Indice

Capitolo 1

Introduzione all'Elaborazione del Linguaggio Naturale

Per linguaggio naturale s'intende quello parlato e scritto dagli esseri umani. Elaborarlo significa progettare e realizzare un sistema informatico automatico che sia in grado di "comprenderlo". La realizzazione di un sistema per l'ELN si concentra su vari aspetti del linguaggio:

- analisi morfologica: studio della forma e struttura interna delle parole
- analisi sintattica: studio di come le parole si dispongono a formare un periodo
- analisi semantica: studio del significato dei termini, dei periodi e studio del ruolo grammaticale all'intermo della frase

La via più naturale per giungere alla comprensione automatica del testo consiste nel concatenare in cascata moduli software che producano, in modo indipendente, l'analisi morfologica, quella sintattica e quella semantica. Questo modo di procedere presenta degli inconvenienti; ad esempio, per risolvere ambiguità linguistiche a livello sintattico sono necessarie conoscenze semantiche.
Attualmente il corpo centrale di tutta questa analisi è l'etichettatura lessicale dei testi (Part-of-Speech Tagging), un processo attraverso il quale, si assegna a ciascuna parola di una frase un'etichetta (sostantivo, aggettivo, verbo, ecc) che ne individua la categoria grammaticale di appartenenza. L'etichettatura può avere più livelli di complessità a seconda del progetto che si intende realizzare. Questa tesi si occupa di *information retrieval*, ovvero di creare un corredo di strumenti che consentano l'individuazione ed il recupero

di informazioni da collezioni di dati testuali. Un esempio immediato sono i diversi motori di ricerca utilizzati in internet.
In particolare, lo scopo della tesi è individuare espressioni quantitative in testi italiani e successivamente mostrare, come partendo da questa prima parte, si possano individuare espressioni temporali anche complesse.

1.1 Text Tokenisation Tool

Il programma utilizzato per lo svolgimento della tesi è il sistema **TTT** (cfr [1]) della University of Edinburgh.
Attraverso tale sistema è possibile contrassegnare e marcare stringhe appartenenti ad un testo. Brevemente il sistema TTT è formato da un *part-of-speech-tagger* che in prima analisi assegna delle etichette a tutte le parole di un testo e da un *sentence boundary disambiguator* che determina il carattere *punto* (.) come parte di una abbreviazione o di fine paragrafo.
Il componente principale del programma è un eseguibile denominato *fsgmatch*.
Fsgmatch processa un flusso di caratteri in ingresso e lo riscrive basandosi su un insieme di regole grammaticali definite in altri file. *Fsgmatch* non è modificabile, ma questo è ininfluente in quanto ciò che vogliamo ottenere dipende unicamente dalle regole che esso deve interpretare. *Fsgmatch* è sviluppato in ambiente XML e per tale motivo le regole grammaticali sono costruite come elementi XML. Queste regole, denominate RULEs, sono contenute in file esterni, che si possono creare a seconda del tipo di esigenza. Così il sistema è modulare in quanto si possono elaborare, in successione, file diversi per ottenere un determinato tipo di etichettatura. In queste pagine viene mostrata principalmente l'etichettatura ottenuta tramite *numbers2.gr* e *timex2.gr*
Ad esempio *numbers2.gr* è un file XML che contiene le regole grammaticali per riconoscere e marcare espressioni numeriche contenute in un file di testo. Tutti i file che contengono le regole grammaticali devono avere estensione *.gr* e possono essere scritti con un qualsiasi editor di testo.

Successivamente verrà spiegato come *fsgmatch* operi a livello carattere ed a livello SGML. In entrambi i casi la distinzione è legata al formalismo con il quale si interpretano i dati in ingresso, ed ai corrispondenti file grammaticali usati.
In definitiva l'etichettatura di un testo si ottiene attraverso una pipeline formata da tutti i file a estensione .gr utilizzati. Per etichettare un testo occorre quindi processarlo attraverso diversi file di estensione .gr.

Dovendo lavorare in ambiente XML vengono di seguito fornite alcune nozioni di base per comprendere al meglio i capitoli successivi.

1.2 Meta-linguaggio XML

XML (Extensibile Markup Language) è un sottoinsieme di SGML, il quale è un meta-linguaggio (un linguaggio che crea altri linguaggi) che viene utilizzato per creare linguaggi di markup, come HTML. Per tale motivo la sintassi di base di XML è simile a quella di HTML (cfr [2]).
XML è una tecnologia che serve a creare linguaggi di markup che descrivono i dati, teoricamente di qualsiasi tipo, in modo strutturato. Diversamente da HTML, che limita l'autore di un documento a un insieme prestabilito di tag, XML può essere utilizzato per creare linguaggi di markup che descrivono i dati in quasi tutti i campi applicativi. Nel capitolo successivo viene fornita una breve descrizione della sintassi base in XML.

Capitolo 2
Fsgmatch

Il cuore del programma nel sistema TTT è chiamato *fsgmatch* (Fast SGml MATCH).
E' un applicativo che processa una stringa di ingresso e la riscrive rispettando una serie di regole grammaticali contenute in un file esterno avente estensione "*.gr*". In questo modo è possibile alterare l'ingresso a piacimento, ma l'uso principale è quello di aggiungere informazioni di markup. *Fsgmatch* ha due differenti modi di operare a seconda che la stringa in ingresso sia considerata come flusso di caratteri (*fsgmatch* a livello carattere) o come flusso di elementi SGML/XML (*fsgmatch* a livello SGML).
Il file in ingresso da analizzare deve essere di tipo XML ed è quindi necessaria una prima elaborazione per renderlo conforme tramite un semplice applicativo *Perl,* denominato *plain2xml.perl* (converte un semplice file di testo in un file XML). I files che contengono le regole grammaticali devono avere un formato consono alla DTD *RuleSpec.dtd.*

2.1 Fsgmatch a livello carattere

Viene di seguito illustrato il funzionamento di *fsgmatch* a livello carattere. La comprensione completa di tali istruzioni prevede una minima familiarità con la struttura di un file XML, la struttura di una DTD XML e la conoscenza base della sintassi nelle espressioni regolari usata nella programmazione in Perl.
A livello carattere *fsgmatch* processa un flusso di caratteri, cercando stringhe che combaciano (in seguito si utilizzerà l'espressione *match*) con espressioni regolari, specificate nelle regole (*RULEs*) contenute nei files della

grammatica (files con estensione .gr). Quando un *match* è verificato, la stringa viene riscritta rispettando le specifiche dettate dalla *RULE*.
Una RULE è un elemento XML RULE che contiene un certo numero di elementi REL. L'elemento RULE deve sempre avere l'attributo *name* ed eventualmente altri attributi. Nell'esempio di seguito l'attributo *targ* specifica cosa riscrivere una volta che si verifica il match descritto dalla rule.

```
<RULE name="cambia"  targ="poker di donne">
   <REL match="tris" rewrite="poker"></REL>
   <REL match="(\n | [ ])+di(\n | [ ])+"></REL>
   <REL match="re" rewrite="donne"></REL>
</RULE>
```

Presentandosi una serie di *match*, occorre specificare se questi devono essere verificati in sequenza o disgiuntamente tramite l'attributo *type.* In questo caso non essendo specificato, per default il valore dell'attributo è SEQ ovvero sequenziale. La RULE sopra descritta cerca una sequenza descritta dalla parola "*tris*" (senza apici), seguita da almeno uno o più caratteri di *newline* o *whitespace*; quindi la parola "*di*", seguita nuovamente da almeno uno o più caratteri *newline* o *whitespace* ed infine la parola "*re*". Se tale sequenza è trovata, essa viene completamente sostituita dalla stringa "*poker di donne*".
Volendo preservare il numero di *whitespace* e *newline*, che compaiono nella stringa cercata, occorre riscrivere la RULE nel seguente modo:

```
<RULE name="cambia2" targ="&A-REW;&B-VAL;&C-REW;">
   <REL var="A" match="tris" rewrite="poker"></REL>
   <REL var="B" match="&WSORNL;+di&WSORNL;+"></REL>
   <REL var="C" match="re" rewrite="donne"></REL>
</RULE>
```

Ogni REL che compone la RULE è individuata unicamente tramite l'attributo *var*. L'attributo *rewrite* riscrive solamente la porzione di stringa che verifica il match per quella unica REL. Il valore dell'attributo *targ* fa riferimento a delle *entità* esterne dichiarate nella DTD, che vengono assegnate al valore attuale della stringa, che verifica il match, oppure al valore specificato dall'attributo *rewrite.* Quindi quello che viene riscritto dalla RULE è il *rewrite* della prima

REL, seguito dal *match* della seconda REL ed infine il *rewrite* della terza REL. Nella seconda REL è stata sostituita l'espressione regolare “*(\n | [ ])*” con l'entità “&WSORNL;” che può essere dichiarata nello stesso file, che contiene tutte le RULE ,tramite l'istruzione:

<!ENTITY WSORNL “(\n | [ ])”>

Ritornando sull'attributo *type* oltre al valore SEQ sono ammessi i valori DISJF e DISJ i quali indicano che solamente una delle REL, che formano una RULE, deve essere verificata per avere match. Nel caso di DISJF la ricerca finisce al primo match valido mentre, nel caso DISJ, la ricerca termina sul match più esteso. Nel seguente esempio si vuole convertire la parola “*due*” con il numero “*2*” e la parola “*duecento*” con il numero “200” :

```
<RULE name="due" type=”DISJ”  targ="&S-REW;">
   <REL match="due"          rewrite="2"></REL>
   <REL match="duecento"  rewrite="200"></REL>
</RULE>
```

In questo caso le operazioni sopra indicate sono eseguite correttamente. Ma se il valore dell'attributo *type* è sostituito con DISJF, il risultato sarebbe “*2*” e “*2cento*” perché ogni volta, il match si fermerebbe alla prima REL, in quanto, mentre si può pensare erroneamente che la prima REL cerchi la parola “*due*” in realtà cerca la sequenza delle tre lettere specificate nel match, indipendentemente dal contesto nel quale si trovino.

Lo scopo principale di *fsgmatch* a livello carattere è quello di produrre dei markup XML, ovvero avvolgere stringhe di caratteri con dei tags iniziali e finali. Come accennato in precedenza, il meccanismo di riscrittura non permette di utilizzare direttamente il carattere “<” per indicare l'inizio della definizione di un elemento, in quanto è un carattere riservato nei costrutti del codice XML. Occorre quindi utilizzare l'entità “*<*” riconoscita da XML come carattere ”<”, quando questo viene utilizzato fuori dai costrutti. “*&lt*;” è un’entità interna di XML che viene espansa nella forma conosciuta “*<*”. Ad esempio, la seguente RULE produce un markup attorno alla virgola di

punteggiatura:

```
<RULE name="virgola"   targ="&lt;W C='CM'>&S-VAL;&lt;/W>”>
   <REL match="," ></REL>
</RULE>
```

Otteniamo:

```
&#60;W C='CM'>,&#60;/W>
```

Occorre utilizzare un semplice programma Perl, denominato *openangle.perl* per sostituire “*&#60*” con il carattere “<”. Il risultato finale, in quanto dobbiamo ottenere comunque un file XML ad elaborazione terminata, risulta essere:

```
<W C='CM'>,</W>
```

In questo modo abbiamo trasformato il carattere *virgola* in un elemento XML. In particolare abbiamo creato l’elemento **W** il cui attributo *C* ha valore **CM**. Il valore dell’attributo identifica il contenuto dell’elemento. Va ricordato che *fsgmatch* è solamente un “interprete” delle RULE definite in un file .gr e che quindi può essere pensato come un generico compilatore per un dato linguaggio, mentre effettivamente il contenuto dei file .gr determina il lavoro che si vuole ottenere. Di seguito vengono riportate alcune RULE contenute in *words2.gr* allo scopo di elencare altre propietà utili per la comprensione del programma.

E' possibile definire REL che chiamino altre RULE:

```
<RULE name="quote-or-br" type="DISJF">
   <REL type="REF" match="quote"></REL>
   <REL type="REF" match="bracket"></REL>
</RULE>
```

La seguente RULE definisce una disgiunzione di due RELs dove l'attributo type=”REF” indica che le stringhe, che devono verificare il match, sono definite da altre RULEs. Il nome di queste altre RULEs sono codificate come il valore dell'attributo *match*, ovvero in questo caso “*quote*” e “*bracket*”. E'

possibile raggruppare insiemi di RELs:

```
<RULE name="words" targ="&A-REW;&B-REW;">
   <REL var="A" type="REF" match="quote-or-br"
                           m_mod="QUEST"></REL>
   <REL var="B" type="GROUP"  match="DISJF">
      <REL type="REF" match="word-punct"></REL>
      <REL type="REF" match="word"></REL>
      <REL type="REF" match="symbol-word"></REL>
   </REL>
</RULE>
```

TYPE=”GROUP” permette ad una REL di contenere un certo numero di RELs che possono essere interpretate sequenzialmente o disgiuntamente, come nel caso sopra. In questo caso la RULE esegue un match quando viene individuata una sequenza descritta da *quote-or-br* e seguita disgiuntamente da *word-punct*, *word* o *symbol-word*.
“&A-REW;” e “&B-REW;” contengono i valori dell'attributo *rewrite,* associati alla RULE “quote-or-br” ed ad una delle rule che seguono. Se le rule chiamate non contengono l'attributo *rewrite*, è possibile per quanto visto precedentemente, sostituire il valore di *targ* con “&A-VAL;” e “&B-VAL;”.
D’importanza fondamentale è l'attributo *m_mod* usato per controllare il match realizzato dalla REL. In questo caso il valore dell'attributo è QUEST ed ha lo stesso significato del simbolo “*?*” in una espressione regolare, ovvero il match è opzionale. Altri valori sono STAR (equivalente di “*”) e PLUS (equivalente di “+”) che specificano rispettivamente zero o più interazioni e una o più interazioni. Infine TEST, come mostrato nell'ultimo esempio,:

```
<RULE name="word-ws" targ="&A-REW;">
   <REL var="A" type="REF" match="words"></REL>
   <REL var="B"  match="&WSORNL;+" m_mod="TEST"></REL>
</RULE>
```

In questo caso la RULE *word-ws* è verificata quando viene trovata una parola definita dalla RULE *words* e seguita da *whitespace* o *newline*. L'attributo m_mod=”TEST” fa sì che la seconda REL venga effettivamente verificata ma non compaia come parte del match completo della RULE *word-ws*. In questo modo si verifica che la parola trovata da *words* sia una parola intera (perchè

seguita da whitespace o newline) e non una porzione di parola.

2.2 Fsgmatch a livello SGML

A livello SGML, *fsgmatch* processa un flusso di elementi per raggrupparli e creare nuovi elementi di estensione maggiore. Come visto in precedenza, l'ultima operazione a livello carattere consiste nel marcare come elementi W, ovvero WORD, parole tramite *words2.gr*. Questo significa che in prima analisi tutti gli elementi hanno lo stesso nome, quindi indistinguibili tra di loro. Per questo motivo a livello SGML è tuttavia possibile verificare il contenuto (PCDATA) di un elemento attraverso il match esatto della stringa marcata oppure attraverso l'utilizzo di espressioni regolari.

```
<REL match="W/#=un"></REL>
```

La REL sopra cerca un elemento W il cui contenuto è esattamente la stringa "*un*". Il simbolo"#" indica che si cerca il contenuto dell'elemento, mentre "=" indica che quello che segue è l'espressione intera ed esatta da cercare.
Nel caso si usassero espressioni regolari, la forma cambia nel modo seguente:

```
<REL match="W/#~^[Uu]n$"></REL>
```

In questo caso si cerca indipendentemente "Un" oppure "un". Il simbolo"~" indica la ricerca di un'espressione regolare mentre "^" e "$", a monte ed a valle dell'espressione regolare, indicano l'inizio e la fine del contenuto dell'elemento.
Nella stesura di un file .gr si farà uso frequente delle *entità* per non dover riscrivere dei match usati molto frequentemente. In tal modo la REL sopra può essere riscritta nel seguente modo, previa definizione della entità a cui fa riferimento.

```
<ENTITY  UN  "W/#~^[Uu]n$">
```

```
<REL match="&UN;"></REL>
```

Qui di seguito una RULE di numbers2.gr illustra come raggruppare elementi per creare un nuovo elemento di dimensioni maggiori.

```
<RULE name="milione" targ_sg="PHR[C='CD']" >
   <REL type="GROUP" match="DISJF">
      <REL          match="&UN;"></REL>
      <REL  match="&CD-DIGIT;/#~^[1-9][0-9]*$"></REL>
      <REL  type="REF" match="textnum"></REL>
   </REL>
   <REL  match="W/#~^milion[ei]$"></REL>
</RULE>
```

Il match è valido quando viene trovata la successione di due elementi descritti dalle RELs sopra. La prima REL (TYPE="GROUP", quindi, contiene a sua volta altre RELs) cerca un elemento che sia disgiuntamente la parola "*un*" oppure un numero cardinale espresso tramite lettere oppure un numero cardinale espresso tramite espressione letteraria. La seconda REL cerca un elemento W il cui contenuto sia la parola "*milione*" oppure "*milioni*". Se tale sequenza viene trovata, essa viene marcata come nuovo elemento XML dove il nome ed il valore dell'attributo sono specificati dall'attributo targ_sg="PHR[C='CD'] definito nella RULE iniziale. In questo caso il valore dell'attributo "CD" indica che il contenuto del nuovo elemento è un numero cardinale.
Supponiamo che la sequenza di elementi in ingresso sia la seguente (precedentemente marcata da words2.gr):

```
<W C='W'>pagato</W> <W C='W'>meno</W> <W C='W'>di</W>
<W C='W'>tre</W> <W C='W'>milioni</W><W C='.'>.</W>
```

IL risultato della RULE *milione* è:

```
<W C='W'>pagato</W> <W C='W'>meno</W> <W C='W'>di</W>
<PHR C='CD'><W C='CD'>tre</W> <W
C='W'>milioni</W></PHR>
<W C='.'>.</W>
```

Da attenta analisi si può notare come la parola "*tre*" abbia cambiato valore

dell'attributo. Questa operazione è stata effettuata dalla RULE *textnum* in quanto a livello SGML è possibile apporre questa modifica ai singoli elementi come mostrato nell'ultimo esempio:

```
<RULE name="1-to-99"  targ_sg="@[C='CD']">
   <REL match="W/#~^&NN;$"></REL>
</RULE>
```

Tale RULE, come spiegato successivamente, identifica numeri cardinali, scritti tramite espressioni letterarie, fino alla cifra "*99*", facendo riferimento ad una entità inserita ad inizio del file *numbers2.gr*. Il simbolo"@" indica che deve essere solamente cambiato il valore dell'attributo dell'elemento che verifica il match.

2.3 Lessico esterno

Negli esempi precedenti è stato introdotto l'uso delle entità per evitare di riscrivere match comuni e largamente utilizzati. Può succedere che alcuni match debbano verificare l'appartenenza di una stringa ad un determinato insieme. Per tale motivo parole o frasi che hanno una caratteristica comune possono essere inserite in un lessico esterno al quale riferirsi per eseguire il match. Ad esempio "*cento*" , "*mille*", "*milione*" e "*miliardo*" possono essere raggruppate ed individuate attraverso un TAG comune ed inserite in un lessico esterno. Tale lessico è un semplice file di testo, dove ogni parola è seguita da uno o più TAG, come riportato nell'esempio:

- cento BIG-UNIT
- mille BIG-UNIT
- milione BIG-UNIT
- miliardo BIG-UNIT

In *numbers2.gr* viene utilizzato un lessico esterno individuato dal file *numbers2.lex*. Per poter utilizzare il lessico occorre creare all'interno di

numbers2.gr un elemento LEX:

```
<LEX type="PHRASE"
   file_name="&TTTDIR;/LEX/numbers2.lex"
   alias="LEX">
</LEX>
```

L'attributo type="PHRASE" indica che il lessico può contenere, oltre a semplici parole anche frasi intere. Successivamente viene fornito il percorso dove si trova tale file ed infine una maniglia con la quale richiamare il file stesso. L'esempio successivo è una dimostrazione di come viene utilizzato il lessico esterno in numbers2.gr:

```
<RULE  name="quant" type="PHRASE"
targ_sg="PHR[C='QUANT']" >
    <REL  match="&WRD;"  >
      <CONSTR   check_in= "LEX"
              subpart='([a-z]+)[ea]$'
              check_tags="QUANT *"
              check_mod="LOWERCASE"  >
      </CONSTR>
   </REL>
</RULE>
```

Per prima cosa attraverso l'entità "*&WRD;*" si verifica che l'elemento cercato sia di tipo W, poi attraverso l'elemento CONSTR si accede al lessico esterno per verificarne il contenuto. L'attributo *Check_in* richiama, tramite la maniglia, il file che contiene il lessico (possono esserci diversi file lessicali). L'attributo *check_tags* indica in quale categoria si deve cercare la parola ovvero, in questo caso, una parola che abbia etichetta QUANT (l'asterisco indica che può avere anche altre etichette oltre a quella specificatamente cercata). L'attributo *subpart* indica che il contenuto del lessico è solamente una parte della parola che si vuole etichettare. In questo caso la RULE cerca una parola i cui caratteri, tranne l'ultimo, appartengano ad una parola definita nel lessico. Tale parola cercata dalla RULE deve terminare poi con la lettera "*e*" oppure con la lettera "*a*".
Attraverso le nozioni fornite sopra si hanno a disposizione tutti gli strumenti per comprendere un *file.gr* e per poter scrivere delle RULEs.

Capitolo 3
Esempio completo di testo etichettato

Il primo passo della sequenza di comandi è quello di trasformare il semplice file di testo in un file XML. Viene utilizzata una procedura scritta in Perl. Supponiamo che il testo da analizzare sia il seguente:

Nel luglio del 2002 la CEG Corp. ha fatturato 100 milioni di euro.

La scorsa notte la compagnia ha annunciato una crescita del 20% .

Il risultato dell'elaborazione risulta essere:

```
<?xml version='1.0'?>
<!DOCTYPE DOCSSYSTEM '/home/faio/TTT_v1.0/RES/general.dtd,xml'>
<DOCS>
<TEXT>
Nel luglio del 2002 la CEG Corp. ha fatturato 100 milioni di euro.

La scorsa notte la compagnia ha annunciato una crescita del 20% .
</TEXT>
</DOCS>
```

Figura 3.1

3.1 Elaborazione a livello carattere

La conversione ad un file XML è ottenuta inserendo un XML *header* ed una linea DOCTYPE che punta ad una DTD esterna per il documento. Il programma Perl inoltre racchiude il contenuto del testo in un elemento TEXT a sua volta racchiuso in un elemento DOCS.

Nel passo seguente vengono marcati paragrafi e titoli utilizzando *fsgmatch* a livello carattere attraverso le regole contenute nel file *paras.gr* . Il risultato dell'elaborazione non altera la stringa d'ingresso eccetto che avvolgere titoli e paragrafi con *tags* iniziali e finali.
Di seguito il risultato dell'elaborazione e la successiva rielaborazione tramite il programma Perl.

```
<?xml version='1.0'?>
<!DOCTYPE DOCS SYSTEM '/home/faio/TTT_v1.0/RES/general.dtd,xml'>
<DOCS>
<TEXT>
&#60;P>Nel luglio del 2002 la CEG Corp. ha fatturato 100 milioni di
euro.&#60;/P>

&#60;P>La scorsa notte la compagnia ha annunciato una crescita del
20%.&#60;/P>
</TEXT>
</DOCS>
```

Figura 3.2

```
<?xml version='1.0'?>
<!DOCTYPE DOCS SYSTEM '/home/faio/TTT_v1.0/RES/general.dtd,xml'>
<DOCS>
<TEXT>
<P>Nel luglio del 2002 la CEG Corp. ha fatturato 100 milioni di euro.</P>

<P>La scorsa notte la compagnia ha annunciato una crescita del20%.</P>
</TEXT>
</DOCS>
```

Figura 3.3

Viene utilizzato nuovamente *fsgmatch* a livello carattere per segmentare paragrafi e titoli in parole individuali attraverso le regole elencate in *words.gr*. Anche in questo caso è necessaria una post-elaborazione attraverso *openangle.perl.*

```
<?xml version='1.0'?>
<!DOCTYPE DOCS SYSTEM '/home/faio/TTT_v1.0/RES/general.dtd,xml'>
<DOCS>
<TEXT>
<P><W C='W'>Nel</W> <W C='W'>luglio</W> <W C='W'>del</W>
<W C='CD'>2002</W> <W C='W'>la</W> <W C='W'>CEG</W>
<W C='W'>Corp.</W> <W C='W'>ha</W> <WC='W'>fatturato</W>
<W C='CD'>100</W> <W C='W'>milioni</W> <W C='W'>di</W>
 <W C='W'>euro.</W></P>

<P><W C='W'>La</W> <W C='W'>scorsa</W><W C='W'>notte</W>
<W C='W'>la</W> <W C='W'>compagnia</W> <W C='W'>ha</W>
<W C='W'>annunciato</W><WC='W'>una</W><W C='W'>crescita</W>
<WC='W'>del</W><W C='CD'>20</W><W C='W'>%</W>
<W C='.'>.</W></P>
</TEXT>
</DOCS>
```

Figura 3.4

Il passo successivo utilizza un *sentence boundary disambiguator*, *ltstop.* Viene esaminato ogni carattere di *full stop (.)* sia che sia marcato come parola separata sia se incluso nella parola precedente. Per ognuno di essi il programma determina quale effettivamente è un carattere di fine frase oppure parte di una abbreviazione. Se appartiene al primo caso esso viene marcato singolarmente come word (<W C='.'>.</W>) altrimenti non viene separato dalla parola della quale indica l'abbreviazione (Corp. nel nostro esempio). Nel caso sia entrambi viene creato un elemento vuoto (<W C='.'></W>) di seguito all'abbreviazione per indicare che è anche un carattere di fine frase.

```
<!DOCTYPE DOCS SYSTEM "/home/faio/TTT_v1.0/RES/general.dtd,xml'>
<DOCS>
<TEXT>
<P><W C='W'>Nel</W> <W C='W'>luglio</W> <W C='W'>del</W>
<W C='CD'>2002</W> <W C='W'>la</W> <W C='W'>CEG</W>
<W C='W'>Corp.</W> <W C='W'>ha</W> <WC='W'>fatturato</W>
<W C='CD'>100</W> <W C='W'>milioni</W> <W C='W'>di</W>
<W C='W'>euro</W><W C='.'>.</W></P>

<P><W C='W'>La</W> <W C='W'>scorsa</W><W C='W'>notte</W>
<W C='W'>la</W><W C='W'>compagnia</W> <W C='W'>ha</W>
<W C='W'>annunciato</W> <W C='W'>una</W><W C='W'>crescita</W>
<W C='W'>del</W><W C='CD'>20</W><W C='W'>%</W>
<W C='.'>.</W></P>
</TEXT>
</DOCS>
```

Figura 3.5

3.2 Elaborazione a livello sgml

L'elaborazione a livello carattere termina con l'operazione precedente. Nei passi successivi si prosegue a livello SGML, ovvero viene utilizzato *fsgmatch* per raggruppare sequenze di elementi XML. Prima di questa operazione ogni singola parola, numero e carattere di punteggiatura è etichettata come singolo elemento XML.

Il primo insieme di regole che operano a livello SGML è descritto in *numbers2.gr*. Queste regole identificano numeri e quantità espresse tramite sequenze di parole. Nel nostro esempio le parole "*100*" e "*milioni*" viste singolarmente come elementi XML vengono raggruppate per formare l'elemento <PHR C='CD'>.

```
<?xml version='1.0'?>
<!DOCTYPE DOCS SYSTEM
'/home/faio/TTT_v1.0/RES/general.dtd,xml'>
<DOCS>
<TEXT>
<P><W C='W'>Nel</W> <W C='W'>luglio</W> <W C='W'>del</W>
<W C='CD'>2002</W> <W C='W'>la</W> <W C='W'>CEG</W>
<W C='W'>Corp.</W> <W C='W'>ha</W><W C='W'>fatturato</W>
<PHR C='CD'><W C='CD'>100</W><W C='W'>milioni</W></PHR>
<W C='W'>di</W><W C='W'>euro</W><W C='.'>.</W></P>


<P><W C='W'>La</W> <W C='W'>scorsa</W><W C='W'>notte</W>
<W C='W'>la</W><W C='W'>compagnia</W> <W C='W'>ha</W>
<W C='W'>annunciato</W> <W C='W'>una</W><W C='W'>crescita</W>
<W C='W'>del</W><W C='CD'>20</W><W C='W'>%</W>
<W C='.'>.</W></P>
</TEXT>
</DOCS>
```

Figura 3.6

Successivamente attraverso l'elaborazione del file *numex2.gr* vengono etichettate le sequenze che individuano quantità monetarie e cifre percentuali. Nel nostro esempio "*100 milioni*" precedentemente marcato viene raggruppato assieme alla sequenza "*di euro*" nell'unico elemento <NUMEX TYPE='MONEY'>.

```
<?xml version='1.0'?>
<!DOCTYPE DOCS SYSTEM'/home/faio/TTT_v1.0/RES/general.dtd,xml'>
<DOCS>
<TEXT>
<P><W C='W'>Nel</W> <W C='W'>luglio</W><W C='W'>del</W> <W
C='CD'>2002</W> <W C='W'>la</W><W C='W'>CEG</W>
<W C='W'>Corp.</W> <W C='W'>ha</W> <W C='W'>fatturato</W>
<NUMEX TYPE='MONEY'><PHR C='CD'><W C='CD'>100</W>
<W C='W'>milioni</W></PHR> <W C='W'>di</W>
<W C='W'>euro</W></NUMEX></P>

<P><W C='W'>La</W> <W C='W'>scorsa</W><W C='W'>notte</W>
<W C='W'>la</W><W C='W'>compagnia</W> <W C='W'>ha</W>
<W C='W'>annunciato</W> <W C='W'>una</W><W C='W'>crescita</W>
<W C='W'>del</W><NUMEX TYPE='PERCNT'><W C='CD'>20</W>
<W C='PTC'>%</W></NUMEX><W C='.'>.</W></P>
</TEXT>
</DOCS>
```

Figura 3.7

Infine *timex2.gr* marca espressioni temporali, dalle più semplici a quelle di maggior complessità. Nel secondo paragrafo viene etichettata la sequenza "*notte scorsa*", mentre nel primo la quantità "*2002*" precedentemente marcata, viene raggruppata assieme alla sequenza "*nel luglio del"* nell'unico elemento <TIMEX TYPE='DATE'>.

```
<?xml version='1.0'?>
<!DOCTYPE DOCS SYSTEM'/home/faio/TTT_v1.0/RES/general.dtd,xml'>
<DOCS>
<TEXT>
<P> <TIMEX TYPE='DATE'><W C='W'>Nel</W><W C='W'>luglio</W>
<W C='W'>del</W><W C='CD'>2002</W></TIMEX> <W C='W'>la</W>
<W C='W'>CEG</W> <W C='W'>Corp.</W> <W C='W'>ha</W>
<W C='W'>fatturato</W> <NUMEX TYPE='MONEY'><PHR C='CD'>
<W C='CD'>100</W><W C='W'>milioni</W></PHR> <W C='W'>di</W>
<W C='W'>euro</W></NUMEX></P>

<P><W C='W'>La</W> <TIMEX TYPE='DATE'><W C='W'>scorsa</W>
<W C='W'>notte</W></TIMEX> <W C='W'>la</W>
<W C='W'>compagnia</W> <W C='W'>ha</W>
<W C='W'>annunciato</W> <W C='W'>una</W> <W C='W'>crescita</W>
<W C='W'>del</W> <NUMEX TYPE='PERCENT'><WC='CD'>20</W>
<W C='PTC'>%</W></NUMEX><W C='.'>.</W></P>
</TEXT>
</DOCS>
```

Figura 3.8

Terminata questa visione d'insieme dei file grammaticali, nei capitoli successivi vengono descritte singolarmente le varie RULEs che permettono di realizzare le etichette mostrate in questi passaggi.

Capitolo 4
Paras.gr e words2.gr

Questo è il primo file grammaticale interpretato da *fsgmatch* ed ha lo scopo di marcare paragrafi e titoli di un documento di testo. Non è stata fatta nessuna modifica alla versione originale. Il testo da etichettare appare come un intero elemento <TEXT>, come riportato in *Figura 3.1*. Vengono etichettati titoli e paragrafi. Un paragrafo termina quando il successivo è separato da almeno due *newline*.

4.1 RULEs di paras.gr

Tali RULEs vengono eseguite in sequenza in modo disgiunto, come spiegato nel *paragrafo 2.1* precedentemente visto.

- **one-line-file**: se il testo è composto da una sola riga, questa viene marcata come singolo elemento **<P>**.

- **initialtitleline**: individua la presenza di un titolo iniziale e lo marca come elemento **<TITLE>**. Un titolo è una riga o due di caratteri solamente maiuscoli che può contenete i principali carattere di punteggiatura.

- **intextitleline**: individua un titolo compreso tra due paragrafi.

- **first-para**: individua il primo paragrafo del testo quando non vi è presente il titolo.

- **last-para**: individua l'ultimo paragrafo del testo.

- **parabreak**: individua la spazio "vuoto" tra due paragrafi, ovvero la presenza di almeno due caratteri *newline*.

Per comprendere il funzionamento di tali RULEs, spieghiamo come viene marcato l'esempio iniziale del capitolo 2.
La prima RULE eseguita è *firstpara* che inserisce, tra la dichiarazione dell'elemento **<TEXT>** e il primo carattere del paragrafo iniziale, un elemento **<P>**. Successivamente viene eseguita *last-para* che inserisce, tra l'ultimo carattere del paragrafo finale e la dichiarazione di fine elemento **</TEXT>,** l'elemento **</P>**. Come ultimo passo parabreak inserisce, a monte ed a valle delle *newline* tra i due paragrafi, **</P>** e **<P>**.
In definitiva tali RULEs non marcano il singolo paragrafo per poi passare al successivo, ma individuano la presenza di paragrafi e aggiungono i caratteri di inizio o fine elemento ogni volta che verificano la presenza di *newline*.

4.2 words2.gr

Vengono di seguito descritte le regole grammaticali che compongono il file *words2.gr*. A differenza di *paras.gr* sono state fatte delle modifiche ad alcune RULEs per permettere un corretto funzionamento con la lingua italiana. D'ora in avanti non si farà più riferimento alla versione originale, evidenziando le differenze e le modifiche apportate, ma si procederà con la descrizione della versione in italiano, in quanto scopo della tesi è di sfruttare il software messo a disposizione e le sue idee base, non quello di adattarlo alla lingua italiana.
Words.gr etichetta ogni singola parola ed ogni singolo carattere di punteggiatura come elemento **<W>** (*Word*). Come detto in precedenza ogni elemento è caratterizzato dal valore dell'attributo. Ad esempio i numeri cardinali hanno valore dell'attributo "CD". I valori degli attributi sono gli stessi

utilizzati nella versione originale.

Nella forma più generale, una parola è una sequenza di caratteri che termina con il carattere *whitespace* o *newline*. Nell'uso comune generalmente, con il termine "parola" intendiamo ad esempio un sostantivo, un verbo, una preposizione priva dei caratteri di punteggiatura, quindi semplici sequenze di caratteri appartenenti all'alfabeto italiano.

4.3 RULE punct-ws

Questa RULES marca ogni carattere di punteggiatura come singolo elemento *word* solo se è seguito da un carattere whitespace oppure newline. Di seguito vengono mostrati i valori degli attributi per ogni singolo carattere:

comma	**,**	**:**	**;**	C=CM
percent	**%**			C=PTC
ampersand	**&**			C=AMP
quote	**‘‘**	**“**		C=QUOTE
braket	**{ }**	**( )**	**[ ]**	C=BR
dots	**…**	**..**		C=DASH
dash	**-**			C=DASH
mark	**?**	**!**		C=.
plus	**+**			C=PLUS

Tabella 4.1

Il carattere di *fullstop*, ovvero il carattere punto, non viene marcato da questa RULE in quanto può appartenere alla parola che lo precede per indicarne l'abbreviazione.

4.4 RULE word

Questa RULE individua come elementi *word* (*figura 3.4*) tutte le sequenze di caratteri che corrispondono alle regole da essa contenute. Occorre ricordare che stiamo lavorando a livello carattere e che quindi è necessario instaurare una gerarchia delle RULEs chiamate in modo da non marcare erroneamente

sequenze, come mostrato nel capitolo precedente. La prima RULE chiamata *Hyphen-rules*, separa e marca singolarmente due parole unite dai caratteri “-“ e “/”. Anche i caratteri di separazione vengono marcati in base alla tabella riportata sopra.
Nella tabella seguente vengono riportate le altre RULE chiamate, esempi delle sequenze che marcano ed il valore dato all’attributo.

ord	1°	1o	20esimo	C=”ORD”
g-m-a	12/1/79	03-12-1975		C=”GMA”
frac	3/4			C=”FRAC”
alphanum	Ita16	57/TG/6F	FF:6-d	C=”AN”
realword	Mario	perché	un'	C=”W”
cd	1	12	123	C=”CD”

Tabella 4.2

Altri esempi si possono dedurre dai listati dei programmi riportati in appendice.

Le due RULEs riportate sopra individuano le principali sequenze di caratteri marcati. Nell’analisi di un testo corretto gli apostrofi separano due parole senza la presenza del carattere *whitespace*.
Gli apostrofi non vengono marcati singolarmente, ma vengono inclusi nell'articolo o nella preposizione che li precedono, come evidenziato da *realword*.

4.5 RULE word-ws

Questa è la RULE principale di questo capitolo. Quando viene eseguita, nel testo sono stati marcati solo i caratteri di punteggiatura individuati da *punct-ws*.
Consideriamo il seguente esempio:

3 colori: verde, bianco e rosso

punct-ws fornisce il seguente risultato:

<P>tre colori<W C='CM'>:</W> verde<W C='CM'>,</W> bianco e rosso

quindi una parola per essere marcata correttamente, prima di terminare con un carattere *whitespace* o *newline* può essere seguita da un carattere di punteggiatura, precedentemente marcato come "*colori*" e "*verde*".
Nel caso il carattere di punteggiatura preceda la parola come parentesi o virgolette, esso non viene precedentemente marcato da *punct-ws* e quindi deve essere fatto da *word-ws*. Solo per questo paragrafo introduciamo i seguenti formalismi:

- **<punct>** = carattere di punteggiatura precedentemente marcato da punct-ws
- **punct** = carattere di punteggiatura riportato nella *tabella 4.1*
- **word** = esempi riportati nella *tabella 4.2*
- **qu_or_bra** = caratteri *quote* o *braket* riportati nella *tabella 4.1*
- **wsornl** = carattere *whitespace* o *newline*

Word-ws marca singolarmente nel modo appropriato le sequenze:

- **word – <punct> – wsornl**
- **qu_or_bra – word – <punct> – wsornl**
- **word – punct – <punct> – wsornl**
- **qu_or_bra – word punct – <punct> – wsornl**

Consideriamo il seguente esempio:

'disse: "evviva!"'

punct-ws fornisce il seguente risultato:

'disse<W C='CM'>:</W> "evviva!"'

Il carattere “ ’ ” non viene marcato perchè non appartiene alla categoria *punct*.
Di seguito *word-ws* fornisce:

<W C='QUOTE'>'</W><W C='W'>disse</W>
<W C='CM'>:</W> "evviva!"'

Per marcare correttamente la frase viene chiamata la RULE *final-word* che etichetta parole e caratteri di punteggiatura ignorando la presenza del carattere *wsornl*.

<W C='QUOTE'>"</W><W C='W'>evviva</W><W C='.'>!</W>
<W C='QUOTE'>"</W>’

viene etichettata in quanto riconosciuta la sequenza:

qu_or_bra – word – punct – punct

mentre il carattere “ ’ ” viene etichettato per ultimo come segue:

<W C='QUOTE'>'</W>

in quanto precedentemente non aveva assunto il significato di apostrofo eventualmente marcato da *realword*.

Capitolo 5
Numbers2.gr

In questo capitolo sono descritte le regole principali che permettono di individuare ed etichettare numeri e quantità rappresentati in modo testuale. Viene applicato *fsgmatch* a livello SGML, ovvero il match non è verificato su sequenze di caratteri, ma su sequenze di elementi XML. Quando una sequenza o un solo elemento verificano il match, allora vengono nuovamente etichettati, ossia viene cambiato il valore dell'attributo che avevano in precedenza (*figura 3.6*). Dovendo riconoscere sequenze di elementi, il problema è stato diviso in due parti: la prima parte consiste nell'individuare i singoli elementi precedentemenenti marcati tramite *words2.gr*; la seconda nel definire le principali strutture delle sequenze usate nella lingua italiana. Occorre individuare per prime le sequenze di parole più lunghe, successivamente quelle più corte e per ultimo i singoli elementi rimasti.
Ad esempio "*3 miliardi e 400 milioni*" non deve essere erroneamente marcato come:

```
<PHR C='CD'><W C='CD'>3</W> <W C='W'>miliardi</W></PHR>
<W C='W'>e</W><PHR C='CD'><W C='CD'>400</W>
<W C='W'>milioni</W></PHR>
```

mentre è corretto:

```
<PHR C='CD'><W C='CD'>3</W> <W C='W'>miliardi</W>
<W C='W'>e</W><W C='CD'>400</W>
<W C='W'>milioni</W></PHR>
```

Numbers2.gr è formata da tre RULEs principali:

- *all-range*
- *all-quant*
- *all-numbers*

5.1 RULE all-range

All-range racchiude sei RULEs di immediata comprensione, il cui scopo è quello di marcare espressioni che intendano intervalli di numeri cardinali oppure di anni temporali, come, ad esempio, le seguenti espressioni:

1999-2002	20-30
1999 o 2002	20 o 30
tra il 1999 ed il 2002	20 al 30

Si rimanda in appendice per la visione di queste RULE, mentre per la spiegazione della RULE *cardinals* occorre proseguire nella lettura.
Successivamente, come descritto nei prossimi capitoli, sono state realizzate un insieme di RULEs per marcare espressioni temporali nelle quali compaiono ore della giornata e numeri romani. Per tale motivo *all-range* oltre ai valori riportati nella tabella è in grado di marcare le seguenti sequenze:

XV – XVI	15:30 – 16:30
XV o Xvi	15,30 o 16,30
XV al XVI	15.30 alle 16.30

Nel *capitolo* 7 (*Timex2.gr*) vengono motivate le scelte che portano a marcare i range nella parte destra della tabella.

5.2 RULE all-quant

La premessa iniziale è che le regole realizzate non controllano la semantica, per cui sequenze di parole che hanno scarso significato possono essere

evidenziate erroneamente. E' fondamentale che il testo da analizzare sia scritto correttamente rispettando la grammatica italiana.

All-quant racchiude quattro RULEs che utilizzano un lessico esterno chiamato *numbers2.lex*, all'interno del quale sono definiti: gli aggettivi indefiniti, individuati tramite l'identificatore AGG (*pochi*, *alcuni*, *tanti*, *molti*); i sostantivi individuati tramite QUANT (*decina*, *dozzina*, *ventina*, *centinaia*, *migliaia*....); ed infine le espressioni numeriche letterarie, individuate tramite BIG-UNITS (*mille*, *milione*, *miliardo*...).

● quantity1

```
<RULE name="quantity1" targ_sg="PHR[C='QUANT']">
   <REL type="REF" match="aggettivo"></REL>
   <REL type="REF" match="quant"></REL>
   <REL                    match="&DI;" m_mod="QUEST"></REL>
   <REL type="REF" match="big" m_mod="QUEST"></REL>
</RULE>
```

Vengono marcate ad esempio le seguenti sequenze:

- alcune decine
- alcune decine di
- alcune decine di milioni

● quantity2

```
<RULE name="quantity2" targ_sg="PHR[C='QUANT']">
    <REL type="REF" match="quant"></REL>
    <REL          match="&DI;" ></REL>
    <REL type="REF" match="big"></REL>
 </RULE>
```

Vengono marcate ad esempio le seguenti sequenze:

- decine di
- decine di milioni

La RULE sopra risulta chiaramente molto simile alla prima, e pertanto si potrebbe unirle, ad esempio, in questo modo:

```
<RULE name="quantity1" targ_sg="PHR[C='QUANT']">
   <REL type="REF" match="aggettivo" m_mod="QUEST"></REL>
   <REL type="REF" match="quant"></REL>
   <REL                match="&DI;" m_mod="QUEST"></REL>
   <REL type="REF" match="big"    m_mod="QUEST"></REL>
</RULE>
```

solamente inserendo *m_mod="QUEST"* nella prima REL di *quantity1*.
Il problema sorge se nel testo compare un' espressione del tipo: "*sette (o 7) centinaia*"
La RULE sopra marcherebbe la parola "*centinaia*" nel seguente modo:

```
<PHR C='QUANT'><W C='W'>centinaia</W></PHR>
```

"Centinaia" è marcata come elemento "PHR" erroneamente in quanto singola parola. Infatti nel momento di scrivere le RULE, si deve pensare a tutte le combinazioni possibili di elementi e, soprattutto, di non marcare erroneamente sequenze di elementi o singoli elementi che concorrono in altre RULEs.
L'esempio proposto deve essere marcato in questo modo:

```
<W C='CD'>sette</W> <W C='QUANT'>centinaia</W>
```

ovvero "*centinaia*" rimasto separato da "*sette*".
Benché l'intera espressione sia considerata come unica quantità, è preferibile mantenerla separata. Questi formalismi si possono cambiare in seguito, ma questo comporta la riscrittura di molte RULE che solo un'attenta analisi può individuare.

- quantity3 e quantity4

Queste ultime due RULEs marcano sequenze di due sole parole ed elementi singoli quali ad esempio:

- pochi milioni
- molte decine
- pochi
- trentina

5.3 RULE all-numbers

La spiegazione immediata di questa RULE risulta molto laboriosa in quanto si presenta con una struttura ad albero molto ramificata, quindi si preferisce elencare le RULEs elementari, ovvero le foglie, per poi dare una visione dell'insieme.

5.3.1 RULE textnum

Dovendo cercare espressioni numeriche in forma letteraria, il primo problema sorge nel marcare numeri cardinali, quali "*trecentoventisette*".
La difficoltà nasce dal fatto che si lavora a livello SGML e non più a livello carattere. Si può comunque eseguire dei match a livello carattere all'interno di un elemento, ovvero di una parola, ma non sfruttare un lessico esterno come in *all-quant*. Questo perchè non è possibile scrivere un numero cardinale con un insieme finito di parole separate, cosa che invece avviene ad esempio per la lingua inglese.
La soluzione usata è quella di definire delle entità alle quali poi riferirsi e, quindi, cercare sequenze di entità all'interno di ogni singolo elemento.
Le entità UNIT, TEEN, TY e BIG-UNIT rappresentano rispettivamente le unità, i numeri da dieci a diciannove, le decine ed i multipli di dieci nella lingua italiana. Per marcare esattamente una decina basta la seguente RULE:

```
<RULE name="decine"  targ_sg="@[C='CD']">
   <REL match="W/#~^&TY;$"></REL>
</RULE>
```

La RULE cerca all'interno di un elemento “W” una stringa che inizi esattamente con il riferimento all'entità “TY” e che termini immediatamente dopo.
Ad esempio, “*trenta*” è definito all'interno dell'entità “TY”, quindi tale parola viene marcata giustamente nel seguente modo:

```
<W C='W'>trenta</W>
```

Volendo marcare “*trentuno*”, si può notare che tale parola è formata dall'unione di “*trent*” ed “*uno*”, quindi la RULE adatta risulta:

```
<RULE name="decine"  targ_sg="@[C='CD']">
   <REL match="W/#~^&TY;&UNIT;$"></REL>
</RULE>
```

In base a questi ragionamenti si costruisce l'entità *&NN;* che permette di marcare inequivocabilmente i numeri cardinali fino a novantanove.
Con lo stesso procedimento si marcano inequivocabilmente anche i numeri fino a novecentonovantanove nel seguente modo:

```
<RULE name="centinaia"  targ_sg="@[C='CD']">
   <REL match="W/#~^&UNIT;?&CENTO;&NN;?$"></REL>
</RULE>
```

Come si può notare in teoria, sfruttando le entità, si possono riconoscere espressioni letterarie molto complesse che contengono le migliaia, i milioni e, volendo, anche i miliardi. Purtroppo inserendo una quantità elevata di entità, la compilazione eseguita da *fsgmatch* fallisce in quanto vengono passati troppi parametri e, quindi, tale approccio oltre “*999*” non funziona. Per ovviare a questo inconveniente si esegue una ricerca di stringhe chiave, quali ad esempio la parola “*mila*” per identificare numeri che rappresentano migliaia, come ventidue***mila***trecentosette. L'inconveniente è che tutte le parole, che contengono la stringa chiave, possono essere marcate erroneamente, anche se hanno significati diversi (e.g. *milanési*). Per ridurre questo tipo di errore, le RULEs, che marcano numeri fino al milione

contengono altre parole chiave.
Numeri sopra il milione, scritti interamente a lettere, difficilmente si trovano nella letteratura italiana, quindi non è stata creata una RULE per identificarli.

5.3.2 RULE textnum2

Questa RULE marca espressioni numeriche in forma letteraria, scritte con più parole, come nell'esempio ad inizio capitolo. Sono marcate, ad esempio, le seguenti espressioni:

- un miliardo
- 20 miliardi e 300 milioni
- due miliardi e trenta milioni
- 40 milioni e 27 mila
- 40 milioni e trecento
- 3 mila e duecentododici
- tremiladuecentododici

Mentre i primi cinque esempi si trovano facilmente in un testo, difficilmente compare il sesto esempio, mentre risulta più probabile il settimo. Viene fatto notare che “*tremiladuecentododici*” viene marcato richiamando la RULE *textnum*, così come la seconda parte del sesto esempio per la parola “*duecentododici*”. Segue che *textnum* è inserita all'interno di *textnum2*.
Di seguito è riportata la struttura ad albero della RULE, evidenziando per il secondo ed il quarto esempio, il percorso fatto dall'algoritmo per eseguire il markup. Esso si basa principalmente sull'attributo TYPE=”DISJ” che, come spiegato anticipatamente, termina la ricerca sul match più esteso.

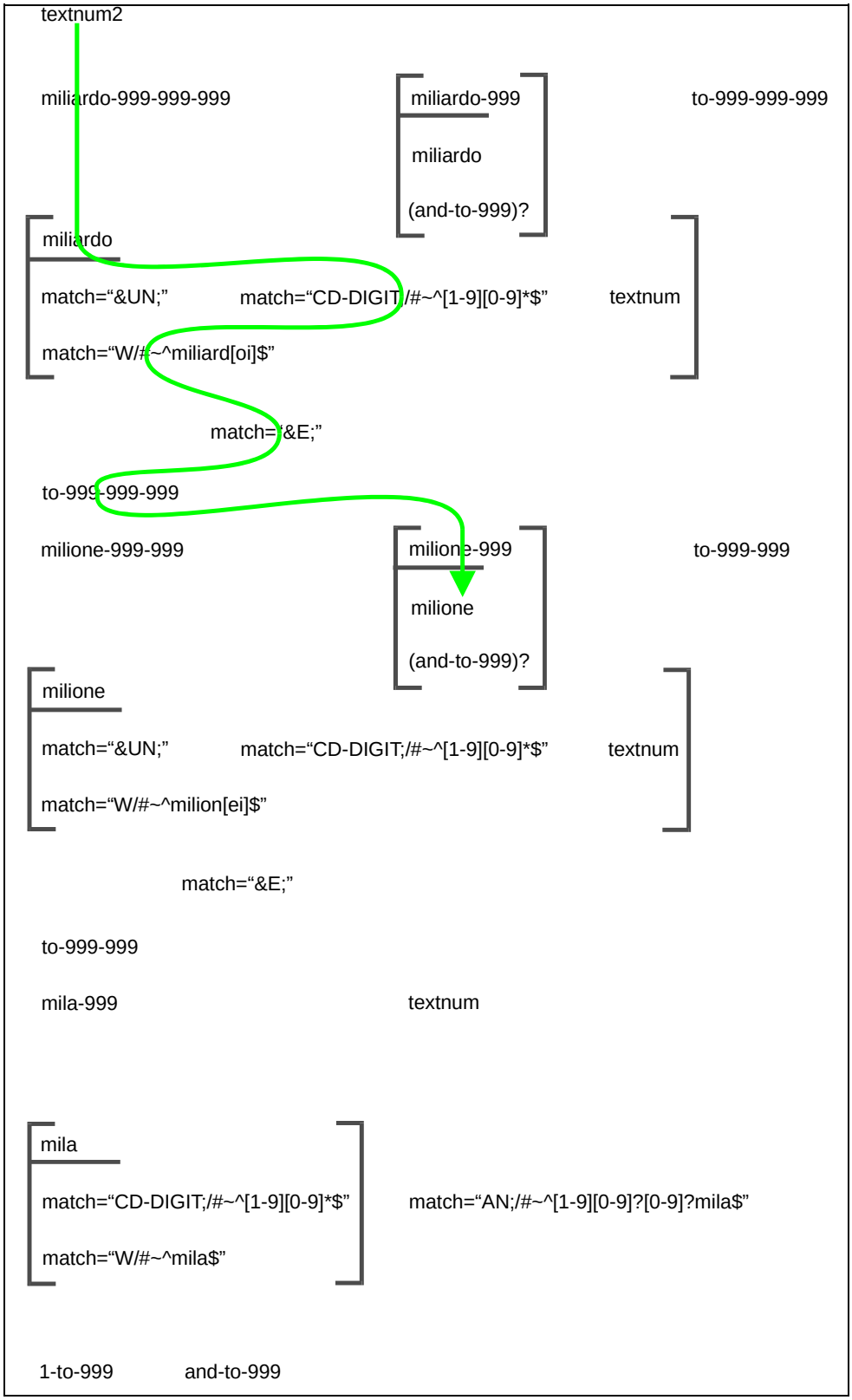

Figura 5.1: 20 miliardi e trecento milioni

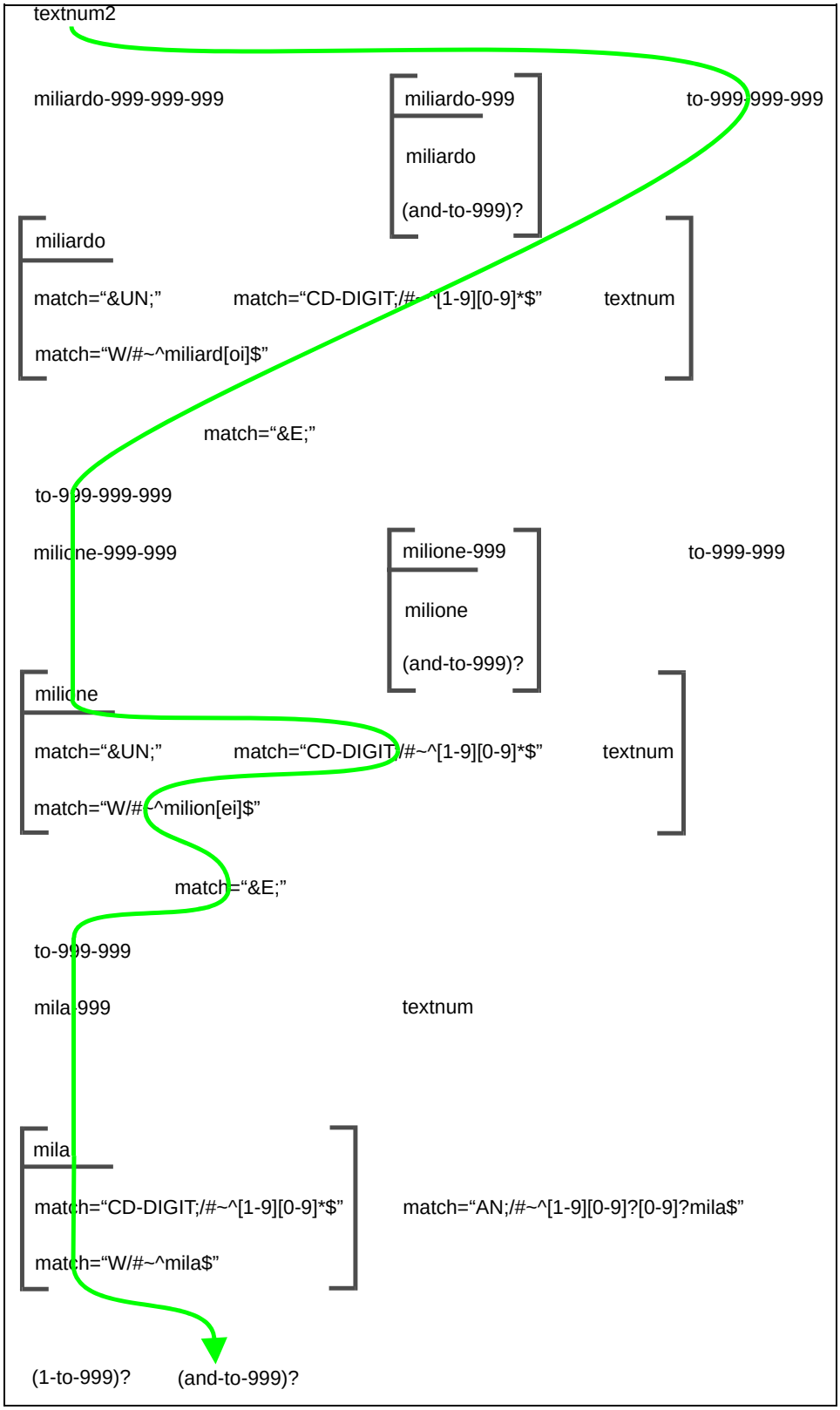

Figura 5.2: 40 milioni e 27 mila

5.3.3 RULE textnum-ordinal

Con questa RULE si apre la trattazione dei numeri ordinali (*primo*, *terzo*, *primi*, *tredicesimo*, *centoventisettesimi*...). Quello che viene subito evidenziato è l'ambiguità di tali espressioni letterarie in quanto, ad esempio, "*un terzo*" può riferirsi, a seconda del contesto dove si trova, ad un numero ordinale (*un terzo posto*), oppure ad una espressione frazionaria (*1/3*). Occorrerebbe un lessico esterno per analizzare sequenze di più parole e marcare in modo non ambiguo tali espressioni. Momentaneamente questa parte non è stata ancora sviluppata quindi le RULEs seguenti distinguono quelle che molto probabilmente sono espressioni frazionarie (*tre quinti*) da quelle che rimangono ambigue (*un terzo*).
Textnum-ordinal marca con l'attributo "C=ORD" tutte le parole che possono essere considerate numeri ordinali, perciò le parole nell'esempio precedente vengono tutte marcate "C=ORD". Questa operazione è ottenuta richiamando altre RULEs che hanno la stessa struttura delle RULEs contenute in *textnum*. Le RULEs elementari utilizzate marcano con l'attributo "C=FRACORD" (indica che può essere un ordinale o parte di un numero frazionario) i singoli match perché, come detto in precedenza, vi è ambiguità in partenza. Successivamente *textnum-ordinal* ne modifica l'attributo come mostrato di seguito:

```
<RULE name="ty-unith-amb"  targ_sg="@[C='FRACORD']">
   <REL match="W/#~^&TY;&UNITESIM;$"></REL>
</RULE>


<RULE name="amb-word-becomes-ord" type="DISJF"
targ_sg="@[C='ORD']">
   <REL type="REF" match="unith-amb"></REL>
   <REL type="REF" match="teenth-amb"></REL>
   <REL type="REF" match="tieth-amb"></REL>
   <REL type="REF" match="ty-unith-amb"></REL>
   <REL type="REF" match="over-100-amb"></REL>
   <REL type="REF" match="big-unith-amb"></REL>
</RULE>
```

```
<RULE name="textnum-ordinal" type="DISJF" >
   <REL type="REF" match="ordinali"></REL>
   <REL type="REF" match="ord-digit"></REL>
   <REL type="REF" match="amb-word-becomes-ord"></REL>
</RULE>
```

In precedenza erano già state marcate a livello carattere da “words2.gr” le espressioni che indicavano inequivocabilmente numeri ordinali quali, “*23esimo*” oppure “*23o*”.

● no-ord

Oltre alla ambiguità legata alle espressioni frazionarie, ci possono essere ambiguità quando la parola “*prima*” è un avverbio di tempo oppure la parola “*secondo*” ha significato di preposizione. Questa RULE marca con l'attributo “C=NONDEF” questi possibili casi semplicemente verificando se le parole sopra enunciate sono seguite da articoli determinativi (*i*, *il*, *le*), preposizioni (*di*, *della*) o dal pronome relativo invariante “*cui*”. La RULE è scritta sfruttando il valore TEST dell’attributo *m_mod* come descritto nel paragrafo 2.2.

```
<RULE name="no-ord-2" targ_sg="@[C='NONDEF']">
   <REL type="REF" match="to-999th-amb"></REL>
   <REL match="W/#~^cui$" m_mod="TEST"></REL>
</RULE>
```

L’insieme di RULEs che compongono *no-ord* sono le prime ad essere eseguite. Successivamente viene elaborata *textnum-ordinal* che etichetta come numeri ordinali tutti i restanti match verificati

● ord-digit

La RULE *ord-digit* verifica semplicemente la presenza di tale elemento. Può sembrare una operazione inutile, ma va ricordato che *textnum-ordinal* viene successivamente richiamata da altre RULEs, alle quali serve questa verifica per marcare delle sequenze.

5.3.4 RULE textnum-fraction

Questa RULE marca tutte le espressioni che si riferiscono a quantità frazionarie con l'attributo C='FRAC'. Una sequenza del tipo “*tre quinti*” verrà riconosciuta come numero frazionario anche se una minima ambiguità rimane in quanto “quinti” può riferirsi ad un numero ordinale. Per tale motivo, mentre l'intera sequenza viene marcata come elemento <PHR C='FRAC'>, la parola “quinti” rimane marcata come <W C='FRACORD'> per indicarne l'ambiguità.

<PHR C='FRAC'><W C='CD'>tre</W>
<W C='FRACORD'>quinti</W></PHR>

Di seguito sono descritte le RULEs appartenenti a *textnum-fraction* che permetto di ottenere il risultato esposto sopra.

- amb-word-denom-ord

Questa RULE verifica se i numeri ordinali sono preceduti dall'articolo indeterminativo “*un*”. In tal caso sono sicuramente ambigui, come spiegato ad inizio paragrafo, quindi vengono rimarcati come elementi <W C='FRACORD'>, escludendo l'articolo. Infatti nel testo può comparire la sequenza “il primo posto”, quindi, non essendoci l'articolo indeterminativo, “primo” rimane marcato <W C='ORD'> .

- fraction

Questa RULE individua espressioni frazionarie letterarie come “tre quinti”. Cerca una sequenza formata da un numero cardinale seguito da un numero ordinale:

```
<RULE name="to-999th-amb" type="DISJF">
   <REL type="REF" match="unith-amb"></REL>
   <REL type="REF" match="teenth-amb"></REL>
   <REL type="REF" match="tieth-amb"></REL>
   <REL type="REF" match="ty-unith-amb"></REL>
   <REL type="REF" match="over-100-amb"></REL>
   <REL type="REF" match="big-unith-amb"></REL>
</RULE>

<RULE name="frac-denom" type="DISJF">
   <REL type="REF" match="to-999th-amb"></REL>
   <REL type="REF" match="ord-digit"></REL>
 </RULE>

<RULE name="fraction" targ_sg="PHR[C='FRAC']">
   <REL type="REF" match="textnum"></REL>
   <REL type="REF" match="frac-denom"></REL>
</RULE>
```

Il risultato di questa elaborazione è quello mostrato nella tabella ad inizio paragrafo.

- frac-digit

Tutte le espressioni matematiche di quantità frazionarie vengono precedentemente marcate da *words2.gr* quindi questa RULE esegue una semplice verifica come quella eseguita da *ord-digit*.

5.3.5 RULE frac-num

Questa RULE marca espressioni che contengono quantità frazionarie, ma che, nel complesso, identificano numeri cardinali quale, ad esempio, “*tre quarti di milione*” oppure “*mezza dozzina*”. La comprensione risulta immediata, in quanto si appoggia interamente sulle RULEs descritte in precedenza.

```
<RULE name="frac-num" targ_sg="PHR[C='CD']">
   <REL type="GROUP"  match="DISJF">
         <REL match="W/#~^[Mm]ezz[ao]$"></REL>
          <REL type="GROUP"  match="SEQ">
              <REL type="REF" match="textnum-fraction"></REL>
              <REL          match="&DI;"></REL>
          </REL>
   </REL>
    <REL type="GROUP"  match="DISJF">
      <REL          match="W/#~iliardo$"></REL>
      <REL          match="W/#~ilione$"></REL>
      <REL          match="W/#~^dozzina$"></REL>
    </REL>
 </RULE>
```

5.3.6 RULE cardinals

Questa è l'ultima RULE di *numbers2.gr* ed ha lo scopo di marcare come numeri cardinali tutti i match eseguiti dalle RULEs *textnum* e *textnum2* che non sono stati marcati da altre RULEs in precedenza.
La RULE principale è *textnum-ordinal* che contiene a sua volta tre RULEs. Tutte le sequenze trovate vengono marcate come <PHR C='CD'> mentre i singoli elementi come <W C='CD'>.

- textnum-mezzo

Viene utilizzata per marcare espressioni quali “*un milione e mezzo*” oppure “*5 miliardi e mezzo*”, come dimostra la semplice costruzione della RULE:

```
<RULE name="textnum-mezzo" targ_sg="PHR[C='CD']">
   <REL type="REF" match="textnum2"></REL>
   <REL          match="&AND;"></REL>
   <REL    match="W/#~^mezzo$"></REL>
 </RULE>
```

Naturalmente, richiamando la RULE *textnum2*, sono marcabili espressioni prive di senso, come “*tre milioni e trecento e mezzo*”. Occorrerebbe scrivere una RULE specifica per ovviare a questo inconveniente, ma questo vale anche per tante altre RULEs viste in precedenza. L'idea base è che il testo

da analizzare sia corretto grammaticalmente e che contenga espressioni sensate ed usate abitualmente nella lingua italiana.

● textnum-and-fraction

Questa semplice RULE marca espressioni quali “ tre e tre quarti”. Sebbene risulterebbe più logica la seguente etichettatura

<W C='CD'>tre</W> <W C='W'>e</W>
<PHR C='FRAC'> <W C='CD'>tre</W> <W C='FRACORD'>quarti</W></PHR>

va ricordato che il significato è lo stesso di “*3,75*” quindi ci si riferisce ad un numero preciso. Per tale motivo tutta la sequenza viene etichettata come <PHR C='CD'>.

● textnum-cardinal

Come ultima, la RULE “textnum-cardinal” marca espressioni numeriche in forma letteraria che non sono state marcate, assieme ad altri elementi dalle RULEs precedenti. Il suo funzionamento è stato accuratamente descritto ad inizio capitolo. E' curioso notare come questa RULE sia utilizzata per ultima da sola, mentre viene richiamata diverse volte da altre RULEs.
In ordine cronologico la prima RULE di *numbers2.gr* creata è stata *textnum* e successivamente *textnum2*.

● digits

Infine l'ultima RULE in assoluto. Come altre viste in precedenza, si limita a verificare elementi precedentemente marcati <W C='CD'> da words2.gr

Capitolo 6
Numex2.gr

Lo scopo di questo insieme di RULEs è di marcare quantità monetarie e numeri percentuali. Sono state apportate solamente delle leggere modifiche al file originale e ad il lessico esterno. In particolare vengono marcate solamente espressioni dove compaiono le parole “*euro*” e “*dollari*” (*figura 3.7*), in quanto sono le monete che vengono maggiormente utilizzate per riferirsi al valore di un introito o di una spesa. E' sufficiente modificare il lessico esterno (cercare la parola in inglese e tradurla in italiano) per poter marcare qualsiasi altra unità monetaria.

6.1 RULE money

Le RULEs principali che formano la RULE *money* sono elencate di seguito seguite da un esempio di quello che marcano. L'ordine con il quale sono elencate è quello con cui vengono eseguite:

- bigsum-and-smallunit
  - 20 euro e 13 centesimi
  - 20 euro ed alcuni centesimi

- currency-number
  - $37

- number-currency

- più di1.300 euro
- meno di un milione di euro
- pochi euro in più
- alcuni euro in meno

● a-currency

- un euro

● number-smallunit

- 20 centesimi di euro
- 20 centesimi

6.2 RULE percent

Di seguito viene mostrata la porzione di codice che etichetta le espressioni percentuali. In particolare la RULE *percent*, evidenzia la struttura utilizzata per marcare le espressioni “*meno di un milione di euro”* ed ”*alcuni euro in meno”* utilizzata nella RULE *money.*

```
<RULE name="quantity" type="DISJF">
   <REL match="W[C='CD']"></REL>
   <REL match="PHR[C='CD']"></REL>
   <REL match="PHR[C='QUANT']"></REL>
   <REL match="PHR[C='RANGE']"></REL>
</RULE>
```

```
<!-- %, percento, per-cento -->
<RULE name="percent_wrd" type="DISJF">
   <REL match="W[C='PCT']"></REL>
   <REL type="GROUP"  match="SEQ" >
      <REL match="W/#~^[Pp]er$"></REL>
      <REL match="W/#~^cento$"></REL>
   </REL>
   <REL match="W/#~^((%)|([Pp]ercento))$"></REL>
</RULE>
<RULE name="percent" targ_sg="NUMEX[TYPE='PERCENT']">
   <REL type="GROUP"  match="SEQ" m_mod="QUEST" >
      <REL match="W/#~^((meno)|(più))$"></REL>
      <REL match="W/#~^del$"></REL>
   </REL>
   <REL    match="W/#=+" m_mod="QUEST"></REL>
   <REL type="REF" match="quantity"></REL>
   <REL type="REF" match="percent_wrd"></REL>
   <REL type="GROUP"  match="SEQ" m_mod="QUEST" >
      <REL match="W/#~^in$"></REL>
      <REL match="W/#~^((meno)|(più))$"></REL>
   </REL>
</RULE>
```

La RULE è estremamente semplice come si può osservare dai listati sopra. In particolare oltre alla quantità percentuale, viene marcato assieme il segno, ovvero il carattere “-” ed il carattere “+”. Mentre per quest'ultimo ne viene verificata la presenza nella sequenza, per il carattere “-” c’è una precedente marcatura, assieme ai numeri cardinali, in *numbers2.gr*., in quanto è una espressione maggiormente utilizzata.

Capitolo 7
Timex2.gr

Il funzionamento di *timex2.gr* è lo stesso di *numbers2.gr*. Viene applicato *fsgmatch* a livello SGML ovvero il match non è verificato su sequenze di caratteri, ma su sequenze di elementi XML. Questa RULE cerca all'interno del testo da analizzare sequenze di parole e, per ultime, singole parole che identificano inequivocabilmente espressioni temporali utilizzate nel lessico comune di tutti i giorni (*Figura 3.8*). Deve essere precisato che l'insieme di regole, che verranno successivamente descritte, sono state create riferendosi a sequenze tipiche riportate nella stesura di un comune quotidiano italiano.
Allo stesso modo di *numbers2.gr*, anche *timex2.gr* utilizza un lessico esterno chiamato *timex2.lex* nel quale sono inserite parole semplici quali i nomi dei giorni e dei mesi fino a espressioni composte come "*l'altro ieri*".
Si è presentato da subito un problema legato alla lunghezza delle sequenze da riconoscere, ovvero all'ampiezza delle finestre da etichettare.
Consideriamo il seguente esempio:

tra le 5 di sabato sera scorso e le 6 di domani pomeriggio

Le parole "*domani*", "*pomeriggio*", "*sabato*" e "*sera*" possono essere etichettate singolarmente in quanto appartengono al lessico esterno ed individuano espressioni temporali. Analogalmente "*domani pomeriggio*" e "*sabato sera*" sono da marcare raggruppate. Lo stesso si può dire di "*5 di sabato sera scorso*" , "*6 di domani pomeriggio*", "*5 di sabato sera*" e "*sabato sera scorso*".
Quindi *timex2.gr* marca interamente la frase dell'esempio sopra come unico

elemento XML che si riferisce ad una espressione temporale. Risulta evidente che la presenza di preposizioni ed avverbi contribuiscono ad aumentare la finestra ed a precisare un determinato intervallo di tempo. Nella stesura delle regole che seguono, non si è fatta nessuna distinzione tra espressioni che identificano precisi istanti temporali, riferiti al singolo giorno, ed espressioni temporali generiche, come intervalli di secoli.
Come in *numbers2.gr* occorre, per prima cosa, definire le finestre più piccole e successivamente creare delle RULEs che le raggruppino per identificare sequenze maggiori. Naturalmente questo approccio non è immediato, in quanto si deve prima classificare tutte le sequenze più lunghe per poi definire quelle più corte. A loro volta quelle più corte devono essere tali da poter essere successivamente utilizzate per creare quelle più lunghe.

Di seguito vengono riportate le RULEs che identificano le finestre elementari formate dalla combinazione di parole presenti nel lessico esterno.

7.1 RULE giorno-stagione-combinate-e-non

Questa RULE marca espressioni temporali che identificano una data precisa, in un determinato anno o mese, oppure un preciso periodo dell'anno.
Vengono eseguiti dei controlli onde evitare di etichettare frasi insensate quale, ad esempio, “*giovedi 32 ottobre*” come riportato di seguito:

```
<RULE name="namegiorno-data-mese"
targ_sg="TIMEX[TYPE='DATE']">
   <REL type="REF" match="nomi-giorni-lex"></REL>
   <REL match="W/#~^([1-9]|1[0-9]|2[0-9]|3[01])$"></REL>
   <REL type="REF" match="nomi-mesi-lex"></REL>
</RULE>
```

La seguente RULE etichetta date espresse tramite numeri cardinali quale, ad esempio, “*3-12-75*”. Come per altre RULEs in seguito, sarebbe comodo che una espressione di questo tipo venisse etichettata precedentemente a livello carattere in modo da dover, in seguito, solo cambiare il valore o il nome dell'attributo a livello SGML. Effettivamente tale sequenza di numeri è

etichettata già da *words2.gr* con il valore di attributo “*GMA*” e quindi la *RULE* in *timex2.gr* deve solamente verificarne il valore dell'attributo per identificarla.

```
<RULE name="giorno-mese-anno" type="DISJF"
targ_sg="TIMEX[TYPE='TIME']">
   <REL match="W[C='GMA']"></REL>
</RULE>
```

Vengono riportati i nomi delle RULEs che compongono la RULE principale *giorno-stagione-combinate-e-non* seguite da un esempio di quello che marcano. L'ordine con il quale vengono elencate è lo stesso con il quale vengono applicate nella ricerca.

● giorno-mese-anno

- 3/12/75
- 3-12-1975

● giorno-data-mese-anno

- giovedì 30 ottobre 2003

● giorno-data-mese

- giovedì 30 ottobre

● namegiorno-data-mese-anno

- giorno 30 ottobre 2003

● data-mese-anno

- 30 ottobre 2003

● namegiorno-data-mese

- giorno 30 ottobre

● data-mese

- 30 ottobre

● mese-anno

- ottobre 2003

● namegiorno-data

- giorno 30

● giorno-data

- giovedi 30

● stagione-anno

- primavera del 2003
- estate 2003
- semestre 2003

7.2 RULE parte-del-giorno-comb-e-non

Questa semplice RULE identifica nomi dei giorni, parti del giorno ed espressioni composte come riportato di seguito:

- martedì mattina
- ieri pomeriggio
- domani sera
- martedì
- domani
- sera

L'insieme di RULEs che marcano gli esempi sopra sono:

```
<RULE name="parte-del-giorno"
targ_sg="TIMEX[TYPE='TIME']">
   <REL type="GROUP" match="DISJF">
      <REL type="REF"    match="nomi-giorni-lex"></REL>
      <REL type="REF    match="ieri-oggi-domani-lex"></REL>
   </REL>
   <REL type="REF"    match="parte-giorno-lex"></REL>
</RULE>

<RULE name="parte-giorno-comb-e-non" type="DISJF" >
   <REL type="REF"    match="parte-del-giorno"></REL>
   <REL type="REF"    match="nomi-giorni-lex"></REL>
   <REL type="REF"    match="parte-giorno-lex"></REL>
   <REL type="REF"    match="ieri-oggi-domani-lex"></REL>
</RULE>
```

Giunti a questo punto, come accennato in precedenza, si è scelto di proseguire oltre, ovvero cercare espressioni composte di maggior complessità e quindi è stata creata la seguente RULE

```
<RULE name="parte-del-giorno-data-mix"
targ_sg="TIMEX[TYPE='TIME']">
   <REL type="REF"    match="parte-giorno-comb-e-non"></REL>
   <REL                    match="W/#~^,$"
m_mod="QUEST"></REL>
   <REL type="REF"    match="giorno-data-mese-anno-stagione-
   mix"></REL>
</RULE>
```

che è in grado di marcare come unico elemento la frase: “*ieri mattina (,) 30 ottobre 2003*”.

7.3 RULE ore-della-giornata

Uno degli aspetti più difficili da gestire nel riconoscere le ore della giornata è individuare le ore ed i minuti rappresentati tramite caratteri numerici. Vi è una forte ambiguità, legata ai diversi caratteri di separazione utilizzati per separare le ore dai minuti, dovuta all'utilizzo alternativo di “*,*” “*.*” e “*:*”.
Ad esempio, “*13,15*” può essere indistintamente un numero cardinale o riferirsi ad un'ora del giorno, mentre “*12:15*” identifica inequivocabilmente

un'ora della giornata. “*13.15*” è una espressione errata per indicare cifre decimale, ma viene spesso utilizzata ed anche per identificare un orario. Per tale motivo “*13:15*” viene marcato mentre le altre espressioni solo se precedute dalla parola “*ore*” oppure si trovano all'interno di sequenze più estese.

Tale RULE non si limita a marcare gli esempi sopra, ma ogni espressione nella quale si faccia riferimento ad un orario preciso della giornata o di un giorno in particolare.

Vengono riportati i nomi delle RULEs che compongono la RULE principale *ore della giornata* seguite da un esempio di quello che marcano. L'ordine con il quale vengono elencate è uguale a quello con vengono applicate nella ricerca.

● ora-parte-giorno-data-mix

- (ore) 15.30 di martedì mattina 30 ottobre

● parte-giorno-data-mix-ora

- martedì mattina 30 ottobre alle 15,30

● parte-giorno-comb-e-non-ora

- domani alle (ore) 5 del pomeriggio
- martedì pomeriggio alle (ore) 15.30

● data-mix-ora

- 30 ottobre alle (ore) 5 del pomeriggio
- 30 ottobre alle (ore) 15,30

● ora-parte-giorno-comb-e-non

- (ore) 6.30 di martedì sera

● ore-ora-digitale

- ore 16.30

- ore 16,30
- ore 15:30

● ora-digitale-certa

- 15:30

7.4 RULE lex-con-avverbi-e-non

Fino ad ora sono state marcate espressioni complesse, ma prive di avverbi, aggettivi, articoli, preposizioni ed altri sostantivi. Inserendo tali parole si ottiene una grande quantità di locuzioni temporali che richiederebbe una attenta classificazione delle combinazioni possibili, prima di poterle marcare. Inoltre occorrerebbe scrivere una grande quantità di RULEs dedicate che solo un linguista è in grado di realizzare. Per tale motivo non è stata fatta una classificazione distinguendo gli avverbi dagli aggettivi. L'unica classificazione presente raggruppa l'insieme di avverbi, aggettivi e sostantivi che indichino l'inizio o la fine di una data temporale. Tale classificazione è riportata nel vocabolario esterno *timex2.lex*. A questo punto vengono chiamate una serie di RULEs che cercano espressioni temporali semplici ("*lunedì*", "*mese*", "*ieri*", "*mattina*", "*secolo*") precedute o seguite da avverbi, aggettivi e sostantivi definiti in *timex2.lex*

Per prima cosa viene definita una RULE (*collezione-sing-plu*) che raggruppa tutti i vocaboli che fanno riferimento ad espressioni temporali semplici, contenute nel lessico esterno ("*lunedì*", "*mese*", "*ieri*", "*mattina*", "*secolo*") oltre alle espressioni trovate dalle RULEs *giorno-stagione-combinate-e-non*, *parte del giorno comb-e-non*, *data-secolo-mix* e *decade*. Brevemente le ultime due marcano i seguenti esempi:

● data-secolo-mix

- VI secolo D.C.
- secolo VI
- nel cinquecento

● decade

- anni trenta
- anni '30
- '30

Inoltre viene definita una RULE che raggruppa tutti i vocaboli che fanno riferimento ad espressioni temporali semplici plurali contenute nel lessico esterno (*collezione-plu*). Queste ultime due RULEs costituiscono le sequenze base con le quali formare espressioni complesse marcate da *lex-con-avverbi-e-non*.
Vegono riportati i nomi delle RULEs che compongono la RULE principale descritta in questo paragrafo, seguite da alcuni esempi di quello che marcano. L'ordine con il quale vengono elencate è uguale a quello con cui vengono applicate nella ricerca.

● Prima-dopo-multi-date-plu

- scorsi 2-3 giorni
- prossime due settimane

● ultimo-primo-multi-date-plu

- ultimi due anni

● prima-dopo-date-sing-plu

- questo mese
- scorso secolo

● ultimo-primo-date-sing-plu

- primi anni '90
- seconda settimana

● multi-date-plu-prima-dopo

- 3 o 4 giorni dopo
- pochi mesi prima

● multi-comp-semplici (espressioni temporali semplici plurali)

- 2-3 giorni
- pochi giorni

● date-sing-plu-prima-dopo

- 30 ottobre scorso
- mese prossimo

● parte-del-giorno

- martedi mattina
- ieri pomeriggio

Infine la RULE *comp-semplici* che raggruppa tutti i vocaboli che fanno riferimento ad espressioni temporali semplici singolari contenute nel lessico esterno.

Terminata la costruzione di questa RULE vengono di seguito descritte le RULEs che marcano le sequenze più corpose. Una spiegazione dettagliata di tale RULEs richiederebbe molto tempo e quindi ci si limita a descriverne brevemente il funzionamento.

7.5 RULE lex-comp-of-al-the-lex-comp

Questa RULE permette di trovare espressioni nelle quali sono presenti internamente alcune preposizioni ed alcuni articoli determinativi.

```
<RULE name="lex-comp-of-al-the-lex-comp"
targ_sg="TIMEX[TYPE='TIME']">
   <REL type="REF"  match="lex-con-avverbi-e-non"></REL>
   <REL type="GROUP" match="DISJF">
      <REL match="&OF;"></REL>
      <REL match="&AL;"></REL>
      <REL match="&THE;"></REL>
   </REL>
   <REL type="REF"    match="lex-con-avverbi-e-non"></REL>
</RULE>
```

In tal modo possono essere marcate espressioni del tipo “*tarda serata di martedì scorso*” oppure “*prime ore di ieri*”
Come avviene per numbers2.gr, tale RULE può trovare delle espressioni prive di senso, se poi adoperate nel linguaggio comune, quindi si presuppone sempre che il testo da analizzare sia formalmente corretto. Collegandoci a quanto detto ad inizio capitolo, questa è una delle RULE che trova le sequenze più lunghe ed è pensando a questa che poi sono state definite le RULEs elementari.

● Prep-misc

Questa RULE permette di marcare gli articoli determinativi, indeterminativi, le preposizioni ed alcune locuzioni temporali (ad esempio: “*nell’arco*”), quando queste precedono una qualsiasi sequenza individuata dalle RULEs precedentemente descritte. Di seguito alcuni esempi delle sequenze marcate:

- nell’arco della giornata
- nel pomeriggio
- per tutto il mese di ottobre
- per le 16:30
- entro il 2004

7.6 RULE tra-fra-tutte-date-possibili

Nei vari articoli di giornale, presi come riferimento molto spesso compaiono espressioni del tipo “*tra il 2 ed il 3 ottobre*”, “*tra oggi e domani*” oppure “*dal 12 al 13 novembre*”. In comune hanno l'uso delle preposizioni ad inizio sequenza. Costruire un insieme di RULEs che tenesse conto di tutte le preposizioni non è stato preso in considerazione a causa della comlpessità del lavoro, quindi sono state realizzate una serie di RULEs per marcare solamente gli esempi sopra.
Il corpo principale di questa RULE è riportato nell'esempio sottostante. E' una struttura molto versatile che permette di marcare qualsiasi combinazione delle sequenze trovate fino ad ora semplicemente decidendo quali parametri passare per la ricerca.

```
<RULE  name="tra-fra-time" arg="$3 $4">
   <REL match="W/#~^([Tt]ra|[Ff]ra)$"></REL>
   <REL match="W/#~^(il?|le)$" m_mod="QUEST"></REL>
   <REL type="REF" match="$3"></REL>
   <REL match="W/#~^(ed?|o)$"></REL>
   <REL match="W/#~^(il|le)$" m_mod="QUEST"></REL>
   <REL type="REF" match="$4"></REL>
</RULE>
```

I parametri che vengono passati sono le RULEs precedentemente descritte attraverso questa semplice RULE:

```
<RULE  name="tra-fra-max" targ_sg="TIMEX[TYPE='TIME']">
   <REL match="tra-fra-time" type="REF">
      <ARG bind='$3'>lex-comp-of-al-the-lex-comp</ARG>
      <ARG bind='$4'>lex-comp-of-al-the-lex-comp</ARG>
   </REL>
</RULE>
```

L'esempio ad inizio capitolo è invece marcato richiamando la seguente RULE:

```
<RULE  name="da-a-ore-della-giornata"
targ_sg="TIMEX[TYPE='TIME']">
   <REL match="da-a-time" type="REF">
      <ARG bind='$5'>ore-della-giornata</ARG>
      <ARG bind='$6'>ore-della-giornata</ARG>
   </REL>
</RULE>
```

Ore-della-giornata raccoglie diverse RULEs come visto in precedenza quindi la RULE sopra può marcare tutte le combinazioni fra i vari esempi riportati nel paragrafo dove viene discussa.

- da-a-tutte-date-possibili

Il funzionamento di questa RULE è identico a quella precedente. Vengono passati gli stessi parametri nello stesso ordine con il quale vengono passati a *tra fra tutte date possibil*

```
<RULE  name="da-a-time" arg="$5 $6">
   <REL match="W/#~^([Dd]alle|[Dd]al|[Dd]a)$"></REL>
   <REL type="REF" match="$5"></REL>
   <REL match="&AL;"></REL>
   <REL type="REF" match="$6"></REL>
</RULE>
```

7.7 RULE tra-fra-range

La rule *tra-fra-tutte-le-date-possibili* riesce a marcare l'espressione “*tra il 2 ottobre ed il 3 ottobre*”, mentre non è in grado di rilevare “*tra il 2 ed il 3 ottobre*” oppure “*tra 2-3 ore*”. Occorre quindi definire una RULE che identifica degli intervalli numerici. La RULE *range* di *numbers2.gr* marca con valore dell'attributo “*range*” le sequenze “*2-3*”, “*2 o 3*”, “*2 al 3*” , mentre per identificare la sequenza “*il 2 ed il 3*” è stata creata una RULE dedicata:

```
<RULE  name="cd-ed-il-cd">
   <REL match="W/#~^(il?|le)$" m_mod="QUEST"></REL>
   <REL type="GROUP" match="DISJF">
      <REL match="&CARD;"></REL>
      <REL match="W/#~^[XVI]+"></REL>
   </REL>
   <REL match="W/#~^(ed?|od?)$"></REL>
   <REL match="W/#~^(il?|le)$" m_mod="QUEST"></REL>
   <REL type="GROUP" match="DISJF">
      <REL match="&CARD;"></REL>
      <REL match="W/#~^[XVI]+"></REL>
   </REL>
</RULE>
```

L'entità *&card;* si riferisce a qualsiasi elemento il cui valore dell'attributo è CD ed inoltre include anche i numeri romani. In questo modo è possibile definire la RULE *tra-fra-range* nel modo seguente:

```
<RULE name="tra-fra-range" targ_sg="TIMEX[TYPE='TIME']">
   <REL match="W/#~^([Tt]ra|[Ff]ra)$"></REL>
   <REL type="GROUP" match="DISJF">
      <REL type="REF"  match="cd-ed-il-cd"></REL>
      <REL type="GROUP" match="SEQ">
         <REL match="W/#~^(il?|le)$" m_mod="QUEST"></REL>
         <REL match="&RANGE;"></REL>
      </REL>
   </REL>
   <REL type="REF" match="lex-con-avverbi-e-non"
   m_mod="QUEST"></REL>
</RULE>
```

Seguono degli esempi delle espressioni marcate correttamente dalla RULE sopra:

- tra 20-30 minuti
- tra 20 o 30 minuti
- tra il 10 ed il 20 ottobre
- tra le 2 o 3 ore
- tra il X ed il XII secolo

7.9 RULEs di contorno

Infine vengono elencate una serie di RULEs che marcano alcune espressioni comuni non riconosciute dalle RULEs precedenti:

● tra-fra-decade-decade

- tra gli anni '20 e '30

● range-anno

- dal 2002 al 2004
- 2003/04

● fino-alla-fine

- fino alla fine del 2003
- fino alla fine del mese

7.10 Problemi ed osservazioni

L'insieme totale delle espressioni temporali esistenti nella lingua italiana è molto vasto e, sicuramente, le RULEs sopra non identificano tutte le sequenze esistenti. Tale vastità è dovuta al fatto che non esistono rigide regole grammaticali per definire sequenze corpose, come ad esempio: "*15:30 di martedi mattina 30 ottobre*" e "*martedi mattina 30 ottobre alle 15:30*" hanno lo stesso significato. Lo stesso si può dire per "*giorni scorsi*" e "*scorsi giorni*" dove è cambiato solamente l'ordine delle parole. Un altro esempio concreto è la frase "*nella notte tra il 2 ed il 3 ottobre*". Marcando solamente "*tra il 2 ed il 3 ottobre*" ci si riferisce ad un intervallo di tempo di quarantotto ore, mentre la frase completa indica un preciso istante temporale più ristretto e quindi ne modifica il significato. *Timex2.gr* etichetta separatamente "*tra il 2 ed il 3 ottobre*" e "*notte*" ma non "*nella notte*" e "*nella notte tra il 2 ed il 3 ottobre*". Seguendo la logica utilizzata in *timex2.gr* si dovrebbe marcare l'ultima espressione, ovvero quella completa. Ragionando in questo modo, si

dovrebbero aggiungere altre RULEs, ma si comincerebbe a perdere il significato di questa tesi. Aggiungere altre RULEs comporterebbe creare una casistica più esaustiva, ma in questo modo entriamo nelle competenze di un linguista. Infatti l'approccio ideale sarebbe quello di marcare solamente sequenze elementari, formate al massimo da due o tre parole, classificarle e quindi creare un altro file di RULEs che elaborino tale sequenze precedentemente marcate. Procedendo con questa struttura a livelli per raggruppare le sequenze più estese si possono creare etichette dettagliate e si possono riconoscere sequenze estese dandone poi un preciso significato.

Capitolo 8
Risultati ed analisi di numbers2.gr

Nel capitolo precedente si è concluso che non è possibile definire in modo esaustivo tutte le possibili espressioni temporali in uso nella lingua italiana. Per tale motivo, non ha senso etichettare un testo e verificare gli errori dovuti ad espressioni temporali non esattamente marcate. Ad esempio la frase "*nella notte tra oggi e domani*" non viene etichettata interamente, ma separatamente in due parti: "*nella notte*" e "*tra oggi e domani*". Questo non può essere considerato un errore in quanto non esiste la RULE che cerca questa sequenza completa. Risulta quindi impossibile creare una statistica degli errori per come è definito *timex2.gr*
Risulta invece possibile creare una statistica sulla RULE numbers2.gr a causa della ambiguità legata ai numeri ordinali (*primo, secondo*…). Inoltre, altra fonte di errori può essere la RULE *textnum* come descritto nel *paragrafo 5.3.1*.

8.1 Corpo del test

Il test è stato eseguito su una raccolta di articoli di giornali prelevati dai maggiori quotidiani italiani. Per semplificare la raccolta del materiale, gli articoli sono stati copiati dai corrispondenti siti internet. In totale sono stati etichettati 23 articoli prevalentemente di economia e di sport per un totale circa di 3400 parole. *Numbers2.gr* ha etichettato 280 sequenze e si sono rivelate errate 14 etichettature, quindi si è ottenuta una precisione del 95%.
In particolare 8 errori sono dovuti alla sequenza "*per cento*" dove "*cento*" individua parte di una quantità frazionaria e non una semplice quantità

numerica. I rimanenti 6 errori riguardano le parole “*prima*”, “*secondo*” e “*tredicesima*” dove nel testo assumevano rispettivamente il significato di avverbio, preposizione e paga mensile straordinaria. Considerando solo questi errori, in quanto di maggior rilevanza, la precisione sale al 97,8%.
Di seguito viene mostrato un articolo utilizzato e successivamente il risultato della elaborazione in modo di poter mettere in evidenza anche gli errori del programma.

8.2 Esempio di articolo etichettato

Viene riportato un articolo di economia, tratto dalla versione on-line del quotidiano “*La Repubblica*” del 26 novembre 2003.

A incrementare la tredicesima dei pensionati contribuiscono anche altre voci per esempio, un aumento generale dell'importo delle pensioni, che si riflette anche sulla tredicesima mensilità. Con la pensione di dicembre, l'Inps pagherà anche 155 euro in più, come stabilito dalla Finanziaria 2001, ai titolari di pensioni il cui importo non superi il trattamento minimo. Per avere diritto, i redditi personali non devono superare l'importo annuo di 7.841,34 euro, mentre quelli cumulati con il coniuge non devono superare 15.682,68 euro annui. L'insieme delle voci comporterà nel 2003 una maggiore spesa di un miliardo e 235 milioni di euro per il pagamento delle tredicesime ai pensionati (il 7,7% in più rispetto al 2002).
I tecnici dell'Inps, quindi, spiegano come è cambiata dall'inizio dell'anno la platea dei pensionati che beneficiano di agevolazioni fiscali. Prima della riforma Tremonti i pensionati che non pagavano tasse (quelli che rientravano nella cosiddetta 'no tax area') erano 5.300.000; dopo la riforma sono 730.000 in più, che fino alla fine del 2002 pagavano in media 90 euro l'anno di tasse. Complessivamente - spiega ancora l'Inps - per effetto dell'applicazione del primo modulo della riforma fiscale, entrato in vigore nel gennaio 2003 ci sono 8.700.000 pensionati (non solo dell'Inps) che da gennaio pagano in media 193 euro in meno l'anno; 3.700 invece, pagano sette euro l'anno in più, con la possibilità di recuperarli - grazie a un dispositivo della scorsa Finanziaria - in sede di dichiarazione di redditi.

A seguito della elaborazione tramite *numbers2.gr* si ottiene:

```
<?xml version='1.0'?>
<!DOCTYPE DOCS SYSTEM "/home/faio/TTT_v1.0/RES/general.dtd,xml" >
<DOCS>
<TEXT>
<P><W C='W'>A</W> <W C='W'>incrementare</W> <W C='W'>la</W>
<W C='ORD'>tredicesima</W> <W C='W'>dei</W> <W C='W'>pensionati</W>
<W C='W'>contribuiscono</W> <W C='W'>anche</W> <W C='W'>altre</W>
<W C='W'>voci</W><W C='CM'>:</W><W C='W'>per</W> <W C='W'>esempio</W>
<W C='CM'>,</W> <W C='W'>un</W> <W C='W'>aumento</W> <W C='W'>generale</W>
```

<W C='W'>dell'</W><W C='W'>importo</W> <W C='W'>delle</W>
<W C='W'>pensioni</W><W C='CM'>,</W><W C='W'>che</W> <W C='W'>si</W>
<W C='W'>riflette</W> <W C='W'>anche</W> <W C='W'>sulla</W>
<W C='ORD'>tredicesima</W> <W C='W'>mensilità</W><W C='.'>.</W>
<W C='W'>Con</W> <W C='W'>la</W> <W C='W'>pensione</W> <W C='W'>di</W>
<W C='W'>dicembre</W><W C='CM'>,</W> <W C='W'>l'</W><W C='W'>Inps</W>
<W C='W'>pagherà</W> <W C='W'>anche</W> **<W C='CD'>155</W>**
<W C='W'>euro</W> <W C='W'>in</W> <W C='W'>più</W><W C='CM'>,</W>
<W C='W'>come</W> <W C='W'>stabilito</W> <W C='W'>dalla</W>
<W C='W'>Finanziaria</W> **<W C='CD'>2001</W>**<W C='CM'>,</W> <W C='W'>ai</W>
<W C='W'>titolari</W> <W C='W'>di</W> <W C='W'>pensioni</W> <W C='W'>il</W>
<W C='W'>cui</W> <W C='W'>importo</W> <W C='W'>non</W> <W C='W'>superi</W>
<W C='W'>il</W> <W C='W'>trattamento</W> <W C='W'>minimo</W><W C='.'>.</W>
<W C='W'>Per</W> <W C='W'>avere</W> <W C='W'>diritto</W><W C='CM'>,</W>
<W C='W'>i</W> <W C='W'>redditi</W> <W C='W'>personali</W> <W C='W'>non</W>
<W C='W'>devono</W> <W C='W'>superare</W> <W C='W'>l'</W><WC='W'>importo</W>
<W C='W'>annuo</W> <W C='W'>di</W> **<W C='CD'>7.841,34</W>**
<W C='W'>euro</W><W C='CM'>,</W> <W C='W'>mentre</W> <W C='W'>quelli</W>
<W C='W'>cumulati</W> <W C='W'>con</W> <W C='W'>il</W> <W C='W'>coniuge</W>
<W C='W'>non</W> <W C='W'>devono</W> <W C='W'>superare</W>
<W C='CD'>15.682,68</W> <W C='W'>euro</W> <W C='W'>annui</W><W C='.'>.</W>
<W C='W'>L'</W><W C='W'>insieme</W> <W C='W'>delle</W> <W C='W'>voci</W>
<W C='W'>comporterà</W> <W C='W'>nel</W> **<W C='CD'>2003</W>**
<W C='W'>una</W> <W C='W'>maggiore</W> <W C='W'>spesa</W> <W C='W'>di</W>
<PHR C='CD'><W C='W'>un</W> <W C='W'>miliardo</W> <W C='W'>e</W>
<W C='CD'>235</W> <W C='W'>milioni</W></PHR> <W C='W'>di</W>
<W C='W'>euro</W> <W C='W'>per</W> <W C='W'>il</W> <W C='W'>pagamento</W>
<W C='W'>delle</W> **<W C='ORD'>tredicesime</W>** <W C='W'>ai</W>
<W C='W'>pensionati</W> <W C='BR'>(</W><W C='W'>il</W> **<W C='CD'>7,7</W>**
<W C='PTC'>%</W> <WC='W'>in</W> <W C='W'>più</W> <W C='W'>rispetto</W>
<W C='W'>al</W> **<W C='CD'>2002</W>**<W C='BR'>)</W><W C='.'>.</W>
<W C='W'>I</W> <W C='W'>tecnici</W> <W C='W'>dell'</W><W C='W'>Inps</W>
<W C='CM'>,</W> <W C='W'>quindi</W><W C='CM'>,</W> <W C='W'>spiegano</W>
<W C='W'>come</W> <W C='W'>è</W> <W C='W'>cambiata</W> <W C='W'>dall'</W>
<W C='W'>inizio</W> <W C='W'>dell'</W><W C='W'>anno</W> <W C='W'>la</W>
<W C='W'>platea</W> <W C='W'>dei</W> <W C='W'>pensionati</W> <W C='W'>che</W>
<W C='W'>beneficiano</W> <W C='W'>di</W> <W C='W'>agevolazioni</W>
<W C='W'>fiscali</W><W C='.'>.</W> **<W C='NONDEF'>Prima</W>**
<W C='W'>della</W> <W C='W'>riforma</W> <W C='W'>Tremonti</W> <W C='W'>i</W>
<W C='W'>pensionati</W> <W C='W'>che</W> <W C='W'>non</W>
<W C='W'>pagavano</W> <W C='W'>tasse</W> <W C='BR'>(</W><W C='W'>quelli</W>
<W C='W'>che</W> <W C='W'>rientravano</W><W C='W'>nella</W>
<W C='W'>cosiddetta</W> <W C='QUOTE'>'</W><W C='W'>no</W> <W C='W'>tax</W>
<W C='W'>area</W><W C='QUOTE'>'</W><W C='BR'>)</W> <W C='W'>erano</W>
<W C='CD'>5.300.000</W><W C='CM'>;</W> <W C='W'>dopo</W> <W C='W'>la</W>
<W C='W'>riforma</W> <W C='W'>sono</W> **<W C='CD'>730.000</W>**
<W C='W'>in</W> <W C='W'>più</W><W C='CM'>,</W> <W C='W'>che</W>
<W C='W'>fino</W> <W C='W'>alla</W> <W C='W'>fine</W> <W C='W'>del</W>
<W C='CD'>2002</W> <W C='W'>pagavano</W> <W C='W'>in</W>
<W C='W'>media</W> **<W C='CD'>90</W>** <W C='W'>euro</W> <W C='W'>l'</W>
<W C='W'>anno</W> <W C='W'>di</W> <W C='W'>tasse</W><W C='.'>.</W>
<W C='W'>Complessivament</W> <W C='DASH'>-</W> <W C='W'>spiega</W>
<W C='W'>ancora</W> <W C='W'>l'</W> <W C='W'>Inps</W> <W C='DASH'>-</W>
<W C='W'>per</W> <W C='W'>effetto</W> <W C='W'>dell'</W>
<W C='W'>applicazione</W> <W C='W'>del</W> **<W C='ORD'>primo</W>**
<W C='W'>modulo</W> <W C='W'>della</W> <W C='W'>riforma</W>
<W C='W'>fiscale</W><W C='CM'>,</W> <W C='W'>entrato</W> <W C='W'>in</W>
<W C='W'>vigore</W> <W C='W'>nel</W> <W C='W'>gennaio</W>
<W C='CD'>2003</W> <W C='W'>ci</W> <W C='W'>sono</W>

<W C='CD'>8.700.000</W> <W C='W'>pensionati</W> <W C='BR'>(</W>
<W C='W'>non</W> <W C='W'>solo</W> <W C='W'>dell'</W><W C='W'>Inps</W>
<W C='BR'>)</W> <W C='W'>che</W> <W C='W'>da</W> <W C='W'>gennaio</W>
<W C='W'>pagano</W> <W C='W'>in</W> <W C='W'>media</W> <W C='CD'>193</W>
<W C='W'>euro</W> <W C='W'>in</W> <W C='W'>meno</W> <W C='W'>l'</W>
<W C='W'>anno</W><W C='CM'>;</W> <W C='CD'>3.700</W> <W C='W'>invece</W>
<W C='CM'>,</W> <W C='W'>pagano</W> <W C='CD'>sette</W> <W C='W'>euro</W>
<W C='W'>l'</W><W C='W'>anno</W> <W C='W'>in</W> <W C='W'>più</W>
<W C='CM'>,</W> <W C='W'>con</W> <W C='W'>la</W> <W C='W'>possibilità</W>
<W C='W'>di</W> <W C='W'>recuperarli</W> <W C='DASH'>-</W> <W C='W'>grazie</W>
<W C='W'>a</W> <W C='W'>un</W> <W C='W'>dispositivo</W> <W C='W'>della</W>
<W C='W'>scorsa</W> <W C='W'>Finanziaria</W> <W C='DASH'>-</W>
<W C='W'>in</W> <W C='W'>sede</W> <W C='W'>di</W> <W C='W'>dichiarazione</W>
<W C='W'>di</W> <W C='W'>redditi</W><W C='.'>.</W> </P>
</TEXT>
</DOCS>

In questo esempio si può notare come la parola “*tredicesima*” venga marcata due volte con valore dell’attributo ORD. Solamente nel primo caso l’etichetta è esatta mentre, quella successiva non indica un numero ordinale.
Successivamente il risultato della elaborazione tramite numex2.gr risulta:

<?xml version='1.0'?>
<!DOCTYPE DOCS SYSTEM "/home/faio/TTT_v1.0/RES/general.dtd,xml" >
<DOCS>
<TEXT>
<P><W C='W'>A</W> <W C='W'>incrementare</W> <W C='W'>la</W>
<W C='ORD'>tredicesima</W> <W C='W'>dei</W> <W C='W'>pensionati</W>
<W C='W'>contribuiscono</W> <W C='W'>anche</W> <W C='W'>altre</W>
<W C='W'>voci</W><W C='CM'>:</W> <W C='W'>per</W> <W C='W'>esempio</W>
<W C='CM'>,</W> <W C='W'>un</W> <W C='W'>aumento</W> <W C='W'>generale</W>
<W C='W'>dell'</W><W C='W'>importo</W> <W C='W'>delle</W>
<W C='W'>pensioni</W><W C='CM'>,</W><W C='W'>che</W> <W C='W'>si</W>
<W C='W'>riflette</W><W C='W'>anche</W> <W C='W'>sulla</W>
<W C='ORD'>tredicesima</W> <W C='W'>mensilità</W><W C='.'>.</W>
<W C='W'>Con</W> <W C='W'>la</W> <W C='W'>pensione</W> <W C='W'>di</W>
<W C='W'>dicembre</W><W C='CM'>,</W> <W C='W'>l'</W><W C='W'>Inps</W>
<W C='W'>pagherà</W> <W C='W'>anche</W> **<NUMEX TYPE='MONEY'>**
<W C='CD'>155</W> <W C='W'>euro</W> <W C='W'>in</W>
<W C='W'>più</W></NUMEX><W C='CM'>,</W> <W C='W'>come</W>
<W C='W'>stabilito</W> <W C='W'>dalla</W> <W C='W'>Finanziaria</W>
<W C='CD'>2001</W><W C='CM'>,</W> <W C='W'>ai</W><W C='W'>titolari</W>
<W C='W'>di</W> <W C='W'>pensioni</W> <W C='W'>il</W> <W C='W'>cui</W>
<W C='W'>importo</W> <W C='W'>non</W> <W C='W'>superi</W> <W C='W'>il</W>
<W C='W'>trattamento</W> <W C='W'>minimo</W><W C='.'>.</W> <W C='W'>Per</W>
<W C='W'>avere</W> <W C='W'>diritto</W><W C='CM'>,</W> <W C='W'>i</W>
<W C='W'>redditi</W> <W C='W'>personali</W> <W C='W'>non</W>
<W C='W'>devono</W> <W C='W'>superare</W> <W C='W'>l'</W><WC='W'>importo</W>
<W C='W'>annuo</W> <W C='W'>di</W>**<NUMEX TYPE='MONEY'>**
<W C='CD'>7.841,34</W> <W C='W'>euro</W></NUMEX><W C='CM'>,</W>
<W C='W'>mentre</W> <W C='W'>quelli</W> <W C='W'>cumulati</W>

<W C='W'>con</W> <W C='W'>il</W> <W C='W'>coniuge</W> <W C='W'>non</W>
<W C='W'>devono</W> <W C='W'>superare</W>**<NUMEX TYPE='MONEY'>**
<W C='CD'>15.682,68</W> <W C='W'>euro</W></NUMEX>
<W C='W'>annui</W><W C='.'>.</W> <W C='W'>L'</W><W C='W'>insieme</W>
<W C='W'>delle</W> <W C='W'>voci</W> <W C='W'>comporterà</W> <W C='W'>nel</W>
<W C='CD'>2003</W> <W C='W'>una</W> <W C='W'>maggiore</W>
<W C='W'>spesa</W> <W C='W'>di</W> **<NUMEX TYPE='MONEY'>**
<PHR C='CD'><W C='W'>un</W> <W C='W'>miliardo</W> <W C='W'>e</W>
<W C='CD'>235</W> <W C='W'>milioni</W></PHR> <W C='W'>di</W>
<W C='W'>euro</W></NUMEX> <W C='W'>per</W> <W C='W'>il</W>
<W C='W'>pagamento</W> <W C='W'>delle</W> <W C='ORD'>tredicesime</W>
<W C='W'>ai</W> <W C='W'>pensionati</W> <W C='BR'>(</W><W C='W'>il</W>
<NUMEX TYPE='PERCENT'><W C='CD'>7,7</W><W C='PTC'>%</W>
<W C='W'>in</W> <W C='W'>più</W></NUMEX> <W C='W'>rispetto</W>
<W C='W'>al</W> <W C='CD'>2002</W><W C='BR'>)</W><W C='.'>.</W>
<W C='W'>I</W> <W C='W'>tecnici</W> <W C='W'>dell'</W><W C='W'>Inps</W>
<W C='CM'>,</W> <W C='W'>quindi</W><W C='CM'>,</W> <W C='W'>spiegano</W>
<W C='W'>come</W> <W C='W'>è</W> <W C='W'>cambiata</W> <W C='W'>dall'</W>
<W C='W'>inizio</W> <W C='W'>dell'</W><W C='W'>anno</W> <W C='W'>la</W>
<W C='W'>platea</W> <W C='W'>dei</W> <W C='W'>pensionati</W> <W C='W'>che</W>
<W C='W'>beneficiano</W> <W C='W'>di</W> <W C='W'>agevolazioni</W>
<W C='W'>fiscali</W><W C='.'>.</W> <W C='NONDEF'>Prima</W> <W C='W'>della</W>
<W C='W'>riforma</W> <W C='W'>Tremonti</W> <W C='W'>i</W>
<W C='W'>pensionati</W> <W C='W'>che</W> <W C='W'>non</W>
<W C='W'>pagavano</W> <W C='W'>tasse</W> <W C='BR'>(</W><W C='W'>quelli</W>
<W C='W'>che</W> <W C='W'>rientravano</W> <W C='W'>nella</W>
<W C='W'>cosiddetta</W> <W C='QUOTE'>'</W><W C='W'>no</W> <W C='W'>tax</W>
<W C='W'>area</W><W C='QUOTE'>'</W><W C='BR'>)</W> <W C='W'>erano</W>
<W C='CD'>5.300.000</W><W C='CM'>;</W> <W C='W'>dopo</W> <W C='W'>la</W>
<W C='W'>riforma</W> <W C='W'>sono</W> <W C='CD'>730.000</W> <W C='W'>in</W>
<W C='W'>più</W><W C='CM'>,</W> <W C='W'>che</W> <W C='W'>fino</W>
<W C='W'>alla</W> <W C='W'>fine</W> <W C='W'>del</W> <W C='CD'>2002</W>
<W C='W'>pagavano</W> <W C='W'>in</W> <W C='W'>media</W>
<NUMEX TYPE='MONEY'><W C='CD'>90</W> <W C='W'>euro</W></NUMEX>
<W C='W'>l'</W><W C='W'>anno</W> <W C='W'>di</W> <W C='W'>tasse</W>
<W C='.'>.</W> <W C='W'>Complessivamente</W> <W C='DASH'>-</W>
<W C='W'>spiega</W> <W C='W'>ancora</W> <W C='W'>l'</W><W C='W'>Inps</W>
<W C='DASH'>-</W> <W C='W'>per</W> <W C='W'>effetto</W> <W C='W'>dell'</W>
<W C='W'>applicazione</W> <W C='W'>del</W> <W C='ORD'>primo</W>
<W C='W'>modulo</W> <W C='W'>della</W> <W C='W'>riforma</W>
<W C='W'>fiscale</W><W C='CM'>,</W> <W C='W'>entrato</W> <W C='W'>in</W>
<W C='W'>vigore</W> <W C='W'>nel</W> <W C='W'>gennaio</W> <W C='CD'>2003</W>
<W C='W'>ci</W> <W C='W'>sono</W> <W C='CD'>8.700.000</W>
<W C='W'>pensionati</W> <W C='BR'>(</W><W C='W'>non</W> <W C='W'>solo</W>
<W C='W'>dell'</W><W C='W'>Inps</W><W C='BR'>)</W> <W C='W'>che</W>
<W C='W'>da</W> <W C='W'>gennaio</W> <W C='W'>pagano</W> <W C='W'>in</W>
<W C='W'>media</W> **<NUMEX TYPE='MONEY'><W C='CD'>193</W>**
<W C='W'>euro</W> <W C='W'>in</W> <W C='W'>meno</W></NUMEX>
<W C='W'>l'</W><W C='W'>anno</W><W C='CM'>;</W> <W C='CD'>3.700</W>
<W C='W'>invece</W><W C='CM'>,</W> <W C='W'>pagano</W>
<NUMEX TYPE='MONEY'><W C='CD'>sette</W>
<W C='W'>euro</W></NUMEX> <W C='W'>l'</W><W C='W'>anno</W>
<W C='W'>in</W> <W C='W'>più</W> <W C='CM'>,</W> <W C='W'>con</W>
<W C='W'>la</W> <W C='W'>possibilità</W> <W C='W'>di</W>
<W C='W'>recuperarli</W> <W C='DASH'>-</W> <W C='W'>grazie</W>
<W C='W'>a</W> <W C='W'>un</W> <W C='W'>dispositivo</W> <W C='W'>della</W>
<W C='W'>scorsa</W> <W C='W'>Finanziaria</W> <W C='DASH'>-</W>
<W C='W'>in</W> <W C='W'>sede</W> <W C='W'>di</W> <W C='W'>dichiarazione</W>
<W C='W'>di</W> <W C='W'>redditi</W><W C='.'>.</W> </P>

</TEXT>
</DOCS>

Infine come ultima operazione il testo viene elaborazione tramite *timex2.gr*. Risulta evidente dall'esempio precedente e successivamente da quello di seguito, che questi ultimi file *.gr* si basano entrambi su *numbers2.gr*. Ovvero è necessario eseguire *numbers2.gr* prima di *numex2.gr* o di *timex2.gr* ma questi ultimi due sono indipendenti tra di loro.

<?xml version='1.0'?>
<!DOCTYPE DOCS SYSTEM "/home/faio/TTT_v1.0/RES/general.dtd,xml" >
<DOCS>
<TEXT>
<P><W C='W'>A</W> <W C='W'>incrementare</W> <W C='W'>la</W>
<W C='ORD'>tredicesima</W> <W C='W'>dei</W> <W C='W'>pensionati</W>
<W C='W'>contribuiscono</W> <W C='W'>anche</W> <W C='W'>altre</W>
<W C='W'>voci</W><W C='CM'>:</W> <W C='W'>per</W> <W C='W'>esempio</W>
<W C='CM'>,</W> <W C='W'>un</W> <W C='W'>aumento</W> <W C='W'>generale</W>
<W C='W'>dell'</W><W C='W'>importo</W> <W C='W'>delle</W>
<W C='W'>pensioni</W><W C='CM'>,</W><W C='W'>che</W> <W C='W'>si</W>
<W C='W'>riflette</W> <W C='W'>anche</W> <W C='W'>sulla</W>
<W C='ORD'>tredicesima</W> <W C='W'>mensilità</W><W C='.'>.</W>
<W C='W'>Con</W> <W C='W'>la</W> <W C='W'>pensione</W>
<TIMEX TYPE='DATE'><W C='W'>di</W> <W C='W'>dicembre</W></TIMEX>
<W C='CM'>,</W> <W C='W'>l'</W><W C='W'>Inps</W> <W C='W'>pagherà</W>
<W C='W'>anche</W> <NUMEX TYPE='MONEY'><W C='CD'>155</W>
<W C='W'>euro</W> <W C='W'>in</W> <W C='W'>più</W></NUMEX><W C='CM'>,</W>
<W C='W'>come</W> <W C='W'>stabilito</W> <W C='W'>dalla</W>
<W C='W'>Finanziaria</W> <W C='CD'>2001</W><W C='CM'>,</W> <W C='W'>ai</W>
<W C='W'>titolari</W> <W C='W'>di</W> <W C='W'>pensioni</W> <W C='W'>il</W>
<W C='W'>cui</W> <W C='W'>importo</W> <W C='W'>non</W> <W C='W'>superi</W>
<W C='W'>il</W> <W C='W'>trattamento</W> <W C='W'>minimo</W><W C='.'>.</W>
<W C='W'>Per</W> <W C='W'>avere</W> <W C='W'>diritto</W><W C='CM'>,</W>
<W C='W'>i</W> <W C='W'>redditi</W> <W C='W'>personali</W> <W C='W'>non</W>
<W C='W'>devono</W> <W C='W'>superare</W> <W C='W'>l'</W>
<W C='W'>importo</W> <W C='W'>annuo</W> <W C='W'>di</W>
<NUMEX TYPE='MONEY'><W C='CD'>7.841,34</W> <W C='W'>euro</W></NUMEX>
<W C='CM'>,</W> <W C='W'>mentre</W> <W C='W'>quelli</W> <W C='W'>cumulati</W>
<W C='W'>con</W> <W C='W'>il</W> <W C='W'>coniuge</W> <W C='W'>non</W>
<W C='W'>devono</W> <W C='W'>superare</W> <NUMEX TYPE='MONEY'>
<W C='CD'>15.682,68</W> <W C='W'>euro</W></NUMEX> <W C='W'>annui</W>
<W C='.'>.</W> <W C='W'>L'</W><W C='W'>insieme</W> <W C='W'>delle</W>
<W C='W'>voci</W> <W C='W'>comporterà</W> **<TIMEX TYPE='DATE'>**
<W C='W'>nel</W> <W C='CD'>2003</W></TIMEX> <W C='W'>una</W>
<W C='W'>maggiore</W><W C='W'>spesa</W> <W C='W'>di</W>
<NUMEX TYPE='MONEY'><PHR C='CD'><W C='W'>un</W> <W C='W'>miliardo</W>
<W C='W'>e</W> <W C='CD'>235</W> <W C='W'>milioni</W></PHR> <W C='W'>di</W>
<W C='W'>euro</W></NUMEX> <W C='W'>per</W> <W C='W'>il</W>
<W C='W'>pagamento</W> <W C='W'>delle</W> <W C='ORD'>tredicesime</W>
<W C='W'>ai</W> <W C='W'>pensionati</W> <W C='BR'>(</W><W C='W'>il</W>
<NUMEX TYPE='PERCENT'><W C='CD'>7,7</W><W C='PTC'>%</W> <W C='W'>in</W>

<W C='W'>più</W></NUMEX> <W C='W'>rispetto</W> **<TIMEX TYPE='DATE'>**
<W C='W'>al</W> <W C='CD'>2002</W></TIMEX><W C='BR'>)</W><W C='.'>.</W>
<W C='W'>I</W> <W C='W'>tecnici</W> <W C='W'>dell'</W><W C='W'>Inps</W>
<W C='CM'>,</W> <W C='W'>quindi</W><W C='CM'>,</W> <W C='W'>spiegano</W>
<W C='W'>come</W> <W C='W'>è</W> <W C='W'>cambiata</W>
<TIMEX TYPE='DATE'><W C='W'>dall'</W><W C='W'>inizio</W>
<W C='W'>dell'</W><W C='W'>anno</W></TIMEX> <W C='W'>la</W>
<W C='W'>platea</W> <W C='W'>dei</W> <W C='W'>pensionati</W> <W C='W'>che</W>
<W C='W'>beneficiano</W> <W C='W'>di</W> <W C='W'>agevolazioni</W>
<W C='W'>fiscali</W><W C='.'>.</W> <W C='NONDEF'>Prima</W> <W C='W'>della</W>
<W C='W'>riforma</W> <W C='W'>Tremonti</W> <W C='W'>i</W>
<W C='W'>pensionati</W> <W C='W'>che</W> <W C='W'>non</W>
<W C='W'>pagavano</W> <W C='W'>tasse</W> <W C='BR'>(</W><W C='W'>quelli</W>
<W C='W'>che</W> <W C='W'>rientravano</W> <W C='W'>nella</W>
<W C='W'>cosiddetta</W> <W C='QUOTE'>'</W><W C='W'>no</W> <W C='W'>tax</W>
<W C='W'>area</W><W C='QUOTE'>'</W><W C='BR'>)</W> <W C='W'>erano</W>
<W C='CD'>5.300.000</W><W C='CM'>;</W> <W C='W'>dopo</W> <W C='W'>la</W>
<W C='W'>riforma</W> <W C='W'>sono</W> <W C='CD'>730.000</W>
<W C='W'>in</W> <W C='W'>più</W><W C='CM'>,</W> <W C='W'>che</W>
<TIMEX TYPE='DATE'><W C='W'>fino</W> <W C='W'>alla</W>
<W C='W'>fine</W> <W C='W'>del</W> <W C='CD'>2002</W></TIMEX>
<W C='W'>pagavano</W> <W C='W'>in</W> <W C='W'>media</W>
<NUMEX TYPE='MONEY'><W C='CD'>90</W> <W C='W'>euro</W></NUMEX>
<TIMEX TYPE='DATE'><WC='W'>l'</W><W C='W'>anno</W></TIMEX>
<W C='W'>di</W> <W C='W'>tasse</W><W C='.'>.</W> <W C='W'>Complessivament</W>
<W C='DASH'>-</W> <W C='W'>spiega</W> <W C='W'>ancora</W> <W C='W'>l'</W>
<W C='W'>Inps</W> <W C='DASH'>-</W> <W C='W'>per</W> <W C='W'>effetto</W>
<W C='W'>dell'</W><W C='W'>applicazione</W> <W C='W'>del</W>
<W C='ORD'>primo</W> <W C='W'>modulo</W> <W C='W'>della</W>
<W C='W'>riforma</W> <W C='W'>fiscale</W><W C='CM'>,</W> <W C='W'>entrato</W>
<W C='W'>in</W> <W C='W'>vigore</W> **<TIMEX TYPE='DATE'><W C='W'>nel</W>**
<W C='W'>gennaio</W> <W C='CD'>2003</W></TIMEX> <W C='W'>ci</W>
<W C='W'>sono</W> <W C='CD'>8.700.000</W> <W C='W'>pensionati</W>
<W C='BR'>(</W><W C='W'>non</W> <W C='W'>solo</W> <W C='W'>dell'</W>
<W C='W'>Inps</W><W C='BR'>)</W> <W C='W'>che</W> **<TIMEX TYPE='DATE'>**
<W C='W'>da</W> <W C='W'>gennaio</W></TIMEX> <W C='W'>pagano</W>
<W C='W'>in</W> <W C='W'>media</W> <NUMEX TYPE='MONEY'><W C='CD'>193</W>
<W C='W'>euro</W> <W C='W'>in</W> <W C='W'>meno</W></NUMEX>
<TIMEX TYPE='DATE'><W C='W'>l'</W><W C='W'>anno</W></TIMEX>
<W C='CM'>;</W> <W C='CD'>3.700</W> <W C='W'>invece</W><W C='CM'>,</W>
<W C='W'>pagano</W> <NUMEX TYPE='MONEY'><W C='CD'>sette</W>
<W C='W'>euro</W></NUMEX> **<TIMEX TYPE='DATE'><W C='W'>l'</W>**
<W C='W'>anno</W></TIMEX> <W C='W'>in</W> <W C='W'>più</W>
<W C='CM'>,</W> <W C='W'>con</W> <W C='W'>la</W> <W C='W'>possibilità</W>
<W C='W'>di</W> <W C='W'>recuperarli</W> <W C='DASH'>-</W> <W C='W'>grazie</W>
<W C='W'>a</W> <W C='W'>un</W> <W C='W'>dispositivo</W> <W C='W'>della</W>
<W C='W'>scorsa</W> <W C='W'>Finanziaria</W> <W C='DASH'>-</W>
<W C='W'>in</W> <W C='W'>sede</W> <W C='W'>di</W> <W C='W'>dichiarazione</W>
<W C='W'>di</W> <W C='W'>redditi</W><W C='.'>.</W> </P>
</TEXT>
</DOCS>

Appendice A
Codice XML delle RULEs

A.1 File paras.gr

```
<!-- Autore:        Federico Faccioni →
<!-- data:          5-12-2003 →

<?xml version='1.0'?>
<!DOCTYPE RULES SYSTEM "../../RES/RuleSpec.dtd" [

<!ENTITY WS             "[ ]" >
<!ENTITY PUNCT          "[#\*\/\{\}\(\)\.\?\!,;:'`$\-]" >
<!ENTITY QUOTE          ['"]' >
<!ENTITY SYMBOL         "(&PUNCT;|&QUOTE;)" >

]>



<RULES  name="paras"  apply="all" >



<!-- These rules assume that SCRIPTS/plain2xml.perl has applied and
    that the plain text file is now wrapped as a TEXT element in an
    xml file. This grammar finds titles and paragraphs.  -->

<!-- Double newline signifies a paragraph break: put a P end tag
    before it and a P start tag after it. -->
<RULE name="pbreak" targ="&lt;/P>&S-VAL;&lt;P>">
    <REL match="\n(\n|&WS;)*\n"></REL>
</RULE>

<!-- A titleline is one or two lines of all caps (and punct). If
    there are more than two lines then it probably isn't a title. -->
<RULE  name="titleline">
    <REL match="[A-Z]([A-Z0-9]|&AMP;|&SYMBOL;|%|&WS;)*"></REL>
    <REL match="\n[A-Z]([A-Z0-9]|&AMP;|&SYMBOL;|%|&WS;)*" m_mod="QUEST"></REL>
</RULE>

<!-- This rule finds a text initial titleline followed by a
    pbreak. The title is marked up and a P start tag is added to
    the following para. -->
<RULE  name="initialtitleline" targ="&A-VAL;&lt;TITLE>&B-VAL;&lt;/TITLE>&C-VAL;&lt;P>">
    <REL match="&START;" m_mod="TEST"></REL>
    <REL var="A" match="(\n|&WS;)*\n" m_mod="QUEST"></REL>
```

```
    <REL var="B" type="REF" match="titleline" ></REL>
    <REL var="C" type="REF" match="pbreak" ></REL>
</RULE>
<!-- This rule finds a titleline preceded by a pbreak or newline and
    followed by a pbreak. The end of the previous para gets a P end
    tag, the title line is marked up and a P start tag is added to
    the following para. -->
<RULE  name="intexttitleline" targ="&lt;/P>&A-VAL;&lt;TITLE>&B-VAL;&lt;/TITLE>&C-
VAL;&lt;P>">
    <REL match="&NOSTART;" m_mod="TEST"></REL>
    <REL  var="A" type="GROUP" match="DISJF">
        <REL type="REF" match="pbreak"></REL>
        <REL match="\n"></REL>
    </REL>
    <REL var="B" type="REF" match="titleline" ></REL>
    <REL var="C" type="REF" match="pbreak" ></REL>
</RULE>

<!--If a text starts with a para (no title) then this rule will put
    in the P start tag. If it is a one line para, it will also put
    in the end tag and start tag of the next para. -->
<RULE  name="first-para" targ="&A-VAL;&lt;P>&B-VAL;&C-REW;">
    <REL match="&START;" m_mod="TEST"></REL>
    <REL var="A" match="(\n|&WS;)*\n" m_mod="QUEST"></REL>
    <REL var="B" match= "([A-zàèìòù0-9 ]|&AMP;|&SYMBOL;|%)+([A-zàèìòù0-
    9]|&AMP;|&SYMBOL;|%|&WS;)*" ></REL>
    <REL var="C" type="GROUP" match="DISJF">
        <REL type="REF" match="pbreak"></REL>
        <REL match="\n" rewrite="&S-VAL;"></REL>
    </REL>
</RULE>

<!-- Deals with end tag of text final para. -->
<RULE  name="last-para" targ="&A-VAL;&lt;/P>&B-VAL;">
    <REL var="A" match= "([A-zàèìòù0-9]|&AMP;|&SYMBOL;|%)+([A-zàèìòù0-
    9]|&AMP;|&SYMBOL;|%|&WS;)*"></REL>
    <REL var="B" match="(\n|&WS;)*" m_mod="QUEST"></REL>
    <REL match="&END;" m_mod="TEST"></REL>
</RULE>

<!-- Deals with files which are only one line long. -->
<RULE  name="one-line-file" targ="&A-VAL;&lt;P>&B-VAL;&lt;/P>&C-VAL;">
    <REL match="&START;" m_mod="TEST"></REL>
    <REL var="A" match="(\n|&WS;)*\n" m_mod="QUEST"></REL>
    <REL var="B" match= "([A-zàèìòù0-9 ]|&AMP;|&SYMBOL;|%)+([A-zàèìòù0-
    9]|&AMP;|&SYMBOL;|%|&WS;)*" ></REL>
    <REL var="C" match="(\n|&WS;)*" m_mod="QUEST"></REL>
    <REL match="&END;" m_mod="TEST"></REL>
</RULE>

<!-- Final catch all. Must ensure that it doesn't mark up text final
    newlines. -->
<RULE  name="parabreak" targ="&A-REW;">
    <REL var="A" type="REF" match="pbreak"></REL>
    <REL var="B" m_mod="TEST" match= "([A-zàèìòù0-9 ]|&AMP;|&SYMBOL;|%)+([A-
    zàèìòù0-9]|&AMP;|&SYMBOL;|%|&WS;)*"></REL>
</RULE>

<!-- If titles are not needed, then comment out the 2nd and 3rd rels
    so that titles will end up as paras/parts of paras. -->
```

```
<RULE name="all" type="DISJF" targ="&S-REW;">
    <REL type="REF" match="one-line-file"></REL>
    <REL type="REF" match="initialtitleline"></REL>
    <REL type="REF" match="intexttitleline"></REL>
    <REL type="REF" match="first-para"></REL>
    <REL type="REF" match="last-para"></REL>
    <REL type="REF" match="parabreak"></REL>
</RULE>

</RULES>
```

A.2 File words2.gr

```
<!-- Autore:          Federico Faccioni →
<!-- data:           5-12-2003 →

<?xml version='1.0'?>
<!DOCTYPE RULES SYSTEM "../../RES/RuleSpec.dtd" [

<!ENTITY % f SYSTEM "../../RES/common-ent">
%f;

<!ENTITY NL             "\n">
<!ENTITY WS             "[ ]">
<!ENTITY WSORNL         "(\n|[ ])">
<!ENTITY PUNCT          "[\(\)\.\?\!,;:'\-]">
<!ENTITY NOTWS          "([A-zàèìòùáéíóú0-9]°|&PUNCT;)">

]>



<RULES  name="words2"  apply="all">



<!--  ======== Punctuation etc ======== -->

<!-- Commas, colons and semi-colons are all marked as C='CM' -->
<RULE name="comma" targ="&lt;W C='CM'>&S-VAL;&lt;/W>">
    <REL match="(,|:|;)"></REL>
</RULE>

<!-- Percent marked as C='PCT'. NB lttok marks these as C='W' -->
<RULE name="percent" targ="&lt;W C='PTC'>&S-VAL;&lt;/W>">
    <REL match="%"></REL>
</RULE>

<!-- Ampersand marked as C='AMP'. NB lttok doesn't mark these -->
<RULE name="ampersand" targ="&lt;W C='AMP'>&S-VAL;&lt;/W>">
    <REL match="&amp;"></REL>
</RULE>

<!-- Quotes of various kinds marked as C='QUOTE' -->
```

```
<RULE name="quote" type="DISJF" targ="&lt;W C='QUOTE'>&S-VAL;&lt;/W>">
    <REL match="(``|'')"></REL>
    <REL match="[`']"></REL>
    <REL match="['']"></REL>
</RULE>

<!-- Brackets of various kinds marked as C='BR' -->
<RULE name="bracket" type="DISJF" targ="&lt;W C='BR'>&S-VAL;&lt;/W>">
    <REL match="[{}]"></REL>
    <REL match="[\(\)]"></REL>
    <REL match="[\[\]]"></REL>
    </RULE>

<RULE name="quote-or-br" type="DISJF">
    <REL type="REF" match="quote"></REL>
    <REL type="REF" match="bracket"></REL>
</RULE>


<!-- sequences of dots marked as C="DASH" -->
<RULE name="dots" targ="&lt;W C='DASH'>&S-VAL;&lt;/W>">
   <REL match="(([\.][\.][\.])|([\.][\.]))"></REL>
</RULE>

<!-- A single dash or sequences of dashes marked as C="DASH" -->
<RULE name="dash" targ="&lt;W C='DASH'>&S-VAL;&lt;/W>">
    <REL match="[\-]+"></REL>
    </RULE>

<RULE name="plus" targ="&lt;W C='PLUS'>&S-VAL;&lt;/W>">
    <REL match="[\+]+"></REL>
</RULE>

<RULE name="dots-or-dash" type="DISJF">
    <REL type="REF" match="dots"></REL>
    <REL type="REF" match="dash"></REL>
    <REL type="REF" match="plus"></REL>
</RULE>

<!-- Question mark and exclamation mark marked as C='.' -->
<RULE name="mark" targ="&lt;W C='.'>&S-VAL;&lt;/W>">
    <REL match="[\?\!]"></REL>
</RULE>

<!-- Full stop marked as C='.' -->
<RULE name="fullstop" targ="&lt;W C='.'>&S-VAL;&lt;/W>">
    <REL match="[\.]"></REL>
</RULE>

<RULE name="punct" type="DISJF">
    <REL type="REF" match="comma"></REL>
    <REL type="REF" match="percent"></REL>
    <REL type="REF" match="ampersand"></REL>
    <REL type="REF" match="quote-or-br"></REL>
    <REL type="REF" match="dots-or-dash"></REL>
    <REL type="REF" match="mark"></REL>
</RULE>


<!--  ======== Cardinals, Ordinals, Fractions ======== -->
```

```
<!-- Cardinal numbers are marked as C='CD' -->
<RULE name="cd" type="DISJ" targ="&lt;W C='CD'>&S-VAL;&lt;/W>">
    <REL match="[0-9][0-9\.]*[0-9](,[0-9]+)?"></REL>
    <REL match="[0-9,]*[0-9][0-9]*"></REL>
</RULE>

<!-- Ordinals are marked as C='ORD' 11esimo a 999esimo-->
<RULE name="ord" type="DISJF" targ="&lt;W C='ORD'>&S-VAL;&lt;/W>">
    <REL match="[0-9]+[Oo°]"></REL>
    <REL match="[1-9][0-9]+esim[oi]"></REL>
</RULE>

<!-- giorno/mese/anno -->
<RULE name="giorno-mese-anno" targ="&lt;W C='GMA'>&S-VAL;&lt;/W>">
    <REL match="(0?[1-9]|[12][0-9]|3[01])[-/](0?[1-9]|1[0-2])[-/]([12][0-9][0-9][0-9]|[0-9][0-
    9])"></REL>
</RULE>

<!-- Fractions are marked as C='FRAC' -->
<RULE name="frac" targ="&lt;W C='FRAC'>&S-VAL;&lt;/W>">
    <REL match="[0-9]+/[0-9]+"></REL>
</RULE>


<!--  ======== Hyphenation ======== -->

<RULE name="special-hyphen-rule1" targ=
"&lt;W C='W'>&A-VAL;&lt;/W>&lt;W C='DASH'>&B-VAL;&lt;/W>&lt;W C='W'>&C-
VAL;&lt;/W>">
    <REL var="A" match="[A-zàèìòùáéíóú°][A-zàèìòùáéíóú°\.]+"></REL>
    <REL var="B" match="[/\-]"></REL>
    <REL var="C" match="[A-zàèìòùáéíóú]+[\.]?"></REL>
</RULE>

<RULE name="special-hyphen-rule2" targ=
"&lt;W C='W'>&A-VAL;&lt;/W>&lt;W C='DASH'>&B-VAL;&lt;/W>&lt;W C='W'>&C-
VAL;&lt;/W>">
    <REL var="A" match="[XVI]+"></REL>
    <REL var="B" match="[\-]"></REL>
    <REL var="C" match="[XVI]+"></REL>
</RULE>

<RULE name="special-hyphen-rules" type="DISJF">
    <REL type="REF" match="special-hyphen-rule1"></REL>
    <REL type="REF" match="special-hyphen-rule2"></REL>
</RULE>


<!--  ======== Miscellaneous ======== -->

<!-- An alphanumeric sequence -->
<RULE name="alphanum"  type="DISJF" targ="&lt;W C='AN'>&S-VAL;&lt;/W>">
    <REL match="[A-z]+[:\/\.\-]*[0-9]+[:\/\.\-]*[A-z]+[0-9]*"></REL>
    <REL match="[0-9]+[:\/\.\-]*[A-z]+[:\/\.\-]*[0-9]+[A-z]*"></REL>
    <REL match="[A-z]+[\/\.\-]*[0-9]+"></REL>
    <REL match="[0-9]+[\/\.\-]*[A-z]+"></REL>
</RULE>

<!-- Mixture of numeric and punctuation - means it's not a simple CD -->
```

```
<RULE name="numeric" type="DISJF" targ="&lt;W C='NUM'>&S-VAL;&lt;/W>">
    <REL match="[0-9]*[:\/\.\-]+[0-9]+[:\/\.\-]+[0-9]+"></REL>
    <REL match="[0-9]*[:\/\.\-]+[0-9]+[:\/\-]+"></REL>
</RULE>
<!--  ======== Words ======== -->


<!-- A realword is all alphabetic except maybe it can contain
    apostrophes, slashes, hyphens or fullstops. Hyphens or fullstops
    may be initial or final as well as internal but apostrophes and
    slashes can only be internal (an initial or final ' is treated as
    a quote.) -->
<RULE name="realword" type="DISJF" targ="&lt;W C='W'>&S-VAL;&lt;/W>">
    <REL match="[Uu][n]['']"></REL>
    <REL match="[Ll]['']"></REL>
    <REL match="[E]['']"></REL>
    <REL match="[Dd]['']"></REL>
    <REL match="[Dd][ae]ll['']"></REL>
    <REL match="[Aa]ll['']"></REL>
    <REL match="[Nn]ell['']"></REL>
    <REL match="[A-zàèìòùáéíóú\.\-][A-zàèìòùáéíóú\/\.\-]*[A-zàèìòùáéíóú\.\-]"></REL>
    <REL match="[A-zàèìòùáéíóú]"></REL>
</RULE>

<!-- Words are either split contracted words, real words, ordinals,
    fractions, cardinals or alphanumeric sequences. Two of the rels
    lines deal with hyphenated words. If all hyphenated words are to
    be split then comment the call to "hyphenated-word" back in. If
    only the special cases are needed then leave the rule as it is. -->
<RULE name="word" type="DISJF">
    <REL type="REF" match="special-hyphen-rules"></REL>
    <REL type="REF" match="ord"></REL>
    <REL type="REF" match="giorno-mese-anno"></REL>
    <REL type="REF" match="frac"></REL>
    <REL type="REF" match="alphanum"></REL>
    <REL type="REF" match="realword"></REL>
    <REL type="REF" match="frac"></REL>
    <REL type="REF" match="cd"></REL>
</RULE>

<!-- Some words have following punctuation: this marks up the word
    and punctuation separately. Nb there can be two punctuation marks
    after a word:  '"Not again," he said' -->
<RULE name="word-punct" targ="&A-REW;&B-REW;&C-REW;">
    <REL var="A" type="REF" match="word"></REL>
    <REL var="B" type="REF" match="punct"></REL>
    <REL var="C" type="REF" match="punct" m_mod="QUEST"></REL>
    </RULE>

<!-- $ marked as C='W' -->
<RULE name="symbol-word" targ="&lt;W C='W'>&S-VAL;&lt;/W>">
    <REL match="[$#]"></REL>
</RULE>

<!-- Words are either words with following punctuation or just words.
    optionally they can be preceded by quotes or brackets. -->
<RULE name="words" targ="&A-REW;&B-REW;">
    <REL var="A" type="REF" match="quote-or-br" m_mod="QUEST"></REL>
    <REL var="B" type="GROUP"  match="DISJF">
        <REL type="REF" match="word-punct"></REL>
```

```
        <REL type="REF" match="word"></REL>
        <REL type="REF" match="symbol-word"></REL>
    </REL>
</RULE>
<!-- This is the key rule: it identifies as a sequence a string that
    matches the rule "words" followed by whitespace. Running text
    will consist of a sequence of these things. -->
<RULE name="word-ws" targ="&A-REW;">
    <REL var="A" type="REF" match="words"></REL>
    <REL var="B"          match="&WSORNL;+" m_mod="TEST"></REL>
</RULE>

<!-- This does the same for punctuation sequences like ... -->
<RULE name="punct-ws" targ="&A-REW;">
    <REL var="A" type="REF" match="punct"></REL>
    <REL var="B"          match="&WSORNL;+" m_mod="TEST"></REL>
</RULE>

<!-- This gets words or punctuation which isn't followed by whitespace:
    it must only be used if word-ws and punct-ws have failed -->
<RULE name="final-word" type="DISJF">
    <REL type="REF" match="words"></REL>
    <REL type="REF" match="punct"></REL>
</RULE>

<!-- Basic sequence is a string of characters followed by
    whitespace. First look just for punctuation sequences
    (e.g. "...") with following whitespace. Then look for a word
    with following whitespace.  Words which are final within an XML
    element have no following whitespace, so the call to
    final-word gets these. This shouldn't get used for any other
    purpose. The final call to "fullstop" catches any fullstops that
    have been missed along the way. -->
<RULE name="all" type="DISJF">
    <REL type="REF" match="punct-ws"></REL>
    <REL type="REF" match="word-ws"></REL>
    <REL type="REF" match="fullstop"></REL>
    <REL type="REF" match="final-word"></REL>
</RULE>

</RULES>
```

A.3 File numbers2.gr

```
<!-- Autore:        Federico Faccioni →
<!-- data:          5-12-2003 →


<?xml version='1.0'?>
<!DOCTYPE RULES SYSTEM "../../RES/RuleSpec.dtd" [

<!ENTITY % f SYSTEM "../../RES/common-ent">
%f;
```

```
<!ENTITY UNIT
    "([Uu]no|[Dd]ue|[Tt]re|[Qq]uattro|[Cc]inque|[Ss]ei|[Ss]ette|[Oo]tto|[Nn]ove)">
<!ENTITY TEEN
    "([Dd]ieci|[Uu]ndici|[Dd]odici|[Tt]redici|[Qq]uattordici|[Qq]uindici|[Ss]edici|[Dd]iciassette|[D
d]iciotto|[Dd]iciannove)">
<!ENTITY TY
    "([Vv]ent[i]?|[Tt]rent[a]?|[Qq]uarant[a]?|[Cc]inquant[a]?|[Ss]essant[a]?|[Ss]ettant[a]?|[Oo]tt
ant[a]?|[Nn]ovant[a]?)">
<!ENTITY BIG-UNIT"
    ([Cc]ent[o]?|[Mm]ille|[Mm]ilione|[Mm]iliard[oi]|[Bb]ilion[ei]|[Tt]rilion[ei])">
<!ENTITY CENTO          "([Cc]ento|[Cc]ent)">
<!ENTITY MILLE          "([Mm]ille|[Mm]ila|[Mm]ill)">
<!ENTITY MILIONE        "([Mm]ilione|[Mm]ilioni)">
<!ENTITY NN             "(&UNIT;|&TEEN;|&TY;|&TY;&UNIT;)">
<!ENTITY NNORD          "(&UNITESIM;|&TEEN-AMB;|&TY-AMB;|&TY;&UNITESIM;)">
<!ENTITY CDM
    "([Cc]entinai[ao]|[Dd]ecin[ea]|[Dd]ozzin[ae]|[Mm]igliai[ao]|[Mm]ilion[ei]|[Mm]iliard[oi])">
<!ENTITY CD-DIGIT       "W[C=CD]|W[C=AN]">
<!ENTITY E              "W/#~^[Ee]$">
<!ENTITY UN             "W/#~^[Uu][nN][Aa]?$">
<!ENTITY DI             "W/#~^[Dd]i$">
<!ENTITY THE            "W/#~^([Ll][aeo]|[Ll]'|[Ii][l]|[Ii]|(gli))$">
<!ENTITY DASH           "W/#=-">
<!ENTITY AL             "W/#~^([Aa]ll['e]|[Aa]l)$">
<!ENTITY WRD            "W/#~^[A-zàèìòùáéíóú]+$">

<!ENTITY UNITESIM
    "(esim[oaei]|unesim[oaei]|duesim[oaei]|treesim[oaei]|quattresim[oaei]|cinquesim[oaei]|sei
esim[oaei]|settesim[oaei]|ottesim[oaei]|novesim[oaei])">
<!ENTITY UNIT-AMB
    "([Pp]rim[oaei]|[Ss]econd[oaei]|[Tt]erz[oaei]|[Qq]uart[oaei]|[Qq]uint[oaei]|[Ss]est[oaei]|[Ss]
ettim[oaei]|[Oo]ttav[oaei]|[Nn]on[oaei])">
<!ENTITY TEEN-AMB
    "([Dd]ecim[oaei]|[Uu]ndicesim[oaei]|[Dd]odicesim[oaei]|[Tt]redicesim[oaei]|[Qq]uattordices
im[oaei]|[Qq]uidicesim[oi]|[Ss]edicesim[oaei]|[Dd]iciassettesim[oaei]|[Dd]iciottesim[oaei]|[Dd]i
ciannovesim[oaei])">
<!ENTITY TY-AMB
    "([Vv]entesim[oaei]|[Tt]rentesim[oaei]|[Qq]uarantesim[oaei]|[Cc]inquantesim[oaei]|[Ss]ess
antesim[oi]|[Ss]ettantesim[oaei]|[Oo]ttantesim[oaei]|[Nn]ovantesim[oaei])">
<!ENTITY BIG-UNIT-AMB
    "([Cc]entesim[oaei]|[Mm]illesim[oiae]|[Mm]ilionesim[oaei]|[Mm]iliardesim[oaei])">

]>



<RULES  name="numbers2" apply="all" type="SGML">


<LEX type="PHRASE"
    file_name="&TTTDIR;/LEX/numbers2.lex"
    alias="LEX">
</LEX>

<LEX type="PHRASE"
    file_name="&TTTDIR;/LEX/timex2.lex"
    alias="TIMLEX">
</LEX>
```

```
<!-- ==========NUMERI IL LETTERE TUTTO ATTACCATO===============-->

<!--1 a 99 -->
<RULE name="1-to-99"  targ_sg="@[C='CD']">
    <REL match="W/#~^&NN;$"></REL>
</RULE>

<!--100 a 999  -->
<RULE name="100-to-999"  targ_sg="@[C='CD']">
    <REL match="W/#~^&UNIT;?&CENTO;&NN;?$"> </REL>
</RULE>

<!--1 a 999 -->
<RULE name="1-to-999" type="DISJF">
    <REL type="REF" match="1-to-99"></REL>
    <REL type="REF" match="100-to-999"></REL>
</RULE>

<!--(1/999)000 a (1/999)099  migliaia senza centinaia-->
<RULE name="1000-to-9099"  targ_sg="@[C='CD']">
    <REL match="W/#~^[a-z]*&MILLE;&NN;?$"> </REL>
</RULE>

<!--(1/999)100 a (1/999)999  migliaia con centinaia -->
<RULE name="1000-to-999999"  targ_sg="@[C='CD']">
    <REL match="W/#~^[a-z]*&MILLE;[a-z]*&CENTO;&NN;?$"> </REL>
</RULE>

<!--oltre il milione -->
<RULE name="milioni"  targ_sg="@[C='CD']">
    <REL match="W/#~^[a-z]*&MILIONE;[a-z]*$"> </REL>
</RULE>

<!--1 a 99.999 -->
<RULE name="textnum" type="DISJF">
    <REL type="REF" match="1-to-999"></REL>
    <REL type="REF" match="1000-to-9099"></REL>
    <REL type="REF" match="1000-to-999999"></REL>
    <REL type="REF" match="milioni"></REL>
</RULE>


<!--==============TEXTNUM2 ================================-->

<RULE name="and-to-999" targ_sg="PHR[C='CD']">
    <REL  match="&E;"></REL>
    <REL type="REF" match="1-to-999"></REL>
</RULE>
<RULE name="mila_sep" targ_sg="PHR[C='CD']" >
    <REL  match="&CD-DIGIT;/#~^[1-9][0-9]?[0-9]?$"></REL>
    <REL  match="W/#~^mila$"></REL>
</RULE>

<RULE name="mila_int" targ_sg="PHR[C='CD']" >
    <REL  match="AN/#~^[1-9][0-9]?[0-9]?mila$"></REL>
</RULE>

<RULE name="mila-999" targ_sg="PHR[C='CD']" >
    <REL type="GROUP" match="DISJF">
```

```
        <REL type="REF" match="mila_sep"></REL>
        <REL type="REF" match="mila_int"></REL>
    </REL>
    <REL type="GROUP" match="DISJF" m_mod="QUEST">
        <REL type="REF" match="1-to-999"></REL>
        <REL type="REF" match="and-to-999"></REL>
    </REL>
</RULE>

<RULE name="to-999-999" type="DISJ">
    <REL type="REF" match="mila-999"></REL>
    <REL type="REF" match="textnum" ></REL>
</RULE>

<RULE name="milione" targ_sg="PHR[C='CD']" >
    <REL type="GROUP" match="DISJF">
        <REL  match="&UN;"></REL>  <!--un, una-->
        <REL  match="&CD-DIGIT;"></REL>
        <REL  type="REF" match="textnum"></REL>
    </REL>
    <REL  match="W/#~^milion[ei]$"></REL>
</RULE>

<RULE name="milione-999-999" targ_sg="PHR[C='CD']" >
    <REL type="REF" match="milione"></REL>
    <REL type="GROUP" match="DISJF">
        <REL type="REF" match="to-999-999"></REL>
        <REL type="GROUP" match="SEQ">
            <REL  match="&E;"></REL>
            <REL type="REF" match="to-999-999"></REL>
        </REL>
    </REL>
</RULE>

<RULE name="milione-999" targ_sg="PHR[C='CD']" >
    <REL type="REF" match="milione"></REL>
    <REL type="REF" match="and-to-999" m_mod="QUEST"></REL>
</RULE>

<RULE name="to-999-999-999" type="DISJ">
    <REL type="REF" match="milione-999-999"></REL>
    <REL type="REF" match="milione-999"></REL>
    <REL type="REF" match="to-999-999"></REL>     <!--inferiore al milione -->
</RULE>

<RULE name="miliardo" targ_sg="PHR[C='CD']" >
    <REL type="GROUP" match="DISJF">
        <REL match="&UN;"></REL>
        <REL match="&CD-DIGIT;"></REL>
        <REL  type="REF" match="textnum"></REL>
    </REL>
    <REL  match="W/#~^miliard[oi]$"></REL>
</RULE>

<RULE name="miliardo-999-999-999" targ_sg="PHR[C='CD']" >
    <REL type="REF" match="miliardo"></REL>
     <REL type="GROUP" match="DISJF">
        <REL type="REF" match="to-999-999-999"></REL>
        <REL type="GROUP" match="SEQ">
            <REL  match="&E;"></REL>
```

```
            <REL type="REF" match="to-999-999-999"></REL>
        </REL>
    </REL>
</RULE>

<RULE name="miliardo-999" targ_sg="PHR[C='CD']" >
    <REL type="REF" match="miliardo"></REL>
    <REL type="REF" match="and-to-999" m_mod="QUEST"></REL>
</RULE>

<RULE name="textnum2" type="DISJ">
    <REL type="REF" match="miliardo-999-999-999"></REL>
    <REL type="REF" match="miliardo-999"></REL>
    <REL type="REF" match="to-999-999-999"></REL>
</RULE>


<!--==================== DIGIT NUMBERS ======================-->


<!-- a <W C='CD'> is recognised by this rule but not
    marked up any further -->
<RULE name="digit" targ="&S-VAL;">
    <REL match="&CD-DIGIT;"></REL>
</RULE>

<!-- il - viene etichettato come C='DASH' da words2.gr -->
<RULE name="minusdigit" targ_sg="PHR[C='CD']">
    <REL match="W/#~^([Mm]eno|-)$"></REL>
    <REL match="&CD-DIGIT;"></REL>
</RULE>

<RULE name="digits" type="DISJF">
    <REL type="REF" match="digit"></REL>
    <REL type="REF" match="minusdigit"></REL>
</RULE>


<!--================= TEXTNUM-ORDINAL ========================== -->


<!-- a <W C='ORD'> is recognised by this rule as a "ord-digit"
    but not marked up any further. -->
<RULE name="ord-digit" targ='&S-VAL;'>
    <REL match="W[C=ORD]"></REL>
</RULE>

<!--1o a 9o in lettere -->
<RULE name="unith-amb"  targ_sg="@[C='FRACORD']">
    <REL match="W/#~^&UNIT-AMB;$"></REL>
</RULE>

<!--10o a 19o  in lettere-->
<RULE name="teenth-amb"  targ_sg="@[C='FRACORD']">
    <REL match="W/#~^&TEEN-AMB;$"></REL>
</RULE>

<!--20o 30o... 90o  in lettere-->
<RULE name="tieth-amb"  targ_sg="@[C='FRACORD']">
    <REL match="W/#~^&TY-AMB;$"></REL>
```

```
</RULE>

<!--21o 32o  78o... in lettere-->
<RULE name="ty-unith-amb"  targ_sg="@[C='FRACORD']">
    <REL match="W/#~^&TY;&UNITESIM;$"></REL>
</RULE>

<!--121o 102o  410o... in lettere-->
<RULE name="over-100-amb"  targ_sg="@[C='FRACORD']">
    <REL match="W/#~^&UNIT;?&CENTO;&NNORD;$"></REL>
</RULE>

<!--100o 1000o  1000o... in lettere-->
<RULE name="big-unith-amb"  targ_sg="@[C='FRACORD']">
    <REL match="W/#~^&BIG-UNIT-AMB;$"></REL>
</RULE>

<RULE name="amb-word-becomes-ord" type="DISJF" targ_sg="@[C='ORD']">
    <REL type="REF" match="unith-amb"></REL>
    <REL type="REF" match="teenth-amb"></REL>
    <REL type="REF" match="tieth-amb"></REL>
    <REL type="REF" match="ty-unith-amb"></REL>
    <REL type="REF" match="over-100-amb"></REL>
    <REL type="REF" match="big-unith-amb"></REL>
</RULE>

<RULE name="textnum-ordinal" type="DISJF" >
    <REL type="REF" match="ord-digit"></REL>
    <REL type="REF" match="amb-word-becomes-ord"></REL>
</RULE>


<!--================== FRAC-DENOM ==================== -->


<RULE name="to-999th-amb" type="DISJF">
    <REL type="REF" match="unith-amb"></REL>
    <REL type="REF" match="teenth-amb"></REL>
    <REL type="REF" match="tieth-amb"></REL>
    <REL type="REF" match="ty-unith-amb"></REL>
    <REL type="REF" match="over-100-amb"></REL>
    <REL type="REF" match="big-unith-amb"></REL>
</RULE>

<!-- Ovvero qui cerco solo i denominatori di espressioni frazionarie
    scritte in lettere. In inglese ci sono parole che hanno questo
    preciso significato, ma non in italiano -->
<RULE name="frac-denom" type="DISJF">
    <REL type="REF" match="to-999th-amb"></REL>
    <REL type="REF" match="ord-digit"></REL>
</RULE>


<!--================ TEXTNUM-FRACTION ========================== -->


<RULE name="frac-digit" targ='&S-VAL;'>
    <REL  match="W[C='FRAC']"></REL>
</RULE>
```

```
<!-- tre quinti -->
<RULE name="fraction" targ_sg="PHR[C='FRAC']">
    <REL type="REF" match="textnum"></REL>
    <REL type="REF" match="frac-denom"></REL>
</RULE>

<!-- quando il numero ordinale è preceduto dall'articolo indeterminativo
    "un" non è possibile determinare univocamente se rappresent una quantità frazionaria -->
<RULE name="amb-word-denom-ord" targ_sg="@[C='FRACORD']">
    <REL match="&UN;" m_mod="TEST"></REL>
    <REL type="GROUP"  match="DISJF">
        <REL type="REF" match="unith-amb"></REL>
        <REL type="REF" match="teenth-amb"></REL>
        <REL type="REF" match="tieth-amb"></REL>
        <REL type="REF" match="ty-unith-amb"></REL>
        <REL type="REF" match="over-100-amb"></REL>
        <REL type="REF" match="big-unith-amb"></REL>
        <REL type="REF" match="ord-digit"></REL>
    </REL>
</RULE>

<!-- Tutte possibili frazioni inclusi casi ambigui-->
<RULE name="textnum-fraction"  type="DISJF">
    <REL type="REF" match="amb-word-denom-ord"></REL>
    <REL type="REF" match="fraction"></REL>
    <REL type="REF" match="frac-digit"></REL>
</RULE>

<!-- vengono etichettati come "NONDEF" i numeri ordinali che molto
    probabilmente, ma non certamente lo sono -->
<RULE name="no-ord-1" targ_sg="@[C='NONDEF']">
    <REL type="REF" match="to-999th-amb"></REL>
    <REL match="&THE;" m_mod="TEST"></REL>
</RULE>

<RULE name="no-ord-2" targ_sg="@[C='NONDEF']">
    <REL type="REF" match="to-999th-amb"></REL>
    <REL match="W/#~^cui$" m_mod="TEST"></REL>
</RULE>

<RULE name="no-ord-3" targ_sg="@[C='NONDEF']">
    <REL type="REF" match="to-999th-amb"></REL>
    <REL match="W/#~^((di)|(del)|(della))$" m_mod="TEST"></REL>
</RULE>

<RULE name="no-ord" type="DISJF">
    <REL type="REF" match="no-ord-1"></REL>
    <REL type="REF" match="no-ord-2"></REL>
    <REL type="REF" match="no-ord-3"></REL>
</RULE>


<!--================== FRAC-NUM ============================== -->


<!--  tre quarti di milione, un quarto di
    milione, 3/4 di milione, mezzo milione -->
<RULE name="frac-num" targ_sg="PHR[C='CD']">
    <REL type="GROUP"  match="DISJF">
        <REL match="W/#~^[Mm]ezz[ao]$"></REL>
```

```
        <REL type="GROUP"  match="SEQ">
            <REL type="REF" match="textnum-fraction"></REL>
            <REL           match="&DI;"></REL>
        </REL>
    </REL>
    <REL type="GROUP"  match="DISJF">
        <REL           match="W/#~iliardo$"></REL>
        <REL           match="W/#~ilione$"></REL>
        <REL           match="W/#~^dozzina$"></REL>
    </REL>
</RULE>


<!--==================== CARDINALS ============================-->


<RULE name="frac-digit" targ='&S-VAL;'>
    <REL  match="W[C='FRAC']"></REL>
</RULE>

<!-- due milioni e mezzo -->
<RULE name="textnum-mezzo" targ_sg="PHR[C='CD']">
    <REL type="REF" match="textnum2"></REL>
    <REL           match="&E;"></REL>
    <REL           match="W/#~^mezzo$"></REL>
</RULE>

<!-- tre e tre quarti -->
<RULE name="textnum-and-fraction" targ_sg="PHR[C='CD']">
    <REL type="REF" match="textnum"></REL>
    <REL           match="&E;"></REL>
    <REL type="REF" match="textnum-fraction"></REL>
</RULE>

<RULE name="textnum-cardinal" type="DISJF">
    <REL type="REF" match="textnum-mezzo"></REL>
    <REL type="REF" match="textnum-and-fraction"></REL>
    <REL type="REF" match="textnum2"></REL>
</RULE>

<!-- 3 3/4 -->
<RULE name="digitnum-and-fraction" targ_sg="PHR[C='CD']">
    <REL match="digits" type="REF"></REL>
    <REL match="frac-digit" type="REF"></REL>
</RULE>

<RULE name="cardinals" type="DISJF">
    <REL type="REF" match="textnum-cardinal"></REL>
    <REL type="REF" match="digitnum-and-fraction"></REL>
    <REL type="REF" match="digits"></REL>
</RULE>

<RULE name="all-number" type="DISJF">
    <REL type="REF" match="no-ord"></REL>
    <REL type="REF" match="textnum-ordinal"></REL>
    <REL type="REF" match="frac-num"></REL>
    <REL type="REF" match="textnum-fraction"></REL>
    <REL type="REF" match="cardinals"></REL>
</RULE>
```

```
<!--============================ RANGE ========================-->

<RULE  name="to-dash" type="DISJF">
    <REL match="&AL;"></REL>
    <REL match="&DASH;"></REL>
</RULE>

<!--1920-1924. Ranges which could be year ranges are found
    before more general ranges. Needed for timex.gr but can
    be removed when dates are not being found.  -->
<RULE  name="card-to-card-yr" targ_sg="PHR[C='YRRANGE']">
    <REL match="W/#~^[12][0-9][0-9][0-9]$"></REL>
    <REL type="REF" match="to-dash"></REL>
    <REL match="W/#~^[12][0-9][0-9][0-9]$"></REL>
</RULE>

<!--1920 or 1921. Ranges which could be year ranges are found
    before more general ranges. Needed for timex.gr but can
    be removed when dates are not being found.  -->
<RULE  name="card-or-card-yr" targ_sg="PHR[C='YRRANGE']">
    <REL match="W/#~^[12][0-9][0-9][0-9]$"></REL>
    <REL match="W/#~^[Oo]$"></REL>
    <REL match="W/#~^[12][0-9][0-9][0-9]$"></REL>
</RULE>

<!--between 1920 and 1922. Ranges which could be year ranges are
    found before more general ranges. Needed for timex.gr but can
    be removed when dates are not being found.  -->
<RULE  name="between-card-and-card-yr" targ_sg="PHR[C='YRRANGE']">
    <REL match="W/#~^([Tt]ra)$"></REL>
    <REL match="W/#~^(il)$"></REL>
    <REL match="W/#~^[12][0-9][0-9][0-9]$"></REL>
    <REL match="W/#~^(ed)$"></REL>
    <REL match="W/#~^(il)$"></REL>
    <REL match="W/#~^[12][0-9][0-9][0-9]$"></REL>
</RULE>

<RULE  name="cardinals-romani" type="DISJF">
    <REL type="REF" match="cardinals"></REL>
    <REL      match="W/#~^[XVI]+"></REL>
</RULE>

<RULE  name="ore-digitali">
    <REL match="W/#~^([1-9]|1[0-9]|2[0-3]|00)$"></REL>
    <REL match="W/#~^[\:]$"></REL>
    <REL match="W/#~^[0-5][0-9]$"></REL>
</RULE>
<!--  20-30 # X-XI # 20,23-30,45 #   20.23-30.45
    20 al 30 # X al XI # 20,23 alle 30,45 #   20.23 alle 30.45 -->
<RULE  name="card-to-card" targ_sg="PHR[C='RANGE']">
    <REL type="GROUP"  match="DISJF">
        <REL type="REF" match="ore-digitali"></REL>
        <REL type="REF" match="cardinals-romani"></REL>
    </REL>
    <REL type="REF" match="to-dash"></REL>
    <REL type="GROUP"  match="DISJF">
        <REL type="REF" match="ore-digitali"></REL>
        <REL type="REF" match="cardinals-romani"></REL>
    </REL>
```

```
</RULE>

<!-- 20 o 30 # X o XI # 20,23 o 30,45 #   20.23 o 30.45-->
<RULE  name="card-or-card" targ_sg="PHR[C='RANGE']">
    <REL type="GROUP"  match="DISJF">
        <REL type="REF" match="ore-digitali"></REL>
        <REL type="REF" match="cardinals-romani"></REL>
    </REL>
    <REL match="W/#~^[Oo]$"></REL>
    <REL type="GROUP"  match="DISJF">
        <REL type="REF" match="ore-digitali"></REL>
        <REL type="REF" match="cardinals-romani"></REL>
    </REL>
</RULE>

<RULE  name="all-range" type="DISJF">
    <REL type="REF" match="card-to-card-yr"></REL>
    <REL type="REF" match="card-to-card"></REL>
    <REL type="REF" match="card-or-card-yr"></REL>
    <REL type="REF" match="card-or-card"></REL>
    <REL type="REF" match="between-card-and-card-yr"></REL>
</RULE>


<!—=========================QUANTITY ===========================-->


<!-- cento, mille milioni... -->
<RULE  name="big"  type="PHRASE"  targ_sg="PHR[C='QUANT']" >
    <REL  match="&WRD;"  >
    <CONSTR   check_in= "LEX" subpart='([a-z]+)[ei]$'
        check_tags="BIG-UNITS *"
        check_mod="LOWERCASE"  >
    </CONSTR>
    </REL>
</RULE>

<!-- ventina, trentina...  -->
<RULE  name="quant" type="PHRASE" targ_sg="PHR[C='QUANT']" >
    <REL  match="&WRD;"  >
    <CONSTR   check_in= "LEX"
        subpart='([a-z]+)[ea]$'   check_tags="QUANT *"
        check_mod="LOWERCASE"  >
    </CONSTR>
    </REL>
</RULE>


<!--pochi, tanti, alcuni -->
<RULE name="aggettivo" type="PHRASE" targ_sg="PHR[C='QUANT']">
    <REL match="&WRD;">
    <CONSTR check_in="LEX"
        check_tags="AGG *"
        check_mod="LOWERCASE">
    </CONSTR>
    </REL>
</RULE>

<!-- alcune decine di milioni, alcune decine -->
<RULE name="quantity1" targ_sg="PHR[C='QUANT']">
```

```
    <REL type="REF" match="aggettivo"></REL>
    <REL type="REF" match="quant"></REL>
    <REL          match="&DI;" m_mod="QUEST"></REL>
    <REL type="REF" match="big"  m_mod="QUEST"></REL>
</RULE>

<!-- decine di, decine di milioni -->
<RULE name="quantity2" targ_sg="PHR[C='QUANT']">
    <REL type="REF" match="quant"></REL>
    <REL          match="&DI;" ></REL>
    <REL type="REF" match="big" m_mod="QUEST"></REL>
</RULE>

<!-- pochi milioni, oltre un milione ... -->
<RULE  name="quantity3" targ_sg="PHR[C='QUANT']">
    <REL type="REF" match="aggettivo"></REL>
    <REL match="&UN;" m_mod="QUEST"></REL>
    <REL type="REF" match="big"></REL>
</RULE>

<!-- pochi milioni, molte decine, pochi trentina  -->
<RULE  name="quantity4" type="DISJF" targ_sg="@[C='QUANT']">
    <REL type="REF" match="aggettivo"></REL>
    <REL type="REF" match="quant"></REL>
</RULE>

<!-- metà ... -->
<RULE  name="quantity5" targ_sg="@[C='QUANT']">
    <REL  match="W/#~^metà$"></REL>
</RULE>

<RULE  name="all-quant" type="DISJF">
    <REL type="REF" match="quantity1"> </REL>
    <REL type="REF" match="quantity2"> </REL>
    <REL type="REF" match="quantity3"> </REL>
    <REL type="REF" match="quantity4"> </REL>
    <REL type="REF" match="quantity5"> </REL> </RULE>


<!--   ========== ALL NUMBERS, RANGES, QUANTS ===========   -->


<RULE name="all" type="DISJF">
    <REL type="REF" match="all-range"></REL>
    <REL type="REF" match="all-quant"></REL>
    <REL type="REF" match="all-number"></REL>
</RULE>
</RULES>
```

A.4 File numex2.gr

```
<!-- Autore:        Federico Faccioni →
<!-- data:          5-12-2003 →
```

```
<?xml version='1.0'?>
<!DOCTYPE RULES SYSTEM "../../RES/RuleSpec.dtd" [

<!ENTITY % f SYSTEM "../../RES/common-ent">
%f;

<!ENTITY WRD            "W/#~^[A-z]+$">
<!ENTITY CARD           "(W[C='CD']|PHR[C='CD'])">
<!ENTITY DASH           "W/#=-">
<!ENTITY AND            "W/#~^([Ee][d]?)$">
<!ENTITY OF             "W/#~^([Dd]i)$">
<!ENTITY OR             "W/#~^([Oo]r|OR)$">
<!ENTITY TO             "W/#~^([Tt]o|TO)$">
<!ENTITY JUSTA          "W/#~^[Aa]$">
<!ENTITY AORAN          "W/#~^[Uu]n[a]?$">

]>



<RULES  name="numex2" apply="all" type="SGML">



<LEX type="PHRASE"
    file_name="&TTTDIR;/LEX/numex.lex"
    alias="LEX"></LEX>


<!-- ================= MONEY ==================== -->


<RULE name="currency-short" type="PHRASE">
    <REL match="W">
        <CONSTR check_in="LEX" check_tags="$- $abb *"></CONSTR>
    </REL>
</RULE>

<RULE name="currency-long" type="PHRASE">
    <REL match="W">
        <CONSTR check_in="LEX" check_tags="$full $unit *"
        check_mod="LOWERCASE">
        </CONSTR>
    </REL>
</RULE>

<RULE name="currency" type="DISJF">
    <REL type="REF" match="currency-long"></REL>
    <REL type="REF" match="currency-short"></REL>
</RULE>

<RULE name="currency-smallunits" type="PHRASE">
    <REL match="W">
    <CONSTR check_in="LEX" check_tags="$c *"
    check_mod="LOWERCASE">
    </CONSTR>
    </REL>
</RULE>

<!-- 25, venticinque, trecento, poche centinaia, 20-30 ... -->
```

```
<RULE name="quantity" type="DISJF">
    <REL match="W[C='CD']"></REL>
    <REL match="PHR[C='CD']"></REL>
    <REL match="PHR[C='QUANT']"></REL>
    <REL match="PHR[C='RANGE']"></REL>
</RULE>

<!-- $700; US$ 700; $US7000 ..... -->
<RULE name="currency-number" targ_sg="NUMEX[TYPE='MONEY']">
    <REL type="REF" match="currency-short"></REL>
    <REL match="W[C='CD']|PHR[C='CD']"></REL>
</RULE>

<!-- duemila euro, un milione di euro, pochi euro... -->
<RULE name="number-currency" targ_sg="NUMEX[TYPE='MONEY']">
    <REL type="REF" match="quantity"></REL>
    <REL match="&OF;" m_mod="QUEST" ></REL>
    <REL type="REF" match="currency"></REL>
</RULE>

<!-- un euro, un dollaro, ... -->
<RULE name="a-currency" targ_sg="NUMEX[TYPE='MONEY']">
    <REL match="&AORAN;"></REL>
    <REL type="REF" match="currency-long"></REL>
</RULE>

<!-- 30 centesimi pochi centesimi..... -->
<RULE name="number-smallunit" targ_sg="NUMEX[TYPE='MONEY']">
    <REL type="REF" match="quantity"></REL>
    <REL type="REF" match="currency-smallunits"></REL>
    <REL type="GROUP" match="SEQ" m_mod="QUEST">
        <REL      match="&OF;"></REL>
        <REL type="REF" match="currency"></REL>
    </REL>
</RULE>

<!-- twenty five dollars and 50 cents, ... -->
<RULE  name = "bigsum-and-smallunit" targ_sg="NUMEX[TYPE='MONEY']">
    <REL type="GROUP"  match="DISJF" >
        <REL type="REF" match="number-currency"></REL>
        <REL type="REF" match="a-currency"></REL>
    </REL>
    <REL match="&AND;"></REL>
    <REL type="REF" match="number-smallunit"></REL>
</RULE>

<RULE name="money" type="DISJF">
    <REL type="REF" match="bigsum-and-smallunit"></REL>
    <REL type="REF" match="currency-number"></REL>
    <REL type="REF" match="number-currency"></REL>
    <REL type="REF" match="a-currency"></REL>
    <REL type="REF" match="number-smallunit"></REL>
</RULE>

<RULE name="money-più-meno" targ_sg="NUMEX[TYPE='MONEY']">
    <REL type="GROUP"  match="SEQ" m_mod="QUEST" >
        <REL match="W/#~^((meno)|(più))$"></REL>
        <REL match="W/#~^di$"></REL>
    </REL>
    <REL type="REF" match="money"></REL>
```

```
    <REL type="GROUP"  match="SEQ" m_mod="QUEST" >
        <REL match="W/#~^in$"></REL>
        <REL match="W/#~^((meno)|(più))$"></REL>
    </REL>
</RULE>


<!-- ================== PERCENT ======================      -->


<!-- %, percento, per-cento -->
<RULE name="percent_wrd" type="DISJF">
    <REL match="W[C='PCT']"></REL>
    <REL type="GROUP"  match="SEQ" >
        <REL match="W/#~^(([Pp]er)|(PER))$"></REL>
        <REL match="W/#~^((cento)|(CENTO))$"></REL>
    </REL>
    <REL match="W/#~^((%)|([Pp]ercento)|(PERCENTO))$"></REL>
</RULE>

<RULE name="percent" targ_sg="NUMEX[TYPE='PERCENT']">
    <REL type="GROUP"  match="SEQ" m_mod="QUEST" >
        <REL match="W/#~^((meno)|(più))$"></REL>
        <REL match="W/#~^del$"></REL>
    </REL>
    <REL match="W/#=+" m_mod="QUEST"></REL>
    <REL type="REF" match="quantity"></REL>
    <REL type="REF" match="percent_wrd"></REL>
    <REL type="GROUP"  match="SEQ" m_mod="QUEST" >
        <REL match="W/#~^in$"></REL>
        <REL match="W/#~^((meno)|(più))$"></REL>
    </REL>
</RULE>


<!--   ========== ALL  ===============         -->


<RULE name="all" type="DISJF">
    <REL type="REF" match="money-più-meno"></REL>
    <REL type="REF" match="percent"></REL>
</RULE>

</RULES>
```

A.5 File timex2.gr

```
<!-- Autore:          Federico Faccioni →
<!-- data:            5-12-2003 →

<?xml version='1.0'?>
<!DOCTYPE RULES SYSTEM "../../RES/RuleSpec.dtd" [

<!ENTITY % f SYSTEM "../../RES/common-ent">
%f;
```

```
<!ENTITY WRD          "W/#~^[A-z']+$">
<!ENTITY CAPWRD       "W/#~^[A-Z][A-z]*$">
<!ENTITY LOWWRD       "W/#~^[a-z]+$">
<!ENTITY CARD         "W[C='CD']|PHR[C='CD']">
<!ENTITY ORD          "W[C='ORD']|PHR[C='ORD']">
<!ENTITY FRAC         "W[C='FRAC']|PHR[C='FRAC']">
<!ENTITY RANGE        "PHR[C='RANGE']">
<!ENTITY RANGE2       "PHR[C='RANGE2']">
<!ENTITY QUANT        "W[C='QUANT']|PHR[C='QUANT']">
<!ENTITY PUNCT        "W/#~^([,.;:?!\(\)]|'|``)$">
<!ENTITY DASH         "W/#=-">
<!ENTITY AND          "W/#~^([Aa]nd|AND)$">
<!ENTITY THE          "W/#~^([Ll][aeo]|[Ll]'|[Ii][l])$">
<!ENTITY OF           "W/#~^([Dd]['i]|[Dd]ell[ea]|[Dd]ell'|[Dd]el|degli)$">
<!ENTITY OR           "W/#~^([Oo]r|OR)$">
<!ENTITY TO           "W/#~^([Tt]o|TO)$">
<!ENTITY AL           "W/#~^[Aa]ll['ae]|[Aa][li]|[Aa]$">
<!ENTITY AORAN        "W/#~^([Aa][n]?|[A][N]?)$">
<!ENTITY AORONE       "W/#~^([Uu]n['a]|[Uu]n)$">

]>



<RULES  name="timex2" apply="all" type="SGML">



<LEX type="PHRASE"
    file_name="&TTTDIR;/LEX/timex2.lex"
    alias="TIMLEX">
</LEX>


<!-- ========== GENERAL PURPOSE RULES ================ -->


<!-- Lexical look-up: look up a lowercase version of the word. -->
<RULE name="look-up" type="PHRASE" arg="$1 $2">
    <REL match="$1">
        <CONSTR   check_in="TIMLEX" check_tags="$2" check_mod="LOWERCASE">
        </CONSTR>
    </REL>
</RULE>

<RULE name="preposizioni-lex">
    <REL match="look-up" type="REF">
    <ARG bind='$1'>&WRD;</ARG>
    <ARG bind='$2'>PRP *</ARG>
    </REL>
</RULE>

<RULE name="misc" type="DISJF">
    <REL  match="W/#~^già$"></REL>
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>MISC *</ARG>
    </REL>
</RULE>
```

```
<!-- festa della mamma ... -->
<RULE name="nomi-feste-lex" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>HOL *</ARG>
    </REL>
</RULE>

<!-- ########################## GIORNI ####################-->

<!-- Lunedi, Martedi, ... -->
<RULE name="nomi-giorni-lex" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>DY *</ARG>
    </REL>
</RULE>

<!-- ########################## PARTE-GIORNO ####################-->

<!-- mattina, mattino, pomeriggio, sera, notte -->
<RULE name="parte-giorno-lex" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>DP *</ARG>
    </REL>
</RULE>

<!-- mattine, mattini, pomeriggi, sere, notti-->
<RULE name="parti-giorni-lex" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>DPP *</ARG>
    </REL>
</RULE>

<!-- colazione, pranzo, merenda, aperitivo, cena -->
<RULE name="ristoro-lex" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>EP *</ARG>
    </REL>
</RULE>

<!-- ########################## MESI ####################-->

<!-- Gennaio, Febbraio, ... -->
<RULE name="nomi-mesi-lex" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>MT *</ARG>
    </REL>
</RULE>

<!-- ########################## STAGIONI ####################-->

<!-- Primavera, estate, autunno, inverno ... -->
<RULE name="nomi-stagioni-lex" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="look-up" type="REF">
```

```
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>SEA *</ARG>
    </REL>
</RULE>
<!-- primaverili, estivi, autunnali, invernali ... -->
<RULE name="agg-nomi-stagioni-lex" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>ASEA *</ARG>
    </REL>
</RULE>

<!-- #########    DATE SINGOLARI e PLURALI ################## -->

<!-- giorno, settimana, mese, anno, weekend, fine settimana
    stagione, bimestre, trimestre, decennio, secolo, millennio -->
<RULE name="date-singolari-lex">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>DU * </ARG>
    </REL>
</RULE>

<!-- giorni, settimane, mesi, anni, weekends, fine settimana
    stagioni, bimestri, trimestri, decenni, secoli, millenni -->
<RULE name="date-plurali-lex">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>DUP *</ARG>
    </REL>
</RULE>

<!-- anno, stagione, bimestre, trimestre, semestre -->
<RULE name="date-singolari-rid-lex">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>DU AN</ARG>
    </REL>
</RULE>

<!-- ##### ULTIMO/ULTIMA/PRIMO/PRIMA/TARDO/TARDA/PRESTO #####-->

<!-- Ultimo, Primo,inizio,fine ... -->
<RULE name="ultimo-primo-lex" type="DISJF">
    <REL  match="W/#~^metà$"></REL>
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>UP *</ARG>
    </REL>
</RULE>

<!-- ##### ULTIMI/ULTIME/PRIMI/PRIME ####################-->

<!-- Ultimi, Primi, ... -->
<RULE name="ultimi-primi-lex" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>UPS *</ARG>
    </REL>
</RULE>
```

```
<!--DOPO/PRIMA/SCORSO(A)/PROSSIMO(A)/PRECEDENTE/SEGUENTE/QUESTO(A)-->

<!-- questo, questa, scorso, scorsa, prossimo, prossima ... -->
<RULE name="dopo-scorso-prossimo-lex">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>BA *</ARG>
    </REL>
</RULE>

<!--#/SCORSI(E)/PROSSIMI(E)/PRECEDENTI/SEGUENTI/QUESTI(E)#-->

<!-- questi, queste, scorsi, scorse, prossimi, prossime ... -->
<RULE name="scorsi-prossimi-lex">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>BAS *</ARG>
    </REL>
</RULE>

<!-- ######### TIME UNITS ################-->

<!-- ora minuto mezzora mezz'ora -->
<RULE name="ora-minuto-lex">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>TU *</ARG>
    </REL>
</RULE>

<!-- ore, minuti, secondi -->
<RULE name="ore-minuti-lex">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>TUP *</ARG>
    </REL>
</RULE>

<!-- mezzogiorno, mezzanotte  -->
<RULE name="mezzogiorno-mezzanotte-lex">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>TN *</ARG>
    </REL>
</RULE>

<!-- ieri, oggi, domani, dopodomani l'altro ieri, ieri l'altro -->
<RULE name="ieri-oggi-domani-lex" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="look-up" type="REF">
        <ARG bind='$1'>&WRD;</ARG>
        <ARG bind='$2'>DD *</ARG>
    </REL>
</RULE>

<!-- ######### QUANT + DATE/TIME UNITS ############################## -->

<!-- 25, twenty-five, three thousand, several thousand, 20-30 ... -->
<RULE name="quant" type="DISJF">
    <REL match="W[C='CD']"></REL>
```

```
    <REL match="PHR[C='CD']"></REL>
    <REL match="PHR[C='QUANT']"></REL>
    <REL match="W[C='QUANT']"></REL>
    <REL match="PHR[C='RANGE']"></REL>
</RULE>

<!-- #########       DECADI ###################################### -->

<!--  '30 -->
<RULE name="decade-apo" targ_sg="TIMEX[TYPE='DATE']" >
    <REL match="W/#~^[']$"></REL>
    <REL match="W/#~^([0-9]0)$"></REL>
</RULE>

<!--  anni '30 -->
<RULE name="decade-sing" targ_sg="TIMEX[TYPE='DATE']" >
    <REL match="W/#~^anni$"></REL>
    <REL match="W/#~^[']$"></REL>
    <REL match="W/#~^[2-9]0$"></REL>
</RULE>

<!--  anni venti, anni ottanta -->
<RULE name="decade-name" targ_sg="TIMEX[TYPE='DATE']" >
    <REL match="W/#~^anni$"></REL>
    <REL match="W/#~(enti|anta)$"></REL>
</RULE>

<!-- anni '30 e '40 -->
<RULE name="decade-decade" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF" match="decade-sing"></REL>
    <REL match="W/#~^e$"></REL>
    <REL match="W/#~^[']$"></REL>
    <REL match="W/#~^[2-9]0$"></REL>
</RULE>

<RULE name="decade" type="DISJF">
    <REL type="REF" match="decade-sing"></REL>
    <REL type="REF" match="decade-name"></REL>
    <REL type="REF" match="decade-apo"></REL>
</RULE>

<!--################## OLD SECOLI ############################-->

<RULE name="acdc" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^(A.C.?|AC|a.c.?|ac|D.C.?|DC|d.c.?|dc)"></REL>
</RULE>

<!-- XX secolo-->
<RULE name="data-secolo" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^[XVI]+"></REL>
    <REL match="W/#~^[Ss]ecolo$"></REL>
    <REL type="REF" match="acdc" m_mod="QUEST"></REL>
</RULE>

<!-- secolo XX-->
<RULE name="secolo-data" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^[Ss]ecolo$"></REL>
    <REL match="W/#~^[XVI]+"></REL>
    <REL type="REF" match="acdc" m_mod="QUEST"></REL>
</RULE>
```

```
<!-- nel/del settecento-->
<RULE name="nel-del-cento" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^([Nn]el|del)$"></REL>
    <REL match="W/#~[a-z]+cento$"></REL>
</RULE>

<RULE name="data-secolo-mix" type="DISJF">
    <REL type="REF" match="data-secolo"></REL>
    <REL type="REF" match="secolo-data"></REL>
    <REL type="REF" match="nel-del-cento"></REL>
</RULE>


<!--################## ORE MINUTI ############################-->


<RULE name="ora-digitale-certa"  targ_sg="TIMEX[TYPE='DATE']" >
    <REL match="W/#~^([1-9]|1[0-9]|2[0-3]|00)$"></REL>
    <REL match="W/#~^[\:]$"></REL>
    <REL match="W/#~^[0-5][0-9]$"></REL>
</RULE>

<!-- 13,45 13:45 13.45-->
<RULE name="ora-digitale"  type="DISJF">
    <REL match="W/#~^([1-9]|1[0-9]|2[0-3]|00)[\.,][0-5][0-9]$"></REL>
    <REL type="GROUP" match="SEQ">
        <REL match="W/#~^([1-9]|1[0-9]|2[0-3]|00)$"></REL>
        <REL match="W/#~^[\:]$"></REL>
        <REL match="W/#~^[0-5][0-9]$"></REL>
    </REL>
    <REL match="W/#~^([1-9]|1[0-9]|2[0-3])$"></REL>
</RULE>

<!-- ore 13,45  ore 13:45-->
<RULE name="ore-ora-digitale" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^ore$"></REL>
    <REL type="REF" match="ora-digitale"></REL>
</RULE>

<RULE name="ore-ora-digitale-mix" type="DISJF" >
    <REL type="REF" match="ore-ora-digitale"></REL>
    <REL type="REF" match="ora-digitale"></REL>
    <REL type="REF"   match="mezzogiorno-mezzanotte-lex"></REL>
</RULE>

<!-- 13.45 o 13.55 #  13.45 alle 17.00 #  13.45 o 17.30 # 13.45-17.30 -->
<RULE  name="ora-ora" type="DISJF" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^ore$"></REL>
    <REL type="GROUP" match="SEQ">
        <REL match="&RANGE;/W[0]/#~^([1-9]|1[0-9]|2[0-3])$"></REL>
        <REL match="&RANGE;/W[1]/#~^[\:]"></REL>
    </REL>
    <REL match="&RANGE;/W[0]/#~^([1-9]|1[0-9]|2[0-3])([,\.][0-5][0-9])?$"></REL>
</RULE>


<!--################## GIORNO DATA MESE ANNO ############################-->
```

```
<!-- 3/12/75 # 3-12-1975 -->
<RULE name="giorno-mese-anno" type="DISJF" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W[C='GMA']"></REL>
</RULE>
<!-- giorno 30-->
<RULE name="namegiorno-data" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^giorno$"></REL>
    <REL match="W/#~^([1-9]|1[0-9]|2[0-9]|3[01])$"></REL>
</RULE>

<!-- giovedi 30-->
<RULE name="giorno-data" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF" match="nomi-giorni-lex"></REL>
    <REL match="W/#~^([1-9]|1[0-9]|2[0-9]|3[01])$"></REL>
</RULE>

<!-- Ottobre 2003 -->
<RULE name="mese-anno" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF" match="nomi-mesi-lex"></REL>
    <REL match="W/#~^([1-2][0-9][0-9][0-9])$"></REL>
</RULE>

<!-- 30 ottobre -->
<RULE name="data-mese" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^([1-9]|1[0-9]|2[0-9]|3[01])$"></REL>
    <REL type="REF" match="nomi-mesi-lex"></REL>
</RULE>

<!--giorno 30 ottobre -->
<RULE name="namegiorno-data-mese" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^giorno$"></REL>
    <REL match="W/#~^([1-9]|1[0-9]|2[0-9]|3[01])$"></REL>
    <REL type="REF" match="nomi-mesi-lex"></REL>
</RULE>

<!--30 Ottobre 2003-->
<RULE name="data-mese-anno" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^([1-9]|1[0-9]|2[0-9]|3[01])$"></REL>
    <REL type="REF" match="mese-anno"></REL>
</RULE>

<!--giorno 30 Ottobre 2003-->
<RULE name="namegiorno-data-mese-anno" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^giorno$"></REL>
    <REL match="W/#~^([1-9]|1[0-9]|2[0-9]|3[01])$"></REL>
    <REL type="REF" match="mese-anno"></REL>
</RULE>

<!-- giovedi 30 Ottobre -->
<RULE name="giorno-data-mese" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF" match="giorno-data"></REL>
    <REL type="REF" match="nomi-mesi-lex"></REL>
</RULE>

<!-- giovedi 30 Ottobre 2003-->
<RULE name="giorno-data-mese-anno" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF" match="giorno-data"></REL>
    <REL type="REF" match="mese-anno"></REL>
</RULE>
```

```
<!-- estate 2003, primavera del 2003, semestre 2003-->
<RULE name="stagione-anno" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="GROUP" match="DISJF">
        <REL type="REF" match="date-singolari-rid-lex"></REL>
        <REL type="REF" match="nomi-stagioni-lex"></REL>
        <REL type="REF" match="nomi-mesi-lex"></REL>
    </REL>
    <REL       match="&OF;" m_mod="QUEST"></REL>
    <REL match="W/#~^([1-2][0-9][0-9][0-9])$"></REL>
</RULE>

<RULE name="giorno-data-mese-anno-stagione-mix" type="DISJF">
    <REL type="REF" match="giorno-mese-anno"></REL>
    <REL type="REF" match="giorno-data-mese-anno"></REL>
    <REL type="REF" match="giorno-data-mese"></REL>
    <REL type="REF" match="namegiorno-data-mese-anno"></REL>
    <REL type="REF" match="data-mese-anno"></REL>
    <REL type="REF" match="namegiorno-data-mese"></REL>
    <REL type="REF" match="data-mese"></REL>
    <REL type="REF" match="mese-anno"></REL>
    <REL type="REF" match="namegiorno-data"></REL>
    <REL type="REF" match="giorno-data"></REL>
    <REL type="REF" match="stagione-anno"></REL>
</RULE>

<RULE name="giorno-mese-stagione" type="DISJF">
    <REL type="REF" match="nomi-giorni-lex"></REL>
    <REL type="REF" match="nomi-mesi-lex"></REL>
    <REL type="REF" match="nomi-stagioni-lex"></REL>
</RULE>

<RULE name="giorno-stagione-combinate-e-non" type="DISJF">
    <REL type="REF" match="giorno-data-mese-anno-stagione-mix"></REL>
    <REL type="REF" match="giorno-mese-stagione"></REL>
</RULE>


<!--######## CD ORD e QUANT prima di DATE-PLURALI-LEX   #######-->


<RULE name="cd-range-ord-quant">
    <REL type="GROUP" match="DISJF">
        <REL match="W[C='CD']"></REL>
        <REL    match="&RANGE;"></REL>
        <REL    match="&QUANT;"></REL>
    </REL>
    <REL match="W/#~^'$" m_mod="QUEST"></REL> <!-- vent'-->
</RULE>


<!--martedi mattina, domani mattina, ieri pomeriggio, domani sera-->
<RULE name="parte-del-giorno" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="GROUP" match="DISJF">
        <REL type="REF"   match="nomi-giorni-lex"></REL>
        <REL type="REF"   match="ieri-oggi-domani-lex"></REL> <!--DD-->
    </REL>
    <REL type="REF"   match="parte-giorno-lex"></REL> <!--DP-->
</RULE>

<RULE name="collezzione-plu" type="DISJF">
```

```
    <REL type="REF"  match="date-plurali-lex"></REL>
    <REL type="REF"  match="parti-giorni-lex"></REL>
    <REL type="REF"   match="ore-minuti-lex"></REL>
</RULE>
<RULE name="collezzione-sing-plu" type="DISJF">
    <REL type="REF"  match="giorno-data-mese-anno-stagione-mix"></REL>
    <REL type="REF"  match="parte-del-giorno"></REL>
    <REL type="REF"  match="data-secolo-mix"></REL>
    <REL type="REF"  match="decade"></REL>
    <REL type="REF"  match="giorno-mese-stagione"></REL>
    <REL type="REF"  match="date-singolari-lex"></REL>
    <REL type="REF"  match="date-plurali-lex"></REL>
    <REL type="REF"  match="parte-giorno-lex"></REL>
    <REL type="REF"  match="parti-giorni-lex"></REL>
    <REL type="REF"  match="ore-minuti-lex"></REL>
    <REL type="REF"  match="ieri-oggi-domani-lex"></REL>
</RULE>


<!-- ######## DOPO/PRIMA/PRECEDENTE(I)/SEGUENTE(I)/SCORSO(AEI)
        PROSSIMO(AEI)/QUESTO(AEI)/FUTURO(I) end match #######-->


<!-- 3 o 4 giorni dopo/prima  3 o 4 ore dopo/prima 2-3 mesi prossimi
    pochi mesi prima  -->
<RULE name="multi-date-plu-prima-dopo" targ_sg="TIMEX[TYPE='DATE']">
    <REL  type="REF" match="cd-range-ord-quant"></REL>
    <REL type="REF" match="collezzione-plu"></REL>
    <REL type="REF" match="scorsi-prossimi-lex"></REL>
</RULE>

<!-- giorni prima, anni dopo, mattino presto,  mese prossimo, secoli prossimi,
    lunedi scorso, aprile scorso, seconda settimana scorsa, 30 ottobre scorso-->
<RULE name="date-sing-plu-prima-dopo" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"  match="collezzione-sing-plu"></REL>
    <REL type="GROUP" match="DISJF">
        <REL type="REF"  match="dopo-scorso-prossimo-lex"></REL>
        <REL type="REF"  match="scorsi-prossimi-lex"></REL>
    </REL>
</RULE>


<!-- ######## start match DOPO/PRIMA/PRECEDENTE(I)/SEGUENTE(I)/SCORSO(AEI)
        PROSSIMO(AEI)/QUESTO(AEI)/FUTURO(I) #######-->


<!--   scorsi 2-3 giorni, prossime due settimane, precedenti 3 o 4 mesi -->
<RULE name="prima-dopo-multi-date-plu" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF" match="scorsi-prossimi-lex"></REL>
    <REL  type="REF" match="cd-range-ord-quant"></REL>
    <REL type="REF"  match="collezzione-plu"></REL>
</RULE>

<!-- questo mese, scorso secolo-->
<RULE name="prima-dopo-date-sing-plu" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="GROUP" match="DISJF">
        <REL type="REF"  match="dopo-scorso-prossimo-lex"></REL>
        <REL type="REF"  match="scorsi-prossimi-lex"></REL>
    </REL>
    <REL type="REF"  match="collezzione-sing-plu"></REL>
```

```
</RULE>


<!--#start match ULTIMO(AEI)/PRIMO(AEI)/TARDO(A)/INIZIO/FINE/MEZZA/META' ###-->


<!--ultimi due anni-->
<RULE name="ultimo-primo-multi-date-plu" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"   match="ultimi-primi-lex"></REL>
    <REL  type="REF" match="cd-range-ord-quant"></REL>
    <REL type="REF"  match="collezzione-plu"></REL>
</RULE>

<!-- primi anni '90 #  seconda settimana # -->
<RULE name="ultimo-primo-date-sing-plu" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="GROUP" match="DISJF">
        <REL       match="&ORD;"></REL>
        <REL type="REF"  match="ultimi-primi-lex"></REL>
        <REL type="REF"  match="ultimo-primo-lex"></REL>
    </REL>
    <REL match="&OF;" m_mod="QUEST"></REL>
    <REL type="GROUP" match="DISJF">
        <REL type="REF"  match="decade"></REL>
        <REL type="REF"  match="collezzione-sing-plu"></REL>
    </REL>

</RULE>

<!-- 2-3 giorni, pochi giorni, 4 giorni -->
<RULE name="multi-comp-semplici">
    <REL  type="REF" match="cd-range-ord-quant"></REL>
    <REL type="REF"  match="collezzione-plu"></REL>
</RULE>

<!--giorni, mesi, settimane anni mattine ore -->
<RULE name="comp-semplici">
    <REL type="REF"  match="collezzione-sing-plu"></REL>
</RULE>


<!--################################################-->


<RULE name="lex-con-avverbi-e-non" type="DISJF" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"   match="prima-dopo-multi-date-plu"></REL>
    <REL type="REF"   match="ultimo-primo-multi-date-plu"></REL>
    <REL type="REF"   match="prima-dopo-date-sing-plu"></REL>
    <REL type="REF"   match="ultimo-primo-date-sing-plu"></REL>
    <REL type="REF"   match="multi-date-plu-prima-dopo"></REL>
    <REL type="REF"   match="multi-comp-semplici"></REL>
    <REL type="REF"   match="date-sing-plu-prima-dopo"></REL>
    <REL type="REF"   match="parte-del-giorno"></REL>
    <REL type="REF"   match="nomi-feste-lex"></REL>
    <REL type="REF"   match="comp-semplici"></REL>
</RULE>

<RULE name="prep-misc">
    <REL type="GROUP" match="DISJF" m_mod="QUEST">
        <REL match="W/#~^più$"></REL>
```

```
        <REL type="REF"   match="misc"></REL>
    </REL>
    <REL type="GROUP" match="DISJF">
        <REL type="REF"   match="preposizioni-lex"></REL>
        <REL       match="&OF;"></REL>
        <REL       match="&AL;"></REL>
        <REL       match="&THE;"></REL>
        <REL       match="&AORONE;"></REL>
    </REL>
    <REL match="&AORONE;" m_mod="QUEST"></REL>
</RULE>

<RULE name="prep-lex-con-avverbi-e-non"  targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"   match="prep-misc"></REL>
    <REL type="REF"   match="lex-con-avverbi-e-non"></REL>
</RULE>

<RULE name="lex-comp-of-al-the-lex-comp" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"   match="lex-con-avverbi-e-non"></REL>
    <REL type="GROUP" match="DISJF">
        <REL       match="&OF;"></REL>
        <REL       match="&AL;"></REL>
        <REL       match="&THE;"></REL>
    </REL>
    <REL type="REF"   match="lex-con-avverbi-e-non"></REL>
</RULE>

<RULE name="prep-lex-comp-of-al-the-lex-comp"  targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"   match="prep-misc"></REL>
    <REL type="REF"   match="lex-comp-of-al-the-lex-comp"></REL>
</RULE>


<!--#####################  PARTE DEL GIORNO ######################-->


<!-- devono avere un apreposizione davanti per avere senso come NEI-->
<!--giorni primaverili, mesi autunnali -->
<RULE name="parte-date-agg-stagioni-plu" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"  match="date-plurali-lex"></REL>
    <REL type="REF"  match="agg-nomi-stagioni-lex"></REL>
</RULE>


<RULE name="parte-giorno-comb-e-non" type="DISJF" >
    <REL type="REF"   match="parte-del-giorno"></REL>
    <REL type="REF"   match="nomi-giorni-lex"></REL>
    <REL type="REF"   match="parte-giorno-lex"></REL>
    <REL type="REF"   match="ieri-oggi-domani-lex"></REL>
</RULE>

<!-- ieri mattina (,) 30 ottobre 2003 # domani giovedi 30 ottobre
    lunedi pomeriggio 30 ottobre -->

<RULE name="parte-del-giorno-data-mix" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"   match="parte-giorno-comb-e-non"></REL>
    <REL       match="W/#~^,$" m_mod="QUEST"></REL>
    <REL type="REF"   match="giorno-data-mese-anno-stagione-mix"></REL>
</RULE>
```

```
<!--(ore) 15.30 di martedi mattina, (ore) 15.30 di domani
    (ore) 15.30 del pomeriggio, mezzanotte di lunedi  15.30 di lunedi-->
<RULE name="ora-parte-giorno-comb-e-non" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"   match="ore-ora-digitale-mix"></REL>
    <REL       match="&OF;"></REL>
    <REL type="REF" match="parte-giorno-comb-e-non"></REL>
</RULE>

<!--(ore) 15.30 del 30 ottobre -->
<RULE name="ora-data-mix" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"   match="ore-ora-digitale-mix"></REL>
    <REL       match="&OF;" ></REL>
    <REL type="REF" match="giorno-data-mese-anno-stagione-mix"></REL>
</RULE>

<!--(ore) 15.30 di martedi mattina 30 otttobre -->
<RULE name="ora-parte-giorno-data-mix"  targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"   match="ore-ora-digitale-mix"></REL>
    <REL       match="&OF;" ></REL>
    <REL type="REF" match="parte-del-giorno-data-mix"></REL>
</RULE>

<!--martedi mattina alle (ore) 15.30, domani alle (ore) 5 del pomeriggio,
     lunedi a mezzanotte-->
<RULE name="parte-giorno-comb-e-non-ora" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF" match="parte-giorno-comb-e-non"></REL>
    <REL       match="&AL;"></REL>
    <REL type="REF" match="ore-ora-digitale-mix"></REL>
    <REL type="GROUP" match="SEQ" m_mod="QUEST">
        <REL       match="&OF;"></REL>
        <REL type="REF"   match="parte-giorno-lex"></REL>
    </REL>
</RULE>

<!--  30 ottobre alle 15,30   30 ottobre alle 5 del pomeriggio-->
<RULE name="data-mix-ora" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF" match="giorno-data-mese-anno-stagione-mix"></REL>
    <REL       match="&AL;"></REL>
    <REL type="REF" match="ore-ora-digitale-mix"></REL>
    <REL type="GROUP" match="SEQ" m_mod="QUEST">
        <REL       match="&OF;"></REL>
        <REL type="REF"   match="parte-giorno-lex"></REL>
    </REL>
</RULE>

<!-- martedi mattina  30 ottobre alle 15,30
    martedi 30 ottobre alle 5 del pomeriggio -->
<RULE name="parte-giorno-data-mix-ora" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF" match="parte-del-giorno-data-mix"></REL>
    <REL       match="&AL;"></REL>
    <REL type="REF" match="ore-ora-digitale-mix"></REL>
    <REL type="GROUP" match="SEQ" m_mod="QUEST">
        <REL       match="&OF;"></REL>
        <REL type="REF"   match="parte-giorno-lex"></REL>
    </REL>
</RULE>

<!--ieri pomeriggio all'ora di merenda, oggi all'ora di pranzo, ora di cena -->
<RULE name="ora-di-pranzo" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"  match="parte-giorno-comb-e-non" m_mod="QUEST" ></REL>
```

```
    <REL       match="&AL;" m_mod="QUEST"></REL>
    <REL       match="W/#~^ora$"></REL>
    <REL       match="&OF;"></REL>
    <REL type="REF"   match="ristoro-lex"></REL>
</RULE>

 <RULE name="ore-della-giornata" type="DISJF">
    <REL type="REF" match="ora-parte-giorno-data-mix"></REL>
    <REL type="REF" match="parte-giorno-data-mix-ora"></REL>
    <REL type="REF" match="parte-giorno-comb-e-non-ora"></REL>
    <REL type="REF" match="data-mix-ora"></REL>
    <REL type="REF" match="ora-parte-giorno-comb-e-non"></REL>
    <REL type="REF" match="ora-data-mix"></REL>
    <REL type="REF" match="ore-ora-digitale"></REL>
    <REL type="REF" match="ora-digitale-certa"></REL>
    <REL type="REF"  match="mezzogiorno-mezzanotte-lex"></REL>
</RULE>

<RULE name="prep-lex-ore-della-giornata" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"   match="prep-misc"></REL>
    <REL type="REF"  match="ore-della-giornata"></REL>
</RULE>


<!--####################  TRA/FRA/DAL ######################-->


<RULE  name="tra-fra-time" arg="$3 $4">
    <REL match="W/#~^([Tt]ra|[Ff]ra)$"></REL>
    <REL match="W/#~^(il?|le)$" m_mod="QUEST"></REL>
    <REL type="REF" match="$3"></REL>
    <REL match="W/#~^(ed?|o)$"></REL>
    <REL match="W/#~^(il|le)$" m_mod="QUEST"></REL>
    <REL type="REF" match="$4"></REL>
</RULE>

<!-- fra lunedi e martedi-->
<RULE  name="tra-fra-max" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="tra-fra-time" type="REF">
        <ARG bind='$3'>lex-comp-of-al-the-lex-comp</ARG>
        <ARG bind='$4'>lex-comp-of-al-the-lex-comp</ARG>
    </REL>
</RULE>

<!-- fra lunedi e martedi-->
<RULE  name="tra-fra-med" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="tra-fra-time" type="REF">
        <ARG bind='$3'>lex-con-avverbi-e-non</ARG>
        <ARG bind='$4'>lex-con-avverbi-e-non</ARG>
    </REL>
</RULE>

<!-- fra lunedi e martedi-->
<RULE  name="tra-fra-ore-della-giornata" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="tra-fra-time" type="REF">
        <ARG bind='$3'>ore-della-giornata</ARG>
        <ARG bind='$4'>ore-della-giornata</ARG>
    </REL>
</RULE>
```

```
<RULE  name="tra-fra-tutte-date-possibili" type="DISJF" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF" match="tra-fra-max"></REL>
    <REL type="REF" match="tra-fra-med"></REL>
    <REL type="REF" match="tra-fra-ore-della-giornata"></REL>
</RULE>

<RULE  name="cd-ed-il-cd">
    <REL match="W/#~^(il?|le)$" m_mod="QUEST"></REL>
    <REL type="GROUP" match="DISJF">
        <REL match="&CARD;"></REL>
        <REL      match="W/#~^[XVI]+"></REL>
    </REL>
    <REL match="W/#~^(ed?|od?)$"></REL>
    <REL match="W/#~^(il?|le)$" m_mod="QUEST"></REL>
    <REL type="GROUP" match="DISJF">
        <REL match="&CARD;"></REL>
        <REL      match="W/#~^[XVI]+"></REL>
    </REL>
</RULE>

<!-- fra i 2 ed i 3 ore/minuti/secondi, fra le 2 e le 3 ore/minuti
    fra 2 o 3 ore, fra 2-3 ore-->
<RULE name="tra-fra-range" targ_sg="TIMEX[TYPE='DATE']">
    <REL      match="W/#~^([Tt]ra|[Ff]ra)$"></REL>
    <REL type="GROUP" match="DISJF">
        <REL type="REF"  match="cd-ed-il-cd"></REL>
        <REL type="GROUP" match="SEQ">
            <REL match="W/#~^(il?|le)$" m_mod="QUEST"></REL>
            <REL       match="&RANGE;"></REL>
        </REL>

    </REL>
    <REL type="REF"  match="lex-con-avverbi-e-non" m_mod="QUEST"></REL>
</RULE>

<!--fra gli anni '30 e '40-->
<RULE name="tra-fra-decade-decade" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^([Ff]ra|[Tt]ra)$"></REL>
    <REL match="W/#~^gli"></REL>
    <REL type="REF"  match="decade-decade"></REL>
</RULE>


<!-- ######################## DA-A ############################-->


<RULE  name="da-a-time" arg="$5 $6">
    <REL match="W/#~^([Dd]alle|[Dd]al|[Dd]a)$"></REL>
    <REL type="REF" match="$5"></REL>
    <REL match="&AL;"></REL>
    <REL type="REF" match="$6"></REL>
</RULE>

<!-- da lunedi a martedi-->
<RULE  name="da-a-max" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="da-a-time" type="REF">
        <ARG bind='$5'>lex-comp-of-al-the-lex-comp</ARG>
        <ARG bind='$6'>lex-comp-of-al-the-lex-comp</ARG>
    </REL>
</RULE>
```

```
<!-- da lunedi e martedi-->
<RULE  name="da-a-med" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="da-a-time" type="REF">
        <ARG bind='$5'>lex-con-avverbi-e-non</ARG>
        <ARG bind='$6'>lex-con-avverbi-e-non</ARG>
    </REL>
</RULE>

<!-- da lunedi a martedi-->
<RULE  name="da-a-ore-della-giornata" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="da-a-time" type="REF">
        <ARG bind='$5'>ore-della-giornata</ARG>
        <ARG bind='$6'>ore-della-giornata</ARG>
    </REL>
</RULE>


<!-- da lunedi a martedi-->
<RULE  name="da-a-ore-ora-digitale-mix" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="da-a-time" type="REF">
        <ARG bind='$5'>ore-ora-digitale-mix</ARG>
        <ARG bind='$6'>ore-ora-digitale-mix</ARG>
    </REL>
</RULE>

<RULE  name="da-a-tutte-date-possibili" type="DISJF" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF" match="da-a-max"></REL>
    <REL type="REF" match="da-a-med"></REL>
    <REL type="REF" match="da-a-ore-della-giornata"></REL>
    <REL type="REF" match="da-a-ore-ora-digitale-mix"></REL>
</RULE>

<!-- dalle 12.30 alle 17.30-->
<RULE name="da-a-range" targ_sg="TIMEX[TYPE='DATE']">
     <REL match="W/#~^([Dd]alle|[Dd]al|[Dd]a)$"></REL>
    <REL      match="&RANGE;"></REL>
    <REL type="REF"  match="lex-con-avverbi-e-non" m_mod="QUEST"></REL>
</RULE>

<!-- dal 1989 al 1993, 2000/03 -->
<RULE  name="range-anno" type="DISJF" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="GROUP" match="SEQ">
        <REL match="W/#~^([Dd]al)$" m_mod="QUEST"></REL>
        <REL       match="PHR[C='YRRANGE']"></REL>
    </REL>
    <REL match="W/#~^[12][0-9][0-9][0-9][/][0-9][0-9]$"></REL>
</RULE>

<RULE name="fino-alla-fine" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^[Ff]ino$" ></REL>
    <REL match="W/#~^alla$" ></REL>
    <REL match="W/#~^fine$" ></REL>
    <REL       match="&OF;"></REL>
    <REL type="GROUP" match="DISJF">
        <REL type="REF" match="lex-con-avverbi-e-non"></REL>
        <REL match="W/#~^([1-2][0-9][0-9][0-9])$"></REL>
    </REL>
</RULE>
```

```
<RULE name="più-tardi" targ_sg="TIMEX[TYPE='DATE']">
    <REL match="W/#~^più$" ></REL>
    <REL match="W/#~^tardi$" ></REL>
</RULE>

<!--per le 16:30, entro il 2004 -->
<RULE name="prep-lex-ore-anni" targ_sg="TIMEX[TYPE='DATE']">
    <REL type="REF"  match="scorsi-prossimi-lex" m_mod="QUEST"></REL>
    <REL type="REF"   match="prep-misc"></REL>
    <REL type="GROUP" match="DISJF">
        <REL match="W/#~^([1-2][0-9][0-9][0-9])$"></REL>
        <REL type="REF"   match="ore-ora-digitale-mix"></REL>
    </REL>
</RULE>


<!-- ##### TUTTE POSSIBILI DATE ################## -->


<RULE name="all" type="DISJF">
    <REL type="REF" match="tra-fra-tutte-date-possibili"></REL>
    <REL type="REF" match="tra-fra-range"></REL>
    <REL type="REF" match="tra-fra-decade-decade"></REL>
    <REL type="REF" match="da-a-tutte-date-possibili"></REL>
    <REL type="REF" match="da-a-range"></REL>
    <REL type="REF" match="prep-lex-comp-of-al-the-lex-comp"></REL>
    <REL type="REF" match="lex-comp-of-al-the-lex-comp"></REL>
    <REL type="REF" match="prep-lex-ore-della-giornata"></REL>
    <REL type="REF" match="ore-della-giornata"></REL>
    <REL type="REF" match="fino-alla-fine"></REL>
    <REL type="REF" match="prep-lex-con-avverbi-e-non"></REL>
    <REL type="REF" match="ora-di-pranzo"></REL>
    <REL type="REF" match="lex-con-avverbi-e-non"></REL>
    <REL type="REF" match="range-anno"></REL>
    <REL type="REF" match="più-tardi"></REL>
    <REL type="REF" match="prep-lex-ore-anni"></REL>
</RULE>

</RULES>
```

Appendice B
Lessico esterno

Di seguito sono riportati i tre lessici esterni utilizzati dai file.gr descritti precedentemente. I nomi dei TAG utilizzati non seguono una particolare logica: alcuni sono abbreviazioni di parole italiane ed inglesi, altri sono anacronismi. Possono essere modificati in qualsiasi momento per rendere la comprensione più immediata.

B.1 Lessico numbers2.lex

```
cento       BIG-UNITS
mille       BIG-UNITS
milion      BIG-UNITS
miliard     BIG-UNITS
biliard     BIG-UNITS
triliard    BIG-UNITS

dozzin        QUANT
decin         QUANT
ventin        QUANT
trentin       QUANT
quarantin     QUANT
cinquantin    QUANT
sessantin     QUANT
settantin     QUANT
ottantin      QUANT
novantin      QUANT
centinai      QUANT
migliai       QUANT

poche     AGG
alcune    AGG
tante     AGG
molte     AGG
pochi     AGG
alcuni    AGG
tanti     AGG
molti     AGG
```

oltre AGG
circa AGG

B.2 Lessico numex2.lex

$ $-
$US $-
$ US :: $-
US$ $-
US $ :: $-
U.S.$ $-
U.S. $ :: $-
$US $-
$us $-
UK# $-
UK # :: $-
uk# $-
uk # :: $-
Mex$ $-
Mex.$ $-
Mex $ :: $-
Mex. $ :: $-

dollaro $unit
dollari $unit
dracma $unit
drachmas $unit
ecu $unit
ecus $unit
ECU $unit
ECUs $unit
Ecu $unit
Ecus $unit
escudo $unit
escudi $unit
euro $unit
fiorino $unit
fiorini $unit
franco $unit
franchi $unit
lire $unit
lira $unit
peseta $unit
pesetas $unit
peso $unit
pesos $unit
rupia $unit
rupie $unit
yen $unit

centesimi $c
centesimo $c
penny $c
pence $c
piastra $c
piastre $c

B.3 Lessico timex2.lex

```
Domenica        DY
Lunedì          DY
Martedì         DY
Mercoledì       DY
giovedì         DY
venerdi         DY
sabato          DY

gennaio         MT MTY
febbraio        MT MTY
marzo           MT MTN
aprile          MT MTN
maggio          MT MTN
giugno          MT MTN
luglio          MT MTY
agosto          MT MTN
settembre       MT MTY
ottobre         MT MTY
novembre        MT MTY
dicembre        MT MTY


minuto          TU
ora             TU
mezzora         TU
mezz ' ora ::   TU

secondi         TUP
minuti          TUP
ore             TUP

mezzogiorno     TN
mezzanotte      TN

colazione       EP
pranzo          EP
merenda         EP
aperitivo       EP
cena            EP

oggi            DD
ieri            DD
domani          DD
dopodomani      DD
l' altro ieri ::    DD
ieri l' altro ::    DD

primavera       SEA
estate          SEA
autunno         SEA
inverno         SEA
```

primaverili	ASEA
estivi	ASEA
autunnali	ASEA
invernali	ASEA
mattina	DP
mattino	DP
pomeriggio	DP
sera	DP
notte	DP
mattinata	DP
serata	DP
nottata	DP
giornata	DP
giorno	DP
mattine	DPP
pomeriggi	DPP
sere	DPP
notti	DPP
mattinate	DPP
serate	DPP
nottate	DPP
giornate	DPP
giorni	DPP
ultimi	UPS
ultime	UPS
primi	UPS
prime	UPS
tardi	UPS
ultimo	UP
ultima	UP
primo	UP
prima	UP
tardo	UP
tarda	UP
inizio	UP
fine	UP
mezza	UP
dopo	BA
prima	BA
precedente	BA
seguente	BA
scorsa	BA
scorso	BA
prossimo	BA
prossima	BA
questo	BA
questa	BA
presto	BA
futuro	BA
fa	BA
dopo	BAS
prima	BAS
precedenti	BAS

seguenti BAS
scorse BAS
scorsi BAS
prossimi BAS
prossime BAS
questi BAS
queste BAS
futuri BAS
fonda BAS

bimestre DU AN
stagione DU AN
anno DU AN
giorno DU
weekend DU
fine settimana :: DU
settimana DU
mese DU
decennio DU
secolo DU
millennio DU

trimestre DU AN
semestre DU AN

giorni DUP
weekends DUP
fine settimana :: DUP
settimane DUP
mesi DUP
stagioni DUP
bimestri DUP
trimestri DUP
anni DUP
decenni DUP
secoli DUP
millenni DUP

della mattina :: ORA
del mattino :: ORA
del pomeriggio :: ORA
della sera : ORA
di notte :: ORA

nel PRP
in PRP
nell' PRP
nella PRP
negli PRP
nei PRP
per tutto :: PRP
il PRP
al PRP
per PRP
di PRP
sul PRP

meno MISC
durante MISC
nell' arco :: MISC

fino MISC
entro MISC
per tutto :: MISC
per MISC

Natale HOL
Epifania HOL
Pasqua HOL
Pentecoste HOL
Avvento HOL
tutti i santi HOL
Ceneri HOL
martedi grasso HOL
festa del papà HOL
festa della mamma HOL
anno nuovo :: HOL
Santa Lucia :: HOL
San Valentino :: HOL
festa delle donne :: HOL
festa della donna :: HOL

Bibliografia

[1] C. Grover, C. Matheson, A. Mikheev, ”*Text Tokenisation Tool – A Flexibe Tokenisation Tool*” http://www.ltg.ed.ac.uk/

[2] H. Deitel, P. Deitel, T. Nieto, T. Lin, P. Sadhu, “*XML Corso di programmazione*“ 2002 APOGEO srl

[3] F. Grandi, F. Mandreoli “*Codifica XML e Gestione di Informazione Temporale in fonti storiche digitalizzate di grandi dimensioni*” http://www-db.deis.unibo.it/~fmandreoli/

[4] “*The semantic Web Agreement Group Home Page*” http://swag.semanticweb.org.