

# Arbeiten mit Flex und Bison

Unser Workshop setzt kaum Linux- und keine C-Kenntnisse voraus. Alle Flex- und Bison-Projekte befinden sich auf unserer Workshopseite bei <http://ddi.ifi.lmu.de/tdi/2016>. Sie sind dort als Zip-Dateien verfügbar. Laden Sie sie jeweils auf Aufforderung der einzelnen Aufgaben hin herunter und entpacken Sie sie in einen eigenen Ordner.

Jeder dieser Ordner enthält ein Makefile, eine Flex-Datei mit namens `parser.l` und eine Bison-Datei mit Name `parser.y`. Der letzte Ordner enthält zudem noch eine Header-Datei `parser.h`. Studieren Sie jeweils die Dateien und insbesondere ihre Kommentare, öffnen Sie dann ein Terminal im entsprechenden Ordner (mit `cd` können Sie in Linux Ordner wechseln), erzeugen Sie mit `make` den jeweiligen Parser, starten Sie ihn mit `./parser` und testen Sie ihn. Im Fehlerfall können Sie ihn mit der Pfeiltaste nach oben leicht noch einmal starten. Wenn Sie den Parser ohne Fehler beenden wollen, drücken Sie `Strg d`.

Einen eigenen Parser erstellen Sie am einfachsten, indem Sie den kompletten Ordner eines anderen Parsers kopieren und in den einzelnen Dateien alles ändern, was geändert werden muss.

1. In den deutschen Städten und Landkreisen folgen die Autokennzeichen im wesentlichen den folgenden Produktionsregeln:

```
<Kennzeichen> → <Landkreis> <Kürzel> <Nummer>.
<Landkreis> → <Buchstabe>|<Buchstabe><Buchstabe>|
<Buchstabe><Buchstabe><Buchstabe>
<Kürzel> → <Buchstabe>|<Buchstabe><Buchstabe>|
<Buchstabe><Buchstabe><Buchstabe>|
<Buchstabe><Buchstabe><Buchstabe><Buchstabe>
<Nummer> → <Ziffer>|<Ziffer><Ziffer>|<Ziffer><Ziffer><Ziffer>|
<Ziffer><Ziffer><Ziffer><Ziffer>
```

Im Ordner `kennzeichen` finden Sie einen passenden Parser. Studieren und testen Sie ihn.

- a) Was ist in Lexer- und Parserdatei anders als in den obigen Regeln? Nennen Sie Gründe.
- b) Bis auf eine Kleinigkeit besteht der Parser nur aus Syntax und nicht aus Semantik. Nennen Sie das einzige semantische Detail.

Im wesentlichen entspricht der so gebaute Parser einem erkennenden Automaten. Erweitern Sie ihn nach folgenden Vorgaben:

- c) Oldtimer haben am Ende des Kennzeichens nach einem Leerraum ein `H`.
- d) Bundeswehrekennzeichen bestehen aus einem `Y`, einen Leerraum und bis zu sechs Ziffern.
- e) Führende Nullen sollen in den Nummern nicht vorkommen.

2. Im Ordner `kennzeichen2` finden Sie eine Erweiterung des Parsers der vorherigen Aufgabe um etliche Semantik.

- a) Wozu dient diese Semantik? Beschreiben Sie ihre Wirkungsweise.
- b) Bauen Sie auch in diesen erweiterten Parser Oldtimer und Bundeswehrekennzeichen ein.

Mit `strcmp (s, t)` kann man in C zwei Zeichenketten auf Gleichheit überprüfen. Bei Gleichheit erhält man als Ergebnis 0. Nutzen Sie das für folgende Anforderung an den Parser:

- c) Viele (vereinfacht: alle) Landkreise wollen `HH` nicht als Kürzel und `88` nicht als Nummer.

3. Der Klassiker unter den Parsern ist der Taschenrechner. Studieren Sie ihn im Ordner `rechner` und erweitern sie ihn um folgende Anforderungen:
  - a) Zahlen wie `.61` sollen erlaubt sein und als `0.61` interpretiert werden.
  - b) Der Rechner soll korrekt mit Klammern `(` und `)` rechnen.
  - c) Der Rechner soll die negativen Zahlen `(-1)`, `(-2)`, `(-3.14)` usw. kennen.
4. Schreiben Sie einen Parser, der die chemischen Summenformeln für Alkane erkennt. Sie folgen dem Muster  $C_nH_{2n+2}$ . Erweitern Sie den Parser auf Alkohole, die auf Alkanen basieren, also auf  $C_nH_{2n+1}OH$ .
5. Die Kinder Anna und Otto haben einen Hund namens Bello und eine Katze namens Sheila. Alle vier Lebewesen und ihre Mitmenschen sind höflich und kontaktfreudig. Ihr Leben wird darum durch die folgenden Produktionsregeln gut beschrieben.

```

<Subjekt> → <Name>|Der <Maskulinum>|Die <Femininum>|Das <Neutrum>
<Prädikat> → sieht|ruft|grüßt
<Objekt> → <Name>|den <Maskulinum>|die <Femininum>|das <Neutrum>
<Name> → Anna|Otto|Bello|Sheila
<Maskulinum> → Lehrer|Schüler|Hund
<Femininum> → Lehrerin|Schülerin|Katze
<Neutrum> → Kind|Tier

```

Einen Parser hierfür zu schreiben ist praktisch ohne Semantik möglich. Probieren Sie es entweder selbst oder schauen sie im Ordner `sprache` nach.

- a) Welches etwas versteckte Detail des Parsers aus den Unterlagen entspricht nicht den Gepflogenheiten der deutschen Rechtschreibung? Wie könnte man den Fehler beheben?
  - b) Erweitern Sie ihren Parser um ein oder zwei zusätzliche Wörter und testen Sie.
6. Natürliche Sprache zu modellieren wird sehr schnell sehr kompliziert. Wir probieren es einmal, indem wir die Grammatik der vorherigen Aufgabe um die Pronomina und um das Präteritum erweitern. Studieren Sie diese Erweiterung im Ordner `sprache2`.
  - a) Welche grammatische Eigenschaft wird geprüft? Welche wird auf andere Weise semantisch verwendet.
  - b) Auf den ersten Blick sieht es so aus, als ob die Verbformen der drei Verben völlig analog gebildet werden. Dem ist aber nicht so. Arbeiten Sie entscheidende Unterschiede heraus.
  - c) Erweitern Sie den Parser so, dass er eine weitere Zeitform beherrscht. (Für das Futur brauchen Sie das Hilfsverb `werden`, für das Perfekt mindestens das Hilfsverb `haben` und das Partizip.)
  - d) Passen Sie den Parser auf eine andere Sprache als Deutsch an.
  - e) Schreiben Sie den Parser so um, dass er auf Sätze wie `Anna ruft Bello.` in der Form `Warum ruft Anna Bello?` antwortet.