RunPgms - ein Metaprogramm zur Ausführung der Auswertung einer klinischen Prüfung auf 'Knopfdruck'

Stefan Beimel
ASTA Medica AG / Frankfurt am Main

Abstract

Die Analyse einer klinischen Prüfung besteht aus einer Reihe von Programmen zum Erzeugen von Formatkatalogen, permanenten Datensätzen, Rohdatenlisten, Auswertungstabellen und Statistiken. Die Anzahl dieser Programme kann je nach Umfang der Studie leicht 30 und mehr betragen. Die Anzahl der Ausgabedateien, die die Listen, Tabellen, Graphiken und Statistiken enthalten, ist in der Regel noch höher, da jedes Programm mindestens eine Ausgabedatei erzeugt. Um eine zusammenfassende Auswertung auf 'Knopfdruck' zu ermöglichen, gibt es für jede klinische Prüfung ein SAS-Programm RunPgms.sas.

Es besteht aus einer Reihe von einfachen Makroaufrufen, die im wesentlichen die einzelnen Programme ausführen, deren Log-Dateien durchsuchen, die Outputs der Programme suchen, die Ausgabedateien zusammenfassen und ein Inhaltsverzeichnis erstellen.

Dieses Vorgehen hat mehrere Vorteile gegenüber dem Lauf der einzelnen Programme. Es erleichtert
- den Überblick über Programm- und Ausgabedateien,
- das schnelle Auffinden von Fehlermeldungen und anderen verdächtigen Informationen,
- die Re-analyse im Falle von Änderungen der Daten,
- das Umsortieren der Tabellen,
- das Ausdrucken der Analyseergebnisse (wenige große statt vieler kleiner Dateien),
- die Dokumentation der Programmläufe und
- damit die Qualitätssicherung der Programmierung.

Einleitung

Zwei zeitraubende, monotone und damit fehlerträchtige Tätigkeiten lassen sich bei der Analyse klinischer Studien immer wieder beobachten:

Zum einen gibt es eine Fülle von Informationen über die Organisation und den Lauf der Programme zu verarbeiten. Es muß die Übersicht über die Programme und deren Output behalten werden, und der fehlerfreie Programmlauf muß mit Hilfe der entstandenen Log-Dateien überprüft werden. Diese Log-Dateien können insgesamt leicht Umfänge von 500 und mehr Seiten annehmen. Außerdem ist das Ausdrucken der Ergebnisse ist aufgrund der vielen Output-Dateien sehr zeitaufwendig.

Zum zweiten besteht oft die Notwendigkeit, eine umfangreiche Analyse wiederholt durchzuführen. Dies ist insbesondere erforderlich, wenn sich Daten geändert haben oder Tabellen hinzugefügt, entfernt oder umsortiert werden müssen mit der Notwendigkeit, die Numerierung zu ändern. Eine wiederholte Analyse kann auch geplant sein im Falle von Interimsanalysen oder regelmäßigen Updates von Berichten (z. B. ISS).

Diese Schritte versucht das Programm RunPgms.sas (Run Programs) zu vereinfachen. Es besteht aus einer Reihen von einfachen Makroaufrufen, die in drei Abschnitten zusammengefaßt werden können.

- Ausführen der Programme
- Numerieren und Einsammeln der Tabellen
- Erstellung der Inhaltsverzeichnisse

Vorgehensweise bei der Programmierung

Zu Beginn jeder Studie wird ein kleines Programm StartUp.sas erstellt, das allgemeine Informationen zur Studie wie Datenverzeichnis und Studien-ID enthält und libnames setzt.

Da RunPgms.sas aufgrund seines Aufbaus (siehe Abb. 7) wie eine Gliederung der Programme und Outputs betrachtet werden kann, sollte es das erste Programm sein, das bei der Analyse einer klinischen Prüfung erstellt wird.

Danach erfolgt die Programmierung der eigentlichen Analyseprogramme. Dazu gehört auch das Programm, das den studienspezifischen Formatkatalog erzeugt und Programme, die aus der ursprünglichen Datenbank permanente Datensätze mit abgeleiteten Variablen erstellen. Diese Programme müssen eingebunden werden, damit bei Änderungen an den Originaldaten die Neuberechnung der abgeleiteten Daten nicht übersehen wird.

Der endgültige Lauf der Programme erfolgt dann innerhalb von RunPgms.sas im Batchmodus: unter VMS mit dem Befehl 'sas runpgms' und unter Windows mit rechter Maustaste - 'BatchSubmit' - der Knopfdruck, der die gesamte Analyse startet. Es ist wichtig, das Programm im Batchmodus zu starten, um ein geschlossenes Log zu erhalten. Das Log von RunPgms.sas enthält die Namen der Ausgabedateien der einzelnen Programme und sämtliche ERRORs, WARNINGs und verdächtigen NOTEs aus deren Log. RunPgms.log muß ausgedruckt und, evtl. mit Kommentaren versehen, archiviert werden. Damit gibt es eine Dokumentation über den Lauf der Programme und die Kontrolle der Logs.

RunPgms.sas faßt sämtliche Lis-Dateien (d. h. den Output der Einzelprogramme) in Out-Dateien zusammen, die gemeinsam mit den ebenfalls erstellten Inhaltsverzeichnissen ausgedruckt werden müssen.

Ausführen der Programme

Führt man ein Programm im Batchmodus 'von Hand' aus, muß man üblicherweise den Befehl 'sas program' abschicken, auf das Programmende warten, das Log öffnen und aufmerksam durchsuchen, den Output suchen und betrachten - und nicht zuletzt aufpassen, daß man kein Programm vergißt. Da die Analyse einer klinischen Prüfung aus 30 und mehr Programmen bestehen kann, ist diese Prozedur sehr mühsam - außerdem gibt es keine Dokumentation darüber, daß das Log auch wirklich durchsucht wurde. Die o.g. Arbeitsschritte werden durch RunPgms.sas sehr vereinfacht.

In RunPgms.sas wird jedes einzelne Programm wird mit Hilfe des Makros %RunPgm ausgeführt. %RunPgm erkennt das Verzeichnis, in dem sich die Programmdatei befindet und führt es mit Hilfe von %include aus. Vorher und nachher wird das komplette Work-Verzeichnis gelöscht, um Fehlfunktionen aufgrund historischer Informationen zu vermeiden.

Das Log wird im gleichen Verzeichnis wie das Programm gespeichert. Dieses Log wird durchsucht nach

```
index ( line, "ERROR"           ) = 1  or
index ( line, "WARNING"         ) = 1  or
index ( line, "AT LEAST ONE W.D")      or
index ( line, "UNINITIALIZED"   )      or
index ( line, "REPEATS OF BY"   )      or
index ( line, "TRUNCATE"        )      or
index ( line, "IGNORED"         )      or
index ( line, "_ERROR_=1"       )      or
index ( line, "INVALID"         )      or
index ( line, "UNKNOWN"         )      or
index ( line, " 0 OBSERVATIONS" )      or
index ( line, " 0 VARIABLES"    )      or
index ( line, " 0 RECORDS"      )      or
index ( line, "NO STATISTICS ARE COMPUTED"          ) or
index ( line, "CHARACTER VALUES HAVE BEEN CONVERTED") or
index ( line, "NUMERIC VALUES HAVE BEEN CONVERTED"  ) or
index ( line, "DIVISION BY ZERO DETECTED"           )
```

Das Ergebnis dieser Suche wird in RunPgms.log geschrieben (Abb. 2).

Zu Beginn jedes %RunPgm wird die Systemzeit in einer Makrovariablen gespeichert. Nachdem das %include beendet ist, werden mit Hilfe von Betriebssystemkommandos die relevanten Verzeichnisse nach Dateien durchsucht, die nach dem Start des Programms entstanden oder verändert wurden. Sie werden als Output des Programmes interpretiert und in RunPgms.log gelistet.

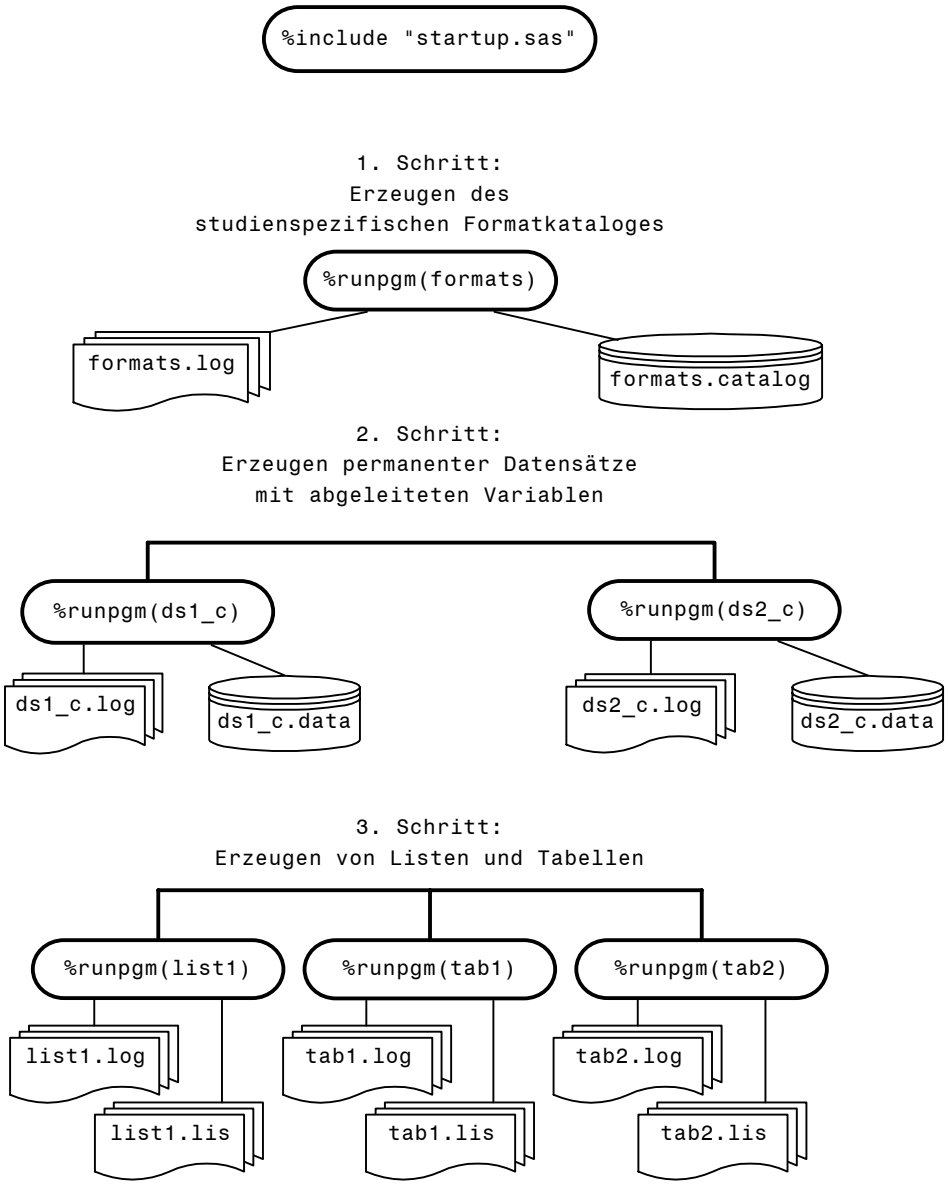

Abbildung 1: Beispiel von Aufrufen des Makros %RunPgm und erzeugte Dateien

```
960
961        %runpgm(formats);
Start of [D18506.ST3109.DATA]FORMATS.SAS _____________________________

Program created BIOMDSK:[D18506.ST3109.DATA]FORMATS.SASEB$CATALOG;1

No Warnings or Suspicious Notes Found in Log
1089
.
.
4277       %runpgm(tparscor);
Start of [D18506.ST3109.PROG]TPARSCOR.SAS _____________________________

Program created BIOMDSK:[D18506.ST3109.PROG]TPARSCOR.LIS;7
Program created BIOMDSK:[D18506.ST3109.PROG]TGLOTOL.LIS;7

Warnings or Suspicious Notes Found in Log:

!! 4656 NOTE: At least one W.D format was too small for the number to be
              printed. The decimal may be shifted by the best format.
!! 4864 WARNING: Not all variables in the list VALUE1-VALUE7 were found.
.
.
4944       %runpgm(trespsix);
```

Abbildung 2: RunPgms.log – von %RunPgm erzeugte Informationen (fett: Programmcode)

Numerieren und Einsammeln der Tabellen

Sind die Tabellennummern in den Einzelprogrammen 'fest verdrahtet', ist es sehr aufwendig, diese Numerierung nachträglich zu ändern. Wird z. B. eine Tabelle eingefügt, müssen alle nachfolgenden Tabellennummern hochgezählt werden. Wenn diese Tabelle weit vorne ergänzt wird - und nach McMurphies Gesetzen ist das der Fall - kann es einen halben Tag und viele Nerven kosten, bis die Numerierung geändert ist.
In RunPgms.sas wird die Originalnumerierung der Tabellen der Reihe nach ersetzt. Das geschieht mit Hilfe des Makros %FOutColl (= collect into output file), das einmal pro erzeugter Lis-Datei aufgerufen wird. Die oben erwähnte Aufgabe des Einfügens einer Tabelle und die Umnumerierung der nachfolgenden Tabellen beschränkt sich damit auf das Einfügen einer Zeile in RunPgms.sas - geschätzter Aufwand 30 Sekunden.
Innerhalb des Makros wird die Haupttabellennummer, d. h. die erste Zahl in jeder ersten Titelzeile, ersetzt. Sie wird erhöht, wenn %FOutColl aufgerufen wird oder wenn sich die ursprüngliche Tabellennummer ändert. Die Information über die aktuelle Tabellennummer wird in der globalen Makrovariablen `glasttno` (= global last table number) gehalten. Es ist möglich, die gewünschte erste Tabellennummer innerhalb einer Datei zu übergeben. Das ist notwendig zu Beginn jedes Appendix, wenn die Nummer über Dateigrenzen festgehalten werden soll oder wenn Tabellennummern übersprungen werden müssen (siehe Abb. 3). Statt einer Nummer kann in der Lis-Datei auch der Platzhalter '???' verwendet werden. Der Bereich der Tabellennummern innerhalb einer Datei wird in RunPgms.log geschrieben (Abb. 6).

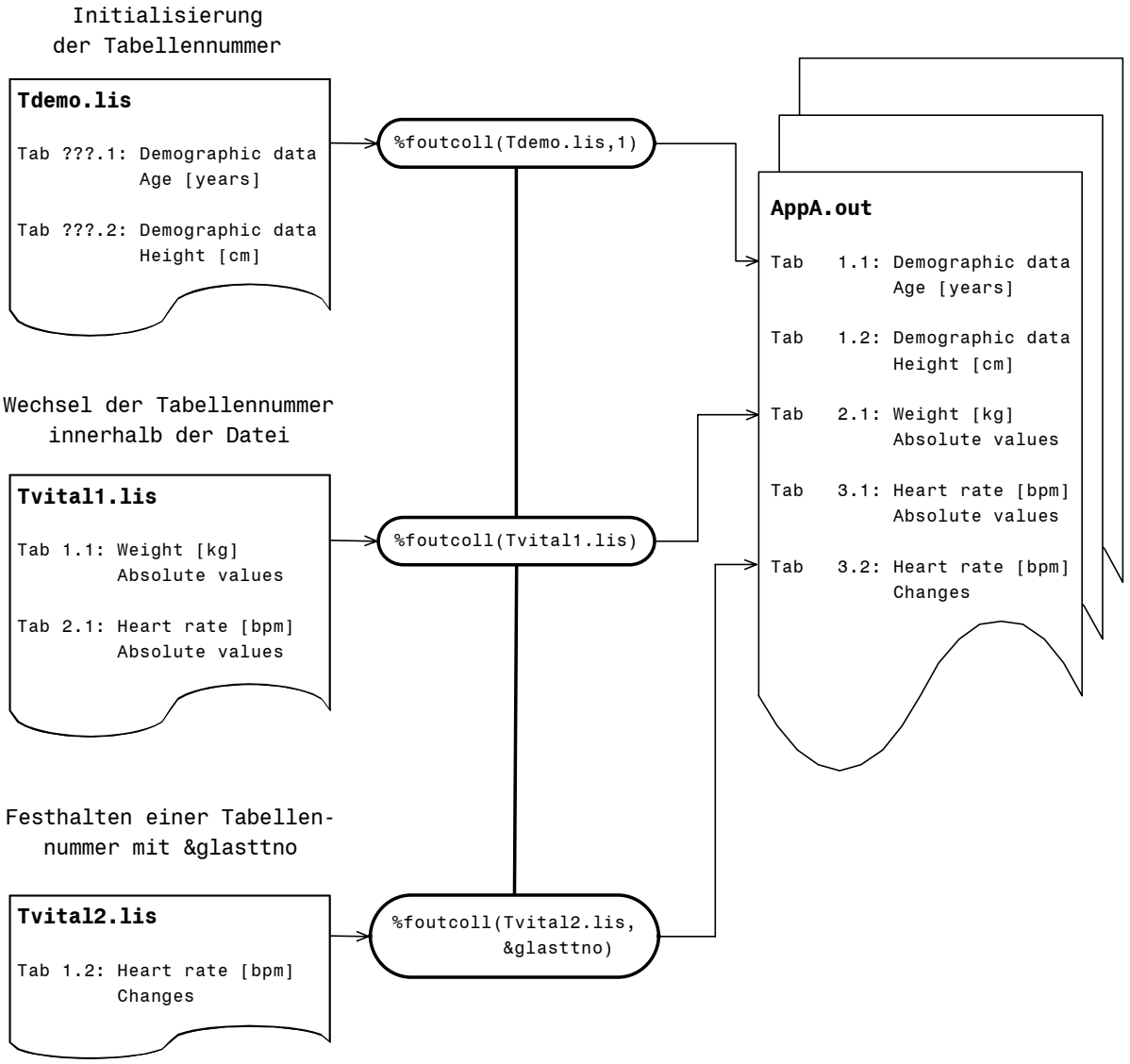

Abbildung 3: Ersetzen der Tabellennummern mit %FOutColl an einem Beispiel

Neben der Numerierung sammelt %FOutColl die Lis-Dateien in Out-Dateien zusammen, die zuvor mit %FOutInit (= initialise output file) erzeugt wurden (Abb. 5).
Dadurch wird das Ausdrucken der Dateien sehr viel schneller und auch sicherer, denn es kann sich zwischen Tabelle 7 und 8 in die Drucker-Warteschlange kein Brief an Mr. X gedrängelt haben.

Zu den Studienberichten der ASTA Medica AG gehören mehrere biometrische Anhänge (Appendix A – Haupttabellen , Appendix B - Zusatztabellen, Appendix C - Rohdatenlisten). Diese Anhänge unterscheiden sich zum Teil in ihrer Schriftgröße - und damit in dem, was auf eine Seite draufpaßt. Üblicherweise wird z. B. für Rohdatenlisten eine kleinere Schriftgröße gewählt und für Tabellen eine etwas größere. Für die Programmierung heißt das, daß sich der Output in `pagesize` und `linesize` unterscheidet. Das Seitenformat kann sich sogar

innerhalb eines Appendix ändern. Leider ist es nicht möglich, verschiedene Formate in einer Out-Datei zu speichern, da sie nicht gemeinsam gedruckt werden können.

Ein Beispiel, wie die Lis- und Out-Dateien organisiert sind und deren Verknüpfung innerhalb von RunPgms.sas zeigt Abb. 5.

Erstellen der Inhaltsverzeichnisse

Nachdem alle Out-Dateien erzeugt wurden, kann für jeden Appendix ein Inhaltsverzeichnis gedruckt werden. Mit Hilfe von %Toc (= table of contents) werden alle für diesen Anhang relevanten Out-Dateien nach Titelzeilen durchsucht. Ein Beispiel für ein Inhaltsverzeichnis ist in Abb. 4 zu sehen. Um diese Inhaltsverzeichnisse zu erstellen, müssen die Numerierung und die entsprechenden Titelzeilen streng hierarchisch gegliedert sein - das ist in SAS normalerweise nicht zwingend notwendig. Das Inhaltsverzeichnis ist eine gute Kontrolle, ob der logische Aufbau und die Reihenfolge der Tabellen den Vorgaben entsprechen.

```
Table of contents for appendix C
                                                                  page

C 1 Demographic data                                                 4

C 2 Adverse events                                                  77
    C 2.1 symptoms                                                  77
    C 2.2 descriptors                                               92

C 3 Concomitant medication                                         107

C 4 Laboratory data                                                109
    C 4.1 chemistries - enzymes, electrolytes                      109
    C 4.2 chemistries - substrates                                 126
    C 4.3 haematology                                              162
    C 4.4 urinalysis, dipstick findings                            197
    C 4.5 urinalysis, sediment                                     214

Last page of this appendix                                         241
```

Abbildung 4: Beispiel eines Inhaltsverzeichnisses

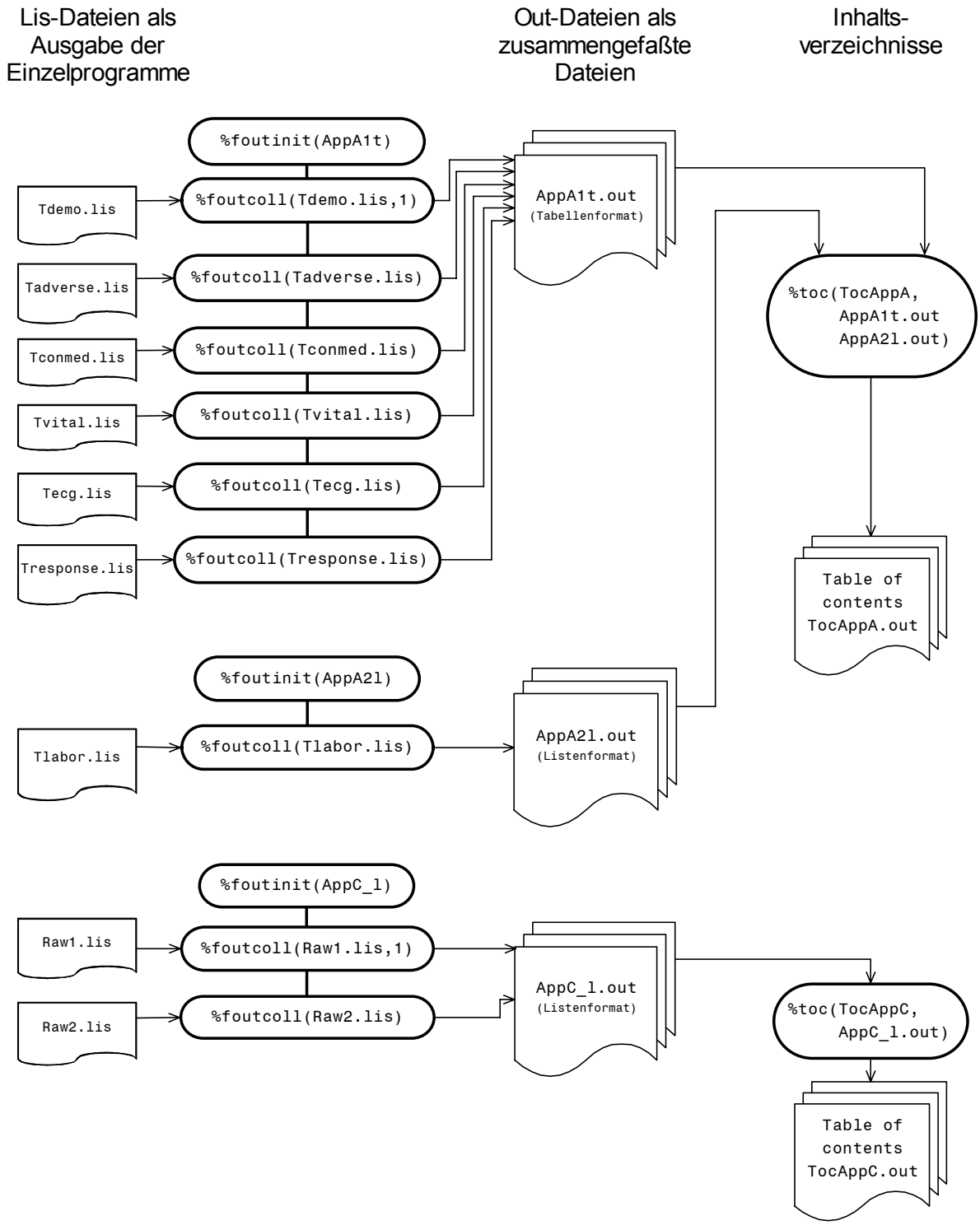

Abbildung 5: Beispiel von Lis-, Out-Dateien und Inhaltsverzeichnissen und deren Verknüpfung

```
________SETTING OF TABLE NUMBERS AND COLLECTING INTO AGGREGATED FILES______

15230      *** tables behind text *********************************;
15231      %foutinit(appa1t);
15232        %foutcoll(tdisp.lis,1);
Appending [D18506.ST3109.PROG]tdisp.lis ____ Table numbers:    1 -    1
15233        %foutcoll(tpred.lis);
Appending [D18506.ST3109.PROG]tpred.lis ____ Table numbers:    2 -    2

15234      %foutinit(appa2l);
15235        %foutcoll(tlabor.lis);
Appending [D18506.ST3109.PROG]tdisp.lis ____ Table numbers:    3 -    7

15261      *** Appendix B *****************************************;
15265      %foutinit(app_b_t);
15266        %foutcoll(enst_sta.lis,1);
Appending [D18506.ST3109.PROG]enst_sta.lis _ Table numbers:    1 -    1
15267        %foutcoll(last_sta.lis,&glasttno);
Appending [D18506.ST3109.PROG]last_sta.lis _ Table numbers:    1 -    1
15268        %foutcoll(vita_sta.lis,2);

________ CREATING TABLE OF CONTENTS ____________________________________

15284      title "Table of contents for tables behind text";
15285      %toc(tocappa, appa1t.out appa2l.out);

NOTE: The infile FILE is:
      File=$4$DKB300:[FB.FBDRV.D18506.ST3109.PROG]APPA1T.OUT

NOTE: The infile FILE is:
      File=$4$DKB300:[FB.FBDRV.D18506.ST3109.PROG]APPA2L.OUT

NOTE: The data set WORK._TOC has 67 observations and 3 variables.

NOTE: (PRINTTO) PRINT= "[D18506.ST3109.PROG]TOCAPPA.OUT" NEW

NOTE: The PROCEDURE REPORT printed pages 1-2.
```

Abbildung 6: RunPgms.log – von %FOutInit, %FOutColl und %Toc erzeugte Informationen (fett: Programmcode)

```
*** include startup.sas ************************************************;
%include "[AXXXXX.STYYYY.prog]startup.sas";

*** Run program which creates formats **********************************;
%runpgm ( formats );

*** Run programs which create permanent data sets **********************;
%runpgm ( pregns_c );
%runpgm ( labor_c  );

*** Run programs which create listings and tables **********************;
%runpgm ( raw      );
%runpgm ( tdemo    );
%runpgm ( tbabdesc );
%runpgm ( tlabor   );
%runpgm ( tpcount  );

*** Replacing table numbers and appending output ************************;

*** Appendix A **************************;
%foutinit ( appa1t );
    %foutcoll ( tage.lis   ,1);
    %foutcoll ( tweight.lis  );
    %foutcoll ( tbabdesc.lis );
    %foutcoll ( tadverse.lis );
    %foutcoll ( tlowpre.lis  );

%foutinit ( appa2l );
    %foutcoll ( tlabor.lis    );
    %foutcoll ( tpcount.lis   );
    %foutcoll ( tpcount2.lis ,&glasttno);  * continue with the same number;

*** Appendix B ********************************;
%foutinit ( appb_t );
    %foutcoll ( raw.lis, 1 );

*** Tables of contents ************************************************;

title "Table of contents for tables behind text";
    %toc ( tocappa, appa1t.out appa2l.out );

title "Table of contents for appendix B";
    %toc ( tocappb, appb_t.out );
```

Abbildung 7: Beispielprogramm RunPgms.sas - Dieses Programm wird als Template benutzt.