Linguaggi di Markup

Violetta Lonati [a]

Dipartimento Scienze dell'Informazione

Università degli Studi di Milano

Linguaggi di Markup

- Sono usati per la descrizione di linguaggi.
- Le parole sono costituite da interi documenti corredati da marcatori, detti tag, che forniscono informazioni su porzioni di testo.
- Esempi: HTML, XML

HyperText Markup Language

L'HTML serve per presentare documenti sul World Wide Web.

- formattazione del testo (veste grafica)
- creazione di collegamenti tra diversi documenti
- Il browser interpreta i tag dell'HTML e visualizza il testo di conseguenza.

HTML - esempi di tag

tag html	cosa caratterizza
<html> ... </html>	documento HTML
<body> ... </body>	parte visibile
...	lista ordinata
...	elemento della lista
...	testo in grassetto
<u> ... </u>	testo sottolineato
...	testo in corsivo

HTML - correttezza

- Non tutti i documenti formati da testo e tag sono descrizioni HTML corrette.
- Le descrizioni HTML corrette possono essere formalmente generate attraverso una grammatica acontestuale.

HTML - grammatica

Porzione di grammatica per generare un elenco numerato:

- Elenco-num $\longrightarrow$ `<ol>` Lista `</ol>`
- Lista $\longrightarrow \epsilon$ | Elemento Lista
- Elemento $\longrightarrow$ `<li>` Testo `</li>`
- Testo $\longrightarrow \epsilon$ | Carattere Testo
- Carattere $\longrightarrow$ `a`|`b`| ... | `A`|`B`|...

HTML - grammatica

Nelle grammatica precedente abbiamo:

- Metasimboli
  Elenco-num, Lista, Elemento, Testo, Carattere.
- Simboli Terminali
  `<ol>`, `</ol>`, `<li>`, `</li>`, a,b, ... , A,B...
- Concatenazione
  indicata dallo spazio.

XML - eXtensible Markup Language

- L'autore ha la possibilità di inventare nuovi tag
- I tag costituiscono delle coppie di parentesi: `<nome-tag>` e' una parentesi aperta `</nome-tag>` e' la corrispondente parentesi chiusa
- Le parentesi (tag) sono potenzialmente infinite
- Una parola XML e' sostanzialmente un testo inserito in una sequenza di parentesizzazioni ben formate.

XML - doc ben formati

Un documento XML si dice ben formato se:

- esiste un tag che contiene tutto il documento
  `<S>` tutto il documento `</S>`
- ogni tag aperto `<x>` è seguito dal corrispondente tag chiuso `</x>`
- i tag sono annidati correttamente:
  se `<x>` è aperta prima di `<y>`
  allora `</x>` va chiusa dopo `</y>`

XML - esempi

I seguenti documenti sono ben formati?

- `<a></a><b></b>` NO
- `<a><b></a>` NO
- `<a><b></a></b>` NO
- `<a><b></b></a>` SI
- `<b><a><b></b></a></b>` SI

XML - linguaggio acontestuale?

- L'insieme dei linguaggi XML ben formati non costituisce un linguaggio acontestuale, perchè i potenziali tipi di parentesi sono infiniti.
- Fissando a priori un insieme finito di possibili parentesi, il linguaggio dei documenti ben formati contententi solo quelle parentesi risulta essere acontestuale.

XML e DTD

Le grammatiche acontestuali possono essere utilizzate per specificare sottoclassi di documenti XML ben formati.

- Si usa una opportuna notazione per definire una grammatica acontestuale, che viene detta DTD (Document Type Definition)
- Un documento XML si dice valido se è generato dalla DTD
- Esistono programmi per la validazione automatica

Notazione usata nelle DTD

Possibili forme per le regole di produzione nella DTD:

```
<!ELEMENT tag (tag1,...,tagN)>
<!ELEMENT tag (#PCDATA)>
```

Le regole precedenti corrispondono, secondo la nostra notazione, alle produzioni:

Esempio: e-mail (1)

Esempio: usiamo XML per caratterizzare dei messaggi di posta elettronica.

- Fissiamo una DTD che definisca la struttura di un e-mail. Per semplicità consideriamo gli elementi `from`, `to`, `subject`, `body`
- Traduciamo la DTD nella notazione standard delle produzioni

Esempio: e-mail (2)

Esempio di DTD per messaggi di posta elettronica:

```
<!ELEMENT mail
(from,to,subject,body)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

Esempio: e-mail (3)

Traduzione delle produzioni nella DTD secondo la notazione standard:

mail $\rightarrow$ `<mail>` from to subject body `</mail>`
from $\rightarrow$ `<from>`Testo`</from>`
to $\rightarrow$ `<to>`Testo`</to>`
subject $\rightarrow$ `<subject>`Testo`</subject>`
body$\rightarrow$ `<body>`Testo`</body>`
Testo $\rightarrow \epsilon$ | Carattere Testo
Carattere $\rightarrow$ `a`|`b`| ... | `A`|`B`|...

Esempio: e-mail (4)

Esempio di file XML valido rispetto alla precedente DTD

```
<?xml version="1.0"?>

<mail>
<from>Alessandra</from>
<to>Fabrizio</to>
<subject>cinema</subject>
<body>Che film andiamo a vedere
questa sera?</body>
</mail>
```

Espressioni regolari nelle DTD

Nella Document Type Definition è possibile anche utilizzare espressioni regolari.

- Sono ammesse produzioni del tipo
  `<!ELEMENT tag (descr)>`
  dove `descr` può contenere sia i metasimboli che gli operatori | * + ? con l'usuale significato visto per le regex in `UNIX`.
- E' possibile tradurre anche queste produzioni più complesse usando un insieme finito di produzioni nella nostra notazione standard.

Esempio di regex nelle DTD

```
<!ELEMENT to (Destinatario+)>
<!ELEMENT Destinatario (#PCDATA)>
```

corrispondono alle produzioni standard

to $\longrightarrow$ `<to>` Destinatari `</to>`
Destinatari $\longrightarrow$ Destinatario
Destinatari $\longrightarrow$ Destinatario Destinatari
Destinatario $\longrightarrow$ Testo
Testo ...