Claudio Beccari

Creare file archiviabili con pdfLaTeX e LuaLaTeX

Versione v.3.0.02 del 2020-05-12

Associati anche tu al GUIT

Fai click per associarti

L'associazione per la diffusione di TEX in Italia, riconosciuta ufficialmente in ambito internazionale, si sostiene *unicamente* con le quote sociali.

Se anche tu trovi che questa guida tematica gratuita ti sia stata utile, il mezzo principale per ringraziare gli autori è diventare socio.

Divenendo soci si ricevono gratuitamente:

- l'abbonamento alla rivista ArsTEXnica;
- il DVD TEX Collection;
- un eventuale oggetto legato alle attività del GUIT.

L'adesione al GUIT prevede un quota associativa compresa tra 12,00 € e 70,00 € a seconda della tipologia di adesione prescelta e ha validità per l'anno solare in corso.

Creare file archiviabili con pdfLATEX e LuaLATEX

Copyright © 2020, Claudio Beccari

Questa documentazione è soggetta alla licenza LPPL (LATEX Project Public Licence), versione 1.3 o successive; il testo della licenza è sempre contenuto in qualunque distribuzione del sistema TEX e nel sito http://www.latex-project.org/lppl.txt.

Questo documento è curato da Claudio Beccari.

Presentazione

Archiviare documenti è una attività che si svolge da millenni; sembra che la praticassero persino i Sumeri prima ancora degli Assiro-Babilonesi. Oggi gli archivi straboccano di documenti cartacei, libri, rapporti, atti giuridici, tesi di laurea, progetti di costruzione, eccetera. Tutti questi documenti dopo molti anni sono spesso preda di roditori, tarli, muffe; la conservazione dei testi archiviati più importanti dà molto lavoro ai restauratori, ma da molto tempo è sentita la necessità di archiviare in formato "elettronico" almeno certi documenti, in modo da smaterializzarli e sottrarli all'usura del tempo.

Per questi motivi l'International Standard Organization ha stabilito le norme per l'archiviabilità a lungo termine di documenti elettronici. Questa guida intende spiegare quali siano i requisiti per l'archiviabilità elettronica e come fare per generare tali documenti.

Nella Guida tematica *Tipocomporre in italiano* (Beccari, 2016) ho descritto per sommi capi il problema dell'archiviabilità a lungo termine. Qui ripeterò alcuni concetti, ma nel corpo di questa guida esporrò come a partire dal 2016 chiunque possa comporre con i programmi del sistema TEX (almeno certi) documenti archiviabili.

Talvolta, però, l'utente del sistema TEX si trova nella necessità di archiviare documenti che non sono composti con i programmi di tipocomposizione di questo sistema; quindi accennerò anche a come usare i programmi di questo sistema per rendere archiviabili tali documenti almeno quando è possibile farlo.

Dal novembre 2016 il pacchetto *pdfx* del sistema TEX, predisposto per comporre documenti conformi alle norme PDF/A, per l'archiviazione, o PDF/X, per la stampa, riesce a fare un ottimo lavoro con le norme PDF/X; invece per quel che riguarda l'archiviazione non riesce ancora a gestire correttamente certe norme, in quanto nessun programma di

tipocomposizione del sistema TEX basato su LATEX[1] per ora è in grado di produrre file PDF che siano classificabili come `Tagged PDF`[2]; non appena i programmi di tipocomposizione del sistema TEX saranno in grado di produrre tali file, il pacchetto *pdfx* è già attrezzato per renderli conformi alle norme (C.V. Radhakrishnan *et al.*, 2016). Oggi si è al corrente che esistono diversi progetti in seno al TEX Users Group per generare questi tipi di file, ma per ora nessuno dà ancora risultati validi con qualunque struttura di documento. Naturalmente continuo a riferirmi ai programmi basati sul mark-up LATEX. ConTEXt, invece, riesce già a produrli, ma non è basato su quel mark-up. Tra gli altri esiste il pacchetto (ancora sperimentale) *tagpdf*, già disponibile nelle distribuzioni del sistema TEX, che potrebbe essere un buon punto per cominciare a sperimentare la creazione dei file `Tagged PDF`, (Fischer, 2018).

Sembra che altri programmi di word processing e di tipocomposizione siano in grado di produrre file di tipo `Tagged PDF`; questa qualifica è necessaria anche per le pubblicazioni elettroniche che passano sotto il nome generico di *e-book* secondo certe norme. Tuttavia per ora il sistema TEX non è ancora in grado di produrre questo tipo di file, tra cui, per esempio, testi in formato E-PUB[3].

Alcuni word processor come Microsoft Office Word, Libre Office Writer, e Open Office Writer dispongono di comandi per salvare i file non solo nei

[1] Sottolineo l'espressione "basato su LATEX". Infatti il programma ConTEXt, che non è basato su LATEX, è in grado di produrre file di tipo `Tagged PDF`. Siccome LuaLATEX usa lo stesso motore di composizione `luatex` di ConTEXt, anche LuaLATEX dovrebbe essere in grado di produrre file di tipo `Tagged PDF`, ma forse gli autori del pacchetto *pdfx* non hanno ancora trovato il modo di conciliare queste due cose.

[2] Un file `Tagged PDF`, è un file che contiene informazioni sulla struttura logica del documento; i "tag" che vengono inseriti nel documento sono comandi in linguaggio XML che ne descrivono la struttura logica. In un certo senso sono analoghi alla gerarchia dei bookmarks generati dal pacchetto *hyperref* quando permette al visualizzatore del file PDF di vedere in una barra laterale la struttura di parti, capitoli, sezioni, e simili, che consentono di eseguire una facile navigazione del documento quando lo si legge sullo schermo. Il tagging dei file PDF, però è molto più complesso di quanto descritto qui in questa nota.

[3] Recentemente TEXLive è stato arricchito di un nuovo pacchetto che afferma di poter creare file HTML trasformabili in formato E-PUB; si tratta del pacchetto *lwarp*, (Burn, 2018); non l'ho controllato, perché ho perso ogni interesse per questo formato, non adatto, secondo me, ai testi tecnico-scientifici. Ma non è vietato che altri lo provino e controllino se davvero il file che questo pacchetto riesce a produrre sia davvero un "tagged file".

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

loro formati aperti o proprietari, ma anche per esportarli in formato PDF conformi alle norme per l'archiviabilità e, mediante opportune impostazioni, affermano di poterli salvare anche come `Tagged PDF`. Le prove che ho fatto con alcuni documenti composti con questi word processor hanno prodotto effettivamente dei file PDF, ma nessuno dei file che ho usato è risultato conforme alle norme, né a quelle relative al formato semplice, né a quello tagged. Questo non vuol dire che quei programmi falliscano sempre (infatti, vedi il paragrafo 5), ma hanno fallito con certi miei documenti non puramente testuali che ho cercato di verificare.

Nel seguito si presenteranno anche dei metodi per trasformare file PDF comunque ottenuti in file compatibili con lo standard PDF/A. Ma "comunque ottenuti" non vuol dire che possono contenere ogni cosa possibile immaginabile che un file PDF possa contenere; quindi quei file "comunque ottenuti" già in partenza non devono contenere cose vietate dalla norma. Leggendo con attenzione le sezioni che seguono, si capisce che cosa un file PDF *non deve* oppure *deve* contenere per poter essere archiviabile. Tra ciò che deve essere contenuto nel file e ciò che è vietato resta un ampio spazio di possibilità alcune conformi e altre non conformi con l'archiviabilità ed è difficile specificare ciò che è *necessario e sufficiente* perché un file PDF sia conforme.

Infatti produrre un file PDF potenzialmente archiviabile non è una cosa semplice; bisogna rispettare molti dettagli delicati e spesso trascurati. Nei primi capitoli cercherò di elencare questi elementi che alla fin fine si rivelano elementi essenziali, tutt'altro che dettagli.

Infine descriverò come verificare che un file PDF potenzialmente archiviabile lo sia davvero.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

Indice

Presentazione 3

1 Le norme PDF/A 9
1.1 I metadati 12
1.2 I colori 13
1.3 I file immessi 16
1.4 I font difettosi 17
1.5 I collegamenti ipertestuali 19

2 Trasformazione di file PDF in file archiviabili 22
2.1 Trasformazione delle immagini 22
2.1.1 Immagini a matrici di punti 23
2.1.2 Immagini vettoriali 23
2.2 Il documento non è archiviabile 26
2.3 Il documento manca solo dei metadati 27
2.4 Verifica della conformità 31

3 Creazione diretta di file archiviabili con pdflatex 32
3.1 Figure e file da includere 32
3.2 Struttura del preambolo con pdflatex 32
3.3 Le opzioni di *hyperref* 38
3.4 La verifica di conformità 39

4 Creazione diretta di file archiviabili con lualatex 40
4.1 Verifica delle figure e dei file testuali 40
4.2 Il preambolo del documento 41
4.3 Commenti alla compilazione con lualatex 43

5 Verifica della conformità 45

Conclusione 49

Bibliografia 50

Indice analitico 51

Elenco delle figure

1.1 Il Bay Bridge di Sidney 15
1.2 Messaggio emesso dai programmi Adobe Acrobat e Adobe Reader quando aprono un file che potrebbe essere PDF/A compatibile 20

5.1 Finestra di dialogo di verapdf-gui con un responso negativo dopo aver eseguito l'analisi di un file 46
5.2 Finestra di dialogo di verapdf-gui con un responso positivo dopo aver eseguito l'analisi di un file 46

Le norme PDF/A 1

L'International Standards Organisation (ISO) ha emanato in anni successivi diverse norme. Le prime nel 2005 (ISO 19005-1, 2005) con il livello 1, suddivise nelle due varianti PDF/A-1a e PDF/A-1b. Poi ha esteso queste norme nel 2011 con il livello 2 (ISO 19005-2, 2011), e nel 2012 con il livello 3 (ISO 19005-3, 2012). Non scendo nei dettagli di tutte queste norme, che alzano via via il livello delle funzionalità accettabili del linguaggio PDF; infatti ammettono la presenza di elementi multimediali all'interno dei file PDF, estendono il numero di profili di colore, eccetera. Una buona descrizione in italiano di ciascuna di queste norme si trova in (C.V. Radhakrishnan *et al.*, 2016).

Qui ci occupiamo solo dei file compatibili con gli standard PDF/A-1b e PDF/A2b, considerati sufficiente per la maggior parte delle necessità di archiviazione; non ci occupiamo degli standard PDF/A-1a e PDF/A-2a che richiedono i file `Tagged PDF`. D'altra parte ogni sottoversione "a" differisce solo per la specificazione di `Tagged PDF` dalla corrispondente sottoversione "b". Disponendo di file `Tagged PDF` che mancano soltanto dei metadati necessari per essere considerati di categoria "a" compatibili, i procedimenti indicati nel seguito possono produrre file conformi alle norme anche in questo caso.

I file compatibili con le norme, devono rispettare un certo numero di requisiti che qui cerco di riassumere senza scendere in troppi dettagli.

1. Con i livelli di compatibilità 1 e 2, di cui ci occupiamo qui, il linguaggio PDF usato all'interno del file deve essere obbligatoriamente di livello 1.4 per il livello 1, ma può essere superiore per il livello 2. Dal 2005 ad oggi il linguaggio PDF è evoluto, così come sono evolute le norme ISO-19005 per l'archiviabilità, cosicché con norme successive si accetta ora anche fino al livello 1.7 del linguaggio PDF. Per quel che riguarda

il sistema TEX non ci sono problemi perché produce file PDF di livello 1.4 del linguaggio PDF.

2. Ogni file che vuole essere conforme con lo standard PDF/A deve disporre di metadati adeguati; la difficoltà maggiore è proprio quella di inserire tutti i metadati necessari nella forma corretta e con i dati corretti; il pacchetto *pdfx*, usato dai programmi di tipocomposizione del sistema TEX introduce nei file da archiviare tutti i metadati necessari e potenzialmente in forma corretta; "potenzialmente" si limita ai pochi metadati forniti obbligatoriamente dall'utente: se questi li specifica in modo scorretto, evidentemente *pdfx* non è in grado di correggerli.
3. L'operatore che vuole produrre file archiviabili secondo la norma mediante il pacchetto *pdfx* dispone di un certo numero di macro per inserire i metadati specifici del documento, come la lista degli autori, l'ente o l'organizzazione che ha prodotto il documento, il suo titolo, una frase descrittiva del suo contenuto, una lista di parole chiave, tutte informazioni che non possono essere impostate in modo predefinito, ma che sono necessarie per poterle estrarre con i programmi di biblioteconomia in dotazione agli archivi e alle biblioteche.
4. Perché questi dati siano accessibili ai suddetti programmi, *non* devono essere contenuti nel file PDF in forma compressa[1], come tutto il resto del documento. Infatti il formato PDF non solo ricorre ad un suo speciale linguaggio per la descrizione di ogni pagina e di ogni oggetto che viene usato nel file, ma è anche fortemente compresso, talvolta criptato, in modo da salvaguardare un certo livello di sicurezza informatica e da ridurre lo spazio occupato nelle memorie (dischi o simili memorie di massa) degli archivi elettronici. La compressione va benissimo, ma i file conformi agli standard PDF/A non devono essere criptati.
5. Fra i metadati deve essere chiaramente specificato un profilo di colore: solo bianco e nero, tonalità di grigio, colori additivi, eccetera; questo

[1] Questo è il motivo per il quale diventa difficile produrre file PDF archiviabili prodotti usando il programma xelatex, in quanto questo, nel suo funzionamento normale, comprime sempre tutto, compresi eventuali metadati. Più avanti si farà un cenno per superare questa difficoltà, ma in buona sostanza quello che può fare xelatex, generalmente lo fa, anche meglio, lualatex; quindi la necessità di usare xelatex per produrre file PDF archiviabili svanisce quasi del tutto.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

perché sia chiaro ed esplicito il modo con cui sono codificati i colori degli elementi contenuti nel file PDF. Di fatto è bene che questo profilo si astenga dalle codifiche con meno bit, e specifichi solo profili di colore RGB (colori additivi Red, Green, Blue). Ne segue che tutti gli oggetti colorati contenuti nel file PDF devono essere coerenti con il profilo di colore dichiarato.

6. Con il livello 1 delle norme di archiviabilità non sono ammesse le trasparenze in qualunque oggetto del documento, generalmente figure o disegni; spesso contengono tali caratteristiche quando sono prodotti con programmi di origini diverse. Con il livello 2 sembra che siano ammesse. Con il livello 1 i file importati di fatto devono essere solo file PDF a loro volta compatibili con le specifiche di questo livello, anche se qualche file in formato JPG viene importato senza problemi; con il livello 2, invece, i tipi di file da importare possono essere di qualunque tipo, anche PNG, JPG o JPEG o J2000.
7. I font usati nel documento direttamente o indirettamente devono essere inclusi nel file PDF, come minimo devono essere inclusi tutti i glifi di ogni font usato nel documento. Quando dico direttamente intendo tutti i font che chi ha predisposto il file PDF ha esplicitamente indicato. Sono invece font usati indirettamente quelli contenuti nei file inclusi. Questa operazione non è tanto ovvia; di solito i programmi che producono file PDF, non includono gli 11 font che sono già presenti in ogni stampante in grado di ricevere in ingresso file PDF. Ora i file archiviabili non devono necessariamente essere stampati, ma devono anche essere letti direttamente sullo schermo e (i driver per) gli schermi generalmente non dispongono di font preinstallati. I programmi di tipocomposizione del sistema TEX provvedono all'inclusione di tutti i font, ma questo potrebbe non essere vero per file PDF prodotti con altri programmi o scaricati dalla rete.
8. Ogni font usato deve avere i suoi glifi, ciascuno con il suo nome ufficiale descritto nello standard UNICODE; per i font conformi alle codifiche OT1, T1, LGR ed alcune altre, il sistema TEX provvede autonomamente ma per font commerciali o poco conformi con UNICODE, questo potrebbe non essere vero.
9. Ogni glifo di ogni font usato nel documento deve essere dotato di dimensioni coerenti e, specialmente, nessuna dimensione deve essere nulla, almeno per i livello 1; per il livello 2 ho potuto constare che

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

non è un requisito obbligatorio. Questo requisito è assente da alcuni glifi matematici coerenti con la codifica OMS (simboli matematici) usati normalmente dai programmi di tipocomposizione del sistema TEX: sempre da pdflatex, talvolta anche da lualatex quando questo è impostato per non usare i font matematici OpenType che, invece, non hanno questi difetti. Se si vuole la compatibilità con il livello 2, ripeto, questo requisito non è necessario.

10. Ci sono delle limitazioni sui collegamenti ipertestuali verso target esterni al file PDF; la cosa si giustifica con l'esperienza quotidiana di incontrare siti internet non più attivi; se alcuni siti non sono più attivi dopo pochi anni, figuriamoci dopo un lungo periodo di archiviazione. A cosa servirebbe inserire in un file archiviabile a lungo termine alcuni indirizzi internet, che non saranno più attivi in un futuro più o meno prossimo?
    Quindi non è prudente usare link a siti esterni. Invece è importantissimo che il file PDF sia autosufficiente e non dipenda da programmi esterni.

Il sistema TEX, con i suoi programmi di tipocomposizione pdflatex e lualatex provvede a quasi tutto il necessario, ma l'operatore deve comunque fare molta attenzione a che tutto ciò che quei programmi di composizione non possono fare da soli sia adeguatamente controllato affinché non violi le norme di archiviabilità.

Nel resto di questo capitolo si parlerà di come controllare e come eventualmente riparare alcuni dei difetti che possono presentarsi nella produzione di un documento PDF da archiviare.

1.1 I metadati

Le norme specificano che i metadati devono essere scritti usando solamente caratteri in codifica UNICODE o UTF-8; altre codifiche, come per esempio ISO 8859-1 (Latin 1) non sono accettabili. In base all'esperienza accumulata ritengo che sia preferibile usare solo caratteri ASCII; essi mancano di qualunque diacritico; se i metadati devono poter essere letti dai programmi di biblioteconomia è preferibile che essi non contengano diacritici, anche se questo rende certi nomi propri in modo incompleto. Per esempio il creatore del programma pdftex si chiama Hàn Thế Thành;

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

come si vede, contiene molti diacritici anche sovrapposti; togliendo gli accenti, che in vietnamita hanno un valore semantico importantissimo, le parole che formano quel nome potrebbero cambiare significato; usando solo font codificati UNICODE che contengano anche i caratteri vietnamiti si potrebbe rendere il nome correttamente, ma poi se il programma biblioteconomico non usa gli stessi font, chi legge le informazioni raccolte dai metadati trova al posto di quei segni dei rettangoli con le diagonali o dei rombi neri contenenti dei punti interrogativi bianchi, insomma trova dei segni privi di qualsiasi informazione utile, oltre a quella ovvia che informa che il glifo è mancante.

La maggior parte dei metadati riguardano certe norme e certi standard definiti a Dublino prima del 2005, oltre a specifiche in merito al programma col quale è stato composto il documento. Ma ci sono alcune specifiche che riguardano il particolare documento, come la lista degli autori, il titolo, eccetera. Questi metadati devono venire specificati dall'utente con modalità speciali che coinvolgono il linguaggio XML e usano speciali *tag* di quel linguaggio definiti in altri metadati.

Per comodità del compositore il pacchetto ***pdfx*** mette a disposizione dei comandi in linguaggio LaTeX che rendono molto più facile l'operazione; tuttavia questi comandi sono usabili quando si compone il documento con uno dei programmi di tipocomposizione che usano il mark-up LaTeX. Usando i word processor che non si appoggiano al sistema TeX, questa separazione fra il compositore e i metadati viene svolta da opportune interfacce grafiche mediante finestre di dialogo.

1.2 I COLORI

Come ho già detto i colori dovrebbero essere sempre specificati con un profilo di colore basato sul modello RGB, anche quando questi colori sono il bianco e il nero o le sfumature di grigio; ovviamente questo profilo è necessario quando sono presenti colori in senso stretto; tuttavia si raccomanda di non usare nessun profilo di colore diverso dal modello RGB.

Se le immagini a colori sono sviluppate con i programmi di disegno del sistema TeX, i colori definiti dal pacchetto ***xcolor*** vanno bene, ma bisogna stare attenti a non usare trasparenze esplicite o implicite se si vuole la conformità di livello 1. I disegni eseguiti con il pacchetto ***tikz*** possono contenere trasparenze, quindi anche se il disegno è sviluppato all'interno

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

delle funzionalità del sistema TEX, sfruttando quelle del linguaggio PDF, è possibile che, anche restando all'interno di quanto si può fare con questo sistema, si possano generare delle incompatibilità con la norme ISO del livello 1; non dovrebbe esserci nessun problema con il livello 2; tuttavia si veda quanto descritto nel capitolo 5.

Quando invece le immagini sono create con altri programmi o sono ricavate dalla rete (attenzione ai diritti di autore) è possibile che contengano trasparenze o si basino su modelli di colore diversi da RGB. Talvolta le trasparenze vengono inserite quando si usa il programma inkscape per rendere conformi certe immagini contenenti legende testuali: bisogna ricordare: livello 1, trasparenze vietate; livello 2, trasparenze consentite.

In ogni caso è opportuno verificare ogni immagine con programmi adatti; fra quelli gratuiti e disponibili per tutte e tre le principali piattaforme di elaborazione (Windows, Mac, Linux) c'è il programma freeware e opensource GIMP (GNU Image Manipulation Program). Ovviamente, se se ne dispone, si possono usare altri programmi di fotoritocco, anche commerciali.

Con GIMP è possibile cambiare modelli e profili di colore; è possibile "appiattire" (flatten) le immagini; togliere le trasparenze, cambiare formato alle figure; è possibile fare tante altre cose tranne una, che ritengo importante: riesce a leggere i file grafici in formato vettoriale (PDF ed EPS), ma non può salvare nessun file in questi formati; perciò di fronte a file vettoriali che hanno bisogno di modifiche o si usa software diverso, oppure si cambia formato scegliendo il formato PNG per disegni al tratto (ricordiamo che il formato PNG può contenere le trasparenze) oppure il formato JPG per le fotografie.

Il formato JPG non dà problemi per quel che riguarda le trasparenze, in quanto non le prevede e non le gestisce, ma visto che comprime le immagini in modo *lossy* perde un poco di informazione e, nel ricostruire le immagini per la visualizzazione o la stampa, vi si possono produrre degli effetti collaterali (immagini ombra); questi effetti sono trascurabili nelle fotografie, ma diventano importanti nei disegni al tratto. Possono apparire immagini ombra anche nelle fotografie se queste contengono elementi fortemente contrastati ripetuti periodicamente anche nell'originale. Per esempio la figura 1.1 contiene i tiranti del ponte ripetuti prospetticamente in modo quasi periodico. Non si vedono immagini ombra perché l'immagine ha una densità di 180 per 180 (pap: pixel al pollice; in inglese, ppi: *pixel per inch*), una densità moderatamente elevata; il file JPG che contiene

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

FIGURA 1.1 Il Bay Bridge di Sidney

questa immagine occupa solo 579 sul disco, ma decompressa in memoria occupa 18,6. Con tale densità di pixel, si ha un ingombro in memoria piuttosto grande; se lo si vuole ridurre, GIMP lo può fare, ma se lo si facesse, potrebbero apparire dei tiranti-ombra che, per quanto evanescenti, disturbano la fotografia. D'altra parte se si ingrandisce la schermata dove compare la figura 1.1 per un fattore di almeno 5, tra un tirante e l'altro si intravedono i tiranti ombra.

Invece il formato PNG comprime le immagini in modo *lossless*, senza perdita di informazione; è adattissimo ai disegni al tratto con pixel adiacenti che possono avere colori fortemente contrastanti. Il metodo di compressione è meno efficace[2] di quello JPG, ma, proprio perché è *lossless*, conserva meglio l'informazione dell'immagine.

Se ne conclude che il formato PNG è preferibile rispetto al formato JPG per i disegni al tratto (forte contrasto) e per pochi tipi di fotografie contenenti dettagli particolari, mentre il formato JPG è più adatto per fotografie che abbiano variazioni di colore con meno contrasto. Con il

[2]Attenzione: una fotografia in formato JPG può essere di mezzo mibibyte, ma trasformato in formato PNG può aumentare anche di venti volte.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

formato PNG bisogna fare attenzione alle trasparenze se si vuole una compatibilità di livello 1.

Talvolta è necessario trasformare anche le immagini vettoriali in formato JPG o PNG a seconda del loro contenuto, cioè se contengono trasparenze per il livello 1 o se contengono la proprietà di interpolazione per i livello 2. Inoltre qualche volta risulta necessario trasformarle in questo modo se contengono scritte i cui font non sono stati incorporati nel file vettoriale. Queste cose si possono fare con il programma inkscape, ma talvolta il programma stesso introduce delle trasparenze a cui bisogna fare attenzione. Si vedrà più avanti come conservare la vettorialità senza introdurre trasparenze.

1.3 I file immessi

I file immessi nei documenti da archiviare non sono solo imagini; potrebbero essere brani di testo ripresi da Internet (attenzione ai diritti di copyright) o ripresi da altri documenti PDF ricavati da Internet (di nuovo: attenzione ai diritti di copyright[3]) o di cui si è autori.

Per le immagini si è già detto come sia eventualmente necessario elaborarle per renderle compatibili con un documento da archiviare.

Per il testo è assolutamente necessario che i font originali siano contenuti nel file che contiene la parte di testo che si intende usare. Se non lo contiene, si può rimediare convertendo il file PDF che contiene il brano di testo in una immagine PNG con definizione sufficientemente alta. Altrimenti, e specialmente se è importante conservare la vettorialità, si può aprire il file suddetto con il programma gratuito inkscape che è in grado di salvare il file in formato PDF trasformando, però, i font che contiene nei loro stessi disegni; il file che si ottiene non contiene più font di nessun genere, ma contiene il testo così come appariva nel file originale. Spesso questa procedura (ottima per recuperare parti di testo che contengono anche

[3]Internet sembra essere considerato un deposito di informazioni di libero uso per chiunque, ma non è così: copiare senza citare l'autore e la fonte significa compiere il reato di plagio, con conseguenze penali e civili che possono essere molto pesanti. Quindi prima di prelevare qualcosa dalla rete, fermo restando l'obbligo di citazione, bisogna sempre esaminare con cura la licenza a cui quel materiale è sottoposto; spesso ha una licenza abbastanza libera che impone a chi copia dei brani solo l'obbligo di citare la fonte; talvolta la licenza è più restrittiva e potrebbe essere necessario il permesso scritto dell'autore.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

immagini) produce un ottimo risultato, purché sia chiaro che i font così disegnati non devono apparire come strati sovrapposti allo sfondo.

1.4 I FONT DIFETTOSI

Talvolta i font usati o contenuti in documenti da importare sono difettosi, nel senso che le loro informazioni metriche sono anomale o comunque non coerenti; un caso che si presenta in modo relativamente frequente è quello in cui certi glifi hanno una dimensione dichiarata nulla. Questo è assolutamente vietato per la compatibilità di livello 1. Ci sono diverse vie per ovviare, una delle quali consiste nell'usare inkscape per eliminare ogni riferimento ai font perché inserisce nel file PDF il disegno di ogni glifo. Però coì facendo il file PDF non consente più di cercarvi dentro determinate stringhe di testo, né di copiare parti di testo da incollare in altri documenti. Può essere utile quando si importano file grafici, ma per brani di testo questa via è fortemente sconsigliabile.

Un altro metodo consiste nel modificare il file metrico che accompagna ogni font, oppure le informazioni metriche contenute all'interno dello stesso file del font. Questo succede con i font TrueType e con quelli OpenType. In questi casi bisogna editare i file dei font con programmi adeguati per correggere le informazioni metriche in essi contenuti. Un programma freeware e opensource che può fare queste cose è il programma FontForge disponibile per tutte e tre le piattaforme principali[4]. Però il difetto di questo metodo è che si possono correggere i dati metrici di interi font e/o di singoli glifi, sempre che la licenza dei font lo consenta, ma si creano degli esemplari di font differenti da quelli che il loro autore distribuisce, e quindi non più usabili da terze parti a cui siano eventualmente stati forniti. In sostanza si violano le norme per l'archiviabilità. La soluzione ideale sarebbe che i loro creatori provvedessero autonomamente alla correzione di questi difetti. Tuttavia ricordo che la conformità di livello 2 non fa caso ai caratteri la cui larghezza sia nulla. Si ripeterà questo concetto diverse volte nel seguito,

[4]Per Windows probabilmente è necessario operare all'interno di un ambiente CygWin, che è semplicemente un emulatore di un sistema operativo di tipo UNIX. Ho usato FontForge moltissimo, e ho anche collaborato a scriverne delle parti molto marginali; ma la potenza del programma sta tutta nella manipolazione interattiva grafica dei singoli glifi vettoriali e delle loro proprietà, a cui io non ho dato nessun contributo.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

ma merita segnalare subito che è meglio scegliere sempre la conformità di livello 2 in ogni caso.

Un terzo metodo usabile solo con i font che accompagnano il sistema TEX, consiste in una sequenza di trucchi per sostituire i font bitmapped con font vettoriali che quelli vettoriali con la codifica OMS, in font virtuali con le dimensioni metriche adeguatamente corrette. Per non confondere i programmi del sistema TEX e i suoi pacchetti di estensione destinati alla scelta dei font, bisogna scambiare i nomi dei font reali e di quelli virtuali in una specie di meccanismo che sembra mordersi la coda. Inoltre quando si aggiorna l'installazione del sistema TEX sulla propria macchina si rischia di perdere tutte le modifiche e di dover ricominciare da capo. In sostanza si cura localmente il problema, ma se ne creano degli altri. Siccome questa soluzione era stata prospettata diversi anni fa dagli stessi autori del programma pdftex e del pacchetto originale *pdfx*, Hàn Thê Thành e C.V. Radhakrishnan, e questa procedura si può ancora trovare descritta in Internet, non solo non ne fornisco l'indirizzo di rete, ma ne scoraggio l'uso per i motivi che ho descritto sopra.

Purtroppo fra i font del sistema TEX alcuni sono difettosi per la compatibilità di livello 1, perché alcuni simboli matematici contenuti nei file con la codifica OMS hanno dimensioni orizzontali nulle. Questi file sono usati solo da pdflatex per comporre la matematica; potrebbero essere usati anche da lualatex quando si omette di specificare nel preambolo del documento l'uso di font matematici OpenType; ovviamente bisogna evitare questa omissione. Con la compatibilità di livello 2 il problema non si pone.

Questo dice subito che cosa bisogna fare nel caso si debba scrivere matematica che contiene quei simboli (precisamente il comando `\not` per negare un operatore di relazione, e il comando `\mapstochar` per costruire il simbolo dell'operazione di iniezione). In sostanza basta usare solo lualatex specificandogli adeguati font matematici OpenType; non tutti i font OpenType testuali sono accompagnati dai corrispondenti font OpenType matematici, ma quelli esistenti si accoppiano molto bene con quasi tutti i font testuali.

Vorrei sottolineare che i font predefiniti per un certo numero di simboli matematici sono i cmsym disegnati per vari corpi: cmsym5, cmsym7, cmsym10, eccetera, tutti con la codifica OMS; questi sono difettosi come detto sopra; ma lo sono anche tutti gli altri font con la stessa codifica, per esempio quelli invocati dal pacchetto *mathdesign* per accompagnare font

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

vettoriali come i Garamond, gli Utopia, i Bookman, ed altri con collezioni di glifi matematici stilisticamente adeguate agli stili dei font testuali. Altri pacchetti che forniscono alternative ai font predefiniti possono avere lo stesso difetto. Invece i recenti font Libertinus-Math, distribuiti con il sistema TEX sia in forma Type 1, sia in forma OpenType, non hanno questo difetto perché usano una codifica diversa e sono stati disegnati in modo da evitare questi problemi. Per quelli OpenType è normale, ma per quelli Type 1, credo che Libertinus sia, se non l'unica, una delle pochissime collezioni di font che usi polizze di 256 caratteri per i caratteri matematici, e quindi sia in grado di ridurre il numero di famiglie matematiche (o gruppi matematici) usate per comporre un dato documento.

Se per qualche motivo non fosse disponibile il file sorgente dei documenti che contengono font difettosi bisogna arrangiarsi con trucchi di vario genere. Nel pacchetto *TOPtesi*, inizialmente creato per lavorare solo con pdflatex, ho condizionatamente inserito dei *patch* in modo da sostituire le definizioni dei comandi `\not` e `\mapstochar` con altre definizioni che sostituiscono i glifi difettosi con altri glifi oppure con disegni costruiti in caso di bisogno con le funzionalità dell'ambiente *picture* di LATEX. Questi *patch* ora sono conservati per retrocompatibilità, ma sono diventati superflui perché basta usare lualatex e i font matematici OpenType; oppure se proprio si vuole usare pdflatex, basta specificare la conformità di livello 2.

Ma quei trucchi funzionano solo se si dispone dei file sorgente del documento da comporre; se non se ne dispone, o si trasforma ogni pagina del documento in immagine PNG o JPG, o si trasforma l'intero documento con il programma inkscape.

1.5 I collegamenti ipertestuali

I collegamenti ipertestuali possono dare dei problemi sostanzialmente di carattere semantico; infatti non ha nessun senso usare collegamenti ipertestuali per riferirsi ad oggetti esterni al documento stesso. Le norme PDF/A sono rigorose per quel che riguarda il fatto che ogni documento archiviato sia autosufficiente. Le citazioni di riferimenti non sono propriamente degli oggetti esterni, ma delle informazioni; ma le informazioni sono utili solo se possono essere usate. Ma l'informazione che un dato riferimento si trova ad un indirizzo di rete non più esistente è una informazione che non serve a niente.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

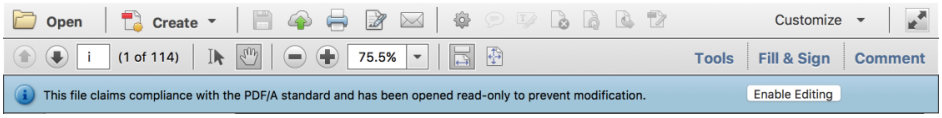

Figura 1.2 Messaggio emesso dai programmi Adobe Acrobat e Adobe Reader quando aprono un file che potrebbe essere PDF/A compatibile

Questo implica che i collegamenti ipertestuali interni al documento sono perfettamente leciti, mentre quelli che fanno riferimento ad oggetti esterni sono da evitare, se non altro perché fra alcuni decenni verosimilmente essi non sarebbero più attivi.

Invece sono assolutamente vietate le dipendenze da *programmi esterni* perché il file conforme alle norme PDF/A dipende solo dall'esistenza di programmi in grado di presentare a schermo e di stampare quel file anche in un futuro molto lontano.

Per quel che riguarda il sistema TEX, il pacchetto ***pdfx*** provvede da solo a caricare il pacchetto ***hyperref*** con l'opzione *pdfa*, che adatta i collegamenti ipertestuali in modo che siano conformi alle specifiche di archiviabilità. Anche questo stesso documento nel retrofrontespizio contiene un link ad un sito esterno, il sito del GUIT, e questo file, verificato come descritto più avanti, risulta conforme alla norme PDF/A-1b. Sarebbe comunque meglio evitare questo genere di collegamenti ipertestuali; tra 50 anni saranno ancora validi?

Va notato anche che il pacchetto ***imakeidx*** (che permette di comporre uno o più indici analitici in modo sincrono) deve essere caricato prima di ***hyperref***; quindi, visto che ***pdfx*** carica ***hyperref***, ne segue che se si vuole caricare ***imakeidx***, lo si deve fare prima di caricare ***pdfx***. Se non si ha questa accortezza, l'indice analitico viene composto lo stesso in modo sincrono, ma i numeri di pagina dei vari lemmi non sono collegati in modo ipertestuale alle rispettive pagine.

Dall'altro lato ci sono programmi di lettura e visualizzazione dei file PDF che, se il file PDF è conforme allo standard, aprono il file in sola lettura; fra questi ci sono il programma gratuito Adobe Reader, e il programma commerciale Adobe Acrobat Pro. Non escludo l'esistenza di altri visualizzatori dal comportamento simile a quello dei due citati.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

I due programmi citati, infatti, quando aprono un documento che contiene i metadati caratteristici dei file PDF/A, prudentemente avvisano che il file viene aperto in sola lettura perché quel documento afferma di essere conforme alla norma per l'archiviabilità (*the document* [...] *claims to be* [...], vedi la figura 1.2). Non bisogna prendere troppo sul serio quella affermazione che lascerebbe intendere che il documento sia davvero PDF/A conforme; in realtà *pretende* solo di esserlo, ma chi l'ha verificato? O meglio, quel file è stato mai verificato per la conformità? Purtroppo esistono moltissimi file in giro che contengono i metadati necessari per l'archiviabilità, ma, come dovrebbe essere chiaro dalla lettura di questo capitolo, i metadati da soli non sono sufficienti.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

Trasformazione di file PDF in file archiviabili 2

Ora parleremo di come si possa trasformare un file PDF *comunque ottenuto* in un file archiviabile secondo le norme PDF/A. Attenzione: è opportuno ripetere che "comunque ottenuto" significa che tale file può essere stato ottenuto con programmi vari ed eterogenei, ma del quale ci interessa solo che si tratta di un file PDF e non con quali mezzi è stato creato.

In primo luogo ci occupiamo di file "comunque ottenuti", poi nel capitolo successivo ci occuperemo di come comporre un documento con i programmi di tipocomposizione del sistema TEX.

Ci sono diversi modi di archiviare file PDF già pronti. In realtà esistono solo pochi casi.

1. Il documento viene sottoposto alla verifica di compatibilità e risulta che tutto è a posto. Evidentemente in questo caso non c'è da fare nulla, ma anche questo caso deve venire esposto. Se infatti manca un certificato di conformità, come si fa a sapere che il file è archiviabile?
2. Il documento viene sottoposto alla verifica di compatibilità e la diagnosi mostra che manca solo dei metadati.
3. Il documento viene sottoposto alla verifica di compatibilità e la diagnosi mostra che manca dei necessari metadati, oltre a presentare altre incompatibilità.

2.1 Trasformazione delle immagini

Quanto esposto in questo capitolo è parzialmente applicabile anche alla trasformazione dei file di testo, che però sono più facili da gestire come

testo; se invece si vuole replicare un testo scandito o ricavato dalla rete come immagine, lo si tratta esattamente come qualunque altra immagine.

2.1.1 Immagini a matrici di punti

Come già detto, le immagini a matrici di punti devono essere in formato JPG o PNG senza trasparenze e devono avere un profilo di colore di tipo RGB; si consideri assolutamente vietato usare un profilo di colore CMYK.

Nel caso che si disponga di un file a matrice di punti con profilo di colore di tipo CMYK lo si apra con un programma di fotoritocco, per esempio GIMP, e lo si trasformi cambiandogli il profilo di colore.

Se si tratta di un file PNG, che è in grado di gestire le trasparenze, gli si cambi anche il formato in JPG, Meglio ancora, si compili con pdflatex un piccolo file con la classe *standlone* che incorpori solo l'immagine; questo modo di procedere in pratica crea un file PDF contenente solo l'immagine; si verifichi questo file con i programmi di verifica PDF/A; esso non sarà conforme, ma la diagnosi dirà se oltre i metadati mancanti, ci sono delle altre carenze da correggere; poi si importi questa immagine PDF, nel documento complessivo da verificare per la compatibilità di livello 2;.

2.1.2 Immagini vettoriali

Se si dispone di una immagine vettoriale con legende scritte con font vettoriali, si analizzi la figura con il programma di verifica e si esaminino i messaggi relativi ai font; se questi non sono inclusi o se corrispondono a font le cui caratteristiche metriche non sono compatibili con la compatibilità di livello 1, si apra il file con il programma inkscape e lo si salvi in formato PDF facendo attenzione che sia marcata l'opzione di sostituire i font con i loro disegni.

A questo punto si dispone di un file PDF senza font ma con possibili trasparenze; se si vuole la compatibilità di livello 1, lo si analizzi con il programma di verifica; se viene segnalata la presenza di trasparenze, si apra nuovamente il file appena salvato in formato PDF, e lo si salvi in formato EPS specificando un profilo di colore di tipo RGB fra quelli disponibili con inkscape; io ho sempre scelto il profilo Adobe RGB e ho sempre ottenuto file conformi.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

Ora si dispone di un file EPS con un profilo di colore giusto, senza trasparenze (che il formato EPS non gestisce), e senza font inclusi, quindi senza incompatibilità con le norme per l'archiviabilità.

Il formato EPS è usabile da tutti e tre i programmi di tipocomposizione del sistema TEX, quindi, a rigore, non ci sarebbe da fare altro. Siccome il linguaggio PostScript con cui è codificato il file EPS è piuttosto prolisso, il file EPS è di grandi dimensioni; si può allora procedere ad una ulteriore trasformazione, come descritto qui di seguito.

1. Su una macchina Mac si può aprire il file con il programma Anteprima o Preview[1] che lo distilla direttamente in formato PDF ai fini della visualizzazione ma non lo salva su disco; lo si potrebbe salvare in questo formato; tuttavia potrebbe aggiungergli dei bordi bianchi; personalmente preferisco usare la cassetta degli attrezzi di Preview, selezionare solo la parte di figura che mi interessa, copiarla nel clipboard con control + C, agire sul menu "File" per selezionare di salvare il contenuto del clipboard in un nuovo file, per poi salvarlo in formato PDF, stando bene attento che non ruoti l'immagine; procedendo come descritto non dovrebbe avere luogo nessuna rotazione; ma se dovesse succedere, lo si riapre con Preview, si clicca command + L o command + R per ruotarlo di 90° rispettivamente a sinistra o a destra, e poi lo si risalva con lo stesso nome.
2. Esistono diversi programmi distiller, come il prototipo Acrobat Distiller, ma ce ne sono altri che hanno finito per chiamarsi nello stesso modo, perché svolgono più o meno la stessa funzione, magari integrati in altri visualizzatori di file PDF. Il prototipo di Distiller sostanzialmente è quasi privo di interfaccia grafica; infatti trovo che sia più facile da usare da linea di comando in una finestra terminale o prompt dei comandi. Prende in entrata un file PS e lo esporta in formato PDF; accetta in entrata anche file in formato EPS. Pur disponendone, sono anni che non lo uso più, perché sul Mac, su cui lavoro, mi è molto più comodo servirmi di Preview. Tuttavia riconosco che Distiller, nella sua funzione primaria è molto professionale.
   Dandogli dunque in pasto il file grafico EPS che non contiene trasparenze, perché questo formato non le consente, si ottiene un file

[1] "Anteprima" è un alias di "Preview"; si tratta dello stesso programma localizzato in italiano invece che in inglese.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

PDF senza trasparenze ma sempre vettoriale se l'input era veramente vettoriale.

3. Un altro metodo che ricorre solo alle funzionalità dei programmi del sistema TEX è il seguente; lo descrivo mediante un esempio. Supponiamo di disporre di una figura in formato EPS ottenuta come descritto in precedenza, senza trasparenze e senza font di nessun genere, e con un profilo di colore RGB; sia `figura.eps` il file in questione.
   Si costruisca allora un semplice file da compilare con pdflatex, salvandolo col nome `figura.tex`:

```
\documentclass{standalone}
\usepackage{graphicx}
\nofiles
\begin{document}
\includegrapics{figura.eps}
\end{document}
```

   Compilando questo semplice documento si ottiene il file `figura.pdf` che, sostanzialmente contiene la figura iniziale descritta con il linguaggio PDF, in formato PDF compresso, in modo da ridurre lo spazio in memoria, e perfettamente scontornato, grazie all'uso della classe *standalone*; questa classe non è molto conosciuta ma è utilissima. Il file `figura.pdf` è esattamente il file della figura che si voleva ottenere. L'occupazione di memoria del file PDF può risultare ridotta anche del 50%, talvolta di più.
   Per non dover riscrivere un file diverso per diverse figure, basta trasformarne il formato una alla volta, cambiando il nome del file `.eps` da convertire e poi cambiando nome al file PDF ottenuto.
   Sembra laborioso, ma semplifica di molto le procedure che si possono fare con un programma come Preview o quelle un po' più manuali da fare con il distiller.
4. Avendo già usato il programma inkscape per creare il file EPS, si può importare in questo programma il file appena esportato per poi salvarlo in formato PDF. Non è la stessa cosa di usare il primo file PDF ottenuto con inkscape, proprio perché si è proceduto con l'intermediario del formato EPS dal quale sono state eliminate le trasparenze che questo formato non gestisce; riprendendolo con inkscape, questo

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

programma non può inserire trasparenze partendo da un file che non ne contiene. Sembra l'uovo di colombo, e in effetti lo è.

Come si vede, dunque, questi procedimenti per rendere compatibili i file grafici possono essere risolutivi oltre che decisamente efficaci.

2.2 Il documento non è archiviabile

Si dispone di un documento non archiviabile: manca dei metadati e/o ha altre incompatibilità. Se proprio deve essere archiviato, non è il caso di "disperarsi". Bisogna leggere le informazioni diagnostiche emesse dal programma di verifica; bisogna cercare di capirle; bisogna poi porre rimedio in qualche modo alle deficienze del file di cui si dispone.

Di solito per porre rimedio a certi semplici errori si può editare il file PDF (supponiamo di non disporre dei file sorgente) con programmi che sono in grado di farlo; so che ne esistono; uno è sicuramente il programma Adobe Acrobat Pro, versione XI; si tratta di un programma commerciale dal costo non indifferente, ma a seconda dell'uso che se ne fa potrebbe essere utile acquistarlo; inoltre gli studenti e il personale delle scuole secondarie superiori e delle università possono fruire dei corposi sconti loro offerti con il programma Education. Non ne parlo nel seguito, perché preferisco riferirmi ai programmi freeware e opensource.

Per errori più consistenti nemmeno Adobe Acrobat Pro, però, riesce a porre rimedio.

Allora in questi casi gravi si deve provvedere a trasformare il file PDF in un file di pagine grafiche; ogni pagina viene "congelata" in una immagine; non contiene più font né collegamenti ipertestuali; non se ne può copiare il testo con l'apposito strumento di selezione; insomma diventa una specie di album fotografico, non dissimile dalle fotocopie fascicolate che venivano fatte anni fa[2]. L'informazione è conservata, magari si riesce ancora a disporre di un file PDF, ma le sue pagine sono immagini. Basta allora aggiungere soltanto i metadati.

[2] Ci sono in rete degli archivi da dove si possono scaricare interi documenti PDF, contenenti testi più o meno antichi, di cui sopravvivono solo pochissimi esemplari originali custoditi gelosamente in alcune biblioteche. Tali testi sono stati fotografati pagina per pagina e con queste foto in formato adeguato è stato assemblato in un file PDF. Non c'è praticamente altro modo per poter studiare questi testi antichi.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

2.3 Il documento manca solo dei metadati

Questo caso è più generale di quanto si pensi. La sua soluzione è semplicissima se si usano i programmi del sistema TEX. Si potrebbe usare anche il programma ghostscript ma la procedura è decisamente più complessa e non assicura affatto risultati migliori.

Supponiamo di disporre di un file che contiene l'intero documento e, a titolo d'esempio, supponiamo che questo file si chiami `file.pdf` e sia composto su pagine virtuali di formato A4[3]. Si crei un file `file-a.tex` in uno dei modi seguenti.

1. Il primo modo, adatto anche per usare una "vecchia" installazione del sistema TEX, cioè anteriore al 2019, si affida al tradizionale ambiente asteriscato *filecontents** che contiene solo i metadati da inserire *prima* di `\documentclass` in questo modo:

```
% !TEX encoding = UTF-8 Unicode
% !TEX TS-program = pdfLaTeX
%
%%%%%% File "file-a.tex"
%
\begin{filecontents*}{\jobname.xmpdata}
\Autor{Mario Rossi\sep
       Carlo Bianchi\sep
       Bruno Neri}
\Title{Il formato archiviabile}
\Subject{Procedure to produce archivable documents}
\Keywords{PDF\sep
          PDF/A\sep
          ISO 19005\sep
          metadata embedding}
\Organisation{Archivi Riuniti}
\end{filecontents*}
%
```

[3] Se fosse composto su pagine di altri formati, nel codice che segue basta sostituire l'opzione *a4paper* della dichiarazione di classe con un'altra opzione di formato della carta; in altre situazioni sarebbe necessario espandere un poco il preambolo per usare il pacchetto *geometry* in modo da impostare le dimensioni giuste della carta virtuale.

```
\documentclass[a4paper]{book}
\usepackage[a-1b]{pdfx}% oppure [a-2b]
\usepackage{pdfpages}
\nofiles
\begin{document}
\pagestyle{empty}
\includepdf[pages=-]{file.pdf}
\end{document}
```

Se si modificano i metadati iscritti nel file sorgente, questi non vengono sostituiti nel file di metadati `\jobname.xmpdata` se preventivamente non si cancella il precedente file con lo sesso nome.

2. Il secondo usa ancora l'ambiente *filecontents* non asteriscato, ma con la sua ridefinizione del 2019 dotato di opzioni, può essere collocato sia prima sia dopo `\documentclass`, ma bisogna esplicitare le due opzioni *overwrite* e *noheader*; tuttavia va sempre inserito prima di invocare il pacchetto ***pdfx*** perché è questo pacchetto che ha bisogno dei metadati. Lo schema diventa il seguente:

```
% !TEX encoding = UTF-8 Unicode
% !TEX TS-program = pdfLaTeX
%
%%%%%% File "file-a.tex"
%
\documentclass[a4paper]{book}
\begin{filecontents}[overwrite,noheader]{\jobname.xmpdata}
\Autor{Mario Rossi\sep
       Carlo Bianchi\sep
       Bruno Neri}
\Title{Il formato archiviabile}
\Subject{Procedure to produce archivable documents}
\Keywords{PDF\sep
          PDF/A\sep
          ISO 19005\sep
          metadata embedding}
\Organisation{Archivi Riuniti}
\end{filecontents}
%
\usepackage[a-1b]{pdfx}% oppure [a-2b]
\usepackage{pdfpages}
```


File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

```
\nofiles
\begin{document}
\pagestyle{empty}
\includepdf[pages=-]{file.pdf}
\end{document}
```

In questo modo non c'è bisogno di cancellare il precedente file di metadati se il contenuto dell'ambiente *filecontents* viene modificato.

3. Il terzo metodo Non dipende dall'eta della versione del sistema TEX di cui si dispone;a parte questa importante piccola differenza, non cambia molto rispetto al secondo metodo se si usa il pacchetto *fancyvrb* che definisce l'ambiente *VerbatimOut*; la classe *guidatematica* con cui è stata creata questa guida tematica, infatti, usa questo metodo; con la classe *book* degli esempi precedenti, basta caricare il pacchetto *fancyvrb* nel preambolo prima di poter usare questo suo ambiente.

```
\documentclass[a4paper, ...]{book}
\usepackage{fancyvrb}
\begin{VerbatimOut}{\jobname.xmpdata}
\Autor{Mario Rossi\sep
       Carlo Bianchi\sep
       Bruno Neri}
\Title{Il formato archiviabile}
\Subject{Procedure to produce archivable documents}
\Keywords{PDF\sep
          PDF/A\sep
          ISO 19005\sep
          metadata embedding}
\Organisation{Archivi Riuniti}
\end{VerbatimOut}
%
\usepackage[a-1b]{pdfx}% oppure[a-2b]
\usepackage{pdfpages}
\nofiles
\begin{document}
\pagestyle{empty}
\includepdf[pages=-]{file.pdf}
\end{document}
```

Nei listati riportati sopra si riconoscono tre parti: la prima contiene due

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

righe di commento del tutto ignorate dal programma di compilazione, ma non ignorate dal programma di editing del file, quando si tratta di un editor moderno capace di autoconfigurarsi in base al contenuto di quelle righe; se l'editor non ne fosse capace esse semplicemente ricordano all'utente che il file va registrato su disco con la codifica UTF-8 e che il file va compilato con pdflatex.

La seconda parte contiene i metadati relativi al documento specifico; vi si notano autori italiani, la casa editrice italiana, ma gli argomenti e le parole chiave sono in inglese. Questo rende i metadati fruibili anche all'estero ed evita almeno in queste stringhe l'uso di segni diacritici poiché usa solamente caratteri ASCII. Si noti che, quando ci sono liste di nomi di autori o liste di parole chiave, esse sono separate dal separatore `\sep` che provvede a generare tutti i separatori in linguaggio XML necessari internamente per la registrazione dei metadati su disco. Questi metadati vengono registrati nella cartella di lavoro mediante un file che si chiama `file-a.xmpdata`. Infatti, quando si compila questo file la variabile `\jobname` viene ad assumere un valore corrispondente al nome del file (principale – in questo caso l'unico file) che viene composto. Si possono gestire i metadati anche in un modo diverso che verrà spiegato più dettagliatamente più avanti

La terza parte è il documento vero e proprio da compilare: il file `file-a.tex` carica pochissimi pacchetti, segnatamente il pacchetto ***pdfx*** usato per inserire i metadati nel file PDF finale `file-a.pdf`, e ***pdfpages*** che serve per immettere nel documento tutte o alcune delle pagine del documento PDF che viene incluso.

L'esecuzione del programma implica di costruire un file PDF che contiene i metadati letti dal file `file-a.xmpdata` registrato nella cartella di lavoro da pdflatex stesso. Successivamente vi immettono una ad una tutte le pagine del file `file.pdf`.

Il risultato è un file PDF archiviabile se e soltanto se, come si è supposto, il file di partenza non aveva nessun difetto a parte la mancanza dei metadati.

Vale la pena di sottolineare che, se il documento iniziale aveva dei collegamenti ipertestuali interni o esterni, tutti questi collegamenti risultano disattivati.

Si noti che ogni file PDF privo di difetti strutturali, ma mancante solo dei metadati, ottenuto con qualsiasi programma capace di produrre file PDF, siano essi gli eseguibili di tipocomposizione del sistema TEX, i word processor, o altro, può essere trattato in questo modo per renderlo

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

archiviabile; il prezzo da pagare è solo l'assenza di collegamenti ipertestuali attivi.

Quindi questo semplicissimo metodo per creare file archiviabili secondo le norme vale sempre. Bene inteso, la parte difficile, non è aggiungere i metadati, ma assicurarsi che non ci siano i difetti strutturali descritti nel capitolo precedente.

2.4 Verifica della conformità

Sembra superfluo dirlo, ma sebbene il semplice metodo descritto produca file archiviabili, è sempre meglio verificarlo ancora, se non altro perché il programma di verifica è in grado di produrre la certificazione che spesso è necessario allegare nel momento in cui si consegna il file archiviabile all'ente che deve custodirlo. Per eseguire la verifica si veda l'apposito capitolo 5.

Creazione diretta di file archiviabili con pdflatex 3

Prima di cominciare, deve essere chiaro che per evitare di dover ricorrere a trucchi bisogna esaminare la matematica che si deve scrivere. Se questa deve fare uso di comandi che invocano glifi difettosi con i font codificati OMS, è meglio comporre il file con lualatex che usa preferibilmente font OpenType; però bisogna specificargli espressamente di usare font matematici OpenType. Altrimenti bisogna ricorrere alla compatibilità di livello 2.

3.1 Figure e file da includere

In ogni caso bisogna verificare che le immagini (con i loro colori, anche in tonalità di grigio) e gli altri file da includere siano a posto; è meglio verificarli per accertare che siano privi di quei difetti di cui si è parlato nel primo capitolo. È vero che queste verifiche si possono fare anche durante la lavorazione del documento, ma bisogna ricordarsi di farle sempre *prima* di usare quei file immagine e quei file testuali.

Il controllo viene eseguito sempre con il programma di controllo che verrà descritto più avanti nel capitolo 5; questo programma però verifica solo le figure e i file in formato PDF. Per file immagine di tipo diverso ci si serva di un un qualunque programma di fotoritocco, per esempio GIMP.

3.2 Struttura del preambolo con pdflatex

Il preambolo del file principale di un documento da archiviare deve caricare certi pacchetti in un ordine preciso; il file principale e gli eventuali file secondari devono essere tutti salvati sul disco con codifica UTF-8, ma i pacchetti devono essere caricati secondo lo schema seguente. Qui si mostra

la struttura del file sorgente per un documento da comporre con pdflatex. Più avanti si indicheranno le piccole modifiche che bisogna apportare per comporre il documento con lualatex. Come già descritto nel paragrafo 2.3, invece di usare *filecontents** prima di `\documentclass` sarebbe meglio servirsi di una versione aggiornata e completa del sistema TEX e usare l'ambiente *filecontents* con le debite opzioni, oppure, dopo aver caricato il pacchetto *fancyvrb* nel preambolo, usare l'ambiente *VerbatimOut* dopo `\documentclass` e prima di invocare il pacchetto *pdfx*. Ecco due esempi abbastanza più dettagliati.

```
% !TEX encoding = UTF-8 Unicode
% !TEX engine = pdfLaTeX
\begin{filecontents*}{\jobname.xmpdata}
⟨metadati⟩
\end{filecontents*}

\documentclass[⟨opzioni⟩]{⟨classe⟩}
\usepackage[⟨opzioni⟩]{imakeidx} % Facoltativo
\usepackage[a-2b]{pdfx}% oppure [a-1b]
...
\usepackage[utf8]{inputenc}
\usepackage[⟨codifica⟩]{fontenc}
\usepackage[⟨opzioni⟩]{⟨pacchetto per la scelta dei font⟩}
...
\usepackage[⟨lingue⟩]{babel}
⟨altri pacchetti⟩
\usepackage{hyperref}% Attenzione: senza opzioni!
\hypersetup[⟨opzioni per {hyperref}⟩]

⟨macro specifiche per questo documento⟩

⟨impostazioni specifiche per la lingua principale⟩
\begin{document}
```

Come di consueto le parti scritte in corsivo fra parentesi acute sono da specificare da parte dell'utente.

Ma si osservi bene l'ordine in cui si susseguono le varie parti.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

1. Le prime due righe sono le solite righe di commenti speciali interpretabili dagli editor intelligenti che sanno autoconfigurarsi interpretando quelle righe; se l'utente non dispone di un editor intelligente, restano come suo promemoria per ricordare la codifica del file e il programma da usare per la compilazione.
2. L'ambiente *filecontents** oltre al nome del file che contiene i metadati, contiene anche i comandi per inserire i metadati stessi. Il nome di questo file deve essere identico a quello del file principale del documento da compilare; il nome del file scritto usando il comando `\jobname` assicura questa identità. L'estensione di questo file deve essere `.xmpdata`.
   Vale la pena di specificare di nuovo il punto seguente. Se si modificano i metadati, questi non vengono modificati nel file `.xmpdata` perché l'ambiente *filecontets** non sovrascrive file già esistenti. Quindi prima di registrare una modifica o si edita il file `.xmpdata` direttamente nella cartella di lavoro, oppure si cancella detto file e nella successiva compilazione del documento l'ambiente provvede a registrare i metadati in una nuova versione rispetto a quella del vecchio file appena cancellato; oppure si usa una versione almeno del 2019 del sistema TEX e si usa l'ambiente *filecontents* nel preambolo prima di caricare il pacchetto ***pdfx*** specificandogli le opzioni descritte nel paragrafo 2.3; oppure, infine, si carica il pacchetto ***fancyvrb*** (che può servire anche per la composizione di altre parti del documento) e si usa direttamente il suo ambiente *VerbatimOut* con i metadati in questo modo:

   ```
   \begin{VerbatimOut}{\jobname.xmpdata}
   ⟨metadati⟩
   \end{VerbatimOut}
   ```

   L'ambiente *VerbatimOut* non ha problemi a riscrivere file preesistenti, anzi potrebbe essere una buona idea quella di usare sempre questo ambiente. In questo modo la parte iniziale del preambolo diventa:

   ```
   % !TEX encoding = UTF-8 Unicode
   % !TEX TS-program = pdfLaTeX
   %
   \documentclass[...]{book}
   ```


File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

```
%
\usepackage{fancyvrb}
\begin{VerbatimOut}{\jobname.xmpdata}
    \Autor{Mario Rossi\sep
           Carlo Bianchi\sep
           Bruno Neri}
    \Title{Il formato archiviabile}
    \Subject{Procedure to produce archivable documents}
    \Keywords{PDF\sep
              PDF/A\sep
              ISO 19005\sep
              metadata embedding}
    \Organisation{Archivi Riuniti}
\end{VerbatimOut}
%
\usepackage[a-2b]{pdfx}% oppure [a-1b]
% Codifiche, lingue e font
...
% Altri pacchetti, impostazioni, definizioni
...
\begin{document}
```

3. Per quel che riguarda i ⟨*metadati*⟩ valgono le stesse osservazioni fatte nel capitolo precedente.
4. L'istruzione `\documentclass` con i suoi argomenti facoltativo e obbligatorio non ha niente di diverso dal solito.
5. Se si vuole caricare il pacchetto ***imakeidx*** lo si deve fare prima di caricare ***pdfx***.
6. La novità, invece, è che il pacchetto ***pdfx*** è bene che sia caricato quasi per primo (a parte l'eventuale pacchetto ***imakeidx***), comunque prima del pacchetto che specifica la codifica di ingresso. Si noti che l'opzione `[a-2b]` serve per specificare che si sta parlando delle specifiche per l'archiviabilità nel loro secondo livello e secondo lo standard che non richiede i file di tipo `Tagged PDF`. L'opzione preimpostata sarebbe `[a-1b]`, che a rigore non sarebbe necessaria, ma è conveniente prendere l'abitudine di specificarla così da ricordare a che livello di conformità si vuole arrivare. Invece, l'opzione `[a-2b]` va espressamente specificata.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

7. Prima di caricare il pacchetto *pdfx* si specificano i metadati come argomento dell'ambiente *VerbatimOuT*; in questo modo se si modificano i metadati non è necessario ricordarsi di cancellare il file `.xmpdata` preesistente.
8. Dopo si possono caricare altri pacchetti, ma il solito gruppetto per le codifiche di ingresso e dei font con cui comporre il documento deve essere caricato dopo *pdfx*.
9. Fra i pacchetti successivi si può caricare *babel* per gestire le lingue come al solito.
10. Per ultimo si può caricare *hyperref* ma *assolutamente senza specificare nessuna opzione*. Le opzioni si possono impostare con `\hypersetup`. A questo proposito si veda il prossimo paragrafo.
11. Si presti particolare attenzione all'uso dei pacchetti *tikz* e *pgfplots*; questi pacchetti sono bellissimi e potentissimi, "tanto" che possono gestire i colori con le trasparenze. Si faccia quindi attenzione a non attivare mai, direttamente o indirettamente, la parola chiave `opacity`. Ma si abbia l'attenzione di stare sul "chi vive" quando si esegue il controllo della compatibilità; col livello 2 non è necessario preoccuparsi delle trasparenze.
12. Dopo *hyperref* normalmente non si possono specificare altri pacchetti; le poche eccezioni riguardano quelli per le lingue che si scrivono da destra a sinistra; un'altra possibile eccezione è quella del pacchetto *glossaries*; questo quindi implica che l'utente deve sempre controllare la documentazione di ciascuno dei pacchetti che usa per vedere se contiene delle limitazioni nell'ordine di caricamento.
13. Quindi si possono definire le eventuali macro utili per la composizione del documento specifico. A rigore queste macro spesso possono essere specificate prima di caricare *hyperref*.
14. Se fra le lingue c'è l'italiano, si possono specificare certi comandi che impostano alcune funzionalità specifiche del modulo *italian.ldf* usato da *babel* per comporre in questa lingua. Per maggiori dettagli si esamini la documentazione leggibile da terminale mediante il comando

    ```
    \texdoc babel-italian
    ```

15. Il preambolo finisce con `\begin{document}`.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

Se si decide invece di servirsi dell'ambiente *filecontents*, se si dispone di una versione sufficientemente aggiornata del sistema TEX, la parte di preambolo che ci interessa cambia semplicemente in questo modo.

```
% !TEX encoding = UTF-8 Unicode
% !TEX TS-program = pdfLaTeX
%
\documentclass[...]{book}
%
\begin{filecontents}[overwrite,noheader]{\jobname.xmpdata}
    \Autor{Mario Rossi\sep
           Carlo Bianchi\sep
           Bruno Neri}
    \Title{Il formato archiviabile}
    \Subject{Procedure to produce archivable documents}
    \Keywords{PDF\sep
              PDF/A\sep
              ISO 19005\sep
              metadata embedding}
    \Organisation{Archivi Riuniti}
\end{filecontents}
%
\usepackage[a-2b]{pdfx}% oppure [a-1b]
% Codifiche, lingue e font
...
% Altri pacchetti, impostazioni, definizioni
...
\begin{document}
```

e, come si vede, le modifiche sono minime rispetto al caso in cui si usi il pacchetto *fancyvrb*.

Non c'è altro da aggiungere relativamente al preambolo, ma è opportuno essere sempre coscienti di quanto viene evidenziato nel prossimo paragrafo relativo ad *hyperref*.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

3.3 Le opzioni di *hyperref*

Si è già detto che *pdfx* carica lui stesso il pacchetto *hyperref* con l'opzione *pdfa*.

Non sarebbe quindi necessario ricaricare il pacchetto *hyperref* nel preambolo; tuttavia durante le prime fasi di lavorazione del documento, il lavoro potrebbe essere più spedito se non ci si preoccupasse dei metadati. Perciò si potrebbe commentare la riga dove si carica *pdfx*. Allora se si omettesse di "ricaricare" *hyperref*, il comando `\hypersetup` produrrebbe un errore. Caricando invece *hyperref* quel comando non produce nessun errore. Tuttavia, se si caricasse *hyperref* direttamente con le sue opzioni con un comando del tipo

`\usepackage[`⟨*opzioni*⟩`]{hyperref}`

quando si decommenti il comando per caricare *pdfx* si avrebbe un altro errore, quello di *Option clash* (conflitto di opzioni), in quanto *pdfx* ha già caricato *hyperref* con la semplice lista di opzioni `[pdfa]`.

Questo "strano" comportamento dipende dal modo con cui il programma di tipocomposizione carica i pacchetti; infatti quando un pacchetto viene caricato vengono memorizzati il suo nome con la data e la versione, e la sua lista di opzioni; se il pacchetto viene invocato di nuovo il programma confronta il nome, la data e la versione con i nomi e le date già memorizzati e, se trova una coincidenza, confronta la lista di opzioni; esso non produce nessun errore, se le opzioni specificate la seconda volta sono un sottoinsieme di quelle specificate la prima volta, ma, se non è questo il caso, emette il messaggio d'errore *Option clash*. Nel nostro caso *hyperref* viene caricato una prima volta con l'opzione `[pdfa]` e una seconda volta senza opzioni, cioè con un insieme di opzioni vuoto; è ovvio che un insieme vuoto è un sottoinsieme di un insieme non vuoto, quindi non viene emesso nessun messaggio d'errore. Le opzioni che interessano vengono invece specificate con `\hypersetup` che può essere usato diverse volte anche con opzioni ripetute e/o persino contrastanti, che vengono accumulate in una lista e nel caso di opzioni contrastanti resta in vigore l'ultima specificata.

Complicato? Un pochino, ma tutto ciò è rilevante per produrre i file archiviabili solo in quanto il pacchetto *hyperref* viene già caricato da *pdfx*. Basta seguire le indicazioni specificate in questo e nel paragrafo precedente per non incorrere in errori.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

Personalmente, quando è il momento di caricare *hyperref*, preferisco usare questo codice[1]:

```
\unless\ifcsname ver@hyperref.sty\endcsname
\usepackage{hyperref}\fi
\hypersetup{⟨opzioni per hyperref⟩}
```

Naturalmente questo codice è funzionale al precedente caricamento di *pdfx*; la stringa `ver@hyperref.sty` costituisce il nome di una macro di cui il precedente comando condizionale controlla l'esistenza; questa macro, infatti, è definita solo se il pacchetto il cui nome segue il prefisso `ver@` è già stato caricato.

3.4 La verifica di conformità

Resta da eseguire la verifica di conformità sia per avere la certezza che non si siano infiltrati degli errori nei font o nei file immessi, sia per poter disporre della certificazione di conformità. Questo argomento verrà trattato nel capitolo 5.

[1] I comandi che seguono, `\unless`, `\ifcsname` e `\endcsname`, fanno parte del nucleo di LaTeX; non sono macro da definire.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

Creazione diretta di file archiviabili con lualatex 4

Vale la pena di segnalare che prima del 2016 non era possibile creare documenti archiviabili con lualatex. Questa affermazione era specificatamente scritta nella documentazione del pacchetto *pdfx*. In realtà ci si riusciva lo stesso se si era disposti ad apportare modifiche (nei limiti della licenza d'uso) ai file ufficiali, ma la cosa non era né elegante né affidabile. Dal 2016 le cose sono cambiate e si possono produrre documenti archiviabili sia con xelatex, sia con lualatex.

Con xelatex bisogna seguire delle strade un po' contorte, e per questo qui accenno solo al fatto che bisogna comporre il documento archiviabile procedendo a mano con la linea di comando e specificando un fattore di compressione nullo. Il file che si ottiene contiene i metadati in chiaro non compressi, ma le sue dimensioni possono essere fino a dieci volte maggiori di quelle che si otterrebbero con un file PDF composto regolarmente. È vero che superata la verifica di conformità, è poi possibile comprimere il file senza comprimere i metadati; questo si può ottenere con Adobe Acrobat Pro; siccome in questa guida voglio privilegiare i programmi freeware e opensource, non ne dico di più.

4.1 Verifica delle figure e dei file testuali

In questo paragrafo si possono ripetere le stesse identiche raccomandazioni già scritte per la composizione mediante pdflatex.

Ripeto solo di verificare i font, i colori, i profili di colore, le trasparenze, esattamente come si è raccomandato di fare nel capitolo precedente. È meglio se si fanno queste verifiche almeno via via che un file grafico o un file di testo deve essere immesso nel flusso di composizione, perché, se

si manifestano errori, li si scopre subito; se la verifica viene fatta solo a documento completato la ricerca degli errori e delle cose da correggere può essere difficile. Questa raccomandazione vale anche se si producono disegni con *tikz* e/o *pgfplots*.

4.2 Il preambolo del documento

Il preambolo di un documento da compilare con lualatex non è molto diverso da quello necessario per comporre con pdflatex.

```
% !TEX encoding = UTF-8 Unicode
% !TEX engine = LuaLaTeX
%
\documentclass[⟨opzioni⟩]{⟨classe⟩}
\usepackage[⟨opzioni⟩]{imakeidx} % Facoltativo
\usepackage{fancyvrb}
\begin{VerbatimOut}{\jobname.xmpdata}
⟨metadati⟩
\end{VerbatimOut}
%
\usepackage[a-2b]{pdfx}% oppure [a-1b]
%
\usepackage{fontspec}[⟨opzioni⟩]
\setmainfont[⟨opzioni per il font principale⟩]{⟨font con grazie⟩}
\setsansfont[⟨opzioni per il font senza grazie⟩]{⟨font senza grazie⟩}
\setmonofont[⟨opzioni per il font teletype⟩]{⟨font teletype⟩}
%
\usepackage{polyglossia}
\setmainlanguage[⟨opzioni per lingua principale⟩]{⟨lingua principale⟩}
⟨eventuali specificazioni di altre lingue⟩
%
\usepackage{amsmath}% se occorre va caricato prima di unicode-math
\usepackage{unicode-math}
\setmathfont[⟨opzioni per il font matematico⟩]{⟨font matematico⟩}
%
⟨altri pacchetti⟩
%
```


File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

```
\usepackage{hyperref}
\hypersetup[⟨opzioni per hyperref⟩]
%
⟨pacchetti, impostazioni e macro specifici per questo documento⟩
\begin{document}
```

Come si vede, le differenze principali rispetto al preambolo descritto per l'uso con il programma pdflatex sono le seguenti.

1. Non viene specificata nessuna opzione né nessun pacchetto per la codifica di ingresso. I file per lualatex *devono* essere scritti e salvati con la codifica UTF-8 e non ci sono alternative.
2. I font di default se non si specifica nessuno dei comandi `\setmainfont`, `\setsansfont` e `\setmonofont` sono i font Latin Modern nella loro versione OpenType. Se si devono usare altri font latini o con altri alfabeti, allora sarebbe preferibile usare questi comandi per specificare font più completi, ma, attenzione! è vero che i font Latin Modern OpenType contengono solo l'alfabeto latino, ma sono dotati dei corpi ottici, per cui i corpi piccoli sono disegnati apposta, e non sono font da 10 rimpiccioliti, col rischio che i tratti sottili svaniscano; così i corpi più grandi sono disegnati apposta e non sono font da 10 ingranditi col rischio che diventino troppo neri.
   È bene ricordarsi di questo fatto; in ogni caso i font CMU OpenType contengono molti alfabeti e molti caratteri speciali e sostituiscono bene i Latin Modern, anche se non dispongono dei corpi ottici. È bene ricordare che se il documento contiene della matematica, bisogna scrupolosamente osservare quanto scritto più sotto, ma è importante definire anche un font matematico OpenType che si accordi con i font testuali. I font OpenType Latin Modern Math si accordano molto bene con i font CMU, ma ritengo che i font XITS Math siano più completi; il loro disegno è leggermente più scuro dei Latin Modern Math, e qualche segno è un po' diverso; tuttavia per accorgersi di queste piccolissime differenze bisogna avere l'occhio molto esercitato.
3. La gestione delle lingue tramite il pacchetto *polyglossia* è diversa da quella che si può ottenere con *babel*; la differenza è che ogni lingua (o gruppi di lingue) viene specificata separatamente con un comando apposito `\setmainlanguage` per la lingua principale, `\setotherlanguage` per ciascuna delle altre lingue oppure

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

`\setotherlanguages` per gruppi di lingue. Ogni lingua può essere specificata con le sue opzioni, tranne quelle che fanno parte di un gruppo di lingue.

4. Se si usa la matematica bisogna usare il pacchetto *unicode-math*; se si desidera usare tutta la collezione di macro di *amsmath*, questo pacchetto deve essere caricato prima di *unicode-math*; inoltre *non bisogna assolutamente* invocare il pacchetto *amssymb*. Invece è opportuno invocare un font matematico OpenType perché questo tipo di font è tale da rendere inutile il concetto di famiglie o gruppi matematici; quindi si elimina il limite massimo di 16 famiglie, limite che invece è presente se si compone con pdflatex.
5. Il pacchetto *hyperref* e le sue opzioni vanno impostate esattamente come si è spiegato per il preambolo dei file da compilare con pdflatex.

4.3 Commenti alla compilazione con lualatex

Come si vede, dunque, le differenze del preambolo dei documenti da compilare con lualatex rispetto a quelli da compilare con pdflatex sono minime e riguardano solo la gestione dei font e quella delle lingue.

Queste gestioni sono più funzionali e performanti di quelle che si possono fare con pdflatex. Per esempio si può specificare anche nel corpo del documento un comando per impostare una lingua, ma si può anche reimpostare una lingua già impostata nel preambolo specificando opzioni diverse. Per i font si possono definire famiglie di font con nomi scelti in modo che siano l'agglutinazione di un nome di lingua con la stringa `font`; così facendo, quando si scrive in quella lingua si usa direttamente quella famiglia di font.

Tastiera e sistema operativo permettendo, si può sempre introdurre nel file sorgente il testo direttamente con l'alfabeto specifico di quella lingua. Nella guida *L'Arte di scrivere in diverse lingue con (Xe)LaTeX* (Beccari e Gordini, 2015) si è composto nello stesso documento in diverse lingue che usano i caratteri latini, ma si è anche scritto in greco, in russo, in ebraico, in arabo, in cinese, in giapponese, e in coreano; il tutto è stato fatto su un Mac Book Pro, con il sistema operativo OS X, versione 10.13.3, con la tastiera USA Extended, ma avendo impostato i driver di tastiera anche per quelle lingue esotiche.

Infine va notato che usare lualatex quando si debbano usare font OpenType è preferibile rispetto a usare xelatex in quanto il primo può usare

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

le funzionalità complete del pacchetto *microtype*, mentre il secondo le può usare solo parzialmente. Come si è detto a inizio capitolo, lualatex può produrre direttamente file PDF archiviabili, mentre per ottenerli con xelatex bisogna procedere a mano con una certa dose di postprocessing.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

Verifica della conformità 5

Dal 2008 ho verificato la conformità con le norme PDF/A di file da archiviare servendomi di successive versioni del programma commerciale Adobe Acrobat Pro.

In questi anni fino al 2016, secondo me, era la via più affidabile per eseguire la verifica di conformità tramite il suo modulo preflight.

Le verifiche on line mi hanno sempre dato dei dispiaceri; inoltre non esistevano programmi freeware e opensource che svolgessero questa funzione.

Ero al corrente che un Consorzio Europeo stesse lavorando ad un progetto freeware e opensource; finalmente nel settembre 2016 ho scaricato dal sito http://www.digitalmeetsculture.net/article/verapdf-0-20-released/[1]. Oggi (2018) questo programma è scaricabile dal suo sito in versione stabile, non più in versione "beta".

In quel sito ci sono le istruzioni per eseguire i download dei pacchetti necessari per compilare e installare il programma verapdf e le librerie accessorie su piattaforme diverse, con sistemi operativi differenti e di differenti versioni. Le piattaforme servite sono le macchine Windows, Mac e Linux, i sistemi operativi sono quelli abbastanza recenti fino agli ultimi distribuiti per quelle piattaforme. L'installazione nel 2016 era ancora un poco "ruspante" perché si trattava di una versione sperimentale, ma ora, nella versione stabile, gli script di installazione eseguono da soli una serie di verifiche che prima era necessario eseguire a mano.

Personalmente ho eseguito tutte le operazioni di decompressione dei pacchetti, di compilazione del software e di installazione in una cartella apposita; ho poi inserito un link simbolico in una cartella accessibile con il mio PATH di utente; questo link simbolico punta al programma che esegue

[1] Nell'indirizzo segnalato, valido nel settembre 2016, la parte 0-20 è il numero della versione sperimentale ma già usabile del programma verapdf

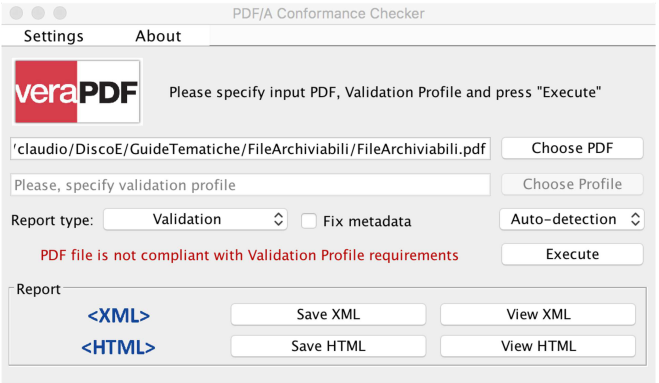

Figura 5.1 Finestra di dialogo di verapdf-gui con un responso negativo dopo aver eseguito l'analisi di un file

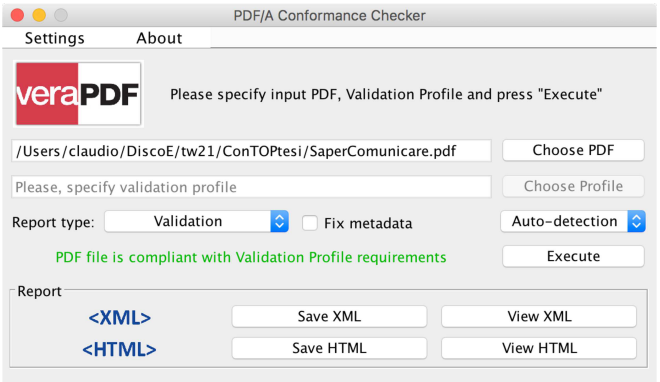

Figura 5.2 Finestra di dialogo di verapdf-gui con un responso positivo dopo aver eseguito l'analisi di un file

l'interfaccia grafica verapdf-gui del programma di verifica. Il tutto funziona velocemente e senza problemi. I link simbolici erano una prerogativa delle piattaforme UNIX; ora anche quelle Windows dispongono di questi comodissimi strumenti.

In un terminale do il comando

```
verapdf-gui
```

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

che mi apre la finestra di dialogo, nella quale basta selezionare la cartella e il nome completo del file da verificare; clicco il bottone Chose PDF, figura 5.1, e mi si apre la solita interfaccia grafica per scegliere la cartella dove si trova il file da verificare; lo marco come file da controllare.

Fatto questo, clicco il bottone Execute e parte la verifica; questa termina con un messaggio nella finestra di dialogo che comunica l'esito della verifica; che questo esito sia positivo o negativo è ora possibile cliccare su uno dei bottoni XML oppure HTML, per leggere ed eventualmente salvare il responso. La figura 5.1 mostra il risultato negativo dell'analisi di questo stesso documento, a cui ho deliberatamente tolto i metadati. Ovviamente ho poi ripristinato i metadati in questo documento, per cui ora esso è conforme alle norme. Nella figura 5.2 compare invece il responso per un altro documento conforme alla normativa PDF/A-1b.

Si noti che i livelli di conformità 1 e 2 sono "diversamente schizzinosi"; Questo stesso documento era conforme al livello 1; ho provato a verificarne la conformità al livello 2 ma è risultato che la figura 1.2 presentava dei problemi con l'interpolazione permessa, metre questa è vietata con il livello 2. Ho corretto questa specifica seguendo le procedure indicate nei paragrafi precedenti, poi ho controllato di nuovo e la conformità era a posto anche per il livello 2. Anche questo piccolo incidente dimostra quanto sia delicato produrre file conformi alle norme ISO per l'archiviabilità.

Quando questo responso è positivo il file salvato funziona da certificazione; se esso è negativo contiene l'elenco di tutte le cose che non vanno, per cui diventa relativamente facile provvedere.

Se si sono creati i file da archiviare secondo le procedure indicate, il responso potrebbe essere negativo nei singoli file da includere; per questo è meglio verificare ciascun file, prima di immetterlo nel documento finale, in modo che lo si possa "correggere"; altrimenti il responso è praticamente sempre positivo sui documenti completati.

Questo dimostra che le difficoltà nel produrre un file archiviabile secondo le norme ISO 19005 non consistono tanto nella compilazione del documento conforme, quanto nella gestione dei singoli elementi che contribuiscono a formare il documento.

Secondo me sarebbe sempre preferibile usare il procedimento indicato con il programma lualatex. Mi rendo conto però che oggi questo programma è usato molto meno di pdflatex e un utente che non l'abbia mai usato potrebbe esserne intimorito; il fatto che lualatex incorpori un interprete del linguaggio

Lua integrato con il linguaggio TEX è un'altra fonte di apprensione, anche se ho mostrato che la creazione diretta di un file archiviabile non richiede nessun comando Lua, almeno per quel che riguarda l'archiviabilità; quindi il corpo del documento può tranquillamente essere privo di tratti in linguaggio Lua e in sostanza può essere identico al corpo di un documento da compilare con pdflatex. Nessun problema; anche pdflatex può produrre file PDF archiviabili, solo che con certi suoi font preimpostati ci possono essere alcuni problemi in più.

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12

Conclusione

In questa breve guida ho mostrato come si possono creare file archiviabili secondo le norme ISO 19005 usando i programmi del sistema TEX e, per la verifica, solo programmi freeware e opensource.

Ho messo in luce i problemi che possono nascere dai font usati, dalle immagini con trasparenze, dai colori usati nel testo e nei file con contenuto grafico, dai collegamenti ipertestuali, dai conflitti fra pacchetti.

Ho concluso che la via più sicura per ottenere un file PDF archiviabile è quella di usare il programma di tipocomposizione lualatex; non è l'unica strada ma è quella che a me sembra più efficace nel produrre file conformi alle norme.

Per onestà devo dire che ho provato un solo esempio che mi ha portato a creare un file conforme alla norma PDF/A-1a che richiede anche la caratteristica di file `Tagged PDF`. Come ho segnalato all'inizio, oggi il programmi del sistema TEX basati su LATEX non sono ancora in grado di produrre file PDF di tipo tagged, anche se esiste già un pacchetto sperimentale, *tagpdf* che è sulla buona strada per raggiungere l'obbiettivo. Non ho provato ancora a usarlo, ma ho fatto un esperimento usando due programmi esterni. Ho composto un file conforme alle norma PDF/A-1b, e l'ho passato a LibreOffice Writer; da questo l'ho esportato come Tagged PDF. Ho poi aperto il file con il modulo Preflight di Acrobat Pro e ne ho verificato la compatibilità con la norma PDF/A-1b a cui il file originale era conforme, ma non lo era più. Ho chiesto a Preflight di convertirlo in PDF/A-1a; con mia sorpresa dopo aver elaborato il file un paio di minuti, ha terminato la conversione, esponendo un lungo elenco di operazioni eseguite, ma alla fine la verifica di conformità del nuovo file era positiva. Non posso dire che questa sia la strada da seguire in generale, almeno finché il programmi del sistema TEX non saranno autosufficienti, perché non è corretto fare affermazioni del genere avendo avuto successo con un solo esempio; al massimo posso affermare che non è impossibile.

Bibliografia

Beccari, C. (2016). *Tipocomporre in italiano*. GuIT. http://www.guitex.org/home/en/guide-tematiche.

Beccari, C. e Gordini, T. (2015). *L'Arte di scrivere in diverse lingue con (Xe)LaTeX*. GuIT. http://www.guitex.org/home/en/doc.

Burn, B. (2018). «Lwarp, LaTeX html5 – The `lwarp` package». PDF document. Leggibile con `texdoc lwarp` su TeX Live.

C.V. Radhakrishnan, Hàn Thế Thành, Ross Moore e Peter Selinger (2016). «Generazione di file PDF/X e PDF/A». *ArsTeXnica*, 21.

Fischer, U. (2018). «The `tagpdf` package». PDF document. Leggible con `texdoc tagpdf` con TeX Live.

ISO 19005-1 (2005). «ISO 19005-1:2005 – Document management – Electronic document file format for long-term preservation». Electronic format. URL http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38920.

ISO 19005-2 (2011). «ISO 19005-2:2011 – Document management – Electronic document file format for long-term preservation». Electronic format. URL http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50655.

ISO 19005-3 (2012). «ISO 19005-3:2012 – Document management – Electronic document file format for long-term preservation». Electronic format. URL http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?ics1=37&ics2=100&ics3=99&csnumber=57229.

Indice analitico

A

ambiente
- *filecontents*, 28, 29, 33, 34, 37
- *filecontents**, 27, 33, 34
- *filecontets**, 34
- *picture*, 19
- *VerbatimOuT*, 36
- *VerbatimOut*, 29, 33, 34

B

`\begin`, 33, 34, 41, 42

C

classe
- *book*, 29
- *guidatematica*, 29
- *standalone*, 25
- *standlone*, 23

D

`\documentclass`, 27, 28, 33, 35, 41

E

`\end`, 33, 34, 41
`\endcsname`, 39

F

`\fi`, 39
file
- `.eps`, 25
- `.xmpdata`, 34, 36
- `file-a.pdf`, 30
- `file-a.tex`, 27, 30
- `file-a.xmpdata`, 30
- `file.pdf`, 27, 30
- Tagged PDF, 4, 5, 9, 35, 49

H

`\hypersetup`, 33, 36, 38, 39, 42

I

`\ifcsname`, 39

J

`\jobname`, 30, 33, 34, 41

M

\mapstochar, 18, 19

N

\not, 18, 19

O

opzione
- *a4paper*, 27
- *noheader*, 28
- *overwrite*, 28
- *pdfa*, 20, 38

P

pacchetto
- *amsmath*, 43
- *amssymb*, 43
- *babel*, 36, 42
- *fancyvrb*, 29, 33, 34, 37
- *geometry*, 27
- *glossaries*, 36
- *hyperref*, 4, 6, 20, 36–39, 43
- *imakeidx*, 20, 35
- *italian.ldf*, 36
- *lwarp*, 4
- *mathdesign*, 18
- *microtype*, 44
- *pdfpages*, 30
- *pdfx*, 3, 4, 10, 13, 18, 20, 28, 30, 33–36, 38–40
- *pgfplots*, 36, 41
- *polyglossia*, 42
- *tagpdf*, 4, 49
- *tikz*, 13, 36, 41
- *TOPtesi*, 19
- *unicode-math*, 43
- *xcolor*, 13

programma
- Anteprima, 24
- distiller, 24, 25
- FontForge, 17
- ghostscript, 27
- GIMP, 14, 15, 23, 32
- inkscape, 14, 16, 17, 19, 23, 25
- lualatex, 6, 10, 12, 18, 19, 32, 33, 40–44, 47, 49
- luatex, 4
- pdflatex, 6, 12, 18, 19, 23, 25, 30, 32, 33, 40–43, 47, 48
- pdftex, 12, 18
- preflight, 45
- Preview, 24
- verapdf, 45
- verapdf-gui, 7, 46
- xelatex, 10, 40, 43, 44

S

\sep, 30
\setmainfont, 41, 42
\setmainlanguage, 41, 42
\setmathfont, 41
\setmonofont, 41, 42
\setotherlanguage, 42
\setotherlanguages, 43
\setsansfont, 41, 42

U

\unless, 39
\usepackage, 33, 38, 39, 41, 42

File archiviabili, versione v.3.0.02, ultima revisione del 2020-05-12