



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement des Innern EDI

Bundesamt für Statistik BFS
Abteilung Gesundheit und Soziales

Silvio Hasler, 22. Januar 2016

Plausibilisierungskonzept der Medizinischen Statistik der Krankenhäuser

Version 5.0.8.1

Dieser Bericht ist ausschliesslich in elektronischer Form in Deutsch und Französisch verfügbar:
<http://www.bfs.admin.ch>

Autor:

*Silvio Hasler
Freudiger EDV-Beratung
Zeughausgasse 16
Postfach 6209
3001 Bern*

Inhaltsverzeichnis

A. Tests auf Einzelfeldebene	3
A.0. Lieferungsmeldung	3
A.1. Minimaldatensatz	4
A.2. Neugeborenen-Zusatzdaten	12
A.3. Psychiatrie-Zusatzdaten	16
A.4. Patientengruppen-Zusatzdaten	22
A.5. Kantonale Zusatzdaten	31
B. Kreuztests zwischen verschiedenen Feldern eines Datensatzes	32
B.0. Lieferungsmeldung	32
B.1. Minimaldatensatz	32
B.2. Neugeborenen-Zusatzdaten	36
B.3. Psychiatrie-Zusatzdaten	37
B.4. Patientengruppen-Zusatzdaten	37
B.5. Kantonale Zusatzdaten	38
C. Plausibilisierungen zwischen Datensätzen	39
C.0. Lieferungsmeldung und vorhandene Datensätze	39
C.1. Existenz von Zusatzdatensätzen	40
C.2. Neugeborenen-Zusatzdaten	41
C.3. Psychiatrie-Zusatzdaten	41
C.4. Patientengruppen-Zusatzdaten	42
D. Plausibilisierung der Diagnose- und Behandlungsfelder	43
D.0. Allgemeines	43
D.1. Tests auf den Diagnosefeldern (ICD-10)	43
D.2. Tests auf den Behandlungsfeldern (CHOP)	48
E. Indikatoren zur Überwachung der Datenqualität	50
E.0. Hilfsmerkmale	50
E.1. Indikatoren zu Patientenmerkmalen	50
E.2. Indikatoren zu Aufenthaltsmerkmalen	51
E.3. Indikatoren betreffend die medizinische Kodierung	52
Anhang A. ICD/CHOP Metadaten-Fileformat	55
Anhang B. Beschreibung der in den Tests verwendeten Funktionen	56
Anhang C. Implementation des Plausibilisierungskonzeptes in MedPlaus	60

A. Tests auf Einzelfeldebene

A.0. Lieferungsmeldung

Als erster Eintrag jeder Datenlieferung steht ein Lieferungs-Header MX mit den allgemeinen Angaben zu den übermittelten Datensätzen und mit dem Schlüssel C der Verbindungs_codes.

Die folgenden Tests werden auf den einzelnen Feldern der Lieferungsmeldung durchgeführt:

A0010 [E] BUR-Nummer: Angabe fehlt

```
strlen(V0002) == 0
```

A0020 [E] Kürzel der Klinik: Angabe fehlt

```
strlen(V0003) == 0
```

```
ACTION: V0003='Dummy';
```

A0021 [E] Kürzel der Klinik: Angabe darf max. 8 Zeichen lang sein

```
strlen(V0003) > 8
```

```
ACTION: V0003[8]=0;
```

A0030 [E] Export-Erstellungsdatum: Angabe fehlt

```
dateV0004.isNull
```

A0031 [E] Export-Erstellungsdatum: Ungültige Datumsangabe

```
!(dateV0004.isNull || dateV0004.isYYYYMMDD)
```

```
ACTION: if (strlen(V0004) > 8) V0004[8]=0;
```

A0040 [E] Laufnummer Datenmeldung: Angabe fehlt

```
strlen(V0005) == 0
```

```
ACTION: V0005='1';
```

A0041 [E] Laufnummer Datenmeldung: Ungültige Angabe

```
strlen(V0005) > 0 && ((!isInt(V0005) || atoi(V0005) < 1 || atoi(V0005) > 99999))
```

```
ACTION: V0005='1';
```

A0050 [E] Meldungsperiode von: Angabe fehlt

```
dateV0006.isNull
```

A0051 [E] Meldungsperiode von: Ungültige Datumsangabe

!(dateV0006.isNull || dateV0006.isYYYYMMDD)

ACTION: if (strlen(V0006) > 8) V0006[8]=0;

A0060 [E] Meldungsperiode bis: Angabe fehlt

dateV0007.isNull

A0061 [E] Meldungsperiode bis: Ungültige Datumsangabe

!(dateV0007.isNull || dateV0007.isYYYYMMDD)

ACTION: if (strlen(V0007) > 8) V0007[8]=0;

A0070 [E] Anzahl Datensätze: Angabe fehlt

strlen(V0008) == 0

A0071 [E] Anzahl Datensätze: Ungültige Angabe

strlen(V0008) != 0 && ((!isInt(V0008) || atoi(V0008) < 3 || atoi(V0008) > 9999999))

A0080 [E] C-Schlüssel Feld: Angabe fehlt

strlen(V0009) == 0

A0081 [E] C-Schlüssel Feld: Falscher Datentyp (nicht hexadezimal)

strlen(V0009) != 0 && !isHex(V0009)

A0082 [E] C-Schlüssel Feld: Länge nicht korrekt

strlen(V0009) > 0 && strlen(V0009) != 256

A.1. Minimaldatensatz

Die Minimaldatensätze werden mit Records vom Typ MB übermittelt. Als erster Datensatz steht der fiktive Rekord, mit dem die Korrektheit der Verbindungscodes überprüft werden kann (vgl. C.0.2).

Die folgenden Tests werden auf den einzelnen Feldern der Minimaldatensätze durchgeführt:

A1010 [E] BUR-Nummer: Angabe fehlt

strlen(V0102) == 0

A1020 [E] Standort: Angabe fehlt

strlen(V0103) == 0

A1022 [E] Standort: Angabe darf max. 2 Zeichen lang sein

strlen(V0103) > 2

A1023 [E] Standort: Ungültige Angabe

strlen(V0103) > 0 && !isIn (V0103,36,_NOGA)

ACTION: V0103='';

A1030 [E] Kanton: Angabe fehlt

strlen(V0104) == 0

A1031 [E] Kanton: Ungültige Angabe

strlen(V0104) > 0 && !isIn (V0104,26,_26_KANTON)

A1040 [E] Anonymer Verbindungscode: Angabe fehlt

strlen(V0201) == 0

A1041 [E] Anonymer Verbindungscode: Zeichenlänge nicht korrekt

strlen(V0201) > 0 && strlen(V0201) != 16

A1042 [E] Anonymer Verbindungscode: Angabe ist nicht hexadezimal

strlen(V0201) > 0 && !isHex(V0201)

A1043 [E] Anonymer Verbindungscode: Angabe ungültig, beruht auf fehlenden oder ungültigen Werten in Name, Vorname, Geburtsdatum (vollständig) oder Geschlecht vor dem Anonymisieren

strcmp(V0201,'0000000000000000') == 0

A1050 [E] Kennzeichnung Statistikfall: Angabe fehlt

strlen(V0202) == 0

A1051 [E] Kennzeichnung Statistikfall: Ungültige Angabe

strlen(V0202) > 0 && !isIn(V0202,3,_ABC)

A1060 [E] Neugeborenen-Zusatzdaten: Angabe fehlt

strlen(V0301) == 0

ACTION: V0301='0';

A1061 [E] Neugeborenen-Zusatzdaten: Ungültige Angabe

strlen(V0301) > 0 && !isIn(V0301,2,_01)

ACTION: V0301='0';

A1070 [E] Psychiatrie-Zusatzdaten: Angabe fehlt

strlen(V0302) == 0

ACTION: V0302='0';

A1071 [E] Psychiatrie-Zusatzdaten: Ungültige Angabe

`strlen(V0302) > 0 && !isIn(V0302,2,_01)`

`ACTION: V0302='0';`

A1080 [E] Patientengruppen-Zusatzdaten: Angabe fehlt

`strlen(V0303) == 0`

`ACTION: V0303='0';`

A1081 [E] Patientengruppen-Zusatzdaten: Ungültige Angabe

`strlen(V0303) > 0 && !isIn(V0303,2,_01)`

`ACTION: V0303='0';`

A1090 [E] Kantonale Zusatzdaten: Angabe fehlt

`strlen(V0304) == 0`

`ACTION: V0304='0';`

A1091 [E] Kantonale Zusatzdaten: Ungültige Angabe

`strlen(V0304) > 0 && !isIn(V0304,2,_01)`

`ACTION: V0304='0';`

A1100 [E] Geschlecht: Angabe fehlt

`strlen(V1101) == 0`

A1101 [E] Geschlecht: Ungültige Angabe

`strlen(V1101) > 0 && !isIn(V1101,2,_12)`

A1110 [E] Geburtsdatum: Angabe fehlt

`dateV1102.isNull`

A1111 [E] Geburtsdatum: Ungültige Angabe

`!dateV1102.isNull && !(dateV1102.isYYYYMMDD || dateV1102.isYYYY0000)`

A1112 [H] Geburtsdatum: Die Länge beträgt 8 Zeichen

`!dateV1102.isNull && strlen(V1102) != 8`

`ACTION: if (strlen(V1102) > 8) V1102[8]=0;`

A1120 [E] Alter bei Eintritt: Angabe fehlt

`strlen(V1103) == 0`

A1121 [E] Alter bei Eintritt: Ausserhalb des gültigen Wertebereichs

`strlen(V1103) > 0 && (atol(V1103) < 0 || atol(V1103) > 120)`

A1130 [E] Wohnort: Angabe fehlt

`strlen(V1104) == 0`

`ACTION: V1104='XXXX';`

A1131 [E] Wohnort: Ungültige Angabe

`!(V1104_regio.inIndex || V1104_plz.inIndex || V1104_wregio.inIndex || V1104_lregio.inIndex || V1104_natio.inIndex)`

A1132 [W] Wohnort: Bei Wohnort in der Schweiz Region (oder allenfalls PLZ) angeben

`strcmp(V1104,'CHE') == 0`

A1140 [E] Nationalität: Angabe fehlt

`strlen(V1105) == 0`

`ACTION: V1105='XXX';`

A1141 [E] Nationalität: Ungültige Angabe

`!V1105_natio.isNull && !(V1105_natio.inIndex || V1105_wregio.inIndex)`

A1150 [E] Eintrittsdatum: Angabe fehlt

`dateV1201.isNull`

A1151 [E] Eintrittsdatum: Ungültige Datumsangabe

`!dateV1201.isNull && !dateV1201.hasDate`

`ACTION: V1201='';`

A1152 [E] Eintrittsstunde: Ungültige Stundenangabe

`dateV1201.Length>8 && dateV1201.hasDate && !dateV1201.hasHour`

A1160 [E] Aufenthalt vor Eintritt: Angabe fehlt

`strlen(V1202) == 0`

`ACTION: V1202='9';`

A1161 [E] Aufenthalt vor Eintritt: Ungültige Angabe

`strlen(V1202) > 0 && !isIn(V1202,9,_9_AUFVOR)`

`ACTION: V1202='9';`

A1170 [E] Eintrittsart: Angabe fehlt

`strlen(V1203) == 0`

ACTION: V1203='9';

A1171 [E] Eintrittsart: Ungültige Angabe

strlen(V1203) > 0 && !isIn(V1203,7,_7_1234589)

ACTION: V1203='9';

A1180 [E] Einweisende Instanz: Angabe fehlt

strlen(V1204) == 0

ACTION: V1204='9';

A1181 [E] Einweisende Instanz: Ungültige Angabe

strlen(V1204) > 0 && !isIn(V1204,8,_8_INSTANZ)

ACTION: V1204='9';

A1190 [E] Behandlungsart: Angabe fehlt

strlen(V1301) == 0

ACTION: V1301='9';

A1191 [E] Behandlungsart: Ungültige Angabe

strlen(V1301) > 0 && !isIn(V1301,4,_4_BEHART)

ACTION: V1301='9';

A1192 [W] Behandlungsart: Unplausible Angabe

strcmp(V1301,'9') == 0

A1193 [E] Behandlungsart: teilstationär ist ab 2009 nicht mehr gültig

strlen(V1301) == 1 && V1301[0] == '2' && dateV1501.Year > 2008 && !(strcmp(V1401,'M500') == 0 && strcmp(V0104,'BE') == 0)

ACTION: doWriteRec=FALSE;

A1200 [E] Klasse: Angabe fehlt

strlen(V1302) == 0

ACTION: V1302='9';

A1201 [E] Klasse: Ungültige Angabe

strlen(V1302) > 0 && !isIn(V1302,4,_4_KLASSE)

ACTION: V1302='9';

A1202 [H] Klasse: Unplausible Angabe

`strcmp(V1301,'3') == 0 && strcmp(V1302,'9') == 0`

A1210 [W] Aufenthalt in Intensivmedizin: Angabe fehlt

`strcmp(MB->V0202,'A') == 0 && strlen(V1303) == 0`

A1211 [E] Aufenthalt in Intensivmedizin: Ungültige Angabe

`strlen(V1303) > 0 && (atol(V1303) < 0 || atol(V1303) > 9999)`

`ACTION: V1303='9999';`

A1220 [W] Administrativer Urlaub, Ferien: Angabe fehlt

`strlen(V1304) == 0`

A1221 [E] Administrativer Urlaub, Ferien: Ungültige Angabe

`strlen(V1304) > 0 && (atol(V1304) < 0 || atol(V1304) > 9999)`

`ACTION: V1304="";`

A1230 [E] Hauptkostenstelle: Angabe fehlt

`strlen(V1401) == 0`

A1231 [E] Hauptkostenstelle: Ungültige Angabe

`strlen(V1401) > 0 && !((StatYear >= 2016 && isln(V1401,16,_16_KST)) || (StatYear < 2016 && isln(V1401,14,_14_KST)))`

A1240 [E] Hauptkostenträger: Angabe fehlt

`strlen(V1402) == 0`

`ACTION: V1402='9';`

A1241 [E] Hauptkostenträger: Ungültige Angabe

`strlen(V1402) > 0 && !isln(V1402,7,_1234589)`

`ACTION: V1402='9';`

A1250 [E] Austrittsdatum: Angabe fehlt (Statistikfall A)

`strlen(V1501) == 0 && V0202[0] == 'A'`

A1251 [E] Austrittsdatum: Ungültige Datumsangabe

`!dateV1501.isNull && !dateV1501.hasDate`

A1252 [E] Austrittsstunde: Ungültige Stundenangabe

`dateV1501.Length > 8 && dateV1501.hasDate && !dateV1501.hasHour`

A1260 [E] Entscheid für Austritt: Angabe fehlt

`strlen(V1502) == 0 && strcmp(V0202,'A') == 0`

`ACTION: V1502='9';`

A1261 [E] Entscheid für Austritt: Ungültige Angabe

`strlen(V1502) > 0 && !isIn(V1502,7,_1234589)`

`ACTION: V1502='9';`

A1270 [E] Aufenthalt nach Austritt: Angabe fehlt

`strlen(V1503) == 0 && V0202[0] == 'A'`

`ACTION: V1503='9';`

A1271 [E] Aufenthalt nach Austritt: Ungültige Angabe

`strlen(V1503) > 0 && !isIn(V1503,10,_0123456789)`

`ACTION: V1503='9';`

A1280 [E] Behandlung nach Austritt: Angabe fehlt

`strlen(V1504) == 0 && strcmp(V0202,'A') == 0`

`ACTION: V1504='9';`

A1281 [E] Behandlung nach Austritt: Ungültige Angabe

`strlen(V1504) > 0 && !isIn(V1504,8,_01234589)`

`ACTION: V1504='9';`

A1290 [E] Hauptdiagnose: Angabe fehlt (Statistikfall A, C)

`DIA[0].KeyLen == 0 && isIn(MB->V0202,2,_AC) && !(MB->hasMN && strcmp(MN->V2201,'0') == 0)`

A1390 [W] Hauptbehandlung (Chirurgie): Angabe fehlt

`BEH[0].isNull && isIn(MB->V0202,2,_AC) && strcmp(MB->V1401,'M200') == 0 && strncmp(DIA[0].Key,'Z38',3) != 0 && strncmp(DIA[0].Key,'Z758',4) != 0`

A1391 [W] Hauptbehandlung: Angabe fehlt

`BEH[0].isNull && isIn(MB->V0202,2,_AC) && !isIn(MB->V1401,5,_5_HKSTH) && strncmp(DIA[0].Key,'Z38',3) != 0 && strncmp(DIA[0].Key,'Z758',4) != 0`

A1392 [H] Hauptbehandlung: Angabe fehlt

`BEH[0].isNull && isIn(MB->V0202,2,_AC) && !isIn(MB->V1401,5,_5_HKSTH) && strcmp(MB->V1401,'M200') != 0 && strncmp(DIA[0].Key,'Z38',3) != 0 && strncmp(DIA[0].Key,'Z758',4) != 0`

A1400 [W] Beginn der Hauptbehandlung (Chirurgie): Angabe fehlt

`dateV1702.isNull && isln(MB->V0202,2,_AC) && strcmp(MB->V1401,'M200') == 0`

A1401 [W] Beginn der Hauptbehandlung: Angabe fehlt

`dateV1702.isNull && isln(V0202,2,_AC) && !isln(V1401,5,_5_HKSTH)`

A1402 [E] Beginn der Hauptbehandlung: Ungültige Datumsangabe

`!dateV1702.isNull && !dateV1702.hasDate`

A1403 [E] MB: Beginn der Hauptbehandlung: Ungültige Stundenangabe

`dateV1702.hasDate && !dateV1702.hasHour`

A1404 [H] MB: Beginn der Hauptbehandlung: Angabe fehlt

`dateV1702.isNull && isln(V0202,2,_AC) && isln(V1401,5,_5_HKSTH) && strcmp(MB->V1401,'M200') != 0`

A.2. Neugeborenen-Zusatzdaten

Der Neugeborenen Datensatz folgt bei der Übermittlung unmittelbar dem Minimaldatensatz, der für das betreffende Neugeborene erstellt wurde.

Dort wird er signalisiert mit der Angabe Neugeborenen-Datensatz = 1 (0.3.V01). Keinesfalls wird ein Neugeborenen Datensatz mit dem Datensatz der Mutter übermittelt, sondern für jedes Neugeborene wird ein normaler Minimaldatensatz erstellt (bei Mehrlingen entsprechend mehrere).

Die folgende Tabelle beschreibt den Inhalt des Minimaldatensatzes des Neugeborenen. Bestimmte Variablen werden von den Angaben der Mutter übernommen. Sie sind in der nachstehenden Tabelle grau markiert.

Variable	Bezeichnung	Angabe im Minimaldatensatz des Kindes
0.1.V01	Rekordart	MB
0.1.V02	BUR-Nummer (BUR-Satellit Gesundheit)	BUR-Nummer der Institution
0.1.V03	Standort	Standort der Institution
0.1.V04	Kanton	Kanton
0.2.V01	Anonymer Verbindungskode	Verbindungskode des Kindes
0.2.V02	Kennzeichnung des Statistikfalls	Betreffend Aufenthaltsdaten des Kindes
0.3.V01	Neugeborenen-Datensatz	Mit Zusatzdaten = 1
0.3.V02	Psychiatrie-Datensatz	Andere Zusatzdaten möglich
0.3.V03	Patientengruppen-Datensatz	Dito
0.3.V04	Kantonaler Datensatz	Dito
1.1.V01	Geschlecht	Geschlecht des Kindes
1.1.V02	Geburtsdatum (JJJJMMTT)	Vollständiges Geburtsdatum des Kindes
1.1.V03	Alter bei Eintritt	Alter = 0
1.1.V04	Wohnort (Region)	Übernommen aus Minimaldatensatz der Mutter
1.1.V05	Nationalität	Übernommen aus Minimaldatensatz der Mutter
1.2.V01	Eintrittsdatum und -stunde	Geburtsdatum und -stunde des Kindes
1.2.V02	Aufenthaltsort vor dem Eintritt	Übernommen aus Minimaldatensatz der Mutter
1.2.V03	Eintrittsart	Eintrittsart = 3 (Geburt)
1.2.V04	Einweisende Instanz	Übernommen aus Minimaldatensatz der Mutter
1.3.V01	Behandlungsart	Behandlungsart des Kindes
1.3.V02	Klasse	Klasse des Kindes
1.3.V03	Aufenthalt in Intensivmedizin	Aufenthalt des Kindes in Intensiv-Neonatalogie
1.3.V04	Administrativer Urlaub & Ferien	Angabe betreffend das Kind
1.4.V01	Hauptkostenstelle	Angabe betreffend das Kind
1.4.V02	Hauptkostenträger für Grundversicherung	Angabe betreffend das Kind
1.5.V01	Austrittsdatum und -stunde	Angabe betreffend das Kind
1.5.V02	Entscheid für Austritt	Angabe betreffend das Kind
1.5.V03	Aufenthalt nach Austritt	Angabe betreffend das Kind
1.5.V04	Behandlung nach Austritt	Angabe betreffend das Kind
1.6.V01	Hauptdiagnose	Hauptdiagnose betreffend das Kind
1.6.V02	Zusatz zu Hauptdiagnose	Weitere Angaben möglich
1.6.V0..	1.-8. Nebendiagnose	Weitere Angaben möglich
1.7.V01	Hauptbehandlung	Weitere Angaben möglich
1.7.V02	Beginn der Hauptbehandlung	Zeitpunkt angeben
1.7.V0..	1.-9. weitere Behandlung	Weitere Angaben möglich

Die folgenden Tests werden auf den einzelnen Feldern der Neugeborenen-Zusatzdaten durchgeführt:

A2010 [H] Interne Geburtsnummer: Angabe fehlt

```
strlen(V2102) == 0
```

```
ACTION: V2102='0';
```

A2011 [E] Interne Geburtsnummer: Ausserhalb des gültigen Wertebereichs

```
strlen(V2102) > 0 && atol(V2102) > 9999
```

ACTION: V2102='0';

A2020 [E] Geburtszeitpunkt: Angabe fehlt

`strlen(V2103) == 0`

A2021 [E] Geburtszeitpunkt: Ungültige Stundenangabe

`dateV2103.Length>8 && dateV2103.hasDate && !dateV2103.hasHour`

A2022 [E] Geburtszeitpunkt: Ungültige Minutenangabe

`dateV2103.Length>10 && dateV2103.hasDate && dateV2103.hasHour && !dateV2103.hasMinute`

A2030 [E] Vitalstatus: Angabe fehlt

`strlen(V2201) == 0`

A2031 [E] Vitalstatus: Ungültige Angabe

`strlen(V2201) > 0 && !isIn(V2201,2,_01)`

A2040 [E] Mehrling: Angabe fehlt

`strlen(V2202) == 0`

A2041 [E] Mehrling: Ungültige Angabe

`strlen(V2202) > 0 && !isIn(V2202,9,_9_MEHRLING)`

A2050 [E] Geburtsrang bei Mehrlingsgeburten: Angabe fehlt oder ist nicht korrekt für Mehrlingsschwangerschaft

`atol(V2202) > 1 && (strlen(V2203) == 0 || atol(V2203) > atol(V2202))`

A2051 [H] Bei einem Einling sollte der Geburtsrang 1 angegeben werden

`strcmp(V2202,'1') == 0 && strlen(V2203) > 0 && strcmp(V2203,'1') != 0`

ACTION: V2203='1';

A2052 [H] Geburtsrang bei Mehrlingsgeburten: Ungültige Angabe

`strlen(V2203) > 0 && !isIn(V2203,9,_9_GEBRANG)`

ACTION: V2203='1';

A2060 [E] Geburtsgewicht (gr): Angabe fehlt

`strlen(V2204) == 0`

A2061 [E] Geburtsgewicht (gr): Ausserhalb des gültigen Wertebereichs

`strlen(V2204) > 0 && (atol(V2204) < 0 || atol(V2204) > 7000)`

A2062 [W] Geburtsgewicht (gr): Unplausible Angabe für ein Lebendgeborenes

`strlen(V2204) > 0 && strcmp(V2201,'1') == 0 && atol(V2204) <= 500`

A2063 [W] Geburtsgewicht (gr)/Gestationsalter 1: Unplausible Angabe für ein Totgeborenes

`strlen(V2204) > 0 && strcmp(V2201,'0') == 0 && (((atol(V2302) <= 217) && (atol(V2204) > 500)) || ((atol(V2302) > 217) && (atol(V2204) <= 500)))`

A2064 [W] Geburtsgewicht (gr): Unplausible Angabe

`strlen(V2204) > 0 && atol(V2204) <= 7000 && atol(V2204) >= 6000`

A2070 [E] Körperlänge (cm): Angabe fehlt

`strlen(V2205) == 0`

A2071 [E] Körperlänge (cm): Ausserhalb des gültigen Wertebereichs

`strlen(V2205) > 0 && strcmp(V2201,'1') == 0 && (strcmp(V2205,'0') == 0 || atol(V2205) > 65)`

A2072 [W] Körperlänge (cm): Unwahrscheinlicher Wertebereich

`strlen(V2205) > 0 && strcmp(V2201,'1') == 0 && (atol(V2205) < 25 && atol(V2205) > 0)`

A2080 [E] Kongenitale Missbildungen: Angabe fehlt

`strlen(V2206) == 0`

A2081 [E] Kongenitale Missbildungen: Ungültige Angabe

`strlen(V2206) > 0 && !isIn(V2206,3,_3_KONGEN_MISSB)`

A2090 [E] Verlegung Kind in anderes Spital: Angabe fehlt

`strlen(V2207) == 0`

A2091 [E] Verlegung Kind in anderes Spital: Ungültige Angabe

`strlen(V2207) > 0 && !isIn(V2207,2,_01)`

A2100 [E] Geburtsdatum der Mutter: Angabe fehlt

`strlen(V2301) == 0`

A2101 [E] Geburtsdatum der Mutter: Ungültige Angabe für die Jahrzahl

`!dateV2301.isNull && (!dateV2301.hasYear || dateV2301.Year < 1900)`

A2102 [E] Geburtsdatum der Mutter: Fehlende/falsche Monatsangabe

`!dateV2301.isNull && !dateV2301.hasMonth`

A2104 [E] Geburtsdatum der Mutter: Ungültige Datumsangabe (JJJJMMTT)

`strlen(V2301) > 6 && dateV2301.hasMonth && (strlen(V2301) != 8 || (dateV2301.Day > 0 && !dateV2301.isYYYYMMDD))`

A2105 [E] Das Geburtsdatum der Mutter darf nicht dem Geburtsdatum des Kindes entsprechen

`strlen(V2301) > 0 && strcmp(V2301,MB->V1102,6) == 0`

A2110 [W] Gestationsalter 1: Angabe fehlt

`strlen(V2302) == 0`

A2111 [W] Gestationsalter 1: Unwahrscheinliche Anzahl Wochen

`strlen(V2302) > 0 && ((atol(V2302)/10) < 16 || (atol(V2302)/10) > 44)`

A2112 [E] Gestationsalter 1: Ungültige Anzahl Tage

`strlen(V2302) > 0 && (strlen(V2302) != 3 || strchr('0123456',V2302[2]) == NULL)`

A2113 [E] Gestationsalter 1: Ungültige Anzahl Wochen

`strlen(V2302) > 0 && ((atol(V2302)/10) < 16 || (atol(V2302)/10) > 44)`

A2120 [W] Gestationsalter 2: Angabe fehlt

`strlen(V2303) == 0`

A2121 [W] Gestationsalter 2: Unwahrscheinliche Anzahl Wochen

`strlen(V2303) > 0 && ((atol(V2303) / 10) < 16 || (atol(V2303) / 10) > 45)`

A2122 [E] Gestationsalter 2: Ungültige Anzahl Tage

`strlen(V2303) > 0 && (strlen(V2303) != 3 || strchr('0123456',V2303[2]) == NULL)`

A2123 [E] Gestationsalter 2: Ungültige Anzahl Wochen

`strlen(V2303) > 0 && ((atol(V2303) / 10) < 16 || (atol(V2303) / 10) > 45)`

A2124 [W] Gestationsalter 2 ist mehr als 4 Wochen grösser als Gestationsalter 1

`strlen(V2302) > 0 && strlen(V2303) > 0 && ((atol(V2303) / 10) - (atol(V2302) / 10) > 4)`

A2130 [H] Anzahl vorausg. Schwangerschaften insgesamt: Angabe fehlt

`strlen(V2304) == 0`

A2131 [E] Anzahl vorausg. Schwangerschaften insgesamt: Ausserhalb des gültigen Wertebereichs

`strlen(V2304) > 0 && (atol(V2304) < 0 || atol(V2304) > 40)`

A2140 [H] Anzahl vorausg. Lebendgeburten: Angabe fehlt

`strlen(V2305) == 0`

A2141 [E] Anzahl vorausg. Lebendgeburten: Ausserhalb des gültigen Wertebereichs

`strlen(V2305) > 0 && (atol(V2305) < 0 || atol(V2305) > 20)`

A2150 [H] Anzahl vorausg. Fehl- oder Totgeburten: Angabe fehlt

`strlen(V2306) == 0`

A2151 [E] Anzahl vorausg. Fehl- oder Totgeburten: Ausserhalb des gültigen Wertebereichs

`strlen(V2306) > 0 && (atol(V2306) < 0 || atol(V2306) > 20)`

A2160 [H] Anzahl vorausg. Schwangerschaftsabbrüche: Angabe fehlt

`strlen(V2307) == 0`

A2161 [E] Anzahl vorausg. Schwangerschaftsabbrüche: Ausserhalb des gültigen Wertebereichs

`strlen(V2307) > 0 && (atol(V2307) < 0 || atol(V2307) > 20)`

A2170 [E] Verlegung Mutter aus anderem Spital: Angabe fehlt

`strlen(V2308) == 0`

A2171 [E] Verlegung Mutter aus anderem Spital: Ungültige Angabe

`strlen(V2308) > 0 && !isIn(V2308,2,_01)`

ACTION: V2308="";

A.3. Psychiatrie-Zusatzdaten

Die Schweizerische Gesellschaft für Psychiatrie (SGP) und die Schweizerische Vereinigung Psychiatrischer Chefärzte (SVPC) hat einen gesamtschweizerischen Fragebogen entwickelt, der mit der Medizinischen Statistik erhoben werden soll. Es handelt sich dabei um das sogenannte Grundmodul Psychiatrie. Dieses ist weitgehend mit dem Minimaldatensatz identisch, mit Ausnahme von einigen Merkmalen, die zusätzlich bzw. in höherem Detailgrad erhoben werden und welche mit dem Zusatzdatensatz Psychiatrie übermittelt werden.

Mit den Daten von Minimaldatensatz und Psychiatrie-Zusatzdaten sind alle Merkmale des Grundmoduls Psychiatrie abgedeckt. Die Erhebung und Übermittlung erfolgt auf den Kanälen der Medizinischen Statistik.

Der Psychiatrie-Zusatzdatensatz folgt bei der Übermittlung der Daten unmittelbar dem Minimaldatensatz, auf den er sich bezieht. Dort wird dies signalisiert mit der Angabe Psychiatrie-Datensatz = 1 (0.3.V02). Einzelne Merkmale sind detaillierter abgebildet als im Minimaldatensatz.

Die folgenden Tests werden auf den einzelnen Feldern der Psychiatrie-Zusatzdaten durchgeführt:

A3010 [W] Zivilstand: Angabe fehlt

`strlen(V3201) == 0`

ACTION: V3201='9';

A3011 [E] Zivilstand: Ungültige Angabe

`strlen(V3201) > 0 && !isIn(V3201,6,_6_ZIVILSTAND)`

ACTION: V3201='9';

A3020 [W] Aufenthaltsort vor Eintritt (Psychiatrie): Angabe fehlt

`strlen(V3202) == 0`

`ACTION: V3202='90';`

A3021 [E] Aufenthaltsort vor Eintritt (Psychiatrie): Ungültige Angabe

`strlen(V3202) > 0 && !isln(V3202,22,_22_AUFEIN_PSY)`

`ACTION: V3202='90';`

A3022 [W] Aufenthaltsort vor Eintritt (Psychiatrie): Detaillierte Angabe erforderlich (2-stellig)

`strlen(V3202) > 0 && isln(V3202,9,_9_AUFEIN)`

A3030 [W] Beschäftigung vor Eintritt (teilzeit erwerbstätig): Angabe fehlt

`strlen(V3203) == 0`

A3031 [E] Beschäftigung vor Eintritt (teilzeit erwerbstätig): Ungültige Angabe

`strlen(V3203) > 0 && !isln(V3203,2,_01)`

A3040 [W] Beschäftigung vor Eintritt (voll erwerbstätig): Angabe fehlt

`strlen(V3204) == 0`

A3041 [E] Beschäftigung vor Eintritt (voll erwerbstätig): Ungültige Angabe

`strlen(V3204) > 0 && !isln(V3204,2,_01)`

A3050 [W] Beschäftigung vor Eintritt (nicht erwerbstätig oder arbeitslos): Angabe fehlt

`strlen(V3205) == 0`

A3051 [E] Beschäftigung vor Eintritt (nicht erwerbstätig oder arbeitslos): Ungültige Angabe

`strlen(V3205) > 0 && !isln(V3205,2,_01)`

A3060 [W] Beschäftigung vor Eintritt (Hausarbeit in eigenem Haushalt): Angabe fehlt

`strlen(V3206) == 0`

A3061 [E] Beschäftigung vor Eintritt (Hausarbeit in eigenem Haushalt): Ungültige Angabe

`strlen(V3206) > 0 && !isln(V3206,2,_01)`

A3070 [W] Beschäftigung vor Eintritt (in Ausbildung): Angabe fehlt

`strlen(V3207) == 0`

A3071 [E] Beschäftigung vor Eintritt (in Ausbildung): Ungültige Angabe

`strlen(V3207) > 0 && !isln(V3207,2,_01)`

A3080 [W] Beschäftigung vor Eintritt (Rehabilitationsprogramm): Angabe fehlt

`strlen(V3208) == 0`

A3081 [E] Beschäftigung vor Eintritt (Rehabilitationsprogramm): Ungültige Angabe

`strlen(V3208) > 0 && !isIn(V3208,2,_01)`

A3090 [W] Beschäftigung vor Eintritt (IV, AHV oder andere Rente): Angabe fehlt

`strlen(V3209) == 0`

A3091 [E] Beschäftigung vor Eintritt (IV, AHV oder andere Rente): Ungültige Angabe

`strlen(V3209) > 0 && !isIn(V3209,2,_01)`

A3100 [W] Beschäftigung vor Eintritt (Arbeit in geschütztem oder beschützendem Rahmen): Angabe fehlt

`strlen(V3210) == 0`

A3101 [E] Beschäftigung vor Eintritt (Arbeit in geschütztem oder beschützendem Rahmen): Ungültige Angabe

`strlen(V3210) > 0 && !isIn(V3210,2,_01)`

A3110 [W] Beschäftigung vor Eintritt (unbekannt): Angabe fehlt

`strlen(V3211) == 0`

A3111 [E] Beschäftigung vor Eintritt (unbekannt): Ungültige Angabe

`strlen(V3211) > 0 && !isIn(V3211,2,_01)`

A3120 [W] Höchste abgeschlossene Schul- oder Berufsbildung: Angabe fehlt

`strlen(V3212) == 0`

A3121 [E] Höchste abgeschlossene Schul- oder Berufsbildung: Ungültige Angabe

`strlen(V3212) > 0 && !isIn(V3212,7,_7_AUSBILD_PSY)`

A3130 [W] Einweisende Instanz (Psychiatrie): Angabe fehlt

`strlen(V3301) == 0`

`ACTION: V3301='90';`

A3131 [E] Einweisende Instanz (Psychiatrie): Ungültige Angabe

`strlen(V3301) > 0 && !isIn(V3301,20,_20_EININST_PSY)`

`ACTION: V3301='90';`

A3132 [W] Einweisende Instanz (Psychiatrie): Detaillierte Angabe erforderlich (2-stellig)

`strlen(V3301) > 0 && !isln(V3301,8,_8_EININST)`

`ACTION: V3302='9';`

A3140 [W] Freiwilligkeit: Angabe fehlt

`strlen(V3302) == 0`

`ACTION: V3302='9';`

A3141 [E] Freiwilligkeit: Ungültige Angabe

`strlen(V3302) > 0 && !isln(V3302,3,_3_FREIWILLIGKEIT)`

A3150 [W] Fürsorgerischer Freiheitsentzug: Angabe fehlt

`strlen(V3303) == 0`

A3151 [E] Fürsorgerischer Freiheitsentzug: Ungültige Angabe

`strlen(V3303) > 0 && !isln(V3303,2,_12)`

A3160 [W] Anzahl Tage / Konsultationen: Angabe fehlt

`strlen(V3401) == 0`

A3161 [E] Anzahl Tage / Konsultationen: Ungültige Angabe

`strlen(V3401) > 0 && (atol(V3401) < 0 || atol(V3401) > 9999)`

A3170 [W] Behandlung (Psychiatrie): Angabe fehlt

`strcmp(MB->V0202,'A') == 0 && strlen(V3402) == 0`

A3171 [E] Behandlung (Psychiatrie): Ungültige Angabe

`strlen(V3402) > 0 && !isln(V3402,12,_12_BEHPHY)`

A3180 [W] Psychopharmakotherapie: Neuroleptika: Angabe fehlt

`strlen(V3403) == 0`

A3181 [E] Psychopharmakotherapie: Neuroleptika: Ungültige Angabe

`strlen(V3403) > 0 && !isln(V3403,2,_01)`

A3190 [W] Psychopharmakotherapie: Depotneuroleptika: Angabe fehlt

`strlen(V3404) == 0`

A3191 [E] Psychopharmakotherapie: Depotneuroleptika: Ungültige Angabe

`strlen(V3404) > 0 && !isln(V3404,2,_01)`

A3200 [W] Psychopharmakotherapie: Antidepressiva: Angabe fehlt

strlen(V3405) == 0

A3201 [E] Psychopharmakotherapie: Antidepressiva: Ungültige Angabe

strlen(V3405) > 0 && !isIn(V3405,2,_01)

A3210 [W] Psychopharmakotherapie: Tranquilizer: Angabe fehlt

strlen(V3406) == 0

A3211 [E] Psychopharmakotherapie: Tranquilizer: Ungültige Angabe

strlen(V3406) > 0 && !isIn(V3406,2,_01)

A3220 [W] Psychopharmakotherapie: Tranquilizer: Angabe fehlt

strlen(V3407) == 0

A3221 [E] Psychopharmakotherapie: Tranquilizer: Ungültige Angabe

strlen(V3407) > 0 && !isIn(V3407,2,_01)

A3230 [W] Psychopharmakotherapie: Hypnotika: Angabe fehlt

strlen(V3408) == 0

A3231 [E] Psychopharmakotherapie: Hypnotika: Ungültige Angabe

strlen(V3408) > 0 && !isIn(V3408,2,_01)

A3240 [W] Psychopharmakotherapie: Antiepileptika: Angabe fehlt

strlen(V3409) == 0

A3241 [E] Psychopharmakotherapie: Antiepileptika: Ungültige Angabe

strlen(V3409) > 0 && !isIn(V3409,2,_01)

A3250 [W] Psychopharmakotherapie: Lithium: Angabe fehlt

strlen(V3410) == 0

A3251 [E] Psychopharmakotherapie: Lithium: Ungültige Angabe

strlen(V3410) > 0 && !isIn(V3410,2,_01)

A3260 [W] Psychopharmakotherapie: Suchtaversionsmittel: Angabe fehlt

strlen(V3411) == 0

A3261 [E] Psychopharmakotherapie: Suchtaversionsmittel: Ungültige Angabe

strlen(V3411) > 0 && !isIn(V3411,2,_01)

A3270 [W] Psychopharmakotherapie: Antiparkinsonmittel: Angabe fehlt

strlen(V3412) == 0

A3271 [E] Psychopharmakotherapie: Antiparkinsonmittel: Ungültige Angabe

strlen(V3412) > 0 && !isIn(V3412,2,_01)

A3280 [W] Psychopharmakotherapie: andere: Angabe fehlt

strlen(V3413) == 0

A3281 [E] Psychopharmakotherapie: andere: Ungültige Angabe

strlen(V3413) > 0 && !isIn(V3413,2,_01)

A3290 [W] Psychopharmakotherapie: Med. z. Beh. körperl. Leiden: Angabe fehlt

strlen(V3414) == 0

A3291 [E] Psychopharmakotherapie: Med. z. Beh. körperl. Leiden: Ungültige Angabe

strlen(V3414) > 0 && !isIn(V3414,2,_01)

A3300 [W] Entscheid für Austritt (Psychiatrie): Angabe fehlt

strlen(V3501) == 0 && strcmp(MB->V0202,'A') == 0

ACTION: V3501='90';

A3301 [E] Entscheid für Austritt (Psychiatrie): Ungültige Angabe

strlen(V3501) > 0 && !isIn(V3501,9,_9_AUSENT_PSY)

ACTION: V3501='90';

A3302 [W] Entscheid für Austritt (Psychiatrie): Detaillierte Angabe erforderlich (2-stellig)

strlen(V3501) > 0 && !isIn(V3501,7,_7_AUSENT)

A3310 [W] Aufenthalt nach Austritt (Psychiatrie): Angabe fehlt

strlen(V3502) == 0 && strcmp(MB->V0202,'A') == 0

ACTION: V3502='90';

A3311 [E] Aufenthalt nach Austritt (Psychiatrie): Angabe ungültig

strlen(V3502) > 0 && !isIn(V3502,13,_13_AUSAUF_PSY)

ACTION: V3502='90';

A3312 [W] Aufenthalt nach Austritt (Psychiatrie): Detaillierte Angabe erforderlich (2-stellig)

strlen(V3502) > 0 && !isIn(V3502,10,_10_AUSAUF)

A3320 [W] Behandlung nach Austritt (Psychiatrie): Angabe fehlt

```
strlen(V3503) == 0 && strcmp(MB->V0202,'A') == 0
```

```
ACTION: V3503='90';
```

A3321 [E] Behandlung nach Austritt (Psychiatrie): Ungültige Angabe

```
strlen(V3503) > 0 && !isIn(V3503,20,_20_AUSBEH_PSY)
```

```
ACTION: V3503='90';
```

A3322 [W] Behandlung nach Austritt (Psychiatrie): Detaillierte Angabe erforderlich (2-stellig)

```
strlen(V3503) > 0 && isIn(V3503,8,_8_AUSBEH)
```

A.4. Patientengruppen-Zusatzdaten

Die Medizinische Statistik wurde ab 1.1.2009 angepasst an die Bedürfnisse einer leistungsorientierten Spitalfinanzierung. Um in der Schweiz die notwendige Kompatibilität zum gewählten deutschen Modell zu erreichen, müssen detailliertere Informationen als bisher erhoben werden. In dem neu gebildeten Patientengruppendatensatz können bis zu 50 Diagnosen und bis 100 Behandlungen erhoben werden. Zudem erhält er weitere für Abrechnungszwecke relevante Daten.

A4100 [E] Lokalisation des Spitals: Angabe fehlt

```
strlen(V4102) == 0
```

A4101 [W] Lokalisation des Spitals: ungültige Angabe (gültige PLZ erforderlich)

```
!V4102_plz.isNull && !V4102_plz.inIndex
```

A4200 [E] Das Reservefeld zur Lokalisation des Spitals ist noch nicht definiert. Es muss leer sein.

```
strlen(V4103) > 0
```

A4210 [W] Bei einer Diagnose mit potentieller Tumoraktivität muss letztere erfasst werden

```
i != 1 && strncmp(DIA[i].Key, 'C00',3) > 0 && strncmp(DIA[i].Key, 'D48',3) < 0 && strncmp(DIA[i].Key, 'C97',3) != 0  
&& strlen(DIA[i].TumorAktivitaet) == 0
```

A4361 [W] Im MB-Datensatz sind IPS-Stunden vorhanden. Ein Teil der Felder MD-Intensivmedizin muss ausgefüllt sein

```
strlen(MB->V1303) > 0 && atol(MB->V1303) != 0 && (strlen(V4401) == 0 || strlen(V4402) == 0)
```

A4370 [E] Die Dauer der künstlichen Beatmung darf nicht kleiner als 0 oder grösser als 99999 Stunden sein

```
strlen(V4401) > 0 && (atol(V4401) < 0 || atol(V4401) > 99999)
```

A4380 [H] Die Dauer der künstlichen Beatmung darf nicht grösser als diejenige der IPS-Stunden sein

```
strcmp(MB->V0202,'A') == 0 && strlen(V4401) > 0 && (atol(V4401) > (atol(MB->V1303)+15)) && atol(MB->V1303) > 0
```

A4390 [E] Die Dauer der künstlichen Beatmung darf nicht grösser als die Aufenthaltsdauer sein

`strcmp(MB->V0202,'A') == 0 && strlen(V4401) > 0 && atol(V4401) > (MB->STAYH+15)`

A4400 [E] Der Schweregrad der akuten Erkrankung darf nicht kleiner als 0 oder grösser als 999 Punkte sein

`strlen(V4402) > 0 && (atol(V4402) < 0 || atol(V4402) > 999)`

A4401 [E] Behandlungsdatum: Angabe fehlt

`!BEH[i].isNull && BEH[i].dateDatum.isNull`

A4402 [E] Behandlungsdatum: Ungültige Datumsangabe

`!BEH[i].dateDatum.isNull && !BEH[i].dateDatum.hasDate`

A4403 [E] Behandlungsdatum: Ungültige Stundenangabe

`BEH[i].dateDatum.hasDate && !BEH[i].dateDatum.hasHour && (BEH[i].dateDatum.Length > 8 || i == 0)`

A4404 [W] Jede Hauptbehandlung benötigt eine Stundenangabe

`!BEH[0].isNull && BEH[0].dateDatum.hasDate && !BEH[0].dateDatum.hasHour`

A4405 [E] Jede Behandlung benötigt ein Behandlungsdatum

`!BEH[i].isNull && BEH[i].dateDatum.isNull`

A4406 [E] Nebenbehandlungen werden ohne Stunde übermittelt

`i > 0 && BEH[i].dateDatum.hasDate && strlen(BEH[i].dateDatum.Value) > 8`

A4410 [E] Wenn ein Schweregrad angegeben wurde, so muss auch die Art des Scores eingetragen sein

`strlen(V4402) > 0 && strlen(V4403) == 0`

A4420 [E] Wenn die Art des Scores angegeben wurde, so muss auch ein Schweregrad eingetragen sein

`strlen(V4403) > 0 && strlen(V4402) == 0`

A4430 [E] Ungültige Art des Scores

`strlen(V4403) > 0 && !isIn(V4403,3,_3_SCOREART)`

A4440 [E] Das Total aller NEMS-Schichten darf nicht kleiner als 0 oder grösser als 999999 sein

`strlen(V4404) > 0 && (atol(V4404) < 0 || atol(V4404) > 999999)`

A4450 [E] Das SGI-Reservefeld 1 ist noch nicht definiert. Es muss leer sein.

`strlen(V4405) > 0`

A4460 [E] Das SGI-Reservefeld 2 ist noch nicht definiert. Es muss leer sein.

`strlen(V4406) > 0`

A4470 [H] Aufnahmegewicht: Angabe fehlt

`strlen(V4501) == 0 && strcmp(MB->V1203,'3') == 0`

A4471 [E] Bei bis zu 28 Tage alten Säuglingen muss das Aufnahmegewicht erfasst werden.

`strcmp(MB->V0202,'A') == 0 && MB->AGED < 28 && strlen(V4501) == 0`

A4475 [W] Aufnahmegewicht eines Säuglings bis 12 Monate: Angabe fehlt

`strcmp(MB->V0202,'A') == 0 && strlen(V4501) == 0 && strcmp(MB->V1203,'3') != 0 && MB->AGEJ < 1`

A4480 [E] Das Aufnahmegewicht darf nicht kleiner als 0 oder grösser als 99999 Gramm sein

`strlen(V4501) > 0 && (atol(V4501) < 0 || atol(V4501) > 16000)`

A4490 [E] Das Aufnahmegewicht eines Säuglings muss - bei Eintrittsart = 3 Geburt - dem Geburtsgewicht im Neugeborenen-Datensatz entsprechen

`MB->hasMN && strlen(V4501) > 0 && strcmp(MB->V1203,'3') == 0 && atol(V4501) != atol(MN->V2204)`

A4500 [W] Kopfumfang: Angabe fehlt

`strcmp(MB->V0202,'A') == 0 && strlen(V4502) == 0 && MB->hasMN`

A4501 [H] Kopfumfang: wird nur bei Neugeborenen erfasst.

`strcmp(MB->V0202,'A') == 0 && strlen(V4502) > 0 && !MB->hasMN`

`ACTION: V4502=";`

A4510 [W] Der Kopfumfang darf bei Lebendgeburten nicht kleiner als 15 oder grösser als 50 cm sein.

`strlen(V4502) > 0 && strcmp(MB->V1502,'5') == 1 && (atol(V4502) < 15 || atol(V4502) > 50)`

A4640 [E] Weitere Wiedereintritte: Ungültige Angabe

`strlen(V4741) > 0 && !(strcmp(V4741,'0') == 0 || strcmp(V4741,'1') == 0)`

A4700 [E] Fallnummer (Fallkostenstatistik): Angabe darf max. 16 Zeichen lang sein

`strlen(V4601) > 16`

A4710 [E] Fallnummer (Fallkostenstatistik): Ungültige Sonderzeichen

`strlen(V4601) > 0 && !isAlphaNumUpperCase(V4601)`

A4800 [E] Das 16. Reservefeld ist noch nicht definiert. Es muss leer sein.

`strlen(V4816) > 0`

A4810 [E] Das 17. Reservefeld ist noch nicht definiert. Es muss leer sein.

`strlen(V4817) > 0`

A4820 [E] Das 18. Reservefeld ist noch nicht definiert. Es muss leer sein.

`strlen(V4818) > 0`

A4830 [E] Das 19. Reservefeld ist noch nicht definiert. Es muss leer sein.

`strlen(V4819) > 0`

A4840 [E] Das 20. Reservefeld ist noch nicht definiert. Es muss leer sein.

`strlen(V4820) > 0`

A4850 [W] DRG-Status: Angabe darf max. 1 Zeichen lang sein

`strlen(V4801) > 1`

A4851 [W] Medikament 1: Angabe darf max. 50 Zeichen lang sein

`strlen(V4802) > 50`

A4852 [W] Medikament 2: Angabe darf max. 50 Zeichen lang sein

`strlen(V4803) > 50`

A4853 [W] Medikament 3: Angabe darf max. 50 Zeichen lang sein

`strlen(V4804) > 50`

A4854 [W] Medikament 4: Angabe darf max. 50 Zeichen lang sein

`strlen(V4805) > 50`

A4855 [W] Medikament 5: Angabe darf max. 50 Zeichen lang sein

`strlen(V4806) > 50`

A4856 [W] Medikament 6: Angabe darf max. 50 Zeichen lang sein

`strlen(V4807) > 50`

A4857 [W] Medikament 7: Angabe darf max. 50 Zeichen lang sein

`strlen(V4808) > 50`

A4858 [W] Medikament 8: Angabe darf max. 50 Zeichen lang sein

`strlen(V4809) > 50`

A4859 [W] Medikament 9: Angabe darf max. 50 Zeichen lang sein

`strlen(V4810) > 50`

A4860 [W] Medikament 10: Angabe darf max. 50 Zeichen lang sein

`strlen(V4811) > 50`

A4861 [W] Medikament 11: Angabe darf max. 50 Zeichen lang sein

`strlen(V4812) > 50`

A4862 [W] Medikament 12: Angabe darf max. 50 Zeichen lang sein

`strlen(V4813) > 50`

A4863 [W] Medikament 13: Angabe darf max. 50 Zeichen lang sein

`strlen(V4814) > 50`

A4864 [W] Medikament 14: Angabe darf max. 50 Zeichen lang sein

`strlen(V4815) > 50`

A4865 [E] Reservefeld 16: Angabe darf max. 50 Zeichen lang sein

`strlen(V4816) > 50`

A4866 [E] Reservefeld 17: Angabe darf max. 50 Zeichen lang sein

`strlen(V4817) > 50`

A4867 [E] Reservefeld 18: Angabe darf max. 50 Zeichen lang sein

`strlen(V4818) > 50`

A4868 [E] Reservefeld 19: Angabe darf max. 50 Zeichen lang sein

`strlen(V4819) > 50`

A4869 [E] Reservefeld 20: Angabe darf max. 50 Zeichen lang sein

`strlen(V4820) > 50`

A4870 [E] DRG Status: Ungültige Angabe

`strlen(V4801) > 0 && !isIn(V4801,2,_01)`

A4871 [W] Der DRG-Status muss erfasst werden

`strcmp(MB->V0202,'A') == 0 && strlen(V4801) == 0`

A4900 [E] 1. Zwischenaustritt: Ungültige Datumsangabe

`!dateV4701.isNull && !dateV4701.isYYYYMMDD`

A4901 [E] 1. Wiedereintritt: Ungültige Datumsangabe

`!dateV4702.isNull && !dateV4702.isYYYYMMDD`

A4902 [E] 2. Zwischenaustritt: Ungültige Datumsangabe

`!dateV4711.isNull && !dateV4711.isYYYYMMDD`

A4903 [E] 2. Wiedereintritt: Ungültige Datumsangabe

`!dateV4712.isNull && !dateV4712.isYYYYMMDD`

A4904 [E] 3. Zwischenaustritt: Ungültige Datumsangabe

`!dateV4721.isNull && !dateV4721.isYYYYMMDD`

A4905 [E] 3. Wiedereintritt: Ungültige Datumsangabe

`!dateV4722.isNull && !dateV4722.isYYYYMMDD`

A4906 [E] 4. Zwischenaustritt: Ungültige Datumsangabe

`!dateV4731.isNull && !dateV4731.isYYYYMMDD`

A4907 [E] 4. Wiedereintritt: Ungültige Datumsangabe

`!dateV4732.isNull && !dateV4732.isYYYYMMDD`

A4910 [E] Der 1. Zwischenaustritt (4.7.V01) liegt vor dem Eintrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4701.isNull && MB->dateV1201.DateValue > dateV4701.DateValue`

A4911 [E] Der 1. Wiedereintritt (4.7.V02) liegt vor dem Eintrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4702.isNull && MB->dateV1201.DateValue > dateV4702.DateValue`

A4912 [E] Der 2. Zwischenaustritt (4.7.V11) liegt vor dem Eintrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4711.isNull && MB->dateV1201.DateValue > dateV4711.DateValue`

A4913 [E] Der 2. Wiedereintritt (4.7.V12) liegt vor dem Eintrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4712.isNull && MB->dateV1201.DateValue > dateV4712.DateValue`

A4914 [E] Der 3. Zwischenaustritt (4.7.V21) liegt vor dem Eintrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4721.isNull && MB->dateV1201.DateValue > dateV4721.DateValue`

A4915 [E] Der 3. Wiedereintritt (4.7.V22) liegt vor dem Eintrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4722.isNull && MB->dateV1201.DateValue > dateV4722.DateValue`

A4916 [E] Der 4. Zwischenaustritt (4.7.V31) liegt vor dem Eintrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4731.isNull && MB->dateV1201.DateValue > dateV4731.DateValue`

A4917 [E] Der 4. Wiedereintritt (4.7.V32) liegt vor dem Eintrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4732.isNull && MB->dateV1201.DateValue > dateV4732.DateValue`

A4920 [E] Der 1. Zwischenaustritt (4.7.V01) liegt nach dem Austrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4701.isNull && dateV4701.DateValue > MB->dateV1501.DateValue`

A4921 [E] Der 1. Wiedereintritt (4.7.V02) liegt nach dem Austrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4702.isNull && dateV4702.DateValue > MB->dateV1501.DateValue`

A4922 [E] Der 2. Zwischenaustritt (4.7.V11) liegt nach dem Austrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4711.isNull && dateV4711.DateValue > MB->dateV1501.DateValue`

A4923 [E] Der 2. Wiedereintritt (4.7.V12) liegt nach dem Austrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4712.isNull && dateV4712.DateValue > MB->dateV1501.DateValue`

A4924 [E] Der 3. Zwischenaustritt (4.7.V21) liegt nach dem Austrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4721.isNull && dateV4721.DateValue > MB->dateV1501.DateValue`

A4925 [E] Der 3. Wiedereintritt (4.7.V22) liegt nach dem Austrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4722.isNull && dateV4722.DateValue > MB->dateV1501.DateValue`

A4926 [E] Der 4. Zwischenaustritt (4.7.V31) liegt nach dem Austrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4731.isNull && dateV4731.DateValue > MB->dateV1501.DateValue`

A4927 [E] Der 4. Wiedereintritt (4.7.V32) liegt nach dem Austrittsdatum

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4732.isNull && dateV4732.DateValue > MB->dateV1501.DateValue`

A4930 [E] Der erste Zwischenaustritt muss vor dem zweiten Zwischenaustritt liegen

`strcmp(MB->V0202,'A') == 0 && !dateV4701.isNull && !dateV4711.isNull && dateV4701.DateValue > dateV4711.DateValue`

A4931 [E] Der zweite Zwischenaustritt muss vor dem dritten Zwischenaustritt liegen

`strcmp(MB->V0202,'A') == 0 && !dateV4711.isNull && !dateV4721.isNull && dateV4711.DateValue > dateV4721.DateValue`

A4932 [E] Der dritte Zwischenaustritt muss vor dem vierten Zwischenaustritt liegen

`strcmp(MB->V0202,'A') == 0 && !dateV4721.isNull && !dateV4731.isNull && dateV4721.DateValue > dateV4731.DateValue`

A4940 [E] Der erste Wiedereintritt muss vor dem zweiten Wiedereintritt liegen

`strcmp(MB->V0202,'A') == 0 && !dateV4702.isNull && !dateV4712.isNull && dateV4702.DateValue > dateV4712.DateValue`

A4941 [E] Der zweite Wiedereintritt muss vor dem dritten Wiedereintritt liegen

`strcmp(MB->V0202,'A') == 0 && !dateV4712.isNull && !dateV4722.isNull && dateV4712.DateValue > dateV4722.DateValue`

A4942 [E] Der dritte Wiedereintritt muss vor dem vierten Wiedereintritt liegen

`strcmp(MB->V0202,'A') == 0 && !dateV4722.isNull && !dateV4732.isNull && dateV4722.DateValue > dateV4732.DateValue`

A4950 [E] Grund des 1. Wiedereintrittes: Ungültige Angabe

`strlen(V4703) > 0 && !isIn(V4703,4,_1239)`

A4951 [E] Grund des 2. Wiedereintrittes: Ungültige Angabe

`strlen(V4713) > 0 && !isIn(V4713,4,_1239)`

A4952 [E] Grund des 3. Wiedereintrittes: Ungültige Angabe

`strlen(V4723) > 0 && !isIn(V4723,4,_1239)`

A4953 [E] Grund des 4. Wiedereintrittes: Ungültige Angabe

`strlen(V4733) > 0 && !isIn(V4733,4,_1239)`

A4960 [E] Wenn ein 1. Zwischenaustritt angegeben wird, muss der 1. Wiedereintritt ausgefüllt sein

`strlen(V4701) > 0 && strlen(V4702) == 0`

A4961 [E] Wenn ein 2. Zwischenaustritt angegeben wird, muss der 2. Wiedereintritt ausgefüllt sein

`strlen(V4711) > 0 && strlen(V4712) == 0`

A4962 [E] Wenn ein 3. Zwischenaustritt angegeben wird, muss der 3. Wiedereintritt ausgefüllt sein

`strlen(V4721) > 0 && strlen(V4722) == 0`

A4963 [E] Wenn ein 4. Zwischenaustritt angegeben wird, muss der 4. Wiedereintritt ausgefüllt sein

`strlen(V4731) > 0 && strlen(V4732) == 0`

A4970 [E] Wenn ein 1. Wiedereintritt angegeben wird, muss auch dessen Grund ausgefüllt sein

`strlen(V4702) > 0 && strlen(V4703) == 0`

A4971 [E] Wenn ein 2. Wiedereintritt angegeben wird, muss auch dessen Grund ausgefüllt sein

`strlen(V4712) > 0 && strlen(V4713) == 0`

A4972 [E] Wenn ein 3. Wiedereintritt angegeben wird, muss auch dessen Grund ausgefüllt sein

`strlen(V4722) > 0 && strlen(V4723) == 0`

A4973 [E] Wenn ein 4. Wiedereintritt angegeben wird, muss auch dessen Grund ausgefüllt sein

`strlen(V4732) > 0 && strlen(V4733) == 0`

A4975 [E] Der Fall hat einen 1. Zwischenaustritt im Vorjahr, einen Austritt im laufenden Jahr. Diese Rehosp-Erfassung entspricht nicht den SwissDRG-Regeln.

`strcmp(MB->V0202,'A') == 0 && dateV4701.isYYYYMMDD && dateV4701.Year != MB->dateV1501.Year`

A4976 [E] Die Dauer des administrativen Urlaubs entspricht der Länge der Abwesenheiten zwischen den Aus-/Wiedereintritten. Wurde die Abwesenheit doppelt erfasst?

`dateV4701.hasDate && UrlaubT > 0 && UrlaubT == ABS`

A4977 [E] Die berechnete Aufenthaltsdauer beträgt weniger als 1 Tag.

`strcmp(MB->V0202,'A') == 0 && LOS < 1`

A4978 [E] Ein Wiedereintritt liegt mehr als 18 Tage hinter dem ersten Zwischenaustritt. Bitte prüfen Sie den Rehosp-Verlauf dieses Falles.

`(dateV4702.hasDate && dateV4702.DateValue - dateV4701.DateValue > 18) || (dateV4712.hasDate && dateV4712.DateValue - dateV4701.DateValue > 18) || (dateV4722.hasDate && dateV4722.DateValue - dateV4701.DateValue > 18) || (dateV4732.hasDate && dateV4732.DateValue - dateV4701.DateValue > 18)`

A4980 [E] Der 1. Wiedereintritt liegt mehr als 18 Tage hinter dem 1. Zwischenaustritt.

`dateV4701.isYYYYMMDD && dateV4702.isYYYYMMDD && dateV4702.DateValue - dateV4701.DateValue > 18`

A4981 [E] Der 2. Wiedereintritt liegt mehr als 18 Tage hinter dem 2. Zwischenaustritt.

`dateV4701.isYYYYMMDD && dateV4712.isYYYYMMDD && dateV4712.DateValue - dateV4701.DateValue > 18`

A4982 [E] Der 3. Wiedereintritt liegt mehr als 18 Tage hinter dem 3. Zwischenaustritt.

`dateV4701.isYYYYMMDD && dateV4722.isYYYYMMDD && dateV4722.DateValue - dateV4701.DateValue > 18`

A4983 [E] Der 4. Wiedereintritt liegt mehr als 18 Tage hinter dem 4. Zwischenaustritt.

`dateV4701.isYYYYMMDD && dateV4732.isYYYYMMDD && dateV4732.DateValue - dateV4701.DateValue > 18`

A4984 [E] 1. Zwischenaustritt/1. Wiedereintritt: unplausible Angaben.

`strcmp(MB->V0202,'A') == 0 && (strlen(V4701) > 0 || strlen(V4702) > 0) && (!dateV4701.isYYYYMMDD || !dateV4702.isYYYYMMDD || dateV4701.DateValue < MB->dateV1201.DateValue || dateV4702.DateValue < dateV4701.DateValue || dateV4702.DateValue > MB->dateV1501.DateValue)`

A4985 [E] 2. Zwischenaustritt/2. Wiedereintritt: unplausible Angaben.

```
strcmp(MB->V0202,'A') == 0 && (strlen(V4711) > 0 || strlen(V4712) > 0) && (!dateV4711.isYYYYMMDD || !dateV4712.isYYYYMMDD || dateV4711.DateValue < dateV4702.DateValue || dateV4712.DateValue < dateV4711.DateValue || dateV4712.DateValue > MB->dateV1501.DateValue)
```

A4986 [E] 3. Zwischenaustritt/3. Wiedereintritt: unplausible Angaben.

```
strcmp(MB->V0202,'A') == 0 && (strlen(V4721) > 0 || strlen(V4722) > 0) && (!dateV4721.isYYYYMMDD || !dateV4722.isYYYYMMDD || dateV4721.DateValue < dateV4712.DateValue || dateV4722.DateValue < dateV4721.DateValue || dateV4722.DateValue > MB->dateV1501.DateValue)
```

A4987 [E] 4. Zwischenaustritt/4. Wiedereintritt: unplausible Angaben.

```
strcmp(MB->V0202,'A') == 0 && (strlen(V4731) > 0 || strlen(V4732) > 0) && (!dateV4731.isYYYYMMDD || !dateV4732.isYYYYMMDD || dateV4731.DateValue < dateV4722.DateValue || dateV4732.DateValue < dateV4731.DateValue || dateV4732.DateValue > MB->dateV1501.DateValue)
```

A.5. Kantonale Zusatzdaten

Gemäss Detailkonzept können die Kantone eigene Zusatzdatensätze definieren und erheben lassen. Zu diesem Zweck ist die Recordart MK definiert.

Als zweite Variable ist die Identifikation des Kantons festgelegt. In der Definition aller weiteren Felder des Datensatzes sind die Kantone frei, wobei die allgemeinen Vorgaben des Schnittstellenkonzeptes eingehalten werden müssen.

Die kantonalen Datensätze werden vom BFS nicht weiterverarbeitet und vor dem Einlesen ausgefiltert.

A5000 [E] Kantonskürzel im kantonalen Zusatzdatensatz: Angabe fehlt

```
isMKCH && strlen(V5102) == 0
```

A5010 [E] Kantonskürzel im kantonalen Zusatzdatensatz: Ungültige Angabe

```
isMKCH && strlen(V5102) > 0 && !isIn(V5102,26,_26_KANTON)
```

B. Kreuztests zwischen verschiedenen Feldern eines Datensatzes

B.0. Lieferungsmeldung

B.0.1. Konsistenzprüfung Datumsangaben

Die in der Lieferungsmeldung vorhandenen Datumsangaben betreffend Beginn und Ende der Meldeperiode und das Datum, an dem der Datenexport erzeugt wurde, müssen kohärent sein.

Der Beginn der Erhebungsperiode liegt zeitlich vor dem Ende der Erhebungsperiode und der Export muss nach dem Ende der Erhebungsperiode durchgeführt worden sein.

B0100 [E] Lieferungsmeldung Inkonsistenz zwischen Meldeperiode von, bis und Export-Erstellungsdatum

```
dateV0007.DateValue < dateV0006.DateValue || dateV0004.DateValue < dateV0007.DateValue
```

B.1. Minimaldatensatz

B.1.1. Statistikfall

Die Variable Statistikfall (0.2.V02) ist abhängig von den Ein- und Austrittsdaten. Es werden drei Fälle unterschieden:

Statistikfall	Beschreibung
A =	Normaler Minimaldatensatz: Spitaleintritt (bzw. Falleröffnung) vor oder während der Erhebungsperiode, Austritt zwischen Beginn und Ende der Erhebungsperiode (normalerweise zwischen 1.1. und 31.12.).
B =	Spitaleintritt während der Erhebungsperiode, andauernde Behandlung am Ende der Erhebungsperiode. In diesem Fall können keine Austrittsangaben gemacht werden. Es erfolgt eine administrative Meldung mit reduziertem Datensatz. Die reduzierten Meldungen sollen mindestens die Variablen bis 1.2.V01 (Eintrittsdatum) enthalten.
C =	Administrative Meldung mit vollständigem Minimaldatensatz ohne Austrittsmerkmale (1.5.V01 - 1.5.V04).

B1100 [E] MB: Statistikfall A-Test: Austritt muss in Erhebungsperiode liegen

```
strcmp(V0202,'A') == 0 && dateV1501.hasDate && (dateV1501.DateValue < MX->dateV0006.DateValue || dateV1501.DateValue > MX->dateV0007.DateValue)
```

ACTION: doWriteRec=FALSE;

B1101 [E] Statistikfall B-Test: Eintritt muss in Erhebungsperiode liegen

```
V0202[0] == 'B' && dateV1201.hasDate && (dateV1201.DateValue < MX->dateV0006.DateValue || dateV1201.DateValue > MX->dateV0007.DateValue)
```

B1102 [E] Statistikfall B-Test: Austritt kann nicht vor Ende der Meldeperiode sein

```
strcmp(V0202,'B') == 0 && dateV1501.hasDate && dateV1501.DateValue < MX->dateV0007.DateValue
```

B1103 [E] Statistikfall C-Test: Eintritt muss vor Erhebungsperiode liegen

```
strcmp(V0202,'C') == 0 && dateV1201.hasDate && dateV1201.DateValue > MX->dateV0006.DateValue
```

B1104 [E] Statistikfall C-Test: Austritt kann nicht vor Ende der Meldeperiode sein

```
V0202[0] == 'C' && dateV1501.hasDate && dateV1501.DateValue < MX->dateV0007.DateValue
```

B1105 [H] Statistikfall B-Test: Angaben werden nur bis 1.4.V02 (Hauptkostenträger) erwartet

```
V0202[0] == 'B' && (strlen(V1501) > 0 || strlen(V1502) > 0 || strlen(V1503) > 0 || strlen(V1504) > 0 || DIACount > 0 || BEHCount > 0)
```

```
ACTION: V1501=""; V1502=""; V1503=""; V1504=""; V1702=""; for (int di=0; di<=DIACount;di++) DIA[di].Key=""; for (int bi=0; bi<=BEHCount;bi++) BEH[bi].Key="";
```

B1106 [H] Statistikfall C-Test: Austrittsangaben werden nicht erwartet

```
strcmp(V0202,'C') == 0 && (strlen(V1501) > 0 || strlen(V1502) > 0 || strlen(V1503) > 0 || strlen(V1504) > 0)
```

```
ACTION: V1501=""; V1502=""; V1503=""; V1504=""; V1702="";
```

B1108 [H] Behandlungsdatum / Meldeperiode: Behandlungsbeginn kann nicht nach Ende der Meldeperiode sein

```
!dateV1702.isNull && dateV1702.DateValue > MX->dateV0007.DateValue
```

B.1.2. Vollständiges Geburtsdatum bei Kindern und bei Todesfällen

Das Geburtsdatum kann mit den Variablen Alter bei Eintritt (1.1.V03), Eintrittsdatum (1.2.V01), Austrittsdatum (1.5.V01) und Beginn der Hauptbehandlung (1.7.V02) plausibilisiert werden. Bei Kindern bis und mit zwei Jahren (Alter bei Eintritt, 1.1.V03) muss das vollständige Geburtsdatum angegeben werden.

B1200 [E] Geburtsjahr: Liegt vor mehr als 120 Jahren

```
dateV1102.hasYear && dateV1102.Year < MX->dateV0004.Year - 120
```

B1201 [E] Geburtsdatum/Eintrittsdatum: Nicht konsistent mit Altersangabe

```
dateV1201.hasYear && dateV1102.hasYear && strlen(V1103) > 0 && !(dateV1201.Year - dateV1102.Year == atoi(V1103) || dateV1201.Year - dateV1102.Year == atoi(V1103)+1)
```

B1202 [E] Geburtsdatum bei Kindern: Bei Kindern bis 2 Jahren muss das vollständige Geburtsdatum angegeben werden

```
isInt(V1103) && atoi(V1103) < 2 && dateV1102.isYYYY0000
```

B1203 [E] Geburtsdatum bei Todesfall: Bei Todesfällen muss das vollständige Geburtsdatum angegeben werden

```
strcmp(V1502,'5') == 0 && dateV1102.isYYYY0000
```

B.1.3. Eintrittsdatum und -stunde

Das Eintrittsdatum soll verglichen werden mit dem aktuellen Datum (nicht mehr als 2 Jahre Differenz), dem Austrittsdatum (1.5.V01) und dem Datum des Beginns der Hauptbehandlung (1.7.V02). Bei Eintrittsart 1 = Notfall muss das Eintrittsdatum inkl. Stunde erfolgen.

B1300 [H] Eintrittsdatum / Beginn Meldeperiode: Liegt mehr als 2 Jahre vor dem Beginn der Meldeperiode

```
dateV1201.hasDate && MX->dateV0006.hasDate && MX->dateV0006.Year - dateV1201.Year > 2
```

B1301 [E] Notfalleintritt: Eintrittsdatum muss inklusive Stundenangabe erfolgen

```
V1203[0] == '1' && dateV1201.hasDate && !dateV1201.hasHour
```

B.1.4. Konsistenz der Datumsangaben

Die Chronologie und Logik der Datumsangaben muss korrekt sein. Das Geburtsdatum muss kleiner sein als das Eintrittsdatum, das Behandlungsdatum und schlussendlich das Austrittsdatum.

B1400 [E] Geburtsdatum darf nicht später als Eintrittsdatum sein

```
dateV1102.hasYear && dateV1201.hasDate && dateV1102.DateValue > dateV1201.DateValue
```

B1401 [E] MB: Eintritt darf nicht später als Behandlung sein

```
strcmp(MB->V0202,'A') == 0 && (dateV1201.hasDate && dateV1702.hasDate && dateV1201.DateValue > dateV1702.DateValue) || (dateV1201.hasHour && dateV1702.hasHour && dateV1201.DateValue == dateV1702.DateValue && dateV1201.Hour > dateV1702.Hour)
```

B1402 [E] MB: Behandlung darf nicht später als Austritt sein

```
(dateV1702.hasDate && dateV1501.hasDate && dateV1702.DateValue > dateV1501.DateValue) || (dateV1702.hasHour && dateV1501.hasHour && dateV1702.DateValue == dateV1501.DateValue && dateV1702.Hour > dateV1501.Hour)
```

B1403 [E] Eintrittsdatum darf nicht später als Austrittsdatum sein

```
dateV1201.hasDate && dateV1501.hasDate && dateV1201.DateValue > dateV1501.DateValue
```

B1404 [E] Eintrittszeit darf nicht später als Behandlungszeitpunkt sein

```
dateV1201.hasHour && dateV1702.hasHour && dateV1201.DateValue == dateV1702.DateValue && dateV1201.Hour > dateV1702.Hour
```

B1406 [E] Eintrittszeit darf nicht später als Austrittszeitpunkt sein

```
dateV1201.hasHour && dateV1501.hasHour && dateV1201.DateValue == dateV1501.DateValue && dateV1201.Hour > dateV1501.Hour
```

B1407 [E] Geburtsdatum darf nicht später als Beginn der Hauptbehandlung sein

```
dateV1102.hasYear && dateV1702.hasDate && dateV1102.DateValue > dateV1702.DateValue
```

B.1.5. Hauptkostenstelle und Alter

Die Hauptkostenstelle Pädiatrie und Geriatrie können mit der Altersangabe des Patienten plausibilisiert werden. Die Pädiatrie (M4..) hat eine obere Altersgrenze von 18 Jahren. In der Geriatrie (M9..) liegt die untere Grenze bei 30 Jahren.

B1500 [W] Hauptkostenstelle / Alter: Obere Altersgrenze überschritten für Hauptkostenstelle Pädiatrie (M400)

```
strlen(V1103) > 0 && strcmp(V1401,'M400') == 0 && atol(V1103) > 18
```

B1501 [W] Hauptkostenstelle / Alter: Unterhalb Altersgrenze für Geriatrie (M900)

`strlen(V1103) > 0 && strcmp(V1401,'M900') == 0 && atol(V1103) < 30`

B.1.6. Eintrittsart Geburt

Wird als Eintrittsart Geburt, Kind in dieser Klinik geboren angegeben, so kann das Alter bei Eintritt nur 0 sein.

B1600 [E] Geburt (Kind)/Altersangabe: Ein Neugeborenes hat das Alter 0

`strcmp(V1203,'3') == 0 && atol(V1103) != 0`

B1601 [W] Eintritts-, Geburtsdatum, Eintrittsart: Bei Eintritt durch Geburt Eintrittsart überprüfen

`strcmp(MB->V1103,'0') == 0 && MB->dateV1102.DateValue == MB->dateV1201.DateValue && strcmp(MB->V1203,'3') != 0 && !hasZ38_147`

B1602 [E] Eine Mutter darf nie Eintrittsart Geburt haben

`strlen(MB->V1103) > 0 && atoi(MB->V1103)>0 && strcmp(MB->V1203,'3') == 0`

B.1.7. Aufenthalt in Intensivmedizin und administrativer Urlaub

Die Dauer, die ein Patient in intensivmedizinischer Betreuung verbracht hat (Angabe in Anzahl Stunden) kann nicht höher sein als die Dauer des ganzen Aufenthalts.

Dies gilt ebenso für die Angabe im Feld administrativer Urlaub und Ferien.

B1700 [E] Aufenthalt in Intensivmedizin: Angabe übersteigt die Dauer des gesamten Aufenthaltes

`dateV1501.hasDate && dateV1201.hasDate && atol(V1303) > 0 && (dateV1501.DateValue > dateV1201.DateValue) && atol(V1303) > (dateV1501.DateValue - dateV1201.DateValue + 1) * 24`

B1701 [E] Administrativer Urlaub, Ferien: Angabe übersteigt die Dauer des gesamten Aufenthaltes

`dateV1501.hasDate && dateV1201.hasDate && atol(V1304) > 0 && (dateV1501.DateValue > dateV1201.DateValue) && atol(V1304) > (dateV1501.DateValue - dateV1201.DateValue + 1) * 24`

B.1.8. Austritt Todesfall

Ein Todesfall wird mit der Angabe Entscheid für Austritt 5 = Todesfall signalisiert. Das Austrittsdatum wird inklusive Zeitangabe in Stunden des Zeitpunkts des Todes angegeben. Die Felder Aufenthalt nach und Behandlung nach sind mit 0 zu initialisieren.

B1800 [W] Todesfall: Aufenthalt und Behandlung nach müssen beide 0 sein

`strcmp(V1502,'5') == 0 && (strcmp(V1503,'0') != 0 || strcmp(V1504,'0') != 0)`

B1801 [E] Austrittsdatum bei Todesfall: Angabe muss inklusive Stunde erfolgen

`strcmp(V1502,'5') == 0 && !dateV1501.hasHour`

B1802 [E] Aufenthalt/Behandlung nach: Angabe nur bei Todesfällen möglich

`(strcmp(V1503,'0') == 0 || strcmp(V1504,'0') == 0) && strcmp(V1502,'5') != 0`

B.1.9. Behandlungsart und Aufenthaltsdauer

Die Behandlungsart muss konsistent sein mit der Dauer des Aufenthaltes, welche sich aus der Differenz von Ein- und Austrittsdatum ergibt. Wird die Aufenthaltsart (1.3.V01) mit 1 = Ambulant angegeben, so kann die Aufenthaltsdauer höchstens einen Tag betragen. Andernfalls wird eine Fehlermeldung angezeigt.

B1900 [H] Aufenthaltsdauer bei ambulanten Fällen: Ungültige Aufenthaltsdauer für ambulante Behandlung

```
strcmp(V1301,'1') == 0 && dateV1201.hasDate && dateV1501.hasDate && dateV1501.DateValue - dateV1201.DateValue > 1
```

B1907 [W] B-/C-Fälle mit MP: Austrittsmerkmale V3501 Entscheid für Austritt, V3502 Aufenthalt nach Austritt, V3503 Behandlung nach Austritt werden nicht ausgefüllt.

```
strcmp(MB->V0202,'A') != 0 && (strlen(V3501) > 0 || strlen(V3502) > 0 || strlen(V3503) > 0)
```

```
ACTION: V3501 = ""; V3502=""; V3503="";
```

B1908 [E] Wenn die Eintrittsart 5 (Verlegung innerhalb 24 Std.) ist, muss der Aufenthaltsort vor dem Eintritt 3 (Krankenheim, Pflegeheim) sein.

```
strlen(V1202) > 0 && strcmp(V1203,'5') == 0 && strcmp(V1202,'6') != 0
```

B.2. Neugeborenen-Zusatzdaten

B.2.0. Statistikfall

Die Variable Statistikfall (0.2.V02) ist abhängig von den Ein- und Austrittsdaten. Es werden drei Fälle unterschieden:

Statistikfall	Beschreibung
A =	Normaler Minimaldatensatz: Spitaleintritt (bzw. Falleröffnung) vor oder während der Erhebungsperiode, Austritt zwischen Beginn und Ende der Erhebungsperiode (normalerweise zwischen 1.1. und 31.12.).
B =	Spitaleintritt während der Erhebungsperiode, andauernde Behandlung am Ende der Erhebungsperiode. In diesem Fall können keine Austrittsangaben gemacht werden. Es erfolgt eine administrative Meldung mit reduziertem Datensatz. Die reduzierten Meldungen sollen mindestens die Variablen bis 1.2.V01 (Eintrittsdatum) enthalten.
C =	Administrative Meldung mit vollständigem Minimaldatensatz ohne Austrittsmerkmale (1.5.V01 - 1.5.V04).

B2000 [E] MN: Statistikfall A-Test: Austritt muss in Erhebungsperiode liegen

```
strcmp(MB->V0202,'A') == 0 && MB->dateV1501.hasDate && (MB->dateV1501.DateValue < MX->dateV0006.DateValue || MB->dateV1501.DateValue > MX->dateV0007.DateValue)
```

B.2.1. Summe aller Schwangerschaften

Die Angabe unter Anzahl vorausgegangene Schwangerschaften insgesamt (2.3.V04) darf die Summe der Felder Anzahl vorausgegangene Lebendgeburten (2.3.V05), Anzahl vorausgegangene Fehl- oder Totgeburten (2.3.V06), Anzahl vorausgegangene Schwangerschaftsabbrüche (2.3.V07) nicht überschreiten.

B2100 [W] Neugeborenen-Zusatzdaten: Inkonsistente Angaben betreffend Schwangerschaften

```
strlen(V2305) > 0 && strlen(V2306) > 0 && strlen(V2307) > 0 && atol(V2304) > (atol(V2305) + atol(V2306) + atol(V2307))
```

B2110 [W] Verhältnis Geburtsgewicht zu Körperlänge: Ausserhalb des gültigen Wertebereichs

```
atol(V2205) != 0 && (atol(V2204) / atol(V2205) < 20 || atol(V2204) / atol(V2205) > 120)
```

B.3. Psychiatrie-Zusatzdaten

B.3.0. Statistikfall

Die Variable Statistikfall (0.2.V02) ist abhängig von den Ein- und Austrittsdaten. Es werden drei Fälle unterschieden:

Statistikfall	Beschreibung
A =	Normaler Minimaldatensatz: Spitaleintritt (bzw. Falleröffnung) vor oder während der Erhebungsperiode, Austritt zwischen Beginn und Ende der Erhebungsperiode (normalerweise zwischen 1.1. und 31.12.).
B =	Spitaleintritt während der Erhebungsperiode, andauernde Behandlung am Ende der Erhebungsperiode. In diesem Fall können keine Austrittsangaben gemacht werden. Es erfolgt eine administrative Meldung mit reduziertem Datensatz. Die reduzierten Meldungen sollen mindestens die Variablen bis 1.2.V01 (Eintrittsdatum) enthalten.
C =	Administrative Meldung mit vollständigem Minimaldatensatz ohne Austrittsmerkmale (1.5.V01 - 1.5.V04).

B3000 [E] MP: Statistikfall A-Test: Austritt muss in Erhebungsperiode liegen

```
strcmp(MB->V0202,'A') == 0 && MB->dateV1501.hasDate && (MB->dateV1501.DateValue < MX->dateV0006.DateValue  
|| MB->dateV1501.DateValue > MX->dateV0007.DateValue)
```

```
ACTION: MB->V0302='0'; MB->hasMP=FALSE;
```

B.4. Patientengruppen-Zusatzdaten

B.4.0. Statistikfall

Die Variable Statistikfall (0.2.V02) ist abhängig von den Ein- und Austrittsdaten. Es werden drei Fälle unterschieden:

Statistikfall	Beschreibung
A =	Normaler Minimaldatensatz: Spitaleintritt (bzw. Falleröffnung) vor oder während der Erhebungsperiode, Austritt zwischen Beginn und Ende der Erhebungsperiode (normalerweise zwischen 1.1. und 31.12.).
B =	Spitaleintritt während der Erhebungsperiode, andauernde Behandlung am Ende der Erhebungsperiode. In diesem Fall können keine Austrittsangaben gemacht werden. Es erfolgt eine administrative Meldung mit reduziertem Datensatz. Die reduzierten Meldungen sollen mindestens die Variablen bis 1.2.V01 (Eintrittsdatum) enthalten.
C =	Administrative Meldung mit vollständigem Minimaldatensatz ohne Austrittsmerkmale (1.5.V01 - 1.5.V04).

B4000 [E] MD: Statistikfall A-Test: Austritt muss in Erhebungsperiode liegen

```
strcmp(MB->V0202,'A') == 0 && MB->dateV1501.hasDate && (MB->dateV1501.DateValue < MX->dateV0006.DateValue  
|| MB->dateV1501.DateValue > MX->dateV0007.DateValue)
```

B4401 [E] MD: Eintritt darf nicht später als Behandlung sein

```
strcmp(MB->V0202,'A') == 0 && (MB->dateV1201.hasDate && BEH[j].dateDatum.hasDate && MB->dateV1201.DateValue > BEH[j].dateDatum.DateValue) || (MB->dateV1201.hasHour && BEH[j].dateDatum.hasHour && MB->dateV1201.DateValue == BEH[j].dateDatum.DateValue && MB->dateV1201.Hour > BEH[j].dateDatum.Hour)
```

B4402 [E] MD: Behandlung darf nicht später als Austritt sein

```
(MB->dateV1501.hasDate && BEH[j].dateDatum.hasDate && MB->dateV1501.DateValue < BEH[j].dateDatum.DateValue) || (MB->dateV1501.hasHour && BEH[j].dateDatum.hasHour && MB->dateV1501.DateValue == BEH[j].dateDatum.DateValue && MB->dateV1501.Hour < BEH[j].dateDatum.Hour)
```

B4411 [W] Dieser ICD-Kode benötigt eine Seitigkeit

```
i != 1 && DIA[i].inIndex && strcmp(DIA[i].Data[ICD_LATERAL],'1') == 0
```

B.5. Kantonale Zusatzdaten

B.5.0. Statistikfall

Die Variable Statistikfall (0.2.V02) ist abhängig von den Ein- und Austrittsdaten. Es werden drei Fälle unterschieden:

Statistikfall	Beschreibung
A =	Normaler Minimaldatensatz: Spitaleintritt (bzw. Falleröffnung) vor oder während der Erhebungsperiode, Austritt zwischen Beginn und Ende der Erhebungsperiode (normalerweise zwischen 1.1. und 31.12.).
B =	Spitaleintritt während der Erhebungsperiode, andauernde Behandlung am Ende der Erhebungsperiode. In diesem Fall können keine Austrittsangaben gemacht werden. Es erfolgt eine administrative Meldung mit reduziertem Datensatz. Die reduzierten Meldungen sollen mindestens die Variablen bis 1.2.V01 (Eintrittsdatum) enthalten.
C =	Administrative Meldung mit vollständigem Minimaldatensatz ohne Austrittsmerkmale (1.5.V01 - 1.5.V04).

B5000 [E] MK: Statistikfall A-Test: Austritt muss in Erhebungsperiode liegen

```
strcmp(MB->V0202,'A') == 0 && MB->dateV1501.hasDate && (MB->dateV1501.DateValue < MX->dateV0006.DateValue || MB->dateV1501.DateValue > MX->dateV0007.DateValue)
```

C. Plausibilisierungen zwischen Datensätzen

C.0. Lieferungsmeldung und vorhandene Datensätze

C.0.0. Anzahl gelieferte Datensätze

Die Angabe der Anzahl vorhandene Datensätze in der Lieferungsmeldung muss der tatsächlichen Zahl entsprechen. Jede Zeile, auch die Lieferungsmeldung selbst, wird mitgezählt.

C0000 [E] Anzahl gelieferte Datensätze: Angabe in Lieferungsmeldung überschritten

```
LineNumberInMX > atol(MX->V0008)
```

C.0.1. Übereinstimmung BUR-Nummer

Die Angabe der BUR-Nummer in der Lieferungsmeldung muss dieselbe sein wie in allen zu der entsprechenden Lieferungsmeldung dazugehörigen Minimaldatensätzen.

C0100 [E] BUR-Nummer: Widerspruch mit der Angabe in der Lieferungsmeldung

```
strlen(V0102) > 0 && strcmp(V0102,MX->V0002) != 0
```

C.0.2. Verbindungscode-Testrecord

Um die Gültigkeit der Verbindungscode sicherzustellen, muss in jeder Datenlieferung als erster Datensatz ein Verbindungscode mit einer fiktiven Person stehen. Es handelt sich um den sogenannten Meier Hans-Record (mit den Angaben Meier Hans, männlich, Geburtsdatum 11.12.1913).

Anhand von diesem wohldefinierten Verbindungscode kann bei der Datenübernahme am BFS überprüft werden, ob die andern Verbindungscode der Lieferung korrekt generiert wurden.

C0200 [E] Pseudo Record "Meier Hans": Verbindungscode im Testrecord fehlt

```
strlen(V0201) == 0
```

C0210 [E] Pseudo Record "Meier Hans": Verbindungscode im Testrecord ist nicht hexadezimal

```
strlen(V0201) > 0 && !isHex(V0201)
```

C0220 [E] Anonymer Verbindungscode: Angabe ungültig, beruht auf fehlenden oder ungültigen Werten in Name, Vorname, Geburtsdatum (vollständig) oder Geschlecht vor dem Anonymisieren

```
strcmp(V0201,'0000000000000000') == 0
```

C0230 [E] Pseudo Record "Meier Hans": Die Zeichenlänge des Verbindungscode ist nicht korrekt

```
strlen(V0201) > 0 && strlen(V0201) != 16
```

C0240 [E] Pseudo Record "Meier Hans": Nur die Rekordart und der Verbindungscode dürfen ausgefüllt sein

```
strlen(V0102) > 0 || strlen(V0103) > 0 || strlen(V0104) > 0 || strlen(V0202) > 0 || strlen(V0301) > 0 || strlen(V0302) > 0
|| strlen(V0303) > 0 || strlen(V0304) > 0 || strlen(V1101) > 0 || strlen(V1102) > 0 || strlen(V1103) > 0 || strlen(V1104) >
0 || strlen(V1105) > 0 || strlen(V1201) > 0 || strlen(V1202) > 0 || strlen(V1203) > 0 || strlen(V1204) > 0 || strlen(V1301)
> 0 || strlen(V1302) > 0 || strlen(V1303) > 0 || strlen(V1304) > 0 || strlen(V1401) > 0 || strlen(V1402) > 0 || strlen(V1501)
> 0 || strlen(V1502) > 0 || strlen(V1503) > 0 || strlen(V1504) > 0 || strlen(DIA[0].Key) > 0
```

C.1. Existenz von Zusatzdatensätzen

Die Felder 0.3.V01 bis 0.3.V04 des Minimaldatensatzes enthalten Informationen (Flags) betreffend allfällige Zusatzdatensätze.

Ist eines der Flags gesetzt, so muss direkt nach dem Minimaldatensatz ein entsprechender Zusatzdatensatz folgen.

Momentan sind vier Typen von Zusatzmeldungen definiert: Neugeborenen- und Psychiatrie-Zusatzdatensätze, Kantonale Zusatzdaten und der Patientengruppen-Zusatzdatensatz.

C1100 [E] Neugeborenen-Zusatzdaten: Fehlender Zusatzdatensatz

```
strcmp(V0202,'A') == 0 && strcmp(V0301,'1') == 0 && !hasMN
```

```
ACTION: V0301='0';
```

C1101 [E] Neugeborenen-Zusatzdaten: Zusatzdatensatz vorhanden ohne Meldung im Basisdatensatz

```
strcmp(MB->V0301,'0') == 0
```

```
ACTION: MB->V0301='1';
```

C1102 [E] Geburt (Kind) / Zusatzdatensatz: Fehlender Neugeborenen-Zusatzdatensatz

```
strcmp(V1203,'3') == 0 && strcmp(V0301,'0') == 0 && strcmp(V1103,'0') == 0 && strcmp(V0202,'A') == 0
```

C1200 [E] Psychiatrie-Zusatzdaten: Fehlender Zusatzdatensatz

```
strcmp(V0302,'1') == 0 && !hasMP
```

```
ACTION: V0302='0';
```

C1201 [E] Psychiatrie-Zusatzdaten: Zusatzdatensatz vorhanden ohne Meldung im Basisdatensatz

```
strcmp(MB->V0302,'0') == 0
```

C1300 [E] Patientengruppen-Zusatzdaten: Fehlender Zusatzdatensatz

```
strcmp(V0303,'1') == 0 && !hasMD
```

```
ACTION: V0303='0';
```

C1301 [E] Patientengruppen-Zusatzdaten: Zusatzdatensatz vorhanden ohne Meldung im Basisdatensatz

```
strcmp(MB->V0303,'0') == 0
```

```
ACTION: MB->V0303='1';
```

C1400 [E] Kantonale Zusatzdaten: Fehlender Zusatzdatensatz

```
strcmp(V0304,'1') == 0 && !hasMK
```

```
ACTION: V0304='0';
```

C1401 [E] Kantonale Zusatzdaten: Zusatzdatensatz vorhanden ohne Meldung im Basisdatensatz

```
strcmp(MB->V0304,'0') == 0
```

```
ACTION: MB->V0304='1';
```

C1410 [E] Patientengruppen-Zusatzdaten: Fehlender obligatorischer MD-Zusatzdatensatz

```
strcmp(V0202,'A') == 0 && strcmp(V1301,'3') == 0 && !hasMD
```

C.2. Neugeborenen-Zusatzdaten

C.2.0. Eintrittsart und Alter bei Fällen mit Neugeborenen-Zusatzdaten

Wird ein Zusatzdatensatz übermittelt, so muss im zugehörigen Minimaldatensatz die Eintrittsart 3=Kind in dieser Klinik geboren lauten. Das Alter des Neugeborenen muss 0 sein (siehe auch B.1.6).

C2000 [E] Neugeborenen-Zusatz / Alter bei Eintritt: Die Altersangabe bei Neugeborenen kann nur 0 sein

```
strcmp(V2101,'MN') == 0 && atoi(MB->V1103) != 0
```

C2100 [H] Neugeborenen-Zusatz / Hauptkostenstelle: Neugeborene sind in der Regel auf der Geburtsabteilung (M300, M400)

```
strcmp(V2101,'MN') == 0 && !isIn(MB->V1401,2,_2_KST_MN)
```

C2200 [E] Vitalstatus / Austrittsart: Angaben stimmen nicht überein

```
strcmp(V2201,'0') == 0 && strcmp(MB->V1502,'5') != 0
```

C2310 [E] ICD-Kode zum Austrittszeitpunkt nicht mehr gültig

```
DIA[i].inIndex && strlen(DIA[i].Data[ICD_VALID_TO]) > 0 && MB->dateV1501.hasYear && MB->dateV1501.Year > atoi(DIA[i].Data[ICD_VALID_TO])
```

C2320 [E] ICD-Kode zum Austrittszeitpunkt noch nicht gültig

```
DIA[i].inIndex && strlen(DIA[i].Data[ICD_VALID_FROM]) > 0 && MB->dateV1501.hasYear && MB->dateV1501.Year < atoi(DIA[i].Data[ICD_VALID_FROM])
```

C.3. Psychiatrie-Zusatzdaten

C.3.0. Übereinstimmung der konfundierten Merkmale

In den Psychiatrie-Zusatzdaten werden einige Merkmale detaillierter erfragt als im Minimaldatensatz. So besteht beispielsweise bei der Angabe Aufenthalt vor dem Eintritt (1.2.V02, 3.2.V02), die Möglichkeit, die Angabe 1=Zuhause zusätzlich mit 11=Zuhause, allein lebend zu detaillieren. Die erste Position der beiden Merkmale müssen jedoch identisch sein.

Auf diese Weise wird die Kongruenz der Merkmale zwischen dem Grundmodul Psychiatrie und der Kombination von Minimaldatensatz + Psychiatrie-Zusatzdaten gewährleistet. Die betroffenen Variablen sind: Aufenthalt vor (1.2.V02 und 3.2.V02), einweisende Instanz (1.2.V04 und 3.3.V01), Entscheid für Austritt (1.5.V02 und 3.5.V01), Aufenthalt nach (1.5.V03 und 3.5.V02), Behandlung nach (1.5.V04 und 3.5.V03).

C3000 [E] Aufenthalt vor: Differenzen in Minimaldaten und Psychiatrie-Zusatz

```
strlen(V3202) > 0 && strcmp(MB->V0202,'A') == 0 && V3202[0] != MB->V1202[0]
```

C3010 [E] Einweisende Instanz: Differenzen in Minimaldaten und Psychiatrie-Zusatz

```
strlen(V3301) > 0 && strcmp(MB->V0202,'A') == 0 && V3301[0] != MB->V1204[0]
```

C3020 [E] Entscheid für Austritt: Differenzen in Minimaldaten und Psychiatrie-Zusatz

```
strlen(V3501) > 0 && strcmp(MB->V0202,'A') == 0 && V3501[0] != MB->V1502[0]
```

C3030 [E] Aufenthalt nach: Differenzen in Minimaldaten und Psychiatrie-Zusatz

```
strlen(V3502) > 0 && strcmp(MB->V0202,'A') == 0 && V3502[0] != MB->V1503[0]
```

C3040 [E] Behandlung nach: Differenzen in Minimaldaten und Psychiatrie-Zusatz

```
strlen(V3503) > 0 && strcmp(MB->V0202,'A') == 0 && V3503[0] != MB->V1504[0]
```

C3100 [H] Psychiatrie-Zusatzdaten / Hauptkostenstelle: Angabe wird in der Regel auf der Abteilung Psychiatrie (M500) erwartet

```
strcmp(MB->V1401,'M500') != 0
```

C.4. Patientengruppen-Zusatzdaten

C.4.0. Vergleich der übereinstimmenden Merkmale

Die Medizinische Statistik wurde ab 1.1.2009 angepasst an die Bedürfnisse einer leistungsorientierten Spitalfinanzierung. Um in der Schweiz die notwendige Kompatibilität zum gewählten deutschen Modell zu erreichen, müssen detailliertere Informationen als bisher erhoben werden. In dem neu gebildeten Patientengruppendatensatz können bis zu 50 Diagnosen und bis 100 Behandlungen erhoben werden. Zudem erhält er weitere für Abrechnungszwecke relevante Daten.

C4010 [W] Die Diagnose im MB muss der Diagnose im MD entsprechen

```
strcmp(MB->V0202,'A') == 0 && strcmp(MB->DIA[i].Key,DIA[i].Key) != 0
```

C4020 [W] Die Behandlung im MB muss der Behandlung im MD entsprechen

```
strcmp(MB->V0202,'A') == 0 && strncmp(BEH[i].Key,MB->BEH[i].Key,5) != 0 && (!MB->BEH[i].inIndex || strncmp(MB->BEH[i].Key,BEH[i].Key,strlen(MB->BEH[i].Key)))
```

D. Plausibilisierung der Diagnose- und Behandlungsfelder

D.0. Allgemeines

Für die Diagnosekodes (ICD-10) ist momentan die Prüfung hinsichtlich Geschlecht, Alter, Kreuz-Stern Kodierung, äussere Ursachen möglich.

Im Bereich der Operationen (CHOP) ist eine Geschlechtsplausibilisierung und eine CHOP-Versionskontrolle der Kodes möglich.

Für die Plausibilisierung der Diagnose- und der Operationskodes stehen Metadaten zur Verfügung, mit deren Hilfe die Angaben in den Minimaldatensätzen geprüft werden kann.

Die Übermittlung der ICD- und CHOP-Kodes im Minimaldatensatz erfolgt ohne Punkte. Dabei ist insbesondere bei den CHOP-Kodes zu beachten, dass die führenden Nullen beibehalten werden.

D.1. Tests auf den Diagnosefeldern (ICD-10)

D.1.1. Gültigkeitstests

Bei der Gültigkeitsprüfung der Diagnosekodes (ICD-10) werden folgende Tests durchgeführt:

- Überprüfung auf eventuell vorhandene Punkte auf der 4. Position in den Kodes
- Existenzprüfung anhand einer Tabelle von allen für die Kodierungen zulässigen drei- und vierstelligen ICD-Kodes
- Da prinzipiell eine vierstellige Kodierung vorgeschrieben ist, werden dreistellige Kodes dort, wo in der ICD-10 eine vierstellige Angabe vorgesehen ist, als Fehler zurückgewiesen. Die Kodes aus Kapitel XX (äussere Ursachen) bilden hier eine Ausnahme. Sie lösen lediglich Warnungen aus, weil die 4. und 5. Stelle nicht immer klar definiert ist bezüglich den Ort oder die Tätigkeit.
- Hinweis bei Erweiterungen auf der 4. und 5. Stelle, die in der offiziellen Schweizer Version der ICD nicht vorgesehen sind.

D1110 [E] Ungültiger Diagnosekode (ICD-10)

```
!DIA[i].isNull && !DIA[i].inIndex
```

D1140 [E] Nicht terminaler ICD 10-Kode

```
DIA[i].inIndex && StatYear > 2009 && StatYear >= atoi(DIA[i].Data[ICD_VALID_FROM]) && StatYear <=
atoi(DIA[i].Data[ICD_VALID_TO]) && (StatYear < atoi(DIA[i].Data[ICD_TERM_FROM]) || StatYear >
atoi(DIA[i].Data[ICD_TERM_TO]))
```

C1140 [E] Nicht terminaler CHOP-Kode

```
BEH[i].inIndex && StatYear > 2009 && StatYear >= atoi(BEH[i].Data[CHOP_VALID_FROM]) && StatYear <=
atoi(BEH[i].Data[CHOP_VALID_TO]) && (StatYear < atoi(BEH[i].Data[CHOP_TERM_FROM]) || StatYear >
atoi(BEH[i].Data[CHOP_TERM_TO]))
```

D1190 [E] Die Diagnosecodes müssen nacheinander aufgelistet werden. Es dürfen innerhalb der übermittelten Codes keine leeren ICD-Felder ('Lücken') enthalten sein.

```
i > 0 && !DIA[i].isNull && (( i < 3 && DIA[0].isNull) || (i > 2 && DIA[i-1].isNull))
```

D.1.2. Kreuz-Stern-Validierungen

Stern Kreuz-Kodes werden in der ICD benutzt, um einerseits die Ätiologie und andererseits die lokale Manifestation einer Krankheit zu bezeichnen. Das eine geschieht mittels eines Kreuz-Kodes, das andere mit einem Stern-Kode.

Der Kreuz-Kode wird immer in der Hauptdiagnose angegeben, der Sternkode im Hauptdiagnose-Zusatz. Die Kodes werden in den Datenfeldern nicht speziell als Kreuz- oder Stern-Kodes gekennzeichnet.

Wird ein Kreuz/Stern-Kodepaar in die Nebendiagnosen gesetzt, so ist zuerst der Kreuzkode anzugeben und im unmittelbar anschliessenden Feld der Sternkode. Daher kann in der 1. Nebendiagnose kein Sternkode und in der letzten Nebendiagnose kein Kreuzkode stehen.

Es kann unterschieden werden zwischen den folgenden Stern-Kreuz Typen:

- Absolute, (1:1) Kreuz-Stern Verbindung: Für einen bestimmten Kreuzkode existiert nur ein einziger obligatorischer Sternkode. Für diese Codes erfolgt eine harte Prüfung (50 Kodepaare in der ICD-10).
- Absoluter (n:1) Sternkode: Dieser Sternkode verlangt nach einem Kreuzkode. Der Sternkode kann nicht in der Hauptdiagnose auftreten.
- Absoluter (1:n) Kreuzkode: Dieser Kreuzkode verlangt nach einem Sternkode. Der Kreuzkode kann nicht im Hauptdiagnose-Zusatz auftreten.

D1210 [E] Sternkode nicht zulässig als Hauptdiagnose

```
DIA[0].inIndex && strcmp(DIA[0].Data[ICD_TYP], '2') == 0
```

D1220 [E] Kreuzkode nicht zulässig als Hauptdiagnose-Zusatz

```
DIA[1].inIndex && (strcmp(DIA[1].Data[ICD_TYP], '1') == 0 || strcmp(DIA[1].Data[ICD_TYP], '4') == 0)
```

D1222 [E] Kode nicht zulässig als Hauptdiagnose-Zusatz

```
DIA[1].inIndex && strcmp(DIA[1].Data[ICD_TYP], '1') != 0 && strcmp(DIA[1].Data[ICD_TYP], '2') != 0 && strcmp(DIA[1].Data[ICD_TYP], '4') != 0 && !(DIA[1].Key[0] == 'V' || DIA[1].Key[0] == 'W' || DIA[1].Key[0] == 'X' || DIA[1].Key[0] == 'Y')
```

D.1.3. Geschlechtsspezifische Codes

Bestimmte Diagnosekodes können grundsätzlich nur bei Männern und bestimmte andere nur bei Frauen angegeben werden.

Im ICD-10 Band 2 sind die folgenden geschlechtsspezifischen Diagnosen angegeben:

Geschlecht	Kodes
männlich	B26.0, C60-C63, D07.4-D07.6, D17.6, D29.-, D40.-, E29.-, E89.5, F52.4, I86.1, L29.1, N40-N51, Q53-Q55, R86, S31.2-S31.3, Z12.5
weiblich	A34, B37.3, C51-C58, C79.6, D06.-, D07.0-D07.3, D25-D28, D39.-, E28.-, E89.4, F52.5, F53.-, I86.3, L29.2, L70.5, M80.0-M80.1, M81.0-M81.1, M83.0, N70-N98, N99.2-N99.3, O00-O99, P54.6, Q50-Q52, R87, S31.4, S37.4-S37.6, T19.2-T19.3, T83.3, Y76.-, Z01.4, Z12.4, Z30.1, Z30.3, Z30.5, Z31.1, Z31.2, Z32-Z36, Z39, Z43.7, Z87.5, Z97.5

D1300 [W] Diagnosekode nicht vereinbar mit Geschlecht

```
DIA[i].inIndex && strlen(DIA[i].Data[ICD_SEX]) > 0 && strcmp(DIA[i].Data[ICD_SEX], MB->V1101) != 0
```

D.1.4. Altersspezifische Codes

Bestimmte Diagnosekodes können grundsätzlich nur in bestimmten Altersbereichen angegeben werden.

D1401 [W] ICD-Kode unvereinbar mit Alter für a) Säugling (hauptsächlich Neugeborene, jedoch auch Patienten zwischen 1 und 364/5 Tagen)

`DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'a') == 0 && MB->AGED > 365`

D1402 [W] ICD-Kode unvereinbar mit Alter für b) Kleinkind (Alter zwischen 1 Tag und 2 Jahren)

`DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'b') == 0 && (MB->AGEJ >= 2 || MB->AGED == 0)`

D1403 [W] ICD-Kode unvereinbar mit Alter für c) Kind (Alter zwischen 1 Tag und 10 Jahren)

`DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'c') == 0 && (MB->AGEJ >= 10 || MB->AGED == 0)`

D1404 [W] ICD-Kode unvereinbar mit Alter für d) Kinder und Jugendliche (Alter zwischen 1 Tag und 19 Jahren)

`DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'd') == 0 && (MB->AGEJ >= 19 || MB->AGED == 0)`

D1405 [W] ICD-Kode unvereinbar mit Alter für e) Post-neonatal (älter als 27 Tage)

`DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'e') == 0 && MB->AGED < 28`

D1406 [W] ICD-Kode unvereinbar mit Alter für f) Älter als Säugling (Alter ab 1 Jahr)

`DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'f') == 0 && MB->AGEJ < 1`

D1407 [W] ICD-Kode unvereinbar mit Alter für g) Pubertät (Alter zwischen 8 und 19 Jahren)

`DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'g') == 0 && (MB->AGEJ < 8 || MB->AGEJ > 19)`

D1408 [W] ICD-Kode unvereinbar mit Alter für h) Gebärfähiges Alter (Patientinnen zwischen 12 und 55 Jahren)

`DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'h') == 0 && (MB->AGEJ < 12 || MB->AGEJ > 55)`

D1409 [W] CHOP-Kode unvereinbar mit Alter für i) Gebärfähiges Alter (Patientinnen zwischen 12 und 55 Jahren)

`BEH[i].inIndex && strcmp(BEH[i].Data[CHOP_AGEGRP], 'i') == 0 && (MB->AGEJ < 12 || MB->AGEJ > 55)`

D1411 [W] ICD-Kode unvereinbar mit Alter für j) Erwachsene (ab 15 Jahren)

`DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'j') == 0 && MB->AGEJ < 15`

D1412 [W] ICD-Kode unvereinbar mit Alter für k) Reife Erwachsene (ab 30 Jahren)

`DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'k') == 0 && MB->AGEJ < 30`

D1420 [W] Kode nicht vereinbar mit Alter (nur für Peri-/Neonatalperiode verwendbar)

`DIA[i].inIndex && DIA[i].Key[0] == 'P' && MB->dateV1201.DateValue - MB->dateV1102.DateValue > 28 && (DIA[i].Key[1] == '1' || isSubKeyIn(DIA[i].Key, 1, 2, 16, _16_DIA_PERINAT_LEN2) || isSubKeyIn(DIA[i].Key, 1, 3, 10, _10_DIA_PERINAT_LEN3))`

D.1.5. Kodes aus ICD-Kapitel XIX und XX

Die Kodes des Kapitels XIX bezeichnen Verletzungen und Vergiftungen aufgrund von äusseren Ursachen. Die Angabe der äusseren Ursache (ein Kode aus Kapitel XX) soll im Hauptdiagnose-Zusatz gemacht werden.

Da es sich um äussere Ursachen handelt, sind demzufolge sämtliche Kodes (V01-Y98) aus Kapitel XX nicht zulässig als Hauptdiagnose. Sie gehören in den Hauptdiagnose-Zusatz und in die Hauptdiagnose wird der Kode gesetzt, der die Verletzung oder Vergiftung beschreibt.

D1500 [E] Hauptdiagnose: Äussere Ursachen (ICD-10 Kap. XX) hier nicht zulässig

```
!DIA[0].isNull && (DIA[0].Key[0] == 'V' || DIA[0].Key[0] == 'W' || DIA[0].Key[0] == 'X' || DIA[0].Key[0] == 'Y')
```

D1652 [W] Das Behandlungsdatum im MD-Record entspricht nicht dem Behandlungsdatum im MB

```
MB->hasMD && strcmp(MB->V1702, MD->BEH[0].Datum) != 0
```

D.1.7. Letale Diagnosen und Austrittsart

Einige Diagnosen implizieren den Tod eines Patienten. Folgedessen kann die Austrittsart bei Vergabe dieser Diagnosen nur 5 (Todesfall) sein.

D1700 [W] Diagnose / Austrittsart: Letale Hauptdiagnose, als Austrittsart nur Tod möglich

```
!DIA[i].isNull && strcmp(MB->V1502,'5') != 0 && (strcmp(DIA[i].Key,'R95',3) == 0 || strcmp(DIA[i].Key,'R96',3) == 0 || strcmp(DIA[i].Key,'R98',3) == 0 || strcmp(DIA[i].Key,'R99',3) == 0 || strcmp(DIA[i].Key,'P95',3) == 0 || strcmp(DIA[i].Key,'O95',3) == 0 || strcmp(DIA[i].Key,'O96',3) == 0 || strcmp(DIA[i].Key,'O97',3) == 0 || strcmp(DIA[i].Key,'I461',4) == 0 || strcmp(DIA[i].Key,'P964',4) == 0 || strcmp(DIA[i].Key,'S18',3) == 0)
```

D.1.8. Codierung von Neugeborenen

In den Diagnosefeldern von neugeborenen Kindern sollte immer ein Code Z38.- angegeben werden, der das Resultat der Geburt dokumentiert. Der Code kann sowohl als Hauptdiagnose erscheinen oder, im Falle von Komplikationen, in den Nebendiagnosen.

Im Falle einer Totgeburt (Vitalstatus V2201 = 0) darf jedoch keine Z38.- Diagnose angegeben werden.

D1800 [E] Lebendgeborene müssen mit dem Diagnosekode Z38 kodiert werden

```
strcmp(MB->V0202,'A') == 0 && MB->hasMN && strcmp(MB->V1203,'3') == 0 && strcmp(MB->MN->V2201,'1') == 0 && !hasDiaLike('Z38')
```

D1810 [E] Vitalstatus (Totgeburt) / Diagnosen: Widersprüchliche Dokumentation eines Lebendgeborenen mit einem Diagnosecode Z38.-

```
DIA[i].inIndex && strcmp(DIA[i].Key,'Z38',3) == 0 && MB->hasMN && strcmp(MB->MN->V2201,'0') == 0
```

D1900 [E] Zusatz zur Hauptdiagnose: nur für Stern-Kodes und für äussere Ursachen (V01!-Y98!) gültig

```
DIA[1].inIndex && (strcmp(DIA[1].Data[ICD_TYP],'1') == 0 || strcmp(DIA[1].Data[ICD_TYP],'4') == 0)
```

D1901 [E] Kodes mit !: immer als Nebendiagnose, ausser die äusseren Ursachen (V01!-Y98!)

```
DIA[0].inIndex && isExkIDia(DIA[0].Key)
```

D1902 [E] Die Kodes R75, Z21, B23.0 und die Gruppe B20-B24 schliessen sich gegenseitig aus und sind während desselben stationären Aufenthaltes nicht zusammen aufzuführen

DIA[i].inIndex && hasB2024 && (strcmp(DIA[i].Key,'R75',3) == 0 || strcmp(DIA[i].Key,'Z21',3) == 0 || strcmp(DIA[i].Key,'B230',4) == 0)

D1905 [E] Der Z37-Code muss im Datensatz der Mutter, nicht in demjenigen des Kindes kodiert werden

strlen(MB->V1103) > 0 && DIA[i].inIndex && strcmp(DIA[i].Key,'Z37',3) == 0 && atoi(MB->V1103) == 0

D1906 [E] Der Z38-Code muss im Datensatz des Kindes, nicht in demjenigen der Mutter kodiert werden

strlen(MB->V1103) > 0 && DIA[i].inIndex && strcmp(DIA[i].Key,'Z38',3) == 0 && atoi(MB->V1103)>0

D1907 [W] Wird einer der folgenden Codes als Haupt- oder Nebendiagnose angegeben, muss der Code O09.-! als Nebendiagnose angegeben werden: O03-O07; O10-O16; O20-O29; O30-O48; O60-O75; O80-O82; Z34-Z37

DIA[i].inIndex && isSSDia(DIA[i].Key) && !hasO09

D1909 [E] Wenn der Code O80 - O82 steht, muss Z37 kodiert werden und gehört in die Nebendiagnose

DIA[i].inIndex && (strcmp(DIA[i].Key,'O80',3) == 0 || strcmp(DIA[i].Key,'O81',3) == 0 || strcmp(DIA[i].Key,'O82',3) == 0) && !hasZ37

D1910 [E] Die Codes O80, O81, O82 sind Hauptdiagnose-Codes.

i > 0 && DIA[i].inIndex && (strcmp(DIA[i].Key,'O80',3) == 0 || strcmp(DIA[i].Key,'O81',3) == 0 || strcmp(DIA[i].Key,'O82',3) == 0)

D1920 [E] Unzulässiger Hauptdiagnosecode

DIA[0].inIndex && strcmp(DIA[0].Data[ICD_VALIDMD], '2') == 0

D1930 [W] Die Diagnosen Z51.0, Z51.1, Z51.82 werden nicht kodiert.

DIA[i].inIndex && (strcmp(DIA[i].Key,'Z510') == 0 || strcmp(DIA[i].Key,'Z511') == 0 || strcmp(DIA[i].Key,'Z5182') == 0)

D1940 [W] Hauptdiagnosecodes T90 - T98 benötigen keine Zusatzdiagnose V - Y mehr.

DIA[0].inIndex && strcmp(DIA[0].Key,'T9',2) == 0 && DIA[1].inIndex && (strcmp(DIA[1].Key,'V',1) == 0 || strcmp(DIA[1].Key,'W',1) == 0 || strcmp(DIA[1].Key,'X',1) == 0 || strcmp(DIA[1].Key,'Y',1) == 0)

D1950 [W] S-/T-Hauptdiagnosecodes benötigen keine Zusatzdiagnose V - Y mehr, wenn einer der folgenden Codes Nebendiagnose ist: Z42, Z44, Z47, Z48, Z50, Z51, Z54.

i > 1 && DIA[i].inIndex && DIA[0].inIndex && DIA[1].inIndex && (strcmp(DIA[0].Key,'S',1) == 0 || strcmp(DIA[0].Key,'T',1) == 0) && (strcmp(DIA[1].Key,'V',1) == 0 || strcmp(DIA[1].Key,'W',1) == 0 || strcmp(DIA[1].Key,'X',1) == 0 || strcmp(DIA[1].Key,'Y',1) == 0) && hasKeyLikeLen(DIA[i].Key,3,'Z42|Z44|Z47|Z48|Z50|Z51|Z54|')

D1960 [E] Der Fall eines Neugeborenen mit dem alleinigen Diagnose-Code Z768 darf nicht fakturiert werden.

MB->hasMN && MB->hasMD && strcmp(MB->MD->V4801,'1') == 0 && strcmp(DIA[0].Key,'Z768') == 0 && DIA[1].isNull && DIA[2].isNull && BEH[0].isNull

D1970 [E] Ein Fall mit Hauptdiagnose Z758 darf nicht fakturiert werden.

`DIA[0].inIndex && strcmp(DIA[0].Key,'Z758') == 0 && MB->hasMD && strcmp(MB->MD->V4801,'1') == 0`

D1980 [E] Ein Fall mit Hauptdiagnose Z758 muss Eintrittsart 4 Interner Übertritt haben.

`DIA[0].inIndex && strcmp(DIA[0].Key,'Z758') == 0 && strcmp(MB->V1203,'4') != 0`

D.2. Tests auf den Behandlungsfeldern (CHOP)

D.2.1. Gültigkeitstests

Bei der Gültigkeitsprüfung der Operationskodes (CHOP) werden folgende Tests durchgeführt:

- Überprüfung auf eventuell vorhandene Punkte auf der 3. Position in den Codes.
- Existenzprüfung anhand einer Tabelle von allen für die Kodierungen zulässigen 3- und 4-stelligen CHOP-Kodes (rund 4300 Einträge).
- Rückweisung von unspezifischen 2- und 3-stelligen Codes, welche in der CHOP exakter zur Verfügung stehen und dementsprechend drei- oder vierstellig angegeben werden sollen.
- Hinweis bei der Verwendung der Angabe 99.99 - sonstige diverse Massnahmen, welche wenn möglich genauer angegeben werden sollten.
- Warnung bei der Übermittlung von erweiterten Codes, die jedoch bis auf die 3. oder 4. Stelle korrekt sind.

D2100 [E] Ungültiger Behandlungskode (CHOP)

`!BEH[i].isNull && !BEH[i].inIndex`

D2110 [E] Unspezifischer 2-stelliger Behandlungskode (CHOP)

`!BEH[i].inIndex && strcmp(BEH[i].Data[CHOP_TYP],'2') == 0`

D2120 [E] Unspezifischer 3-stelliger Behandlungskode (CHOP)

`BEH[i].inIndex && strcmp(BEH[i].Data[CHOP_TYP],'3') == 0`

D2130 [H] 4-stellige Restkategorie (99.99) genauer spezifizieren

`strcmp(BEH[i].Key,'9999') == 0`

D2140 [H] Der CHOP-Kode im MB-Datensatz darf nicht länger als 5 Zeichen sein

`strlen(MB->BEH[i].Key) > 5`

D2190 [E] Die Behandlungscodes müssen nacheinander aufgelistet werden. Es dürfen innerhalb der übermittelten Codes keine leeren CHOP-Felder ('Lücken') enthalten sein.

`i > 0 && !BEH[i].isNull && BEH[i-1].isNull`

D.2.2. Geschlechts- und altersspezifische Codes

Bestimmte Behandlungskodes können grundsätzlich nur bei Männern und bestimmte andere nur bei Frauen angegeben werden. Andere hingegen machen nicht bei jedem Alter Sinn.

D2200 [W] Behandlungskode nicht vereinbar mit Geschlecht

`BEH[i].inIndex && strlen(BEH[i].Data[CHOP_SEX]) > 0 && strcmp(BEH[i].Data[CHOP_SEX],MB->V1101) != 0 && !hasF64`

D2210 [E] Behandlungscode aus Kapitel 13 unvereinbar mit Geschlecht (Mann)

```
strcmp(MB->V0202,'A') == 0 && BEH[i].inIndex && strcmp(MB->V1101,'1') == 0 && strncmp(BEH[i].Key,'72',2) >= 0 && strncmp(BEH[i].Key,'75',2) <= 0
```

D2250 [W] Behandlungscode aus Kapitel 13 unvereinbar mit Alter (< 12 Jahre)

```
strcmp(MB->V0202,'A') == 0 && BEH[i].inIndex && atoi(MB->V1103) < 12 && strncmp(BEH[i].Key,'72',2) >= 0 && strncmp(BEH[i].Key,'75',2) <= 0
```

D.2.3. CHOP Versionskontrolle

Die CHOP-Klassifikation wird jährlich revidiert. Dabei fallen einige Codes weg, andere kommen dazu und ersetzen eventuell ältere.

D2301 [E] Ungültiger CHOP-Code (CHOP10 oder früher)

```
!BEH[i].isNull && StatYear < 2009 && strcmp(BEH[i].Data[CHOP_TERM4],'0') == 0
```

D2302 [E] Ungültiger CHOP11-Code

```
!BEH[i].isNull && !BEH[i].inIndex)
```

D2310 [E] CHOP-Code zum Austrittszeitpunkt nicht mehr gültig

```
BEH[i].inIndex && strlen(BEH[i].Data[CHOP_VALID_TO]) > 0 && MB->dateV1501.hasYear && MB->dateV1501.Year > atoi(BEH[i].Data[CHOP_VALID_TO])
```

D2320 [E] CHOP-Code zum Austrittszeitpunkt noch nicht gültig

```
BEH[i].inIndex && strlen(BEH[i].Data[CHOP_VALID_FROM]) > 0 && MB->dateV1501.hasYear && MB->dateV1501.Year < atoi(BEH[i].Data[CHOP_VALID_FROM])
```

D.2.4. CHOP - Kode 89.81 - Autopsie und Austrittsart

Der CHOP Kode 89.81 bezeichnet eine Autopsie und kann demzufolge nur bei einem Todesfall (= 5) angegeben werden.

D2410 [W] Behandlungskode nur bei Todesfall möglich

```
strcmp(MB->V1502,'5') != 0 && strncmp(BEH[i].Key,'8981',4) == 0
```

D2500 [E] Beginn der Hauptbehandlung MD muss dem Beginn der Hauptbehandlung MB entsprechen

```
strcmp(MB->V0202,'A') == 0 && strcmp(BEH[0].dateDatum.Value,MB->V1702) != 0
```

E. Indikatoren zur Überwachung der Datenqualität

Angesichts der Komplexität und Vieldeutigkeit des Datenmaterials und der gestiegenen Anforderungen an die inhaltliche Korrektheit der Angaben in den Minimaldatensätzen wurde ein Hilfsmittel eingeführt, mit dem die Datenqualität der Lieferungen beurteilt werden kann. Das Resultat sind verschiedene Indikatoren, die sich als aussagekräftig erwiesen haben bei der Beurteilung der Qualität der Daten.

Einerseits werden bestimmte Kennzahlen berechnet (mittlere Aufenthaltsdauer, Durchschnittsalter), andererseits wird der Anteil an relativ unpräzisen Angaben und Restkategorien (wie andere und unbekannt) in den Variablen ausgezählt. Ausgehend von den medizinischen Informationen können Analysen gemacht werden bezüglich die Vollständigkeit, Gültigkeit (Validität), Präzision, Plausibilität, Korrektheit, die Vollständigkeit und die Verlässlichkeit der Kodierung, indem die Raten von bestimmten wichtigen Krankheiten und Eingriffen ausgewiesen werden.

Die Indikatoren werden nur für die Statistik Typ A-Fälle ausgewertet und für nichtambulante Fälle.

E.0. Hilfsmerkmale

E.0.0. Berechnung der Anzahl Fälle (Statistiktyp A)

Die Anzahl Fälle vom Statistiktyp A werden berechnet. Es handelt sich um ein Hilfsmerkmal, das für die Berechnung der prozentualen Angaben und der Mittelwerte pro Fall in den folgenden Indikatoren benutzt wird.

E0000 [C] Anzahl Fälle (Statistiktyp A): Austritte während der Beobachtungsperiode

```
strcmp(V0202,'A') == 0
```

E0001 [C] Anzahl Fälle (Statistiktyp B): Eintritte während des Jahres und am 31.12 hospitalisiert

```
strcmp(V0202,'B') == 0 && strcmp(V1301,'3') == 0
```

E0002 [C] Anzahl Fälle (Statistiktyp C): Langzeitaufenthalte

```
strcmp(V0202,'C') == 0 && strcmp(V1301,'3') == 0
```

E0003 [C] Anzahl Austritte (stationäre A-Fälle)

```
strcmp(V0202,'A') == 0 && strcmp(V1301,'3') == 0
```

E0004 [C] Ambulanter Fall

```
strcmp(V1301,'1') == 0
```

```
ACTION: doWriteRec=FALSE;
```

E.1. Indikatoren zu Patientenmerkmalen

E.1.0. Analyse nach Geschlecht

Der Anteil von weiblichen Patienten wird ausgezählt.

E1000 [C] Zusammensetzung nach Geschlecht: Anzahl weibliche Patienten

```
strcmp(V0202,'A') == 0 && strcmp(V1101,'2') == 0
```

E.1.1. Analyse nach Alter

Das Durchschnittsalter der Patientinnen und Patienten wird berechnet durch aufsummieren aller Altersangaben und Division durch die Anzahl Fälle.

E1100 [C] Zusammensetzung nach Alter (Statistiktyp A): Mittelwert Alter

```
strcmp(V0202,'A') == 0
```

E.2. Indikatoren zu Aufenthaltsmerkmalen

E.2.0. Analyse der Angaben zum Aufenthalt

In der Variable Behandlungsart wird der Anteil von teilstationären Patienten analysiert. Bei den Angaben Liegeklasse (1.3V02) und Hauptkostenträger (1.4V02) werden die Fälle mit der Angabe unbekannt (9) oder andere (8) ausgeschieden. Schliesslich wird die Aufenthaltsdauer aufsummiert zur Berechnung der mittleren Aufenthaltsdauer.

E2000 [C] Zusammensetzung nach Behandlungsart: Anzahl teilstationäre Patienten

```
strcmp(V0202,'A') == 0 && strcmp(V1301,'2') == 0
```

E2001 [C] Liegeklasse 9 unbekannt (unpräzise Angabe)

```
strcmp(V0202,'A') == 0 && strcmp(V1302,'9') == 0
```

E2002 [C] Hauptkostenträger 9 unbekannt (unpräzise Angabe)

```
strcmp(V0202,'A') == 0 && strcmp(V1402,'9') == 0
```

E2003 [C] Hauptkostenträger 8 andere (unpräzise Angabe)

```
strcmp(V0202,'A') == 0 && strcmp(V1402,'8') == 0
```

E2010 [C] Zusammensetzung nach Aufenthaltsdauer: Mittlere stat. Aufenthaltsdauer

```
strcmp(V0202,'A') == 0 && strcmp(V1301,'3') == 0
```

E.2.1. Analyse der Eintrittsangaben

Der Anteil von unpräzisen Angaben in den Variablen Aufenthalt vor Eintritt (1.2V02), Eintrittsart (1.2V03), Einweisende Instanz (1.2V04) und Behandlungsart (1.3V01) wird auf die Werte unbekannt (9) oder andere (8) untersucht.

E2101 [C] Aufenthalt vor Eintritt 9 unbekannt (unpräzise Angabe)

```
strcmp(V0202,'A') == 0 && strcmp(V1202,'9') == 0
```

E2102 [C] Aufenthalt vor Eintritt 8 andere (unpräzise Angabe)

```
strcmp(V0202,'A') == 0 && strcmp(V1202,'8') == 0
```

E2103 [C] Eintrittsart 9 unbekannt (unpräzise Angabe)

`strcmp(V0202,'A') == 0 && strcmp(V1203,'9') == 0`

E2104 [C] Eintrittsart 8 andere (unpräzise Angabe)

`strcmp(V0202,'A') == 0 && strcmp(V1203,'8') == 0`

E2105 [C] Einweisende Instanz 9 unbekannt (unpräzise Angabe)

`strcmp(V0202,'A') == 0 && strcmp(V1204,'9') == 0`

E2106 [C] Einweisende Instanz 8 andere (unpräzise Angabe)

`strcmp(V0202,'A') == 0 && strcmp(V1204,'8') == 0`

E2107 [C] Behandlungsart 9 unbekannt (unpräzise Angabe)

`strcmp(V0202,'A') == 0 && strcmp(V1301,'9') == 0`

E.2.2. Analyse der Austrittsangaben

Der Anteil von Angaben unbekannt (9), andere (8) und Todesfall (5) in der Variablen Entscheid für Austritt (1.5V02), sowie der Anteil von unpräzisen Angaben in den Variablen Aufenthalt nach Austritt (1.5V03), Behandlung nach Austritt (1.5V04) wird analysiert.

E2200 [C] Entscheid für Austritt: Todesfall

`strcmp(V0202,'A') == 0 && strcmp(V1502,'5') == 0`

E2201 [C] Entscheid für Austritt: 9 unbekannt (unpräzise Angabe)

`strcmp(V0202,'A') == 0 && strcmp(V1502,'9') == 0`

E2202 [C] Entscheid für Austritt: 8 andere (unpräzise Angabe)

`strcmp(V0202,'A') == 0 && strcmp(V1502,'8') == 0`

E2203 [C] Aufenthalt nach Austritt: 9 unbekannt (unpräzise Angabe)

`strcmp(V0202,'A') == 0 && strcmp(V1503,'9') == 0`

E2204 [C] Aufenthalt nach Austritt: 8 andere (unpräzise Angabe)

`strcmp(V0202,'A') == 0 && strcmp(V1503,'8') == 0`

E2205 [C] Behandlung nach Austritt: 9 unbekannt (unpräzise Angabe)

`strcmp(V0202,'A') == 0 && strcmp(V1504,'9') == 0`

E2206 [C] Behandlung nach Austritt: 8 andere (unpräzise Angabe)

`strcmp(V0202,'A') == 0 && strcmp(V1504,'8') == 0`

E.3. Indikatoren betreffend die medizinische Kodierung

Vollständigkeit und Anzahl der medizinischen Codes (Diagnosen und Behandlungen). Anzahl Codes pro Fall, Diversivität der Kodierung, d.h. die Anzahl verschiedenartige Codes.

E.3.0. Analyse der Diagnosekodes

Ein wichtiges Merkmal sind die Diagnosekodes pro Fall, die Diversität der verwendeten Diagnosekodes, der Anteil an korrekten Diagnosen mit vollständigem Detailgrad. Zusätzlich werden Diagnosekodes analysiert, die Angaben von Komplikationen medizinischer Behandlungen, Chemotherapien, Strahlentherapien, Nachkontrollen und Tumoren bezeichnen. Als letzte Gruppe werden sehr ungenaue und ungenaue Codes gemäss ihrer Bedeutung ausgewiesen.

E3000 [C] Hauptdiagnose: Angabe vorhanden

```
strcmp(MB->V0202,'A') == 0 && !DIA[0].isNull
```

E3001 [C] Diagnose: vorhanden

```
strcmp(MB->V0202,'A') == 0 && !DIA[i].isNull
```

E3101 [C] Behandlung: vorhanden

```
strcmp(MB->V0202,'A') == 0 && !BEH[j].isNull
```

E3002 [C] Zusatz zur Hauptdiagnose: Angabe vorhanden

```
strcmp(MB->V0202,'A') == 0 && !DIA[1].isNull
```

E3003 [C] Nebendiagnose: Angabe vorhanden

```
i > 1 && strcmp(MB->V0202,'A') == 0 && !DIA[i].isNull
```

E3006 [C] Terminaler ICD GM-Code

```
strcmp(MB->V0202,'A') == 0 && DIA[i].inIndex && strcmp(DIA[i].Data[ICD_TERMGM],'1') == 0
```

E3010 [C] Hauptdiagnose: Vollständiger Detailgrad

```
strcmp(MB->V0202,'A') == 0 && DIA[i].inIndex && strlen(DIA[i].Data[ICD_QUA]) > 0
```

E3015 [C] Kode nicht vereinbar mit Alter (unter- oder oberhalb des zulässigen Bereiches)

```
DIA[i].inIndex && ((strlen(DIA[i].Data[ICD_MINAGE]) > 0 && atol(MB->V1103) < atol(DIA[i].Data[ICD_MINAGE]))  
|| (strlen(DIA[i].Data[ICD_MAXAGE]) > 0 && atol(MB->V1103) > atol(DIA[i].Data[ICD_MAXAGE])))
```

E3020 [H] Meldepflichtige Diagnose

```
DIA[i].inIndex && (hasKeyLikeLen(DIA[i].Key,4,'A392|A393|A394|A395|A398|A399|A403|A413|A481|A482|A491|A492|A810|  
A841|B160|B161|B162|B169|B180|B181|B182|G000|G001') || hasKeyLikeLen(DIA[i].Key,3,'J09|J13|J14'))
```

E.3.1. Analyse der Behandlungskodes

Auch bei den Behandlungskodes können die Anzahl Codes pro Fall, die Diversität der verwendeten Behandlungskodes und die Detailliertheit analysiert werden. Anschliessend wird die Rate an nicht invasiven (konservativen) Behandlungen untersucht. Sehr unpräzise und unpräzise Behandlungskodes werden unterschieden.

E3100 [C] Hauptbehandlung: Angabe vorhanden

```
strcmp(MB->V0202,'A') == 0 && !BEH[0].isNull
```

E3105 [C] Terminaler 4-stelliger Code (CHOP)

```
strcmp(MB->V0202,'A') == 0 && BEH[i].inIndex && strcmp(BEH[i].Data[CHOP_TERM4],'1') == 0
```

E3106 [C] Terminaler 6-stelliger Code (CHOP)

```
strcmp(MB->V0202,'A') == 0 && BEH[i].inIndex && strcmp(BEH[i].Data[CHOP_TERM6],'1') == 0
```

E3110 [C] Hauptbehandlung: Vollständiger Detailgrad

```
strcmp(MB->V0202,'A') == 0 && BEH[i].inIndex && strlen(BEH[i].Data[CHOP_QUA]) > 0
```

E3120 [C] Hauptbehandlung: Wenig invasive Behandlungen

```
strcmp(MB->V0202,'A') == 0 && BEH[i].inIndex && strcmp(BEH[i].Data[CHOP_QUA],'7') == 0
```

E3130 [C] Hauptbehandlung: Sehr unpräziser Behandlungscode

```
strcmp(MB->V0202,'A') == 0 && BEH[i].inIndex && strcmp(BEH[i].Data[CHOP_QUA],'9') == 0
```

E3140 [C] Hauptbehandlung: Unpräziser Behandlungscode

```
strcmp(MB->V0202,'A') == 0 && BEH[i].inIndex && strcmp(BEH[i].Data[CHOP_QUA],'10') == 0
```

Anhang A. ICD/CHOP Metadaten-Fileformat

Die Prüfung der Diagnose- und Operationscodes erfolgt mit Hilfe von Metadaten. Es handelt sich dabei um eine Tabelle für die ICD-10 und eine Tabelle für die CHOP.

In den Plausibilitätstests wird mit den Funktionen INDEXCOL(Kode,Col) jeweils auf die entsprechende Kolonne zugegriffen. Die genaue Beschreibung der Funktionen befindet sich in Anhang C.

ICD-10:

1. Kode	2. KodeTyp	3. Komplementärkode	4. Sex	5. AlterMin	6. AlterMax
Liste der 3-/4-stelligen ICD-10 GM-Kodes, die für die Tests verwendet werden	0 = Äussers Ursache / ohne Kap. Z 1 = Absoluter + Kode mit * (1:1) im Titel 2 = Absoluter * Kode (1:n) 3 = Anderer, konventioneller Kode 4 = Absoluter + ohne * im Titel 5 = Zusatz aus Kap. XX erforderlich 6 = Unspezifischer 3-stelliger ICD-Kode	Gültige Komplementärkodes zum entsprechenden Stern-/Kreuzkode	Zulässiges Geschlecht für Codevergabe. Leer, falls nicht geschlechtsspezifisch	Zulässige untere Altersgrenze für Vergabe der Codes	Zulässige obere Altersgrenze für Vergabe der Codes

Die Codes des Typs 6 können grösstenteils durch gültige vierstellige Codes mit Endung .9 ersetzt werden. In diesen Fällen erscheint im Feld Komplementärkode der entsprechende vierstellige Kode.

CHOP:

1. Kode	2. KodeTyp	3. Komplementärkode	4. Sex	5. AlterMin
Liste der 2-, 3-, und 4-stelligen CHOP-Kodes.	1 = nicht definiert 2 = unspezifischer 2-Steller 3 = unspezifischer 3-Steller 4 = gültiger 4-Steller 5 = gültiger 3-Steller 6 = gültiger 6-Steller 7 = 5-Steller	Liste der gültigen Komplementärkodes. Zur Zeit keine Einträge.	Zulässiges Geschlecht für Codevergabe. Leer, falls nicht geschlechtsspezifisch 1 = nur Männer 2 = nur Frauen	Noch nicht definiert.

6. AlterMax	7. ICD-10 Hauptdiagnose	8. gültig ab	9. gültig bis	10. Ersatzcode
Noch nicht definiert.	Angabe der Hauptdiagnose, die dieser Behandlung entspricht. Noch nicht definiert.	Jahrzahl, ab der der Kode verwendet werden darf.	Jahrzahl, bis und mit der der Kode verwendet werden darf.	Kode, in den ein ungültiger/revidierter Kode überführt wird.

Anhang B. Beschreibung der in den Tests verwendeten Funktionen

Die in den vorhergehenden Kapiteln beschriebenen Tests verwenden eine Anzahl von Funktionsaufrufen, die im folgenden beschrieben werden. Generell orientieren sich die MedPlaus-Tests an der C++-Syntax

=

Zuweisung var = Wert (is)

V0301='0';

==

entspricht, bei Vergleich (equal)

strlen(V0002) == 0

!=

entspricht nicht, bei Vergleich (not equal)

strlen(V0008) != 0

<

kleiner als (less than)

atoi(V0008) < 3

>

größer als (greater than)

strlen(V0103) > 2

!

nicht (NOT)

!(dateV0004.isNull)

&&

und (AND)

strlen(V0009) != 0 && !isHex(V0009)

||

oder (OR)

!(dateV0004.isNull || dateV0004.isYYYYMMDD)

+

Addition

(MB->STAYH+15)

*

Multiplikation

*(dateV1501.DateValue - dateV1201.DateValue +1) * 24*

/

Division

(atoi(V2302)/10) > 44

for ()

For-Schleife. Führt Anweisung aus, solange Bedingung in For-Schleife erfüllt ist

++ zählt hoch

for (int di=0; di <=DIACount;di++)

->

Verweist auf Record, der Variable beinhaltet („Pointer“)

strcmp(MB->V0202, 'A') == 0

atoi

Wandelt String- in Integerwerte um (StringToInt)

isInt(V1103) && atoi(V1103)

atoI

Wandelt String- in Integerwerte um (StringToInt)

atoI(V0008) < 3

BEH[pos]

Behandlungsfunktionen. Prüfen die Behandlungsvariablen

pos gibt die Position an (0 = HB, 1 = 1. NB, 2 = 2. NB, 3 = 3. NB etc.)

.isNull: Behandlungsfeld leer

.Data[CHOP_]: Attribute des Codes

.Key: Feldinhalt

BEH[0].isNull

datevar

Datumsfunktionen. Prüfen die Datumsvariable

.isNull: Datumsfeld leer

.isYYYYMMDD: Datumsfeld im Format JJJJMMTT

.isYYYY0000: Datumsfeld im Format JJJJ0000

.DateValue: Datumswert

.hasDate: enthält gültiges Datum

.hasYear: enthält gültiges Jahr

.hasMonth: enthält gültigen Monat 1 – 12

.hasHour: enthält gültige Stunde 0 – 23

.hasMinute: enthält gültige Minute 0 - 59

.Hour: Stundenwert

!(dateV0004.isNull)

DIA[pos]

Diagnosefunktionen. Prüfen die Diagnosevariablen

pos gibt die Position an (0 = HD, 1 = ZHD, 2 = 1. ND, 3 = 2. ND etc.)

.isNull: Diagnosefeld leer

.Data[ICD_]: Attribute des Codes

.Key: Feldinhalt

strlen(DIA[0].Key) > 0

DIACount

Zählt die Anzahl Diagnosen eines Records

for (int di=0; di <=DIACount;di++)

doWriteRec

Steuert, ob der Record geschrieben wird

= FALSE (nein)

= TRUE (ja)

doWriteRec=FALSE;

hasDiaLike('DIA')

Prüft, ob Record die angegebene Diagnose enthält, die mit 'DIA' beginnt

hasDiaLike('Z38')

hasMN

Prüft, ob Record einen Neugeborenen-Zusatzdatensatz enthält
hasMN

hasMP

Prüft, ob Record einen Psychiatrie-Zusatzdatensatz enthält
hasMP=FALSE

hasF64

Prüft, ob Record F64-Diagnose enthält
hasF64

hasZ38_147

Prüft, ob Record Z38-Diagnose enthält
hasZ38_147

BOOL hasKeyLikeLen(char* key, int len, char* keylist)

Prüft, ob substr(key,0,len) in der Liste enthalten ist. Die Listenelemente sind mit Pipe abgetrennt.
hasKeyLikeLen(DIA[i].Key,3,'Z42|Z44|Z47|Z48|Z50|Z51|Z54|')

isAlphaNumUpperCase(var)

Prüft, ob der String nur Zeichen der Menge ['0'..'9','A'..'Z'] enthält
isAlphaNumUpperCase(V4601)

isExklDia(DIA[pos].Key)

Prüft das Vorhandensein einer Reihe von spezifischen, festgelegten Diagnosecodes
isExklDia(DIA[0].Key)

isHex(var)

Liefert den booleschen Wert true, falls die Angabe var hexadezimal ist
isHex(V0009)

isIn(var,nr,_list)

Liefert den logischen Wert des Resultats der Suche des Wertes var in einer indizierten Liste (true: gefunden, false: nicht gefunden)
isIn(V0202,3_ABC)

isInt(var)

Liefert den booleschen Wert true, falls die Angabe var eine Ganzzahl ist
isInt(V1103)

isMKCH

Prüft, ob Record ein kantonaler Zusatzdatensatz MK ist
isMKCH $\&\&$ strlen(V5102) == 0

isSSDia(DIA[pos].Key)

Prüft ob die Diagnose eine Schwangerschaftsdiagnose ist: O03-O07; O10-O16; O20-O29; O30-O48; O60-O75; O80-O82; Z34-Z37
isSSDia(DIA[i].Key)

BOOL isSubKeyIn(char *str, int offset, int len, int size, char* array[])

SubKey = substr(str,offset,len). Liefert Wahr, wenn das Array mit size Einträgen einen Wert enthält, der mit SubKey beginnt.

isSubKeyIn(DIA[i].Key,1,2,16,_16_DIA_PERINAT_LEN2)

LineNumberInMX

Zählt die Anzahl der gelieferten Datensätze
LineNumberInMX > atol(MX->V0008)

STAYH

Aufenthaltsdauer in Stunden
(*MB->STAYH+15*)

char * strchr(char * str, int character)

Lokalisiert das erste Auftreten eines bestimmten Zeichens in einer Zeichenkette
strchr('0123456',V2302[2]) == NULL

strcmp(var,'value')

Liefert das Resultat eines Vergleichs von 2 Zeichenketten (StringCompare)
strcmp(V1104,'CHE') == 0

strncmp(var,'value',pos)

Liefert das Resultat eines Vergleichs einer bestimmten Position von 2 Zeichenketten (StringCompare)
strncmp(DIA[0].Key,'Z38',3) != 0

strlen(var)

Liefert die Anzahl Zeichen, die in var enthalten sind (StringLength)
strlen(V0002) == 0

var_list.inIndex

Prüft, ob der Variablenwert in einer vorgegebenen Indexliste enthalten ist
!(V1104_regio.inIndex)

Bemerkung

ACTION

Funktionen hinter der Bezeichnung ACTION: werden nur im speziellen Korrekturmodus von MedPlaus ausgeführt. Und dies auch nur, wenn die spezifische Testbedingung erfüllt ist.
V0301='0';

Anhang C. Implementation des Plausibilisierungskonzeptes in MedPlaus

Die in den vorhergehenden Kapiteln beschriebenen Tests sind im Auftrag des BFS in der Plausibilisierungssoftware MedPlaus implementiert worden.

Datenexporte im BFS-Format (gemäss Schnittstellenkonzept Version 1.2 vom 26.11.1997) / Format 2006 / Format 2009 können mit diesem Hilfsmittel auf ihre Korrektheit überprüft werden.

Die Software generiert Hinweise, Warnungen und Fehlermeldungen. Nach erfolgter Prüfung können drei Typen von Berichten ausgedruckt werden:

- Gesamtübersichten mit absoluten und relativen Fehlerhäufigkeiten
- Listen mit den Datensätzen
- Detailausdrucke der einzelnen Fälle inklusive Fehler und Warnungen

Eine Fehlerfilter-Funktion ermöglicht die Selektionierung von Fällen mit einem bestimmten Fehler.

Das Programm ist zweisprachig (d/f) und enthält ein umfassendes Hilfe-System.

Im weiteren steht eine Windows-DLL Implementation von MedPlaus zur Verfügung, mit der es möglich ist, aus Fremdapplikationen die Plausibilisierung aufzurufen.

MedPlaus steht kostenlos zur Verfügung und kann per Download bezogen werden unter:

www.freudiger.com/medplaus.html