

Linuxdoc Reference

Un'introduzione al dtd di linuxdoc

Uwe Böhme, <uwe@hof.baynet.de>

v1.1, 30 Gennaio 2000

Questo testo è destinato ad essere una guida di riferimento per la definizione del tipo di documento (dtd) di SGML linuxdoc, fornito insieme al sistema di formattazione testi SGML versione 1.0. Dovrebbe essere valido anche per versioni successive, che possono essere trovate presso *la mia homepage* <<http://www.hof.baynet.de/~uwe>>. Traduzione e adattamenti in italiano a cura di Hugh Hartmann hhartmann@libero.it e Vieri Giugni v.giugni@gmail.com, revisione a cura di Elisabetta Galli lab@kkk.it

Contents

1 Making of	4
1.1 Aspetti Legali	4
1.2 Genesi	4
2 Introduzione	5
3 Un documento minimale	5
3.1 Passo dopo passo	5
3.2 Un documento per iniziare	6
4 Classi di documento	7
4.1 Il tag Article	8
4.1.1 Il tag Titlepage	8
4.1.2 Il tag Header	9
4.1.3 Il tag Table Of Contents	9
4.1.4 Il tag List Of Figures	10
4.1.5 Il tag List Of Tables	10
4.1.6 Body	10
4.1.7 Il Tag Appendix	10
4.1.8 Il tag Bibliography	10
4.1.9 Il tag Footnote	10
4.2 Il tag Report	10
4.3 Il tag Book	11
4.4 Il tag Letter	11
4.5 Il tag Telefax	11
4.6 Il tag Slides	12
4.6.1 Il tag Slide	12

4.7	Il tag Note	12
4.8	Il tag Manual Page	12
5	Inline	12
6	Creare le sezioni	13
7	Paragrafi	14
7.1	Paragrafo normale	14
7.1.1	Tag di paragrafo	14
7.1.2	Nuova riga vuota	15
7.2	Paragrafi stile elenco	15
7.2.1	Il tag List	15
7.2.2	Il tag Itemize	15
7.2.3	Il tag Enum	16
7.2.4	Il tag Descrip	16
7.3	Illustrazioni e Tabelle	17
7.3.1	Il tag Table	17
7.3.2	Il tag Figure	18
7.4	Il tag Tabular	19
7.5	Paragrafi matematici	21
7.5.1	Il tag Displayed Formula	21
7.5.2	Il tag Equation	21
7.6	Paragrafo Teorema	21
7.7	Paragrafi Code e Verbatim	22
7.7.1	Il tag Code	23
7.7.2	Il tag Verbatim	23
8	I tag Inline	24
8.1	Enfatizzazione	24
8.2	Il tag Short-quote	24
8.3	Il tag Formula	25
8.4	Il tag External	25
9	Formule matematiche	25
9.1	Il tag Fraction	25
9.2	I tag Product, Integral e Summation	26
9.3	Il tag Limited	26
9.4	Il tag Array	27

9.5	Il tag Root	27
9.6	Il tag Figure	28
9.7	Il tag Realfont	28
9.8	Altri tag matematici	28
10	Etichette e riferimenti	28
10.1	Il tag Label	29
10.2	Il tag Reference	29
10.3	Il tag Page Reference	29
10.4	Il tag Url	29
10.5	Il tag Htmlurl	30
10.6	Il tag Cite	30
10.7	Il tag Ncite	30
11	Indici	30
11.1	Includere un indice analitico	31
11.1.1	Manualmente	31
11.1.2	Hacked	31
12	Programmazione istruita	31
13	Bibliografia	32
A	Simboli nominati	32
A.1	Caratteri nominati	32
A.2	Spazi bianchi nominati	32
B	Figure matematiche	33
C	Il dtd di Linuxdoc per il sorgente	34

List of Figures

1	Inclusione di un file encapsulated postscript TM	19
2	Uno spazio vuoto per incollare una fotografia	19

List of Tables

1	Mappatori di SGML-Tools per documenti sgml	6
2	Classi di documento	8
3	Tag in una lettera	11

4	Posizioni della tabella	17
5	Una tabella di esempio	18
6	Allineamento di colonna	20
7	Esempio di tabella per il tag Tabular	20
8	Posizioni delle formule matematiche	25
9	Tag con limite superiore, limite inferiore e operatore	26
10	Tag matematici senza tag inclusi	28
11	Elementi indice analitico	31
12	Caratteri nominati	33
13	Figure matematiche	33

1 Making of

1.1 Aspetti Legali

Copyright © 1997-2000 by Uwe Böhme. This document may be distributed under the terms set forth in the Linux Documentation Project License at *LDP* <<http://metalab.unc.edu/LDP/COPYRIGHT.html>>. Please contact the authors if you are unable to get the license. This is free documentation. It is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose.

Questo documento non fa parte di *ldp* (anche se è stato usato il loro tipo di licenza). Ancora non ne faccio parte.

1.2 Genesi

Questo documento nasce dal tentativo di imparare di più sulla scrittura di testi su un sistema Linux. L'unico sistema che sembrava adatto alle mie necessità era *sgml-tools* *l'organizzazione SGML-Tools* <<http://www.sgml-tools.org>> e il *linuxdoc* dtd.

Nella [*SGML-Tools Guida dell'Utente 1.0 (\$Revision: 1.1 \$)*] (vedere la sezione 13 (Riferimenti)) la struttura generale è descritta in modo facile e piacevole. Anche [*Esempi rapidi di SGML, v1.0*] (vedere la sezione 13 (Riferimenti)) è stata utile, **ma**:

Molte caratteristiche non sono menzionate.

Cercando di imparare di più, ho incontrato la [*Definizione del tipo di documento qwertz*] (vedere la sezione 13 (Riferimenti)). È dettagliata come speravo, ma non è pensata per il di *linuxdoc* (anche se *linuxdoc* è basato su *qwertz*).

Ho tentato un nuovo approccio: osservare il file stesso dtd¹, e tentare di comprenderlo.

Con il passare del tempo ho notato di essermi dimenticato alcune cose o, almeno, di non averle fatte notare abbastanza. Questo cambierà con la prossima revisione.

Ogni commento sarà il benvenuto (in particolare ogni aiuto con l'ortografia o la grammatica inglese) via e-mail, all'indirizzo *Uwe Böhme* <<mailto:uwe@hof.baynet.de>>.

¹dtd = definizione del tipo di documento

2 Introduzione

Il principio di ogni documento sgml (linuxdoc, docbook, html) è più o meno lo stesso:

Non scrivere come dovrebbe *apparire*, ma scrivere cos'è.

Questo è un approccio diverso rispetto allo standard "wysiwyg"² one³. Non si dice al programma che questa riga dovrebbe essere scritta in un carattere più grande, **per apparire come** un titolo. Ciò che si fa è dire che questa riga è un titolo. Non si tenta di far apparire il proprio documento **come se fosse** un report, ma lo si etichetta **per essere** un report. Così si *etichetta* il testo con il <tag> adatto.

I grandi vantaggi di questo approccio sono:

1. Non è necessario pasticciare con impostazione dei font, interlinee o altre cose connesse direttamente al layout
2. Si descrive il proprio documento in modo più astratto, così che sia maggiormente riutilizzabile e possa essere mappato in diversi tipi di supporti.⁴

In aggiunta, in tutti i documenti in stile sgml si potranno trovare i simboli nominati. Questo è un concetto di espansione del set di caratteri del documento per evitare inconsistenze nelle decisioni dell'analizzatore, nell'interpretazione o mappatura di alcuni caratteri speciali.

Come potrebbe l'analizzatore sapere se un carattere < indica l'inizio di un tag o se deve essere stampato direttamente? Questo problema viene risolto dal carattere nominato *lt*. Scrivendo `<`; si otterrà < nel proprio testo. Per un elenco dei simboli nominati vedere A (Simboli nominati).

Suggerimento per il nuovo utente

Potrebbe essere una buona idea scaricare questo documento non solo come documento dvi o ps, ma scaricare anche il sorgente sgml. Questo darà la possibilità di osservare i sorgenti: si potrebbe trovare in questo articolo qualcosa di adatto alle proprie necessità.

3 Un documento minimale

In questa sezione si troverà ciò di cui si avrà bisogno per creare un documento minimale conforme al dtd di linuxdoc. Vuole essere un primo contatto. Si può saltare questa sezione, se si conoscono già i fondamentali.

3.1 Passo dopo passo

I passi da fare per creare un grazioso documento linuxdoc e mapparli nella forma desiderata sono:

- Scegliere un semplice editor di testo a piacimento.
- Creare un file e dargli un nome (o salvarlo successivamente), per esempio `iniziare.sgml`.
- Scrivere il documento

²Quello che si vede è quello che (forse) si ottiene (se si ha molta fortuna e il proprio computer vince la guerra contro il software bacato)

³Si potrebbe chiamarlo per esempio `wysiwym`, cioè "What you see is what you mean" (quello che si vede è quello che significa)

⁴Se si è già tentato di riutilizzare un documento scritto in un layout wysiwy specializzato in html allora si sa già di cosa si sta parlando.

- Salvare il file e chiudere l'editor.
- Eseguire il correttore digitando `sgmlcheck iniziare.sgml`.
- Se vengono restituiti degli errori, riaprire nuovamente il documento nell'editor e provare a correggerlo.⁵
- Eseguire nuovamente il correttore finché non si otterranno più errori.
- Ora si deve decidere a cosa serve il documento. Prendere la corretta combinazione di mappatore e analizzatore e tradurre il documento. Per trovare i mappatori disponibili negli "SGML-Tools", vedere la tabella 1 (Mappatori di SGML-Tools per documenti sgml).

digitare	per ottenere
<code>sgml2html start.sgml</code>	Linguaggio di markup di ipertesti per browser web
<code>sgml2lyx start.sgml</code>	Formato di testo Lyx o KLyx WYSIWYG
<code>sgml2info start.sgml</code>	Pagina info per UN*X info
<code>sgml2latex start.sgml</code>	Output DVI
<code>sgml2latex -output=tex start.sgml</code>	Output in puro tex
<code>sgml2latex -output=ps start.sgml</code>	Output postscript
<code>sgml2rtf start.sgml</code>	Formato rich text
<code>sgml2txt start.sgml</code>	Puro testo

Table 1: Mappatori di SGML-Tools per documenti sgml

3.2 Un documento per iniziare

Si inizierà con un semplice documento (i numeri e i due punti all'inizio della riga sono solo esplicativi, non vanno digitati):

```

1: <!doctype linuxdoc system>
2: <notes>
3: <title>Un piccolo esempio di Linuxdoc</title>
4: <p>Ciao <em>mondo</em>.</p>
5: <p><bf>Ci</bf> siamo.</p>
6: </notes>

```

Ora si darà un'occhiata alle singole righe:

1. Un documento linuxdoc deve iniziare, come tutti i documenti conformi allo SGML, con il *preambolo*. Volendo si può considerarlo come un necessario pezzo di magia, o si può provare a trovare maggiori informazioni riguardo SGML. Il preambolo indica all'analizzatore SGML quale dtd (definizione del tipo di documento) usare per verificare la sintassi del documento.
2. Si apra la *classe di documento*: si deve decidere quale tipo di documento si vuole scrivere. Vedere la sezione 4 (Classi di documento) per una descrizione dettagliata riguardo le *classi di documento*. Le necessarie informazioni di intestazione, che dipendono dalla *classe di documento*, vengono spiegate in questa sezione. Nel nostro caso useremo un tag `<notes>`, che crea una nota indicante un semplice documento non strutturato.

⁵I messaggi di errore di `sgmlcheck` daranno un suggerimento sul tipo di errore, insieme alla riga e alla colonna in cui è avvenuto.

3. Anche se non è indispensabile, è una buona idea fornire un *titolo* al documento. Per farlo si usa il tag `<title>`.
4. Un paragrafo marcato con il tag `<p>`, contenente la parola *mondo* che è testo *inline enfatizzato* dal tag ``.
5. Un altro paragrafo totalmente marcato, con un'altra parola *inline* e resa in *grassetto* tramite il tag `<bf>`.
6. Qui viene chiuso il tag *classe di documento* aperto.

Lo stesso esempio può essere scritto più brevemente eliminando i tag automaticamente posizionati dall'analizzatore, e utilizzando tag abbreviati:

```
1: <!doctype linuxdoc system>
2: <notes>
3: <title>Un piccolo esempio Linuxdoc
4: <p>Ciao <em/mondo/.
5:
6: <Ci bf/siamo/.
7: </notes>
```

Ora rivediamo ogni singola riga:

1. Il *preambolo*.
2. La classe del documento (anch'essa invariata).
3. Il *titolo*. Non è chiuso, perchè il tag `p` nella riga successiva lo chiude implicitamente.
4. Il paragrafo chiude implicitamente il *titolo*. Il tag *emphasize* è indicato in forma abbreviata. La notazione breve si può utilizzare solo se il testo nel tag non contiene il carattere `/`. Il *paragrafo* non è chiuso esplicitamente in questa riga.
5. La riga vuota in questo punto è il motivo per cui non è necessario chiudere il precedente *paragrafo* e aprire il prossimo. Una riga vuota viene interpretata come la fine del paragrafo attuale, e l'inizio di uno nuovo.
6. Un altro paragrafo (non aperto direttamente), con un altro tag abbreviato *inline*.
7. Chiusura del tag *classe di documento* aperto, che a sua volta chiude implicitamente anche il paragrafo ancora aperto.

Ora lavorare con i tag dovrebbe essere un pò più chiaro.

4 Classi di documento

```
<!element linuxdoc o o
    (sect | chapt | article | report |
     book | letter | telefax | slides | notes | manpage ) >
```

Descrive complessivamente la classe del documento, e naturalmente (lasciando stare la definizione doctype) deve essere il primo tag e racchiudere l'intero documento. Alcuni dei tag, cioè `sect` e `chapt` (vedere la sezione 6 (Creare le sezioni)) non hanno alcun senso presi singolarmente, nonostante siano inclusi come parti

di classi di documento più, complete: dunque saranno descritti successivamente come parte di altre classi di documento. È meglio decidere prima quale delle classi di documento menzionate sopra è adatta al tipo di documento che si vuole scrivere.

Per una dettagliata descrizione delle classi di documento vedere la tabella 2 (Classi di documento).

Capitolo	Tag della classe
4.1 (Tag per l'articolo)	<article>
4.2 (Tag per il report)	<report>
4.3 (Tag per il libro)	<book>
4.4 (Tag per la lettera)	<letter>
4.5 (Tag per il telefax)	<telefax>
4.6 (Tag per le slide)	<slides>
4.7 (Tag per le note)	<notes>
4.8 (Tag per le pagine di manuale)	<manpage>

Table 2: Classi di documento

Per me la *classe article* è la più importante, ecco perché è descritta per prima e molto dettagliatamente.

4.1 Il tag Article

```
<!element article - -
  (titlepag, header?,
   toc?, lof?, lot?, p*, sect*,
   (appendix, sect+)?, biblio?) +(footnote)>

<!attlist article
  opts cdata "null">
```

Come si può vedere, il tag *article* necessita l'inclusione di alcuni tag, che verranno di seguito spiegati.

L'attributo *options* (*opts*) accetta un elenco separato da virgole di differenti fogli di stile (L^AT_EX .sty) da includere nel documento.

4.1.1 Il tag Titlepage

```
<!element titlepag o o (title, author, date?, abstract?)>
```

Il tag *Titlepage* (*titlepag*) è implicitamente inserito all'inizio della propria *classe di documento*. Non è necessario scriverlo esplicitamente. Si noti comunque che il tag è obbligatorio. Il suo scopo è quello di descrivere l'impaginazione e gli elementi dei titoli delle pagine.

Il tag Title

```
<!element title - o (%inline, subtitle?) +(newline)>
```

Ogni *classe di documento* con un proprio titolo di pagina certamente necessiterà di un *titolo*, denotato da un tag <title>. Non è necessario chiuderlo. Un titolo può contenere un *sottotitolo* che inizia con il tag <subtitle>.

Se si osserva l'intestazione di questo documento si vedrà che è mappato dai tag:

```
<title>Linuxdoc Reference
<subtitle>Un'introduzione al dtd di linuxdoc
```

Il tag `Author`

```
<!element author - o (name, thanks?, inst?,
                      (and, name, thanks?, inst?)*)>
```

Solitamente qui si mette il (proprio) nome. La gente dovrebbe sapere chi ha scritto il documento, per questo si mette un tag `<author>`. Se non lo si annota, il tag `name` è inserito implicitamente. L'*autore* ha anche degli elementi opzionali che possono essere etichettati all'interno del tag `author`.

Se si desidera fare dei ringraziamenti a qualcuno (potrebbe essere qualcuno che ha fornito delle informazioni utili), inserirlo all'interno del tag `<thanks>`. Inoltre, se si sta scrivendo il proprio documento come membro dello staff di una *istituzione*, inserirlo all'interno del tag `<inst>`.

Il tag `<and>` ripete le stesse cose, come se si fosse inserito un secondo tag `author`. Chiaramente serve per i coautori.

Il tag `Date` Volendo, si può marcare il proprio documento con una *data* usando il tag `<date>`.⁶

Il tag `Abstract` Questo tag è destinato ad un *riassunto* della descrizione del proprio documento. Non si confonda il tag `<abstract>` con l'*introduzione*, che più probabile vada inserita nella prima *sezione* del proprio documento (vedere la sezione 6 (Creare le sezioni)).

4.1.2 Il tag `Header`

```
<!element header - - (lhead, rhead) >
<!element lhead - o (%inline)>
<!element rhead - o (%inline)>
```

Un tag `<header>` specifica cosa dovrebbe essere stampato nella parte più alta di ogni pagina. Esso consiste di una *intestazione sinistra* cioè `<lhead>` e una *intestazione destra* cioè `<rhead>`). Entrambi gli elementi sono obbligatori se si usa un'intestazione, ma possono essere lasciati vuoti: in questo modo si può facilmente ottenere l'effetto di avere solo un'intestazione sinistra o solo un'intestazione destra.

Come si vedrà, l'intestazione iniziale può essere inserita dopo il titolo della pagina. In seguito può essere aggiunta una nuova intestazione per ogni nuovo capitolo o sezione. L'intestazione stampata su una pagina è quella che ha effetto alla fine della pagina attuale. In questo modo l'intestazione sarà quella dell'ultima sezione della pagina.

4.1.3 Il tag `Table Of Contents`

Usando il tag `<toc>`, verrà generato un *indice*, osservando l'intestazione della sezione e aggiungendo i riferimenti.⁷ Verranno incluse solo le sezioni superiori alla `sect3`.

⁶Non è possibile verificare la validità della data inserita, quindi meglio non abusare di questo tag.

⁷In un documento con collegamenti ipertestuali si vedranno i collegamenti, in un documento \LaTeX si vedranno i numeri di pagina.

4.1.4 Il tag List Of Figures

Usando il tag `<lof>`, verrà generato un *elenco di illustrazioni*, osservando le didascalie delle illustrazioni e aggiungendo i riferimenti.

4.1.5 Il tag List Of Tables

Usando il tag `<lot>`, verrà generata una *lista di tabelle*, osservando le didascalie delle tabelle e aggiungendo i riferimenti.

4.1.6 Body

Qui si inseriscono varie sezioni stando a quanto dice la sezione 6 (Creare le sezioni). Non c'è un *tag body*. Il corpo del testo inizia dal primo *capitolo, sezione o paragrafo*.

4.1.7 Il Tag Appendix

Alla fine dell'articolo si può inserire il tag `<appendix>`⁸, che apre un'area di sezioni aggiuntive. Il tag `appendix` implica un tipo di numerazione delle sezioni diverso per i successivi tag di sezione.

4.1.8 Il tag Bibliography

È destinato a raccogliere tutte le `<cites>` e `<ncites>` usate all'interno del proprio documento. Il tag `<biblio>` sarà sostituito da una *bibliografia* a seconda del tipo di mappatura del documento, a volte tramite collegamenti ipertestuali, a volte tramite numeri di sezione o qualunque cosa possa essere utile.⁹

4.1.9 Il tag Footnote

Una *nota a piè di pagina* può essere messa in ogni punto del proprio documento. Per la precisione, si dovrebbe piazzare il tag `<footnote>` nel punto del proprio documento in cui dovrebbe essere visualizzato il testo etichettato. Questo tag dovrebbe essere usato per informazioni aggiuntive, che non sono necessarie per comprendere gli scopi principali del proprio documento ma potrebbero essere utili, interessanti o divertenti.¹⁰ dovunque all'interno dell'articolo.

4.2 Il tag Report

```
<!element report - -
  (titlepag, header?, toc?, lof?, lot?, p*,
   chapt*, (appendix, chapt+)?, biblio?) +(footnote)>
```

La classe di documento *report* ha un approccio orientato al capitolo. Quindi, in un documento classificato con un tag `<report>` il livello più alto è raggruppato dal tag `<chapt>` (vedere 6 (Creare le sezioni)). Il resto della struttura è identico alla classe *article* 4.1 (Il tag Article).

⁸Really you shouldn't think about people (e.g. m.d.s knifing your belly here).

⁹Fino ad ora non sono stato in grado di creare un file `.bbl`, dunque non ho potuto verificare.

¹⁰D'altra parte l'ultima non è sempre vera, anche se ci si prova.

4.3 Il tag Book

```
<!element book - -
  (titlepag, header?, toc?, lof?, lot?, p*, chapt*,
  (appendix, chapt+)?, biblio?) +(footnote) >
```

Come si può notare, l'elemento *book* è identico all'elemento *report* 4.2 (Il tag Report). Quindi quanto detto in precedenza vale anche se si classifica il proprio documento con un tag `<book>`.

4.4 Il tag Letter

```
<!entity % addr "(address?, email?, phone?, fax?)" >

<!element letter - -
  (from, %addr, to, %addr, cc?, subject?, sref?, rref?,
  rdate?, opening, p+, closing, encl?, ps?)>
```

Anche lo scopo della classe di documento *letter* dovrebbe essere abbastanza autoesplicativa. Inserire un tag `<letter>` se si vuole scrivere una lettera.

I tag della lettera sono descritti nella tabella 3 (Tag in una lettera)

tag	obbligatorio	che cos'è
from	si	mittente
address	no	indirizzo del mittente
email	no	email del mittente
phone	no	telefono del mittente
fax	no	fax del mittente
to	si	destinatario
address	no	indirizzo del destinatario
email	no	email del destinatario
phone	no	telefono del destinatario
fax	no	fax del destinatario
cc	no	copia per conoscenza
subject	no	oggetto della lettera
sref	no	riferimenti del mittente
rref	no	riferimenti del destinatario
rdate	no	data di ricevimento??
opening	si	apertura
paragraphs	si	vedere 7 (Paragrafi)
closing	si	chiusura
encl	no	allegato
ps	no	post scriptum

Table 3: Tag in una lettera

4.5 Il tag Telefax

```
<!element telefax - -
  (from, %addr, to, address, email?,
```

```
phone?, fax, cc?, subject?,
opening, p+, closing, ps?)>
```

La struttura globale è la stessa della classe *letter*. La sola differenza è che con il tag `<telefax>` il tag del destinatario `<fax>` diventa obbligatorio.¹¹

4.6 Il tag Slides

```
<!element slides - - (slide*) >
```

La classe *slides* è destinata per slide e trasparenze di livello superiore. Quindi la struttura di un documento classificato con un tag `<slides>` è molto semplice: contiene singole slide che iniziano con un tag `<slide>`. Niente altro. Se non esplicitamente scritto, il primo tag *slide* inizia implicitamente.

4.6.1 Il tag Slide

```
<!element slide - o (title?, p+) >
```

Un tag `<slide>` è permesso solo all'interno della classe di documento *slides*. Una *slide* può contenere:

Un *titolo* (vedere la sezione 4.1.1 (Il tag Title)) e uno o più *paragrafi* (vedere la sezione 7 (Paragrafi)). Questo è tutto.

4.7 Il tag Note

```
<!element notes - - (title?, p+) >
```

È una classe destinata ad appunti personali, la sua struttura è perfino più semplice della classe di documento *slides* (vedere 4.6.1 (Il tag Slide)). Dopo aver classificato un documento con il tag `<notes>`, sono permessi solo un *titolo* (vedere la sezione 4.1.1 (Il tag Title)) e uno o più *paragrafi* (vedere la sezione 7 (Paragrafi)).

4.8 Il tag Manual Page

```
<!element manpage - - (sect1*)
-(sect2 | f | %mathpar | figure | tabular |
table | %xref | %thrm )>
```

Questa classe di documento è destinata alla scrittura di *pagine di manuale*, ed è adatta alle necessità del programma *man*. In un documento classificato dal tag `<manpage>` il tag di sezione di livello più alto è il tag `sect1` (vedere la sezione 6 (Creare le sezioni)), per un facile inserimento di pagine di manuale all'interno di una classe di documento *article* o *book*. L'eccezione alla normale creazione di sezioni è che viene permesso un solo livello di sottosezione (`sect2`).

5 Inline

```
<!entity % inline
" (#pcdata | f | x | %emph; |sq| %xref | %index | file )* " >
```

¹¹Il perché dovrebbe essere ovvio.

Inline può trovarsi ovunque dentro il testo, e non ha alcuna influenza sul flusso del testo o sulla struttura logica del documento.

#pcdata

Il *parsed character data* è semplicemente testo normale all'interno del flusso che può contenere altri inline.

f

Formule matematiche inline secondo il `maths.dtd`. Vedere 8.3 (Il tag Formula).

x

Il tag *external* aggira l'analizzatore. I dati nei tag finiscono direttamente nel file mappato. Vedere il capitolo 8.4 (Il tag External) per informazioni dettagliate.

%emph;

Enfatizzazione del testo. Vedere il capitolo 8.1 (Enfatizzazione).

sq

Citazioni nel flusso di testo. Vedere il capitolo 8.2 (Il tag Short Quote).

%xref

XRiferimenti nel testo o riferimenti esterni. Vedere il capitolo 10 (Etichette e Riferimenti).

%index

Anche questo non lo so spiegare. Se ne sapete di più, scrivetemi.

file

Anche questo non lo so spiegare (posso solo supporre riguardi file di immagine eps). Se ne sapete di più, scrivetemi.

6 Creare le sezioni

```
<!element chapt - o (%sect, sect*) +(footnote)>
<!element sect - o (%sect, sect1*) +(footnote)>
<!element sect1 - o (%sect, sect2*)>
<!element sect2 - o (%sect, sect3*)>
<!element sect3 - o (%sect, sect4*)>
<!element sect4 - o (%sect)>
```

La creazione di *sezioni*¹² si ottiene tramite elementi concordi che formano l'albero delle sezioni. Servono a unire i vari paragrafi del nostro documento, formando un grazioso albero. Il tag di livello più alto e la profondità ammessa varia a seconda della *classe di documento* (vedere la sezione 4 (La classe di documento)).

La normale gerarchia prevede:

chapt

sect

sect1

¹²Anche il tag `chapt` è un tag che crea *sezioni*.

```

sect2
  sect3
    sect4

```

Si prenda un libro e si guardi l'indice per vederlo.

Ciascuno dei tag tra le *sezioni* ha più o meno la stessa sintassi. Ognuno di essi ha un'*intestazione*. Il tag `intestazione` è implicito se non lo si annota. Inoltre ogni tag che crea sezioni può contenere un tag `intestazione`, che modifica l'intestazione attuale del documento (vedere la sezione 4.1.2 (Il tag Header)).

In esso si possono posizionare sezioni subordinate e *paragrafi* (vedere 7 (Paragrafi)).

Alcuni dei tag che creano sezioni possono apparire soltanto in speciali classi di documento (4 (Classi di documento)).

Suggerimento:

È cosa saggia posizionare un tag *etichetta* dopo il testo del tag *sezione*, anche se non si vuole fare riferimento alla sezione 10 (Etichette e riferimenti). Più avanti, quando il proprio documento aumenterà di dimensioni, si potrebbe volerlo.

7 Paragrafi

```

<!entity % sectpar
  " %par; | figure | tabular | table | %mathpar; |
    %thrm; | %litprog; ">

<!entity % par
  " %list; | comment | lq | quote | tscreen " >

<!entity % litprog " code | verb " >

```

Ciascuno dei tag qui descritti forma un paragrafo.

Per ovvie ragioni, un paragrafo normalmente¹³ inizia e finisce con una riga vuota.¹⁴

Ci sono alcuni tag che formano sempre un paragrafo, e un modo per formare implicitamente un paragrafo. Ci sono vari tipi di paragrafi, perchè non tutti i tipi di paragrafo possono apparire in ogni classe di documento, in ogni posto.

I diversi tipi di paragrafi sono spiegati nelle prossime sezioni. Per maggiori dettagli su `%litprog`; si veda 12 (Programmazione istruita).

7.1 Paragrafo normale

Un paragrafo normale può essere creato in due modi:

7.1.1 Tag di paragrafo

Il tag `<p>` inizia un nuovo *paragrafo*. Questo tag è obbligatorio se si vuole terminare un'intestazione di sezione senza chiudere esplicitamente il tag `sect`. In questo caso il tag `<p>` chiude automaticamente il tag `<sect>`.

¹³Qui si spiega il comportamento delle eccezioni `figure` e `tabular`.

¹⁴In quale altro modo si potrebbe sapere che è un paragrafo?

7.1.2 Nuova riga vuota

Una riga vuota tra due paragrafi inizia implicitamente un nuovo *paragrafo*. Si faccia attenzione con gli elenchi descrittivi. In questo caso un tag `<tag>` vuoto non diventerà un paragrafo usando una riga vuota.

7.2 Paragrafi stile elenco

```
<!entity % list
    " list | itemize | enum | descrip " >
```

Questi quattro tag indicano l'inizio di un paragrafo in stile elenco. All'interno di ciascun elenco le singole voci sono separate da un tag *item*.

```
<!element item o o ((%inline; | %sectpar;)*, p*) >
```

Come si può vedere, una voce può contenere a sua volta paragrafi (e quindi anche altri elenchi, perfino di tipo differente).

7.2.1 Il tag List

```
<!element list - - (item+)>
```

Il tag *list* verrà mappato come un elenco spoglio senza punti, numeri o altre cose.

Per vederlo, ecco un piccolo esempio:

```
<list>
<item>Un punto
<item>Un altro
<item>L'ultimo
</list>
```

Si vedrà (a seconda della mappatura) qualcosa come:

```
Un punto
Un altro
L'ultimo
```

7.2.2 Il tag Itemize

```
<!element itemize - - (item+)>
```

Il tag *itemize* verrà mappato come un elenco puntato, che viene solitamente usato per elenchi nei quali l'ordine delle voci non è importante.

Un piccolo esempio:

```
<itemize>
<item>Un punto
<item>Un altro
<item>L'ultimo
</itemize>
```

Si vedrà (a seconda della mappatura) qualcosa come:

- Un punto
- Un altro
- L'ultimo

7.2.3 Il tag Enum

```
<!element enum - - (item+)>
```

Il tag *enum* verrà mappato come una lista numerata.

Un piccolo esempio:

```
<enum>
<item>Un punto
<item>Un altro
<item>L'ultimo
</enum>
```

Si vedrà (a seconda della mappatura) qualcosa come:

1. Un punto
2. Un altro
3. L'ultimo

7.2.4 Il tag Descrip

```
<!element descrip - - (tag?, p+)+ >
```

Il tag *descrip* verrà mappato come un elenco descrittivo. Il concetto in questo caso è leggermente diverso rispetto agli altri tipi di elenchi citati prima.

Qui si posiziona un *tag* (questa volta il nome del tag è letteralmente **tag**), che è descritto più avanti.

Un piccolo esempio:

```
<descrip>
<tag/sgml/structured general markup language.
<tag/html - hypertext markup language/
Una implementazione di sgml.
Contiene alcuni concetti su come collegare informazioni in modo molto conveniente.
È questo che lo ha reso tanto importante, nonchè lo standard per i
documenti pubblicati tramite internet.
```

```
<tag/internet/Una internet connessa globalmente (internet intesa come termine
tecnico)
</descrip>
```

Si vedrà (a seconda della mappatura) qualcosa come:

sgml

structured general markup language.

html - hypertext markup language

Una implementazione di sgml. Contiene alcuni concetti su come collegare informazioni in un modo molto conveniente. È questo che lo ha reso tanto importante, nonché lo standard per i documenti pubblicati tramite internet.

internet

Una internet connessa globalmente (internet intesa come termine tecnico)

7.3 Illustrazioni e Tabelle

I tag `<figure>` e `<table>` creano paragrafi molto speciali. Non sempre sono inseriti nel normale flusso di testo. Entrambi i tag possono contenere un attributo `loc` (*location*) che indica come manipolare il flusso di questo paragrafo particolare.

Il valore dell'attributo `loc` è una stringa di quattro lettere al massimo, in cui ogni lettera dichiara una posizione nella quale l'illustrazione o la tabella **può** apparire, come descritto nella tabella 4 (Posizioni della tabella).

h	here	Nella stessa posizione del file SGML
t	top	All'inizio di una pagina
b	bottom	Alla fine di una pagina
p	page	In una pagina separata con sole illustrazioni e tabelle

Table 4: Posizioni della tabella

Il valore predefinito dell'attributo `loc` è `top`.

7.3.1 Il tag Table

```
<!element table - - (tabular, caption?) >
```

Come si può vedere, una *tabella* consiste nello stesso tag `<table>`, che include un tag `<tabular>` e opzionalmente un tag `<caption>`.

Il tag `<tabular>` può anche essere posizionato senza un tag `<table>`, come descritto in dettaglio nella sua sezione (si veda 7.4 (Il tag Tabular)).

Il tag *caption* si usa anche per posizionare la voce per l' *elenco delle tabelle*, se ne è stato dichiarato uno (si veda 4.1.5 (Il tag lista di tabelle)).

Un breve esempio mostrerà come lavorano insieme.

```
<table loc="ht">
<tabular ca="lcr">
Guarda|questa|tabella@
Non|è|carina@
1.234|colonne|miste
</tabular>
<caption>Una tabella di esempio
</table>
```

Guarda	questa	tabella
Non	è	carina
1.234	colonne	miste

Table 5: Una tabella di esempio

Il *caption* “Una tabella di esempio” sarebbe il nome nella *lista delle tabelle*.

7.3.2 Il tag Figure

```
<!element figure - - ((eps | ph ), img*, caption?)>
```

L'utilizzo del tag `<figure>` è equivalente a quello del tag `<table>`. Invece del tag `<tabular>` si può utilizzare o il tag `<eps>` o il tag `<ph>`.

Il tag Encapsulated PostscriptTM

```
<!attlist eps
    file cdata #required
    height cdata "5cm"
    angle cdata "0">
```

Il tag `<eps>` serve a includere un file esterno in formato *encapsulated postscriptTM* nel documento.

Gli attributi del tag `<eps>` sono:

file

L'attributo `file` richiede il *nome* di un file encapsulated postscriptTM che termini con il suffisso `.ps`. Il suffisso obbligatorio `.ps` non deve essere scritto.

height

L'*altezza* dello spazio che il file andrà ad occupare. Se non lo si specifica, il valore predefinito sarà 5 cm. Si faccia attenzione a non inserire spazi tra il numero e l'unità di misura (i, cm).

angle

L'*angolo* viene indicato in gradi sessagesimali (0-360) e all'aumento del numero il file viene ruotato in senso orario.

Un esempio:

```
<figure loc="here">
<eps file="logo" height="4cm" angle="15">

<caption>Inclusione di un encapsulated postscript(TM)
</figure>
```

Il tag *img* verrà ignorato dalla mappatura L^AT_EX e sarà utile per l'html, poichè molti browser non riconoscono l'eps.

Figure 1: Inclusione di un file encapsulated postscriptTM.

Il tag *caption* andrà nella *lista delle illustrazioni* come descritto nella sezione 4.1.4 (Il tag lista di immagini).

Il tag placeholder

```
<!attlist ph
      vspace cdata #required>
```

Questo tag non posiziona alcunchè, ma inserisce uno spazio vuoto adatto per incollare manualmente figure nel buon vecchio modo. L'attributo *vspace* assegna lo spazio lasciato libero. **Caveat:** L'argomento numerico dell'attributo *vspace* necessita di un'unità di misura direttamente dietro al numero. Non si lasci uno spazio (come detto per l'attributo *height* in 7.3.2 (Il tag Encapsulated PostscriptTM)).

```
<figure loc="ht">
<ph vspace="5cm">
<caption>Uno spazio vuoto.
</figure>
```

Restituisce:

Figure 2: Uno spazio vuoto per incollare una fotografia

A questo punto potrebbe essere il caso di cercare forbici e colla.

7.4 Il tag Tabular

```
<!element tabular - -
      (hline?, %tabrow, (rowsep, hline?, %tabrow)*, caption?) >
```

Il tag `<tabular>` viene interpretato come un paragrafo a se stante, se viene scritto da solo. Insieme al tag `<table>` diventa parte del paragrafo del tag `<table>` (vedere 7.3.1 (Il tag Table)).

All'interno del tag `tabular` possono esserci righe e colonne che separano il testo. Devono esserci almeno una colonna e una riga.¹⁵

Il tag `<tabular>` ha un attributo `ca` obbligatorio per *l'allineamento di colonna*. L'allineamento di colonna contiene un solo carattere per ogni colonna nel loro ordine da sinistra a destra. I caratteri che si possono posizionare per colonna sono descritti nella tabella 6 (Allineamento di colonna)

carattere	allineamento
l	left (sinistra)
c	centered (centrato)
r	right (destra)

Table 6: Allineamento di colonna

In teoria si dovrebbe poter posizionare un `|` nell'attributo `ca` per disegnare una linea orizzontale di separazione per due colonne. Il problema: non funziona. L'analizzatore lo accetta bene, solo che l'output in \LaTeX mapperà `|` come `{\$}`, che naturalmente è l'impostazione di quattro colonne erroneamente allineate per tutte e quattro le colonne. Proverò a capire come risolverlo.

Le colonne all'interno del tag `<tabular>` sono separate da un *separatore di colonna*, il tag `<colsep>`. Il carattere `|` viene tradotto come `<colsep>`, così lo si può usare al suo posto.¹⁶

Ciò che vale per le colonne vale anche per le righe. Si separano le righe con un *separatore di riga*, il tag `<rowsep>`. Il carattere `@` viene tradotto con `<rowsep>`.

Opzionalmente si può posizionare una *linea orizzontale* con il tag `<hline>`. Si faccia attenzione a questo tag: gli strumenti SGML lo trasformeranno correttamente sia posizionandolo davanti alla riga che si vuole sotto la linea, che dietro la fine della riga che si vuole al di sopra. Ma l'unico modo per scriverlo senza provocare un "errore" dell'analizzatore è scriverlo direttamente e senza spazi o riga vuota dietro il separatore di riga.

```
<tabular ca="lcr">
Guarda|questa|tabella@<hline>
Non|è|gradevole@
1.234|colonne|miste@
</tabular>
```

I risultati sono visibili nella tabella 7 (Esempio di tabella per il tag Tabular)

Guarda	questa	tabella
Non	è	gradevole
1.234	colonne	miste

Table 7: Esempio di tabella per il tag Tabular

¹⁵Altrimenti non risulterebbe molto utile.

¹⁶Meno battitura, più divertimento.

Attenzione:

nella mappatura per L^AT_EX tutto funziona bene se si posiziona un tag *tabular* senza un tag *table*; solo nelle altre mappature (per esempio html) verrà disordinato.

7.5 Paragrafi matematici

```
<!entity % mathpar " dm | eq " >
```

Un *paragrafo matematico* consiste o di una *formula visualizzata* tramite un tag `<dm>`,¹⁷ o di una *equazione* tramite il tag `<eq>`. Lavorano entrambi allo stesso modo.

Entrambi i tag contengono una formula matematica. Vedere 9 (Formule matematiche) per i tag validi.

Nota:

Poichè nè Netscape nè Microsoft hanno ritenuto necessario aggiungere la mappatura matematica ai loro browser (come richiesto e definito da *w3c*), non c'è in html un modo piacevole di mappare, o almeno di mostrare a video, i simboli matematici. Quindi, guardando la versione online, ci si può liberamente meravigliare di questo controsenso. Forse si vorrà dare uno sguardo alla versione postscript.

7.5.1 Il tag Displayed Formula

Questo tag mostra una *formula matematica* come un *paragrafo*. La formula viene mappata al centro come linea singola.¹⁸

```
<dm>(a+b)<sup/2/=a<sup/2/+2ab+b<sup/2/</dm>
```

Viene mappato come:

$$(a + b)^2 = a^2 + 2ab + b^2$$

7.5.2 Il tag Equation

```
<dm>(a+b)<sup/2/=a<sup/2/+2ab+b<sup/2/</dm>
```

Viene mappato come:

$$(a + b)^2 = a^2 + 2ab + b^2$$

7.6 Paragrafo Teorema

```
<!entity % thrm
      " def | prop | lemma | coroll | proof | theorem " >
```

```
<!element def - - (thtag?, p+) >
<!element prop - - (thtag?, p+) >
<!element lemma - - (thtag?, p+) >
<!element coroll - - (thtag?, p+) >
<!element proof - - (p+) >
<!element theorem - - (thtag?, p+) >
```

¹⁷No, scusate, non per il marco tedesco! ;-)

¹⁸Non ci sono garanzie per questo. Si sa, la mappatura è una questione di gusti.

Come si può vedere i diversi tipi di paragrafi *theorem* sono praticamente identici. L'unica eccezione leggermente diversa è *proof*, che non possiede un `thtag`. Per tutte le altre versioni il `thtag` fornisce il `tag` del paragrafo teorema.

Si provi a utilizzare solo questo, che è adatto al significato di ciò che si sta digitando.

```
<thrm>
<thtag>Alexander's thrm</thtag>
Sia  $\mathcal{G}$  un insieme di obiettivi secondari ottenibili non banali,
e  $\mu$  un ordinamento su  $\mathcal{G}$ .  $\mu$ 
astrattamente indicativo se e solo se è una linearizzazione di
 $\lim_{\mu} \mathcal{G}$ .
</theorem>
```

`thrm` viene rimpiazzato dal tag adeguato.

Forse qualcuno esperto di matematica rimarrà scioccato dal mio abuso dei tipi, ma sono pigro e ho semplicemente copiato gli esempi:

Definizione (`def`):

Definition 1 (Definizione di Alessandro) *Sia \mathcal{G} un insieme di obiettivi secondari ottenibili non banali, e μ un ordinamento su \mathcal{G} . μ è astrattamente indicativo se e solo se è una linearizzazione di $\mu_{\mathcal{G}}^*$.*

Proposizione (`prop`):

Proposition 1 (Proposizione di Alessandro) *Sia \mathcal{G} un insieme di obiettivi secondari ottenibili non banali, e μ un ordinamento su \mathcal{G} . μ è astrattamente indicativo se e solo se è una linearizzazione di $\mu_{\mathcal{G}}^*$.*

Lemma (`lemma`):

Lemma 1 (Lemma di Alessandro) *Sia \mathcal{G} un insieme di obiettivi secondari ottenibili non banali, e μ un ordinamento su \mathcal{G} . μ è astrattamente indicativo se e solo se è una linearizzazione di $\mu_{\mathcal{G}}^*$.*

Corollario (`coroll`):

Corollary 1 (Corollario di Alessandro) *Sia \mathcal{G} un insieme di obiettivi secondari ottenibili non banali, e μ un ordinamento su \mathcal{G} . μ è astrattamente indicativo se e solo se è una linearizzazione di $\mu_{\mathcal{G}}^*$.*

Theorem 1 (Teorema di Alessandro) *Sia \mathcal{G} un insieme di obiettivi secondari ottenibili non banali, e μ un ordinamento su \mathcal{G} . μ è astrattamente indicativo se e solo se è una linearizzazione di $\mu_{\mathcal{G}}^*$.*

La riprova è questa, senza `thtag`:

Proof. Sia \mathcal{G} un insieme di obiettivi secondari ottenibili non banali, e μ un ordinamento su \mathcal{G} . μ è astrattamente indicativo se e solo se è una linearizzazione di $\mu_{\mathcal{G}}^*$.

7.7 Paragrafi Code e Verbatim

Sono entrambi tag di un paragrafo e con un comportamento molto simile. All'interno di questi tag molti caratteri speciali non necessitano della forma nominata come nella sezione A (Simboli nominati). Le eccezioni sono:

1. `&etago;` → `</` → fine del tag aperto

Forse più avanti l'elenco si allungherà.

A differenza della normale mappatura di un paragrafo, gli spazi vuoti e gli a capo verranno mappati letteralmente (come saranno scritti nel sorgente).

Inoltre (con riferimento al layout manuale), il tipo di carattere per la mappatura sarà non-proporzionale.¹⁹

Nota:

Ripeto, non sono un oratore nato e nemmeno amo molto la matematica. Così ho solo scritto alcuni controsensi, che possono provocare il mal di testa e far venire i capelli grigi alle persone che vogliono utilizzare questo documento per imparare a formulare teorie matematiche o fisiche.

Ci si senta liberi di inviare esempi migliori.

7.7.1 Il tag Code

```
<!element code - - rcdata>
```

Si utilizzi il *tag code* per scrivere un esempio di codice sorgente all'interno del testo.

Un esempio di codice

```
<code>
```

```
#include <stdio.h>
int main() {
    printf("Ciao mondo");
    return 1;
}
```

```
</code>
```

7.7.2 Il tag Verbatim

```
<!element verb - - rcdata>
```

Si utilizzi il *tag verbatim* per qualunque cosa che non sia codice sorgente (si utilizzi 7.7.1 (Il tag code) per questo) che necessita la buona vecchia spaziatura, tipo la stampata del terminale, la grafica ASCII ecc.

Un esempio di verbatim:

```
<verb>
```

```
/////////
| * * |
| | |
| <---> |
| \_____/
```

```
</verb>
```

¹⁹Si veda la differenza tra IIWW e IWW.

8 I tag Inline

Qui gli *inline* astratti vengono spezzati finchè restano solo tag reali e utilizzabili. Si richiami alla mente:

```
<!entity % inline
    " (#pcdata | f | x | %emph; | sq | %xref | %index | file ) * " >
```

I tag inline non influenzano i paragrafi, la creazione di sezioni o le classi di documento. Modificano solamente il testo nel suo normale scorrimento.

8.1 Enfaticizzazione

```
<!entity % emph
    " em | it | bf | sf | sl | tt | cparam " >
```

L'*enfaticizzazione* è la riunione di tag per enfaticizzare il testo inline.

Diversi tipi di enfaticizzazione sono:

em → Il tag **Emphasize**

Odio essere ridondante ma devo ripetere: il tag *emphasize* si usa per enfaticizzare il testo, che normalmente verrà mappato in italiano. Quindi scrivendo la frase `<em/un testo enfaticizzato/`, verrà mappata come *un testo enfaticizzato*.

it → Il tag **Italic**

Il tag *italic* si usa per la mappatura in corsivo. Scrivendo la frase `<it/un testo in italiano /`, verrà mappata come *un testo in italiano*.

bf → Il tag **Boldface**

Il tag *boldface* si usa per la mappatura in grassetto. Scrivendo la frase `<bf/un testo in grassetto/`, verrà mappata come **un testo in grassetto**.

sf → Il tag **Swissfont**

Tom Gordon del GMD afferma che questo è il tag *sans serif*. La mia interpretazione del tag *sf* è *swissfont*, che per me è più facile da ricordare. Questo mapperà il testo inline a un font non appartenente alla famiglia helvetica. Quindi la frase `<sf/un testo swissfont/` verrà mappata come un testo *swissfont*.

sl → Il tag **Slanted**

Penso che salterò la spiegazione. `<sl/un testo slanted /` verrà mappato come *un testo slanted*.

tt → Il tag **Terminaltype**

Il testo etichettato con il tag *terminaltype* verrà inserito inline, proprio come tutto il resto del testo all'interno di un paragrafo. Non verrà incluso nell'output del sorgente se si sta lavorando come descritto nella sezione 12 (Programmazione istruita), perfino se appare come codice scritto. `<tt/Un testo scritto al terminale/` verrà mappato come un testo scritto al terminale.

8.2 Il tag Short-quote

Normalmente questo tag dovrebbe essere visualizzato allo stesso livello dei tag *emphasize*, ma la definizione del dtd di linuxdoc lo mette allo stesso livello della enfaticizzazione, e così mi sono comportato.

Il tag *shortquote* è una citazione inline che non forma un proprio paragrafo. Il testo `<sq/una breve citazione/` verrà mappato come "una breve citazione".

8.3 Il tag Formula

Il tag *formula* permette di prendere nota di una formula matematica all'interno del testo normale, che non appare in una propria riga. Quindi il testo `<f>x=y²</f>` verrà visualizzato come $x = y^2$. Vedere 9 (Formule matematiche) per i tag validi all'interno di *formula*.

8.4 Il tag External

Il tag *external* fa passare i dati così etichettati direttamente dall'analizzatore, senza modificarli. Un esempio è \LaTeX .

9 Formule matematiche

Possono apparire all'interno dei tag elencati nella tabella 8 (Posizioni delle formule matematiche)

tag	descrizione	vedere
f	formula inline	8.3 (il tag Formula)
dm	formula visualizzata	7.5 (paragrafo matematico)
eq	equazione	7.5 (paragrafo matematico)

Table 8: Posizioni delle formule matematiche

Guardando questo documento mappato in html, si noterà che l'html non ha un modo gradevole per visualizzare le formule matematiche.

Dopo una piccola trasformazione manuale il contenuto di un tag *matematico* assomiglierà a:

```
<!element xx      - -
      ((fr|lim|ar|root) |
      (pr|in|sum) |
      (#pdata|mc|(tu|phr)) |
      (rf|v|fi) |
      (unl|ovl|sup|inf))*>
```

xx sta per f, dm o eq. Sono tutti la stessa cosa.

Nota:

Poichè nè Netscape nè Microsoft hanno ritenuto necessario aggiungere la mappatura matematica ai loro browser (come richiesto e definito da *w3c*), non c'è in html un modo piacevole di mappare, o almeno di mostrare a video, i simboli matematici. Quindi, guardando la versione online, ci si può liberamente meravigliare di questo controsenso. Forse si vorrà dare uno sguardo alla versione postscript.

9.1 Il tag Fraction

```
<!element fr      - - (nu,de) >
<!element nu      o o ((%fbtxt;)* >
<!element de      o o ((%fbtxt;)* >
```

Come si può vedere, una *frazione* consiste di un tag *numeratore* e di un tag *denominatore*, ognuno dei quali può contenere una *formula matematica*.

Un esempio dirà di più:

```
<dm><fr><nu>7/<de>13/</fr></dm>
```

risulterà come:

$$\frac{7}{13}$$

In caso si voglia posizionare $\frac{1}{2}$ invece del numeratore senza ripulirlo, si digiterà:

```
<dm><fr><nu><fr><nu>1/<de>2/</fr></nu><de>13/</fr></dm>
```

che risulterà come:

$$\frac{\frac{1}{2}}{13}$$

9.2 I tag Product, Integral e Summation

```
<!element pr      - - (ll,ul,opd?) >
<!element in      - - (ll,ul,opd?) >
<!element sum     - - (ll,ul,opd?) >
```

Ciascuno di essi ha un *limite inferiore* (il tag ll), un *limite superiore* (il tag ul) e un *operando* opzionale, e ciascuno di essi può consistere in una formula. I tag sono simili nella sintassi, come mostrato nella tabella 9 (Tag con limite superiore, limite inferiore e operatore).

nome	esempio	risultato
Prodotto	<code><f>y=<pr><ll>i=1n<opd>x<inf/i/</pr></f></code>	$y = \prod_{i=1}^n x_i$
Integrale	<code><f>y=<in><ll>ab<opd>x<sup/2/</in></f></code>	$y = \int_a^b x^2$
Sommatoria	<code><f>y=<sum><ll>i=1n<opd>x<inf/i/</sum></f></code>	$y = \sum_{i=1}^n x_i$

Table 9: Tag con limite superiore, limite inferiore e operatore

9.3 Il tag Limited

```
<!element lim     - - (op,ll,ul,opd?) >
<!element op      o o (%fcstxt;|rf|%fph;) -(tu) >
<!element ll      o o ((%fbutxt;)* >
<!element ul      o o ((%fbutxt;)* >
<!element opd     - o ((%fbutxt;)* >
```

Questo tag si può usare per operatori con limite superiore e inferiore diversi da prodotti, somme o integrali. Per gli altri tipi definiti *operatori* è assegnato mediante il tag `op`, che può a sua volta contenere una formula matematica.

$$B_{i=0}^n x_i$$

9.4 Il tag Array

```

<!element ar      - - (row, (arr, row)*) >
<!attlist ar
  ca      cdata   #required >
<!element arr    - o empty >
<!element arc    - o empty >
<!entity  arr    "<arr>" >
<!entity  arc    "<arc>" >

```

Ovviamente un ragionevole documento matematico necessita di un metodo per descrivere gli array e le matrici. Un *array* (*ar*) è annotato come equivalente a un *tabular* (vedere la sezione 7.4 (Il tag Tabular)). Le differenze di gestione sono:

- non c'è il tag `<hline>`.
- Non permesso l'attributo di carattere `ca |`.
- Le colonne non sono separate dal tag `colsep` ma dal tag `arc` (*colonna di array*).
- Le righe non sono separate dal tag `rowsep` ma dal tag `arr` (*riga di array*).

Anche in questo caso i caratteri `|` e `@` vengono mappati con l'adeguato tag di separazione, in questo modo si può davvero annotare un array come tabular.

```

<dm><ar ca="clcr">
a+b+c | uv   <arc> x-y | 27   @
a+b   | u+v  | z   | 134 <arr>
a     | 3u+vw | xyz | 2,978
</ar></dm>

```

Viene mappato come:

$$\begin{array}{cccc}
 a + b + c & uv & x - y & 27 \\
 a + b & u + v & z & 134 \\
 a & 3u + vw & xyz & 2,978
 \end{array}$$

9.5 Il tag Root

```

<!element root    - - ((%fbutxt;)* >
<!attlist root
  n cdata "">

```

root viene annotato con il tag `root`, che contiene un attributo `n` contenente il valore per "n'th" root.

```

<dm><root n="3"/x+y/</dm>

```

viene mappato come:

$$\sqrt[3]{x+y}$$

9.6 Il tag Figure

```
<!element fi - o (#pcdata) >
```

Con il tag figure si possono posizionare figure matematiche. I caratteri così marcati sono mappati direttamente come figura matematica. Si può sapere quale carattere viene mappato a quale figura guardando B (Figure matematiche).

9.7 Il tag Realfont

```
<!element rf - o (#pcdata) >
```

Questo tag posiziona un carattere reale dentro una formula matematica.²⁰ Le formule non sono permesse in questo tag.

```
<dm><rf/Binom:/ (a+b)<sup/2/=a<sup/2/+2ab+b<sup/2/</dm>
```

viene mappato come:

$$\text{Binom: } (a + b)^2 = a^2 + 2ab + b^2$$

9.8 Altri tag matematici

I rimanenti tag modificano semplicemente la formula marcata, senza richiedere altri tag. L'effetto è mostrato nella tabella 10 (Tag matematici senza tag inclusi)

nome	tag	esempio	risultato
<i>vector</i>	v	$\langle f \rangle \langle v/a \rangle \times \langle v/b \rangle = \langle v/0 \rangle \langle /f \rangle$	$\vec{a} \times \vec{b} = \vec{0}$
<i>overline</i>	ovl	$\langle f \rangle \langle ovl/1+1 \rangle = \langle ovl/2 \rangle \langle /f \rangle$	$\overline{1+1} = \overline{2}$
<i>underline</i>	unl	$\langle f \rangle \langle unl/1+1 \rangle = \langle unl/2 \rangle \langle /f \rangle$	$\underline{1+1} = \underline{2}$
<i>superior</i>	sup	$\langle f \rangle e = m \times c \langle sup/2 \rangle \langle /f \rangle$	$e = m \times c^2$
<i>inferior</i>	inf	$\langle f \rangle x \langle inf/i \rangle := 2x \langle inf/i-1 \rangle + 3 \langle /f \rangle$	$x_i := 2x_{i-1} + 3$

Table 10: Tag matematici senza tag inclusi

10 Etichette e riferimenti

```
<!entity % xref
  " label|ref|pageref|cite|url|htmlurl|ncite " >
```

Un documento più sofisticato necessiterà di riferimenti ad altri punti all'interno del documento.

²⁰Non sono proprio sicuro su rf. Cosa dovrebbe essere?

10.1 Il tag Label

```
<!element label - o empty>
<!attlist label id cdata #required>
```

Per fare riferimento a un punto, capitolo o sezione all'interno del proprio documento si usa un tag *label*.

Un esempio potrebbe essere:

```
<sect1>Benvenuti all'articolo<label id="intro">
<p>...
```

10.2 Il tag Reference

```
<!element ref - o empty>
<!attlist ref
      id cdata #required
      name cdata "{refnam}">
```

Con questo tag è possibile fare riferimento a un punto del proprio documento etichettato come in 10.1 (Il tag Label).

Il modo in cui il riferimento verrà mappato nel proprio documento dipende anche in questo caso dal mappatore. Può apparire come collegamento ipertestuale (HTML) o come numero di sezione (L^AT_EX).

10.3 Il tag Page Reference

```
<!element pageref - o empty>
<!attlist pageref
      id cdata #required>
```

Un esempio per pageref:

```
<pageref id="intro">
```

Nella mappatura HTML *pageref* non è utilizzabile, perché non ci sono numeri di pagina. Nella mappatura L^AT_EX il tag verrà mappato al numero di pagina dell'etichetta di riferimento.

10.4 Il tag Url

```
<!element url - o empty>
<!attlist url
      url cdata #required
      name cdata "{urlnam}" >
```

Un esempio per un *url*:

```
<url url="http://www.gnu.org" name="Organizzazione GNU">
```

Organizzazione GNU <http://www.gnu.org>

La mappatura HTML visualizza un collegamento ipertestuale nel proprio documento. Il riferimento è il valore dell'attributo *url*, il testo presente nel collegamento ipertestuale è il valore dell'attributo *name*.

La mappatura L^AT_EX visualizza il nome seguito dall'url.

10.5 Il tag Htmlurl

```
<!element htmlurl - o empty>
<!attlist htmlurl
    url cdata #required
    name cdata "{urlnam}" >
```

Un esempio per htmlurl:

```
<htmlurl url="http://www.gnu.org" name="Organizzazione GNU">
```

Organizzazione GNU

La sola differenza tra questo tag e il 10.4 (Tag Url) è nella mappatura \LaTeX .

La mappatura \LaTeX semplicemente elimina l'attributo url ed evidenzia il nome.

In tutti gli altri casi è assolutamente identico al *tag url*.

10.6 Il tag Cite

```
<!element cite - o empty>
<!attlist cite
    id cdata #required>
```

Per quanto ne so questo tag necessita di bibTeX per funzionare bene. Quindi mi spiace, ma non sono stato capace di usarlo. Per questa ragione sicuramente non sono la persona giusta per spiegare qualcosa al riguardo.

10.7 Il tag Ncite

```
<!element ncite - o empty>
<!attlist ncite
    id cdata #required
    note cdata #required>
```

Uguale al 10.6 (Tag Cite).

11 Indici

```
<!entity % index "idx|cdx|nidx|ncdx" >
```

```
<!element idx - - (#pcdata)>
<!element cdx - - (#pcdata)>
<!element nidx - - (#pcdata)>
<!element ncdx - - (#pcdata)>
```

I tag index servono a creare un indice analitico del proprio documento. Sono utili solo se si vuole la mappatura in \LaTeX . Differiscono leggermente come accennato nella tabella 11 (Elementi indice analitico).

tag	la mia traduzione
idx	indice analitico
cdx	indice analitico del codice (indice analitico via terminale)
nidx	indice analitico invisibile
ncdx	indice analitico del codice invisibile (indice analitico via terminale)

Table 11: Elementi indice analitico

11.1 Includere un indice analitico

Ci sono due modi per includere un indice nel proprio documento. Si guardino entrambi per decidere.

11.1.1 Manualmente

1. Impostare gli attributi opzionali della propria classe di documento in modo da contenere i pacchetti *makeidx*. Si può fare mediante: `<article opts="makeidx">`.
2. Evidenziare tutte le parole che si vogliono successivamente nell'indice con il *tag idx* o il *tag cdx*. Se la parola che si vuole indicizzare a una posizione nel proprio documento non si trova nel testo, basta semplicemente scriverla nella posizione che si vuole indicizzare con il *tag nidx*. Funziona come il normale *idx*, solo che il testo etichettato verrà silenziosamente eliminato nel documento normale.
3. Processare il proprio file con il comando `makeindex sgml2latex -m ilmiodocumento.sgml`. Questo produrrà un `miodocumento.idx` aggiuntivo.
4. Processare `miodocumento.idx` con il comando `makeindex`, così: `makeindex miodocumento.idx`. Questo produrrà un `miodocumento.ind` aggiuntivo.
5. Per includere l'indice così generato nel proprio documento lo si processi con il comando `sgml2latex -o tex -m miodocumento.sgml`. Il risultato è l'output di `miodocumento.tex`.
6. Modificare `miodocumento.tex` con il proprio editor preferito. Cercare la riga `\end{document}` (dovrebbe essere da qualche parte verso la fine del file) e inserire il testo `\printindex` davanti a questa riga.
7. Processare il file modificato con il comando `latex miodocumento.tex`. Ciò produrrà il `miodocumento.dvi` finale che, di nuovo, si può elaborare con il comando `dvips` per generare un documento postscript.

Un bel pasticcio, non è vero?

11.1.2 Hacked

Attualmente sto lavorando a una patch per gli strumenti sgml per automatizzare l'inclusione e la generazione di un indice. Per saperne lo stato attuale, vedere <http://www.bnhof.de/~uwe/lnd/indexpatch/index.html>.

12 Programmazione istruita

```
<!entity % litprog " code | verb " >
```

Questa è una cosa divertente. L'idea è non scrivere alcun commento testuale all'interno di un programma, e forse successivamente si useranno alcuni strumenti speciali, per estrarre il testo²¹, ma scrivere un grosso documento e poi estrarne il codice.²² Il principio è: tutto il testo all'interno dei tag `verb` e `code` verrà riunito in un file sorgente.

Questo è tutto, perché al momento non ricordo il nome dello strumento che fa questo.

13 Bibliografia

- *Definizione del tipo di documento quertz*
Norman Welsh
- *SGML-Tools Guida dell'utente 1.0 (\$Revision: 1.1 \$)*
Matt Welsh e Greg Hankins e Eric S. Raymond
Novembre 1997
- *Esempi rapidi di SGML, v1.0*
Matt Welsh, <mdw@cs.cornell.edu>
Marzo 1994

A Simboli nominati

A.1 Caratteri nominati

Quest è una lista leggermente modificata presa da [*SGML-Tools Guida dell'Utente 1.0 (\$Revision: 1.1 \$)*]. Se si nota qualche mancanza non si esiti a inviarmi una mail. Molti caratteri nominati mostrati nella tavola 12 (Caratteri nominati) sono gli stessi del dtd di html.

A.2 Spazi bianchi nominati

C'è un piccolo numero di "comunque li si vogliono chiamare". L'apparenza è simile ai caratteri nominati, ma verranno stampati solo alcune volte, o addirittura per niente.

thinsp

Spazio sottile:

`d D` → d D

emsp

Spazio evidenziato: `d D` → d D

ensp

Spazio normale: `/d D/` → d D

nbsp

Spazio non interrotto: Uno spazio al quale la linea non può essere interrotta. Due parole separate dal `nbsp` verranno trattate dall'analizzatore e mappatore come una lunga unica parola.

²¹Si pensi a `perlpod`.

²²Chi non ama documentare il proprio codice non l'apprezzerà.

AElig	Æ	Aacute	Á	Acirc	Â	Ae	Ä	Agrave	À	Atilde	Ã
Auml	Ä	Ccedil	Ç	Eacute	É	Egrave	È	Euml	Ë	Iacute	Í
Icirc	Î	Igrave	Ì	Iuml	Ï	Ntilde	Ñ	Oacute	Ó	Ocirc	Ô
Oe	Ö	Ograve	Ò	Oslash	Ø	Ouml	Ö	Uacute	Ú	Ue	Û
Ugrave	Û	Uuml	Ü	Yacute	Ý	aacute	á	acirc	â	ae	ä
aelig	æ	agrave	à	amp	&	apos	'	aring	â	arr	↓
ast	*	atilde	ã	auml	ä	bsol	\	bull	•	ccedil	ç
cir	o	circ	^	clubs	♣	colon	:	comma	,	commat	@
copy	©	darr	↓	deg	°	diams	◇	divide	÷	dollar	\$
dquot	"	eacute	é	ecirc	ê	egrave	è	equals	=	etago	</
euml	ë	excl	!	frac12	$\frac{1}{2}$	frac14	$\frac{1}{4}$	frac18	$\frac{1}{8}$	frac34	$\frac{3}{4}$
frac38	$\frac{3}{8}$	frac58	$\frac{5}{8}$	frac78	$\frac{7}{8}$	gt	>	half	$\frac{1}{2}$	hearts	♥
hellip	...	horbar	—	hyphen	-	iacute	í	icirc	î	iexcl	¡
igrave	ì	iquest	¿	iuml	ï	laquo	«	larr	←	lcub	{
ldquo	“	lowbar	_	lpar	(lsqb	[lsquo	‘	lt	<
mdash	—	micro	μ	middot	·	mu	μ	ndash	—	not	¬
ntilde	ñ	num	#	oacute	ó	ocirc	ô	oe	ö	ograve	ò
ohm	Ω	ordf	$\overset{a}{_}$	ordm	$\overset{e}{_}$	oslash	ø	otilde	õ	ouml	ö
para	¶	percent	%	period	.	plus	+	plusmn	±	pound	£
quest	?	quot	"	raquo	»	rarr	→	rcub	}	rdquo	”
reg	®	rpar)	rsqb]	rsquo	'	sect	§	semi	;
sol	/	spades	♠	sup1	¹	sup2	²	sup3	³	sz	ß
szlig	ß	tilde	~	times	×	trade	™	uacute	ú	uarr	↑
ucirc	û	ue	ü	ugrave	ù	uuml	ü	verbar		yacute	ý

Table 12: Caratteri nominati

shy

Trattino suggerito: Se il mappatore sta per interrompere una parola che ha al suo interno il tag `shy`, probabilmente interromperà la parola dove c'è il tag `shy` e metterà al suo posto un *trattino*. Se non è necessaria alcuna interruzione di parola, il tag `shy` non espande niente.

B Figure matematiche

$a - \dagger$	$b - [$	$c -]$	$d - \lceil$	$A - \mathcal{A}$	$B - \mathcal{B}$	$C - \mathcal{C}$	$D - \mathcal{D}$
$e - \rfloor$	$f - \{$	$g - \}$	$h - \langle$	$E - \mathcal{E}$	$F - \mathcal{F}$	$G - \mathcal{G}$	$H - \mathcal{H}$
$i - \rangle$	$j - $	$k - $	$l - \ddagger$	$I - \mathcal{I}$	$J - \mathcal{J}$	$K - \mathcal{K}$	$L - \mathcal{L}$
$m - \Updownarrow$	$n - \backslash$	$o - \wr$	$p - \surd$	$M - \mathcal{M}$	$N - \mathcal{N}$	$O - \mathcal{O}$	$P - \mathcal{P}$
$q - \amalg$	$r - \nabla$	$s - \int$	$t - \sqcup$	$Q - \mathcal{Q}$	$R - \mathcal{R}$	$S - \mathcal{S}$	$T - \mathcal{T}$
$u - \sqcap$	$v - \sqsubseteq$	$w - \sqsupseteq$	$x - \S$	$U - \mathcal{U}$	$V - \mathcal{V}$	$W - \mathcal{W}$	$X - \mathcal{X}$
$y - \dagger$	$z - \ddagger$			$Y - \mathcal{Y}$	$Z - \mathcal{Z}$		

Table 13: Figure matematiche

La mappatura speciale per i caratteri utilizzabili per costruire figure matematiche sono mostrate nella tabella 13 (Figure matematiche).

C Il dtd di Linuxdoc per il sorgente

È il `linuxdoc.dtd` usato per analizzare questo documento. Il registro di revisione, i commenti di revisione e alcune altre righe ridondanti sono state tolte per risparmiare carta e spazio dello schermo.

```

<!-- This is a DTD, but will be read as -*- sgml -*- -->
<!-- ===== -->
<!-- $Id: Linuxdoc-Reference.sgml,v 1.1 2001/04/02 13:52:31 gferg Exp $

This is LINUXDOC96 DTD for SGML-Tools.

This was LINUXDOC.DTD,
a hacked version of QWERTZ.DTD v1.3 by Matt Welsh,
Greg Hankins, Eric Raymond, Marc Baudoin and
Tristan Debeaupuis; modified from QWERTZ.DTD by
Tom Gordon.

<!entity % emph
    " em|it|bf|sf|sl|tt|cparam " >

<!entity % index "idx|cdx|nidx|ncdx" >

<!-- url added by HG; htmlurl added by esr -->
<!entity % xref
    " label|ref|pageref|cite|url|htmlurl|ncite " >

<!entity % inline
    " (#pcdata | f| x| %emph; |sq| %xref | %index | file )* " >

<!entity % list
    " list | itemize | enum | descrip " >

<!entity % par
    " %list; | comment | lq | quote | tscreen " >

<!entity % mathpar " dm | eq " >

<!entity % thrm
    " def | prop | lemma | coroll | proof | theorem " >

<!entity % litprog " code | verb " >

<!entity % sectpar
    " %par; | figure | tabular | table | %mathpar; |
    %thrm; | %litprog; ">
<!element linuxdoc o o
    (sect | chapt | article | report |
    book | letter | telefax | slides | notes | manpage ) >

<!-- 'general' entity replaced with ISO entities - kwm -->
<!entity % isoent system "isoent">

```

```

%isoent;

<!entity urlnam sdata "urlnam" >
<!entity refnam sdata "refnam" >
<!entity tex sdata "[tex ]" >
<!entity latex sdata "[latex ]" >
<!entity latexe sdata "[latex]" >
<!entity tm sdata "[trade ]" >
<!entity dquot sdata "[quot ]" >
<!entity ero sdata "[amp ]" >
<!entity etago '</' >
<!entity Ae '&Auml;' >
<!entity ae '&auml;' >
<!entity Oe '&Ouml;' >
<!entity oe '&ouml;' >
<!entity Ue '&Uuml;' >
<!entity ue '&uuml;' >
<!entity sz '&szlig;' >
<!element p o o (( %inline | %sectpar )+) +(newline) >
<!entity ptag '<p>' >
<!entity psplit '</p><p>' >

<!shortref pmap
    "&#RS;B" null
    "&#RS;B&#RE;" psplit
    "&#RS;&#RE;" psplit
--    ''' qtag --
    "[" lsqb
    "~" nbsp
    "_" lowbar
    "#" num
    "%" percent
    "^" circ
    "{" lcub
    "}" rcub
    "|" verbar >

<!usemap pmap p>
<!element em - - (%inline)>
<!element bf - - (%inline)>
<!element it - - (%inline)>
<!element sf - - (%inline)>
<!element sl - - (%inline)>
<!element tt - - (%inline)>
<!element sq - - (%inline)>
<!element cparam - - (%inline)>

<!entity ftag '<f>' -- formula begin -- >
<!entity qendtag '</sq>'>

```

```

<!shortref sqmap
    "&#RS;B" null
--    '"" qendtag --
    "[" lsqb
    "~" nbsp
    "_" lowbar
    "#" num
    "%" percent
    "^" circ
    "{" lcub
    "}" rcub
    "|" verbar >

<!usemap sqmap sq >

<!element lq - - (p*)>
<!element quote - - ((%inline; | %sectpar;)*, p*)+ >
<!element tscreen - - ((%inline; | %sectpar;)*, p*)+ >
<!element itemize - - (item+)>
<!element enum - - (item+)>
<!element list - - (item+)>

<!shortref desmap
    "&#RS;B" null
    "&#RS;B&#RE;" ptag
    "&#RS;&#RE;" ptag
    "~" nbsp
    "_" lowbar
    "#" num
    "%" percent
    "^" circ
    "[" lsqb
    "]" rsqb
    "{" lcub
    "}" rcub
    "|" verbar >

<!element descrip - - (tag?, p+)+ >
<!usemap desmap descrip>

<!element item o o ((%inline; | %sectpar;)*, p*) >

<!element tag - o (%inline)>
<!usemap desmap tag>

<!usemap global (list,itemize,enum)>
<!entity space " ">
<!entity null "">

<!--

```

```

<!shortref bodymap
  "&#RS;B&#RE;" ptag
  "&#RS;&#RE;" ptag
  ''' qtag
  "[" lsb
  "~" nbsp
  "_" lowbar
  "#" num
  "%" percent
  "^" circ
  "{" lcub
  "}" rcub
  "|" verbar>
-->

<!element figure - - ((eps | ph ), img*, caption?)>
<!attlist figure
  loc cdata "tbp"
  caption cdata "Caption">

<!-- eps attributes added by mb and td -->
<!element eps - o empty >
<!attlist eps
  file cdata #required
  height cdata "5cm"
  angle cdata "0">

<!element ph - o empty >
<!attlist ph
  vspace cdata #required>

<!element img - o empty>
<!attlist img
  src cdata #required>

<!element caption - o (%inline)>

<!shortref oneline
  "B&#RE;" space
  "&#RS;&#RE;" null
  "&#RS;B&#RE;" null
--  ''' qtag --
  "[" ftag
  "~" nbsp
  "_" lowbar
  "#" num
  "%" percent
  "^" circ
  "{" lcub
  "}" rcub

```

```

    "|" verbar>

<!usemap oneline tag>
<!usemap oneline caption>

<!entity % tabrow "(%inline, (colsep, %inline)*)" >
<!element tabular - -
    (hline?, %tabrow, (rowsep, hline?, %tabrow)*, caption?) >

<!attlist tabular
    ca cdata #required>

<!element rowsep - o empty>
<!element colsep - o empty>
<!element hline - o empty>

<!entity rowsep "<rowsep>">
<!entity colsep "<colsep>">

<!shortref tabmap
    "&#RE;" null
    "&#RS;&#RE;" null
    "&#RS;B&#RE;" null
    "&#RS;B" null
    "B&#RE;" null
    "BB" space
    "@" rowsep
    "|" colsep
    "[" ftag
--    "'" qtag --
    "_" thinsp
    "~" nbsp
    "#" num
    "%" percent
    "^" circ
    "{" lcub
    "}" rcub >

<!usemap tabmap tabular>
<!element table - - (tabular, caption?) >
<!attlist table
    loc cdata "tbp">

<!element code - - rcdada>
<!element verb - - rcdada>

<!shortref ttmap -- also on one-line --
    "B&#RE;" space
    "&#RS;&#RE;" null
    "&#RS;B&#RE;" null

```

```

    "&#RS;B" null
    '#'      num
    '%'      percnt
    '~'      tilde
    '_'      lowbar
    '^'      circ
    '{'      lcub
    '}'      rcub
    '|'      verbar >

<!usemap tmap tt>
<!element mc - - cdata >
<!entity % sposs      "tu" >
<!entity % fcs        "%sposs;|phr" >
<!entity % fcstxt     "#pcdata|mc|/%fcs;" >
<!entity % fscs       "rf|v|fi" >
<!entity % limits    "pr|in|sum" >
<!entity % fbu        "fr|lim|ar|root" >
<!entity % fph        "unl|ovl|sup|inf" >
<!entity % fbutxt     "(%fbu;) | (%limits;) |
                      (%fcstxt;)|(%fscs;)|(%fph;)" >
<!entity % fphtxt     "p|#pcdata" >
<!element f          - - ((%fbutxt;)* >

<!entity fendtag '</f>' -- formula end -- >

<!shortref fmap
    "&#RS;B" null
    "&#RS;B&#RE;" null
    "&#RS;&#RE;" null
    "_" thinsp
    "~" nbsp
    "]" rsqb
    "#" num
    "%" percnt
    "^" circ
    "{" lcub
    "}" rcub
    "|" verbar>

<!usemap fmap f >

<!element dm - - ((%fbutxt;)*>
<!element eq - - ((%fbutxt;)*>

<!shortref dmmmap
    "&#RE;" space
    "_" thinsp
    "~" nbsp
    "]" rsqb

```

```

    "#" num
    "%" percent
    "^" circ
    "{" lcub
    "}" rcub
    "|" verbar>

<!usemap dmmmap (dm,eq)>
<!element fr      - - (nu,de) >
<!element nu      o o ((%fbutxt;)* >
<!element de      o o ((%fbutxt;)* >
<!element ll      o o ((%fbutxt;)* >
<!element ul      o o ((%fbutxt;)* >
<!element opd     - o ((%fbutxt;)* >
<!element pr      - - (ll,ul,opd?) >
<!element in      - - (ll,ul,opd?) >
<!element sum     - - (ll,ul,opd?) >
<!element lim     - - (op,ll,ul,opd?) >
<!element op      o o (%fcstxt;|rf|%fph;) -(tu) >
<!element root    - - ((%fbutxt;)* >
<!attlist root
    n cdata "">
<!element col o o ((%fbutxt;)* >
<!element row o o (col, (arc, col)* >

<!element ar      - - (row, (arr, row)* >
<!attlist ar
    ca cdata #required >
<!element arr     - o empty >
<!element arc     - o empty >
<!entity arr "<arr>" >
<!entity arc "<arc>" >

<!shortref armap
    "&#RE;" space
    "@" arr
    "|" arc
    "_" thinsp
    "~" nbsp
    "#" num
    "%" percent
    "^" circ
    "{" lcub
    "}" rcub >

<!usemap armap ar >
<!element sup     - - ((%fbutxt;)* -(tu) >
<!element inf     - - ((%fbutxt;)* -(tu) >
<!element unl - - ((%fbutxt;)* >
<!element ovl - - ((%fbutxt;)* >

```

```

<!element rf - o (#pcdata) >
<!element phr - o ((%fphtxt;)* >
<!element v - o ((%fcstxt;)*
      -(tu|%limits;|%fbu;|%fph;) >
<!element fi - o (#pcdata) >
<!element tu - o empty >

<!usemap global (rf,phr)>
<!element def - - (thtag?, p+) >
<!element prop - - (thtag?, p+) >
<!element lemma - - (thtag?, p+) >
<!element coroll - - (thtag?, p+) >
<!element proof - - (p+) >
<!element theorem - - (thtag?, p+) >
<!element thtag - - (%inline)>

<!usemap global (def,prop,lemma,coroll,proof,theorem)>
<!usemap oneline thtag>
<!entity qtag ' <sq>' >

<!shortref global
      "&#RS;B" null -- delete leading blanks --
      -- ''' qtag --
      "[" ftag
      "~" nbsp
      "_" lowbar
      "#" num
      "%" percent
      "^" circ
      "{" lcub
      "}" rcub
      "|" verbar>

<!usemap global linuxdoc>
<!element label - o empty>
<!attlist label id cdata #required>

<!-- ref modified to have an optional name field HG -->
<!element ref - o empty>
<!attlist ref
      id cdata #required
      name cdata "&refnam">

<!-- url entity added to have direct url references HG -->
<!element url - o empty>
<!attlist url
      url cdata #required
      name cdata "&urlnam" >

<!-- htmlurl entity added to have quieter url references esr -->

```

```
<!element htmlurl - o empty>
<!attlist htmlurl
    url cdata #required
    name cdata "&urlnam" >

<!element pageref - o empty>
<!attlist pageref
    id cdata #required>
<!element comment - - (%inline)>
<!element x - - ((#pcdata | mc)*) >
<!usemap #empty x >

<!-- Hacked by mdw to exclude abstract; abstract now part of titlepag -->
<!element article - -
    (titlepag, header?,
    toc?, lof?, lot?, p*, sect*,
    (appendix, sect+)?, biblio?) +(footnote)>

<!attlist article
    opts cdata "null">

<!-- Hacked by mdw to exclude abstract; abstract now part of titlepag -->
<!element report - -
    (titlepag, header?, toc?, lof?, lot?, p*,
    chapt*, (appendix, chapt+)?, biblio?) +(footnote)>

<!attlist report
    opts cdata "null">
<!element book - -
    (titlepag, header?, toc?, lof?, lot?, p*, chapt*,
    (appendix, chapt+)?, biblio?) +(footnote) >

<!attlist book
    opts cdata "null">

<!-- Hacked by mdw, abstract now part of titlepag -->
<!element titlepag o o (title, author, date?, abstract?)>
<!element title - o (%inline, subtitle?) +(newline)>
<!element subtitle - o (%inline)>
<!usemap oneline titlepag>
<!element author - o (name, thanks?, inst?,
    (and, name, thanks?, inst?)*)>
<!element name o o (%inline) +(newline)>
<!element and - o empty>
<!element thanks - o (%inline)>
<!element inst - o (%inline) +(newline)>
<!element date - o (#pcdata) >

<!usemap global thanks>
```

```
<!element newline - o empty >
<!entity nl "<newline>">

<!-- Hacked by mdw -->
<!element abstract - o (%inline)>
<!usemap oneline abstract>

<!element toc - o empty>
<!element lof - o empty>
<!element lot - o empty>
<!element header - - (lhead, rhead) >
<!element lhead - o (%inline)>
<!element rhead - o (%inline)>
<!entity % sect "heading, header?, p* " >
<!element heading o o (%inline)>
<!element chapt - o (%sect, sect*) +(footnote)>
<!element sect - o (%sect, sect1*) +(footnote)>
<!element sect1 - o (%sect, sect2*)>
<!element sect2 - o (%sect, sect3*)>
<!element sect3 - o (%sect, sect4*)>
<!element sect4 - o (%sect)>
<!usemap oneline (chapt,sect,sect1,sect2,sect3,sect4)>
<!element appendix - o empty >
<!element footnote - - (%inline)>
<!usemap global footnote>
<!element cite - o empty>
<!attlist cite
    id cdata #required>

<!element ncite - o empty>
<!attlist ncite
    id cdata #required
    note cdata #required>

<!element file - - (#pcdata)>

<!element idx - - (#pcdata)>
<!element cdx - - (#pcdata)>
<!element nidx - - (#pcdata)>
<!element ncdx - - (#pcdata)>

<!element biblio - o empty>
<!attlist biblio
    style cdata "linuxdoc"
    files cdata "">
<!element slides - - (slide*) >

<!attlist slides
    opts cdata "null">
```

```
<!element slide - o (title?, p+) >
<!entity % addr "(address?, email?, phone?, fax?)" >

<!element letter - -
    (from, %addr, to, %addr, cc?, subject?, sref?, rref?,
     rdate?, opening, p+, closing, encl?, ps?)>

<!attlist letter
    opts cdata "null">

<!element from          - o (#pcdata) >
<!element to            - o (#pcdata) >

<!usemap oneline (from,to)>

<!element address      - o (#pcdata) +(newline) >
<!element email        - o (#pcdata) >
<!element phone        - o (#pcdata) >
<!element fax          - o (#pcdata) >

<!element subject      - o (%inline;) >
<!element sref         - o (#pcdata) >
<!element rref         - o (#pcdata) >
<!element rdate        - o (#pcdata) >

<!element opening      - o (%inline;) >
<!usemap oneline opening>

<!element closing - o (%inline;) >
<!element cc - o (%inline;) +(newline) >
<!element encl - o (%inline;) +(newline) >

<!element ps - o (p+) >

<!element telefax - -
    (from, %addr, to, address, email?,
     phone?, fax, cc?, subject?,
     opening, p+, closing, ps?)>

<!attlist telefax
    opts cdata "null"
    length cdata "2">

<!element notes - - (title?, p+) >
<!attlist notes
    opts cdata "null" >
<!element manpage - - (sect1*)
    -(sect2 | f | %mathpar | figure | tabular |
     table | %xref | %thrm )>
```

```
<!attlist manpage
    opts cdata "null"
    title cdata ""
    sectnum cdata "1" >
<!shortref manpage
    "&#RS;B" null
--    ''' qtag --
    "[" ftag
    "~" nbsp
    "_" lowbar
    "#" num
    "%" percent
    "^" circ
    "{" lcub
    "}" rcub
    "|" verbar>

<!usemap manpage  manpage >
```
