

Parametervalidierung mit regulären Ausdrücken - verpackt in ein SAS Einleseformat

Johannes Lang
HMS Analytical Software
Rohrbacher Str.26
69115 Heidelberg
johannes.lang@analytical-software.de

Zusammenfassung

Für die Formulierung von Gültigkeitsregeln für Parameterwerte bieten sich oftmals SAS Ausgabe- bzw. Einleseformate an. Bereits seit SAS 9.3 gibt es hierfür ein zusätzliches Feature: Die Angabe von regulären Ausdrücken innerhalb eines SAS Einleseformats. Die Idee dabei: Die Gültigkeitsregel wird als regulärer Ausdruck formuliert, und alle gültigen bzw. ungültigen Werte werden jeweils auf einen Wert (1 bzw. 0) abgebildet, der dann im Programm abfragbar ist. Dies ermöglicht die Erstellung eines wiederverwendbaren Frameworks zur Parameterprüfung, dessen Einsatz die Software-Qualität erhöht und den künftigen Implementierungsaufwand senkt.

Schlüsselwörter: SAS Einleseformat, reguläre Ausdrücke, Parametervalidierung

1 Motivation

Jeder SAS Programmierer¹ mit Makro-Kenntnissen kennt das Thema Parametervalidierung: Erstens unter dem Aspekt der Herausforderung, unterschiedliche Prüfarten für alphanumerische Werte zu implementieren (z.B. „Wert muss ganzzahlig numerisch sein“, „Wert darf keine Sonderzeichen enthalten“). Die zweite Herausforderung besteht darin, solche Prüfungen wiederverwendbar zu machen, ohne die Prüflogik immer wieder kopieren zu müssen. Für beide Herausforderungen bietet dieser Kurzbeitrag einen Lösungsansatz.

2 Vorgehen

2.1 Kapselung der Validierungsregel als Einleseformat

Regeln für die Parametervalidierung können in vielen Fällen als Geschäftsregeln verstanden werden, die vom Fachbereich vorgegeben werden. Dazu ist es hilfreich, wenn dieser die Regeln selbst ändern kann, ohne in den Programmcode eingreifen zu müssen.

¹ egal ob männlich oder weiblich ☺

Hierfür bieten sich oftmals SAS Ausgabe- bzw. Einleseformate an. Die Formatdefinitionen können außerhalb des Programms (z.B. in einer CSV-Datei) verwaltet und zur Laufzeit eingelesen werden (PROC FORMAT mit CNTLIN-Option).

Bis zur Version 9.2 mussten jedoch bei der Formatdefinition die gültigen Werte explizit aufgezählt werden, da keine regulären Ausdrücke als Gültigkeitsmuster unterstützt wurden. Dies ist seit SAS 9.3 möglich (vgl. [1]), und soll hier kurz an einem Beispiel gezeigt werden.

Die Idee dabei: Die Gültigkeitsregel wird als regulärer Ausdruck formuliert, und alle gültigen bzw. ungültigen Werte werden jeweils auf einen Wert (1 bzw. 0) abgebildet. Damit kann mit Hilfe einer einfachen Konvention (Erfolgreiche Prüfung liefert immer 1 zurück) ein allgemeiner Prüfungsablauf implementiert werden.

Als Beispiel werden Makroparameter verwendet, die RGB-Farbwerte enthalten, um das Layout von Berichten zu steuern. In der SAS-Notation besteht ein gültiger RGB-Farbwert aus den Zeichen ‚cx‘ gefolgt von einer sechsstelligen Zeichenkette, welche den Hexadezimalcode des Farbtons angibt (vgl. [2]). Dieser setzt sich zusammen aus den Ziffern 0-9, sowie den Buchstaben A-F, die in Groß- oder Kleinschreibung angegeben werden können.

Listing 1: Beispiel eines Einleseformats für die Validierung von RGB-Farbwerten

```
PROC FORMAT LIBRARY=Work;
  invalue validrgb
    '/^cx[0-9a-fA-F]{6}$/' (REGEXP) = 1
    OTHER = 0
;
RUN;
```

Listing 2: Beispielprogramm für die Anwendung des Einleseformaten zur Parameterprüfung:

```
%Macro checkValue (p_sValue = );
  %IF %SYSFUNC(inputn(&p_sValue., validrgb.)) EQ 1 %THEN %DO;
    %PUT Value &p_sValue. is valid;
  %END;
  %ELSE %DO;
    %PUT Value &p_sValue. is invalid;
  %END;
%Mend checkValue;

%checkValue (p_sValue = 0012345); /* Präfix fehlt */
%checkValue (p_sValue = cx000000); /* Gültiger Ausdruck */
%checkValue (p_sValue = cx0012345); /* Ausdruck zu lang */
```

Ausgabe im SAS-Log nach Absenden der Programme aus Listing 1 und 2:

```
Value 0012345 is invalid
Value cx000000 is valid
Value cx0012345 is invalid
```

2.2 Kapselung der Validierung über ein metadatengestütztes Framework

Im Folgenden wird ein metadatengestütztes Framework zur Parametervalidierung vorgestellt, das bei HMS Analytical Software entwickelt wurde und in vielen unserer SAS-Projekte zum Einsatz kommt².

Das Framework besteht aus SAS Makros, die anhand von Steuertabellen die Werte von bekannten Parametern prüfen und auch gleich die Fehlerberichterstattung durchführen. Die letztgenannte Funktion ist auch separat als Messaging-Framework nutzbar.

2.2.1 Vorbereitung der Parameter-Metadaten

Um das Framework nutzen zu können, müssen alle Parameter, deren Werte geprüft werden sollen, in einer Steuertabelle registriert werden. Aktuell werden drei Parametertypen unterstützt:

- NUM_RANGE: Numerische Werte in einem bestimmten Bereich
- ALPHA_LIST: Vordefinierte Werteliste
- FMT_VALUE: Gültige Werte werden durch ein Einlese- bzw. Ausgabeformat definiert

Je nachdem welcher Parametertyp eingetragen wird, müssen unterschiedliche Zusatzinformationen hinterlegt werden. Für das hier verwendete Beispiel wird der Typ FMT_VALUE genutzt, die Steuertabelle sieht dann aus wie in Tabelle 1.

Tabelle 1: Beispiel für Steuertabelle des HMS-Frameworks zur Parameterprüfung

Name	Value
PARAM_NAME_LICHTGRAU	g_sLichtGrau
PARAM_TYPE_LICHTGRAU	FMT_VALUE
PARAM_DEFAULT_LICHTGRAU	cxc0c0c0
PARAM_FMTNAME_LICHTGRAU	validrgb
PARAM_FMTFUNCTION_LICHTGRAU	INPUTN
PARAM_INVALIDVALUE_LICHTGRAU	0
PARAM_DESCR_LICHTGRAU	Lichtgrau für Dashboards
PARAM_MISSINGALLOWED_LICHTGRAU	0
PARAM_NAME_HELLGRAU	g_sHellGrau
PARAM_TYPE_HELLGRAU	FMT_VALUE

² Die Makro-Frameworks zur Parameterprüfung und zur Fehlerbehandlung wurden von meinem geschätzten Kollegen Andreas Bachert entwickelt.

PARAM_DEFAULT_HELLGRAU	cx969696
PARAM_FMTNAME_HELLGRAU	validrgb
PARAM_FMTFUNCTION_HELLGRAU	INPUTN
PARAM_INVALIDVALUE_HELLGRAU	0
PARAM_DESCR_HELLGRAU	Hellgrau für Dashboards
PARAM_MISSINGALLOWED_HELLGRAU	0

Anmerkungen zu Tabelle 1:

- Für jeden Parameter wird ein logischer Name definiert, der als Suffix bei den Schlüsseln der Steuertabelle verwendet wird (z.B. LICHTGRAU). Als Wert muss der Name der zu prüfenden Makrovariablen genannt werden (g_sLichtGrau).
- Der Name des Einlese- bzw. Ausgabeformats für die Validierung wird unter dem Schlüssel PARAM_FMTNAME_<ParamName> hinterlegt.
- Wie das Format angewendet werden soll, wird unter PARAM_FMTFUNCTION_<ParamName> angegeben.
- Der resultierende Wert nach Anwendung des Formats, der einen ungültigen Wert angibt, steht hinter PARAM_INVALIDVALUE_<ParamName>.
- Zusatzinformationen pro Parameter:
 - Defaultwert (PARAM_DEFAULT_<ParamName>)
 - Angabe, ob der Wert leer sein darf (PARAM_MISSINGALLOWED_<ParamName>)
 - Beschreibungstext, der z.B. in Fehlermeldungen angezeigt wird (PARAM_DESCR_<ParamName>)

2.2.2 Allgemeiner Aufruf des Prüfmakros

Die Prüfung aller bekannten Parameter wird durch ein allgemeines Makro %CheckAllParamsInList gekapselt. Dieses konsultiert anhand der übergebenen Variablenennamen die obige Steuertabelle (vgl. Tabelle 1) und wendet die dort hinterlegten Gültigkeitsprüfungen auf die aktuellen Parameterwerte an. Beim ersten erkannten ungültigen Wert wird eine formatierte Fehlermeldung ausgegeben, sowohl ins SAS-Log als auch in eine konfigurierbare Tabelle. Der Meldungstext ist ebenfalls über eine eigene Steuertabelle konfigurierbar.

Listing 3: Beispielprogramm zur Nutzung des Frameworks zur Parametervalidierung

```

/*
Parameter g_sLichtGrau, g_sHellGrau werden im produktiven Szenario
von außen gesetzt, z.B. über globale Makrovariablen.
*/
%LET g_sLichtGrau = cxc0c0c0; /* Gültiger Wert */
%LET g_sHellGrau = 0012345; /* Präfix fehlt */

```


Die Meldungstexte werden in einer sprachspezifischen Datei definiert und in permanente SAS-Formate kompiliert. Je nachdem welche Sprache festgelegt ist, werden dann die jeweils passenden SAS-Formate zur Laufzeit verwendet.

7 Fazit

SAS Einleseformate eignen sich sehr gut, um Validierungsregeln sauber von der sonstigen Programmlogik zu trennen. Da seit SAS 9.3 auch reguläre Ausdrücke in Einleseformaten unterstützt werden, können auch musterbasierte Prüfungen auf diese Weise ausgelagert werden, was die innere Software-Qualität erhöht.

Durch die Nutzung eines metadatengestützten Makro-Frameworks zur Parameterprüfung kann auch die Anwendung der Validierungsregeln einmalig implementiert und über ein allgemeines Prüfmakro aufgerufen werden. Hierfür hat HMS Analytical Software eine Beispielimplementierung erstellt, die in unterschiedlichen Projekten eingesetzt wird. Dadurch wird die Anzahl der Codezeilen pro Programm oft deutlich reduziert, und man kann sich besser auf die wesentliche Aufgabe des jeweiligen Programmes konzentrieren.

Die skizzierten SAS Makro-Frameworks können gerne auf Anfrage zur Verfügung gestellt werden (Stichwort „HMS-Toolbox für SAS BI Anwender“).

Literatur

- [1] Langston, Rick: Using the New Features in PROC FORMAT, in: Proceedings of SAS Global Forum 2012.
<https://support.sas.com/resources/papers/proceedings12/245-2012.pdf> [23.02.2017]
- [2] SAS Institute Inc.: SAS/GRAPH 9.4: Reference, Fifth Edition. SAS Institute Inc., Cary, NC, 2016.