.Ganzinger, R.Giegerich, U.Moncke, R.Wilhelm, **A Truly Generative Semantics-Directed Compiler Generator**, *Proc. of the SIGPLAN 82 Symposium on Compiler Construction*, ACM, pp. 172-184, June, 1982.

[17] A.Giurescu, **Problemi della formazione delle parole in Italiano**, *Atti del IV Convegno di Studi della Societa' Linguistica Italiana*, Bulzoni, Roma, 1970.

[18] G.Heidorn, **Natural language inputs to a simulation programming system**, *Technical Report NPS-55HD72101A, Naval Postgraduate School*, Monterey, California, 1972.

[19] G.Hirst, E.Charniak, **Word sense and case slot disambiguation**, *Proc. of AAAI-82*, pp 95-98, 1982.

[20] J.Hopcroft, J.Ullman, **Introduction to automata theory, languages and computation**, Addison Wesley, pp. 248-279, 1979.

[21] P.Jacobs, **Generation in a natural language interface**, *Proc. of IJCAI-83*, 1983.

[22] K.Jensen, G.Heidorn, L.A.Miller, Y.Ravin, **Parse Fitting and Prose Fixing: Getting a Hold on Ill-formedness**, *IBM Technical Report RC 10457 (#46757)*, 4, 4, 1984.

[23] S.C.Johnson, **YACC- Yet Another Compiler Compiler**, *Technical Report CSTR 32*, Bell Laboratories, 1975.

[24] M.Kay, **Functional Grammar**, *Proc. of the 5th Meeting of the Berkeley Linguistic Society*, pp. 142-158, 1979.

[25] D.E.Knuth, **Semantics of context-free Languages**, *Mathematical Systems Theory*, vol.2, 1968.

[26] D.E.Knuth, **Semantics of context-free Languages: Correction**, *Mathematical Systems Theory*, vol.5, 1971.

[27] M.Lebowitz, **Generalization and Memory in an Integrated Understanding System**, *Yale Univ. Dept. of Computer Sc., Research Report*, n. 186, 1980.

[28] S.Leuzzi, M.Russo, **Un Analizzatore Morfologico della Lingua Italiana**, *Atti del I Convegno Nazionale sulla Programmazione Logica*, Genova, 12-14 marzo, 1986.

[29] S.L.Lytinen, **Dinamically combining syntax and semantics in natural language processing**, *Proc. of AAAI-86*, vol.1, 1986.

[30] B.Lorho, **Semantic attribute processing in the system DELTA**, *In A.Ershov e C.H.A.Koster ed. "Methods of Algorithmic Language Implementation"*, Springer Verlag, pp. 21-40, 1977.

[31] P.Monachesi, **Definizione e realizzazione di una grammatica formale per l'analisi sintattica di testi, ottenuta mediante tecniche computazionali**, *Tesi di laurea in Lingue e Letterature Straniere*, Macerata, 1987.

[32] M.C.McCord, **Slot Grammars**, *American Journal of Computational Linguistics*, pp. 31-43, 1980.

[33] M.C.McCord, **Focalizers, the scoping problem, and semantic interpretations rules in logic grammar**, *Technical Report, University of Kentucky*, 1981.

[34] M.C.McCord, **Using slots and modifiers in logic grammar for natural language**, *Artificial Intelligence*, vol. 18, pp. 327-367, 1982.

[35] R.Moore, **The role of logic in knowledge representation and commonsense reasoning**, *in Brachman e Levesque ed. "Readings in Knowledge Representation"*, Morgan Kaufmann Publisher, 1985.

[36] A.Newell, **Heuristic programming: ill-structured problems**, *Progress in Operations Research*, vol.3, 1969.

[37] J.C.Oden, **On the use of semantic constraints in guiding syntactic analysis** *Int. Journal on Man-Machine Studies*, n.19, Academic Press, pp.335-357, 1983.

[38] M.T.Pazienza, P.Velardi, **Knowledge on Word Uses for Semantic Analysis of Texts,** *Workshop on Conceptual Graphs,* Thornwood, NY, 18-20 agosto, 1986.

[39] M.T.Pazienza, P.Velardi, **A structured representation of word-senses for semantic analysis,** *Proc. of the 3rd Conference of the European Chapter of the ACL,* Copenhagen, 1-3 aprile, 1987.

[40] M.T.Pazienza, P.Velardi, **Integrating Conceptual Graphs and Logic in a Natural Language Understanding System,** *in "Natural Language Understanding and Logic Programming",* North-Holland, 1987.

[41] F.Pereira, D.Warren, **Definite clause grammars for language analysis - a survey of the formalism and a comparison with Augmented Transition Networks,** *Artificial Intelligence,* vol. 13, pp. 231-278, 1980.

[42] S.D.Richardson, **Enhanced Text Critiquing using a Natural Language Parser,** *Proc. of the 7th Int. Conference on Computers and Humanities,* Provo, Utah, June 26-28, 1985.

[43] C.Riesbeck, **Conceptual Analysis,** *in R.Schank ed. "Conceptual Information Processing",* North-Holland, 1975.

[44] G.Ritchie, **Semantics in Parsing,** *in M.King ed. "Parsing natural Language",* Academic-Press, 1983.

[45] M.Russo, **A Generative Grammar Approach for the Morphologic and Morphosyntactic Analysis of Italian,** *Proc. of the 3rd Conference of the European Chapter of the ACL,* Copenhagen, 1-3 aprile, 1987.

[46] M.Russo, **A Rule Based System for the Morphologic and Morphosyntactic Analysis of the Italian Language,** *in "Natural Language Understanding and Logic Programming",* North-Holland, 1987.

[47] M.Russo, **Integrazione di Lessico e Morfologia per l'analisi dell'Italiano,** *Congresso Annuale AICA,* Trento, 30 settembre-2 ottobre, 1987.

[48] R.C.Schank, M.Lebowitz, L.Birnbaum, **An Integrated Understander,** *American Journal of Computational Linguistics,* vol. 6, n. 1, 1980.

[49] J.F.Sowa, **Conceptual Structures: Information Processing in Mind and Machine,** Addison-Wesley, Reading, 1984.

[50] O.Stock, **Dynamic Unification in Lexically Based Parsing,** *PCIA/27,* Roma, dicembre 1984.

[51] O.Stock, F.Cecconi, C.Castelfranchi, **Analisi morfologica integrata in un parser a conoscenze linguistiche distribuite,** *Congresso Annuale AICA,* Palermo, 24-26 settembre 1986.

[52] P.Velardi, M.T.Pazienza, M.DeGiovannetti, **Conceptual Graphs for the Analysis and generation of sentences,** *IBM Journal of Research and Development,* January, 1988.

[53] R.Wilensky, **A knowledge-based approach to language processing: a progress report,** *Proc. of IJCAI-81,* 1981.

[54] Y.Wilks, **Deep and Superficial Parsing,** *in M. King ed. "Parsing natural Language",* Academic Press, 1983.

INTERPRETAZIONE DELL'ITALIANO IN TERMINI DI LOGIC PROGRAMMING

Giovanna Dondossola[+], Giacomo Ferrari[*]

[+] Dipartimento di Scienze dell'Informazione, Universita' di Milano
[*] Dipartimento di Linguistica, Universita' di Pisa

Abstract: In this paper a system for the syntactic analysis and semantic interpretation of natural language, in particular Italian, is presented. The system consists of three components, i) a syntactic parser based on extraposition grammar, which translates Italian sentences into (a set of) 3-BQ trees, ii) a disambiguator which eliminates some of the non admissible 3-BQs, and iii) a semantic interpreter which translates the tree into a logical representation in terms of LRS, a logic language specifically designed. A description of this language is also provided.

1. Introduzione

1.1. Obiettivi generali

Questo articolo presenta un sistema di analisi sintattica e semantica del linguaggio naturale, in particolare l'italiano, fondato su tecniche sviluppate nell'ambito della programmazione logica.

La programmazione logica e, in particolare, il Prolog e' stato utilizzato per implementare vari sistemi di analisi per l'inglese,