pädagogische hochschule schwyz

Michael Hielscher, Beat Döbeli Honegger

01.09.21

Was hat Informatik mit Sprache zu tun?

Warum dieses Dokument?

Mit Informatik verbinden die meisten Menschen Computer, Programmieren, Algorithmen und Nullen und Einsen. Doch Informatik hat auch sehr viel mit Sprache zu tun. Für den Schulunterricht kann die Verknüpfung zum Sprachunterricht bewusst herausgearbeitet werden, um dem eher mathematisch geprägten Bild von Informatik entgegenzuhalten. Die Informatik beschäftigt sich mit Sprache und Schrift aus unterschiedlichen Perspektiven und je nach Teildisziplinen mit anderen, damit verbundenen Herausforderungen.

Dieses Dokument richtet sich primär an Lehrpersonen und Dozierende des Lehrplanmoduls "Medien und Informatik". Es eignet sich damit nicht zur Bearbeitung mit Schülerinnen und Schülern, erklärt jedoch Hintergründe und Konzepte im Sinne einer fachlichen Überhöhung. An besonders gekennzeichneten Stellen sind Hinweise und Ideen für eine mögliche Umsetzung auf der Zielstufe angegeben.

Ziel ist die Verknüpfung insbesondere der Kompetenzen "MI.2.1 Datenstrukturen" und "D.5 Sprache(n) im Fokus" aus dem Lehrplan 21. Auf der einen Seite geht es darum, Daten aus der Umwelt darzustellen, zu strukturieren und auszuwerten und über geeignete Datenstrukturen zur Informationsspeicherung und -verarbeitung zu sprechen und auf der anderen Seite den Gebrauch und die Wirkung von Sprache zu untersuchen und dabei zum Beispiel Sprachstrukturen in Wörtern und Sätzen zu analysieren.

Bedeutung von Sprache

In der Menschheitsgeschichte spielte die Sprache zur Kommunikation und die Schrift zum dauerhaften Speichern von Informationen schon immer eine wesentliche Rolle. Die vergleichsweise junge Informatik als Wissenschaft der systematischen Darstellung, Speicherung, Verarbeitung und Übertragung von Informationen, beschäftigt sich damit per Definition auch mit Sprache und deren Notation - insbesondere mit Sprachen, die Computer verarbeiten und verstehen können.

Zur Unterscheidung verwenden wir im Folgenden die Begriffe *natürliche Sprache* wenn gesprochene Sprachen wie Deutsch, Englisch usw. gemeint sind und *formale Sprache* für künstliche Sprachen wie einer Programmiersprache (z.B. JavaScript, Python, Logo), einer Beschreibungssprache (z.B. Flussdiagramme, UML, HTML, CSS) oder einer Datenbanksprache (z.B. SQL). Die Sprachwissenschaften unterscheiden zwischen der gesprochenen und der geschriebenen Sprache. Bei der Kommunikation mit gesprochener Sprache spielen zusätzlich Aspekte wie Gestik, Mimik oder Tonfall eine Rolle, die hier nicht weiter betrachtet werden. Bei natürlichen Sprachen gibt es zudem Mehrdeutigkeiten, die bei formalen Sprachen i.d.R. von vornherein ausgeschlossen werden. Zum Beispiel ist beim Begriff "Mars" unklar, ob damit ein Planet, ein Schokoriegel oder ein Kriegsgott gemeint ist. Wir

können die Bedeutung meist aus dem Kontext herauslesen oder hineininterpretieren. Je nach Textart ist eine Interpretation sogar explizit erwünscht (z.B. Lyrik). Auch auf Satzebene können Mehrdeutigkeiten entstehen. Die Schlagzeile der Frankfurter Allgemeine Zeitung "Wie viele Deutsche vertragen Schweizer Universitäten?" lässt verschiedene Interpretationen zu. Eine Verträglichkeit im Sinne einer Allergie war aber wohl nicht gemeint. Was für uns durch Sprachgefühl und Intuition klar erscheint (Pragmatik), ist für eine Maschine schwer zu verstehen. Bei Maschinen ist eine Interpretation in der Regel sogar unerwünscht, da Computer-Programme immer gleich arbeiten sollten. Damit unterscheiden sich natürliche und formale Sprache in ihrem Zweck als Kommunikationsmittel.

Umsetzung in der Volksschule:

Anhand von Beispielen aus verschiedenen Sprachen kann die allgemeine Funktion von Sprache erarbeitet werden. Welche Sprachen kennen die Schülerinnen und Schüler? Wie sieht eine einfache Begrüssung in diesen Sprachen aus?

Was macht eine Sprache aus (Konzept von Sprache)? Was ist der Unterschied zwischen Dialekt und Sprache? Diskussion um den verbindenden, aber auch trennenden Charakter von Sprache (Kulturräume). Mehr als 7000 Sprachen auf der Welt, warum gibt es so viele? https://www.zeit.de/wissen/2013-04/s39-infografik-sprachen.pdf

Lehrplan 21:

Die Schülerinnen und Schüler können Sprache erforschen und Sprachen vergleichen.

D.5.C.1.d: können Lautung, Wort- und Satzbau in verschiedenen Sprachen (der Klasse) vergleichen.

Wir beschränken uns im Folgenden auf eine Funktion von Sprache: Einen Sachverhalt möglichst präzise beschreiben zu können. Im Sprachunterricht der Volksschule ist etwa die Sachbeschreibung als Textmuster, bei der ein Sachgegenstand mit möglichst treffenden Adjektiven in einem Aufsatz dargestellt werden soll, ein Beispiel für diesen beschreibenden Aspekt von Sprache. Bei Computersprachen geht es ebenso um eine präzise Beschreibung von Informationen - etwa Anweisungen, die der Computer in einer definierten Reihenfolge vornehmen soll oder Farbangaben zu den einzelnen Pixeln eines Bildes.

Für natürliche Sprachen existieren vielfältige Dialekte und Abwandlungen, die sich stetig verändern - neue Wörter kommen hinzu, andere veralten. Durch grössere Reformen werden zuweilen auch grammatikalische Anpassungen an der Sprache vorgenommen. Sobald ein neues Computerprogramm entwickelt wird und dessen Daten auf der Festplatte gespeichert werden müssen, wird häufig ein neues Datenformat erfunden. Datenformate werden für Anwenderinnen und Anwender primär durch ihre Dateierweiterungen wie pdf, docx, xlsx, svg, jpeg, png usw. sichtbar und zeigen an, dass die Daten innerhalb der Datei einen ganz bestimmten Aufbau verwenden, um von einem passenden Programm verarbeitet werden zu können. Das Erfinden von neuen formalen Sprachen zur Repräsentation von Information ist ein schöpferischer Prozess von Menschenhand. Viele Datenformate sind dabei aus Effizienzgründen so gestaltet, dass nur Maschinen sie lesen und

schreiben können. Auch formale Sprachen werden gelegentlich überarbeitet. Etwa wenn neue Programmversionen erscheinen (z.B. Word 97, Word 2004 …) oder durch Standardisierungsgremien wie die W3C, die u.a. neue Version des HTML-Standards herausgibt (HTML4, HTML5). Ältere Programmversionen kennen die neusten Sprachen (Datenformate) in der Regel nicht und sind damit auch nicht in der Lage entsprechende Dateien zu verarbeiten. Für formale Sprachen gibt es meist eine umfangreiche Spezifikation, die diese präzise beschreibt.

Umsetzung in der Volksschule:

Schüler/innen sollen den Zusammenhang zwischen Datenformaten am Computer und natürlichen Sprache erkennen.

- In einer Tabelle werden die bereits bekannten Datentypen auf dem Computer zusammentragen. Welche Programme werden verwendet und welche Dateierweiterungen besitzen die mit dem Programm verarbeiteten Dateien?
- Die Unterscheidung zwischen für Menschen lesbaren und unlesbaren Datenformate lassen sich anhand von Beispielen aufzeigen (z.B. Umbenennen eines .png in .txt -> unlesbar, .html und .svg Dokumente im Texteditor anschauen -> lesbar).
- Computersprachen als Kommunikationsmittel zwischen Mensch und Computer und zwischen Computern verdeutlichen. Datenformate sind wie verschiedene Sprachen, die der Computer verstehen kann.

Lehrplan 21:

MI.2.1.d: kennen analoge und digitale Darstellungen von Daten (Text, Zahl, Bild und Ton) und können die entsprechenden Dateitypen zuordnen.

MI.2.1.e: kennen die Bezeichnungen der von ihnen genutzten Dokumententypen.

In den Anfängen der Digitalisierung verwendete man zum Programmieren von Computer unmittelbar den Maschinencode (auch als Maschinensprache bezeichnet). Ein kurzer Zahlencode steht dabei für eine bestimmte Operation (addiere, lese, schreibe usw.), die der Hersteller für den jeweiligen Prozessor vorgesehen und dokumentiert hat. Eine Programmiererin musste damit zunächst eine lange Liste von Zahlencodes und deren Bedeutung für jeden Prozessortyp erlernen. Um diese mühsame Arbeit zu vereinfachen wurden sogenannte Hochsprachen (Programmiersprachen) erfunden, die für Menschen leichter zu lesen, zu schreiben und zu erlernen sind und von Compilern automatisiert in einen äquivalenten Maschinencode des jeweilig verwendeten Prozessors übersetzt werden können. Auch bei natürlichen Sprachen gab es Versuche solche Hochsprachen (sogenannte Plansprachen) zu entwickeln, um die internationale Kommunikation zu vereinfachen. Das wohl bekannteste Beispiel für eine Plansprache ist Esperanto, welches weltweit von geschätzt einer Millionen Menschen gesprochen wird.

Ziel eines fächerverbindenden Unterrichts von Deutsch und Medien und Informatik ist die Verknüpfung von natürlichen und künstlichen (formalen) Sprachen bei der Konzeptbildung von Sprache und ihrer Bedeutung.

Syntax und Semantik

Wie bei natürlichen Sprachen benötigen formale Sprachen grammatikalische Regeln (Syntax) und eine Festlegung der Bedeutung (Semantik). Müsste man eine vollständige Definition der Sprache "Deutsch" aufschreiben, so käme wohl ein Art Grammatik- und Wörterbuch heraus. Dies mit der Gewissheit, wohl nie alle Besonderheiten, lokalen Eigenheiten, Ausnahmefälle usw. berücksichtigt zu haben. Ein Kompromiss, mit dem wir Menschen leben können - Maschinen, die unsere natürlichen Sprachen verstehen sollen, benötigen aber präzise und vollständige Beschreibungen davon, wie die Sprache aufgebaut ist und welche Bedeutung die Wörter und Sätze haben. Bereits in den 1960er Jahren ging man davon aus, dass Computer bald die menschlichen Sprachen verstehen würden und sich das Programmieren mit Programmcode erübrigen würde, da eine einfache Erklärung des Problems per Mikrofon genügen würde. Auch ein halbes Jahrhundert später ist diese Vision nicht erfüllt. Eine präzise Problembeschreibung bleibt auch mit Spracherkennung eine Herausforderung. Zudem scheitern Computer auch heute noch am Verständnis unserer komplexen natürlichen Sprache und ihrer Feinheiten.

Doch wie beschreibt man Sprachen so vollständig und präzise, dass keine Interpretation mehr möglich und kein Spezialfall vergessen geht? Es braucht eine systematische, formale Beschreibung für Sprachen. Ziel der Informatik ist es, eine Sprache so präzise zu beschreiben, sodass ein Computer die Regeln automatisiert anwenden kann, um mindestens zu entscheiden, ob eine beliebige Eingabe zur gegebenen Sprache gehört oder nicht. Der Bereich der formalen Sprachen und abstrakten Automaten ist einer der fünf zentralen Inhaltsbereiche der *Bildungsstandards für Informatik* der Gesellschaft für Informatik (GI, https://www.informatikstandards.de/).

Es gibt auch abseits der Computer und der Informatik vielfältige, formal definierte Sprachen, die genau wie unserer natürlichen Sprache gewissen Regeln folgen. Ein Beispiel sind die Schilder im Strassenverkehr. Diese bestehen aus einer Grundform und einer bestimmten Farbgebung:

Die Grundformen stehen in diesem Beispiel für ein Verbotszeichen, ein Gefahrenzeichen und ein Gebotszeichen, die mit weiteren Symbolen zu einem Strassenschild ergänzt werden:

Die Verkehrszeichen bilden damit eine eigene Sprache, die wir interpretieren und lesen können, auch wenn wir ein bestimmtes Schild noch nie zuvor gesehen haben.

Die Syntax einer Sprache beschreibt den Aufbau einzelner Sätze in der Sprache. Im Beispiel der Verkehrszeichen hätten wir eine Grundform, ein Symbolbild und eventuell noch ein kleines rechteckiges Zusatzzeichen unterhalb der Grundform (z.B. in 800m, 6-18 Uhr). Wenn diese regelkonform kombiniert werden, erhalten wir ein syntaktisch korrektes Verkehrsschild.

Obwohl es sich bei den beiden folgenden Schildern "Wildverbot" und "Staugebot" um syntaktisch korrekte Verkehrsschilder handelt, ergeben sie von ihrer Semantik, also ihrer Bedeutung, keinen Sinn und finden sich deshalb nicht im Strassenverkehr:

Die Semantik ist eine Teildisziplin der Sprachwissenschaft und befasst sich mit der Beschreibung und der Erklärung der Bedeutung von Sprachelementen und deren Kombination zu komplexen Äusserungen, sodass ganze Sätze oder noch grössere Einheiten entstehen, die in der Kommunikation eine Bedeutung haben. Bei einer Programmiersprache kann mit einer formalen Semantik jeder Programmieranweisung ein eindeutiges Maschinenverhalten zugeordnet werden.

Umsetzung in der Volksschule:

Die Schülerinnen und Schüler erhalten eine Übersicht der gebräuchlichsten Verkehrsschilder. Die wiederkehrenden Bestandteile der Schilder sollen identifiziert und einer Bedeutung zugeordnet werden (Grundform, Farben, Symbolen). Die Schüler konstruieren einige Verkehrsschilder nach den Grundregeln der Sprache, die es bislang noch nicht gibt. Z.B.: Smartphone-Verbotsschild, Achtung Hausaufgaben, nur für Schüler/innen usw. Die Lösungen können anschliessend verglichen und diskutiert werden.

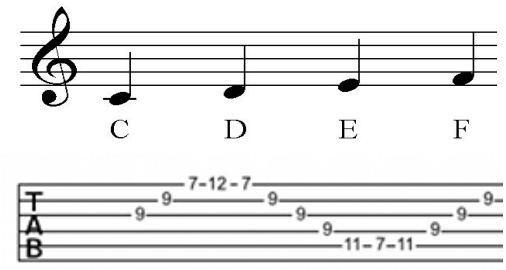
Formale Sprachen und ihre Syntax / Semantik kann am Beispiel mit jenen der natürlichen Sprache in Bezug gesetzt werden (Lehrplan 21 Bereich D.5.D. Grammatikbegriffe).

Lehrplan 21:

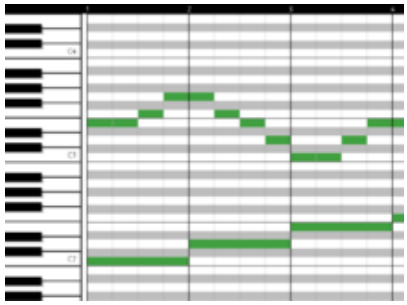
D.5.B.1.c: können sich unter Anleitung mit verschiedenen sprachlichen Themen auseinandersetzen (z.B. Spracherwerb, Verständlichkeit/Internationalität von Piktogrammen, Geheimsprachen/-schriften).

Formalisierte Sprachen erlauben auch die Kommunikation über natürliche Sprachgrenzen hinweg. Piktogramm-Sprachen wie die Verkehrsschilder müssen in

der Regel nicht ins Englische, Französische und Italienische übersetzt werden. Ein weiteres Beispiel ist Musik und ihre verschiedenen Notationen. Die gebräuchliche Notenschrift mit Notenschlüssel, Notenzeile und Noten mit Notenwerten ist eine Informationsdarstellung von Tönen. Auch dafür gibt es je nach Instrument weitere Darstellungsvarianten, die alle zum Ziel haben Tonfolgen möglichst präzise zu beschreiben. Die Darstellungen lassen sich in der Regel ineinander überführen (siehe Abb. 1).



Tabulatur (Gitarre)



Piano-Roll

Abbildung 1: Beispiele für unterschiedliche Musiknotationen

Das wohl bekannteste Beispiele für eine formale Sprachen ist die Mathematik. Arithmetische Ausdrücke wie: $y = 4x^2 + 3x + 10$ werden weltweit verstanden und müssen nicht übersetzt werden. Ihre Regel sind klar beschrieben, womit auch Computer in der Lage sind Gleichungen zu lesen und zu berechnen. Computer-Algebra-Systeme können komplexe mathematische Aufgaben selbstständig lösen, weit über die Fähigkeiten eines einfachen Taschenrechners hinaus. Gleiches gilt für chemische Gleichungen wie:

$$C_6H_{12}O_6 + 6\,O_2 \longrightarrow 6\,CO_2 + 6\,H_2O$$

Der Aufbau solcher Sprache lässt sich mit grammatikalischen Regeln genau beschreiben. Die Sprache definiert dabei die Bedeutung der einzelnen Symbole und Zeichen. Bereits am Beispiel der arithmetischen Gleichungen und chemischen Formeln wird deutlich, dass in unterschiedlichen Sprachen die gleichen Symbole eine ganz andere Funktion haben können.

Umsetzung in der Volksschule:

Die Schülerinnen und Schüler sammeln weitere Alltagsbeispiele für formale Sprachen. Zum Beispiel: Autokennzeichen, Kleider- und Schuhgrössen, Postleitzahlen, Telefonnummern, Kontonummern, ISBN-Nummern, Seriennummern bei Geräten ...

Einfache formale Sprachen weisen überschaubare Regeln auf und können damit von Schülerinnen und Schülern vollständig beschrieben werden. Im Gegensatz zu natürlichen Sprachen erlaubt dies eine Vollständigkeitsbetrachtung (wurden alle relevanten Regeln der Sprache erkannt und festgehalten?)

Formale Sprachen sind nicht nur für Programmiersprachen relevant, sondern überall da, wo Informationen strukturiert gelesen und verarbeitet werden müssen. Der Strichcode auf der Milchpackung, die IBAN-Nummer bei der Überweisung oder die Autokennzeichen, alle folgen präzisen Regeln.

Ziel eines fächerverbindenden Unterrichts von Deutsch und Medien und Informatik ist es, die Bedeutung von Syntax und Semantik sowohl für natürliche als auch für formale Sprachen herauszuarbeiten. Formale Sprachen aus dem Alltag werden dazu analysiert und ihre bereits bekannten Regeln zur Wortkonstruktion angewendet. Davon wird der Wunsch nach einer präzisen Beschreibung der Regeln einer Sprache abgeleitet.

Syntax in natürlichen Sprachen

Um an die Erfahrungswelt der Schülerinnen und Schüler anzuknüpfen wird zunächst die Syntax natürlicher Sprachen betrachtet. Im Sprachunterricht werden zur Rechtschreibung und Grammatik vielfältige Inhalte zu diesem Thema erarbeitet. In Ergänzung dazu kann mit der "Informatik-Brille" der formale Aufbau der natürlichen Sprachen betrachtet werden.

Der Begriff Syntax wird in der Sprachwissenschaft insbesondere mit dem Satzbau verbunden. Man unterscheidet die Syntax von der linguistischen Morphologie, die den inneren Aufbau der Wörter behandelt (etwa Wortendungen, Pluralformen usw.). Betrachten wir an einem Beispiel den Satzbau in der deutschen Sprache (aus dem Wikipedia-Artikel zum Thema Satzglieder):

Hans baut im Wald mit seinem Freund eine riesige Baumhütte.
Im Wald baut Hans mit seinem Freund eine riesige Baumhütte.
Mit seinem Freund baut Hans im Wald eine riesige Baumhütte.
Eine riesige Baumhütte baut Hans im Wald mit seinem Freund.

Die Satzglieder können nach ihrer Funktion farbig eingeteilt werden:

Subjekt Prädikat Objekt Adverbialbestimmung

Ein Satz wird somit aus mehreren Satzgliedern gebildet, die in einer definierten Reihenfolge aneinandergefügt werden müssen. Eine abweichende Aneinanderreihung ist nicht zulässig und gilt als grammatikalisch falsch.

Wir können für diese vier Variante eine formale Beschreibung erstellen, anhand deren man die obigen Sätze bilden kann. Dazu definieren wir einen "Satz" als eine von vier möglichen Varianten wie oben angegeben:

```
Satz -> Subjekt Prädikat Adverb1 Adverb2 Objekt .

Satz -> Adverb1 Prädikat Subjekt Adverb2 Objekt .

Satz -> Adverb2 Prädikat Subjekt Adverb1 Objekt .

Satz -> Objekt Prädikat Subjekt Adverb2 Adverb1 .
```

Für die Platzhalter auf der rechten Seite können wiederum mögliche Ersetzungen definiert werden:

```
Subjekt -> Hans

Prädikat -> baut

Adverb1 -> im Wald

Adverb2 -> mit seinem Freund

Objekt -> eine riesige Baumhütte
```

Damit haben wir einen formalen Aufschrieb aller bildbarer Sätze der "Hans-Sprache" gebaut. Die Hans-Sprache ist eine einfache Sprache und bildet einen winzig kleinen Teil der deutschen Sprache ab. Um einen "Satz" zu erstellen, können wir eine der vier bei "Satz" angegebenen Varianten auswählen und danach alle Platzhalter ersetzen. Dieser Vorgang lässt sich mit einem Computerprogramm automatisieren. Unter https://programmingwiki.de/Sprache lässt sich das Beispiel ausprobieren und zufällige Sätze generieren. Wirklich spannend ist das nicht, da nur genau vier verschiedene Sätze entstehen können. Erweitern wir die Platzhalter um einige weiteren Auswahlmöglichkeiten:

```
Subjekt -> Hans
Subjekt -> Egon
Subjekt -> Max

Prädikat -> baut
Prädikat -> bastelt
Prädikat -> streicht

Adverb1 -> im Wald
Adverb1 -> im Garten
Adverb1 -> hinter dem Haus

Adverb2 -> mit seinem Freund
Adverb2 -> mit Susi
Adverb2 -> mit jeder Menge
           Werkzeug

Objekt -> eine riesige
          Baumhütte
Objekt -> ein Co-Kart
Objekt -> eine Rakete
```

Wenden wir die gleichen Ersetzungsstrategie erneut an, erhalten wir nun Sätze wie:

Mit seinem Freund baut Max im Garten eine Rakete .
Mit jeder Menge Werkzeug bastelt Hans hinter dem Haus eine Rakete .
Eine riesige Baumhütte baut Egon mit seinem Freund im Wald .

Es lassen sich bereits mehrere hundert verschiedene Sätze generieren, die alle syntaktisch korrekte und semantisch sinnvolle deutsche Sätze darstellen. Auf der Webseite kann man etwas experimentieren und eigene Wörter für die Platzhalter eintragen. So ganz willkürlich wurden die weiteren Wörter aber nicht ausgewählt. Es fällt zum Beispiel auf, dass alle Subjekte männliche Vornamen sind. Wenn wir die weiblichen Vornamen Maria und Susi bei Subjekt hinzufügen, so entstehen grammatikalisch falsche deutsche Sätze.

Mit seinem Freund baut Maria im Garten eine Rakete.
Eine riesige Baumhütte bastelt Susi mit seinem Freund im Wald.

Wir benötigen eine Fallunterscheidung für weibliche und männliche Subjekte. Ergänzen wir den ersten Platzhalter "Satz" jeweils um eine männliche und eine weibliche Variante:

```
Satz -> SubjektM Prädikat Adverb1 Adverb2M Objekt .

Satz -> SubjektW Prädikat Adverb1 Adverb2W Objekt .

Satz -> Adverb1 Prädikat SubjektM Adverb2M Objekt .

Satz -> Adverb1 Prädikat SubjektW Adverb2W Objekt .

Satz -> Adverb2M Prädikat SubjektM Adverb1 Objekt .

Satz -> Adverb2W Prädikat SubjektW Adverb1 Objekt .

Satz -> Objekt Prädikat SubjektM Adverb2M Adverb1 .

Satz -> Objekt Prädikat SubjektW Adverb2W Adverb1 .
```

```
SubjektM -> Hans
SubjektM -> Karl
SubjektM -> Egon
SubjektM -> Max

Adverb2M -> mit seinem Freund
Adverb2M -> mit seiner Freundin
Adverb2M -> sehr schnell
Adverb2M -> mit viel Geduld
```

```
SubjektW -> Maria
SubjektW -> Susi
SubjektW -> Lara
SubjektW -> Eva

Adverb2W -> mit ihrem Freund
Adverb2W -> mit ihrer Freundin
Adverb2W -> sehr schnell
Adverb2W -> mit viel Geduld
```

Erneut können wir grammatikalisch korrekte Sätze generieren. Natürlich definiert unsere Sprach-Beschreibung nur einen ganz kleinen Teil der deutschen Sprache. Mit Hilfe eines Wörterbuchs könnte man für einzelne Platzhalter tausende möglicher Wörter finden und eintragen.

Umsetzung in der Volksschule:

Die Satzgliedstellung in der deutschen Sprache wird im Sprachunterricht erarbeitet. Im Informatikunterricht experimentieren die Schülerinnen und Schüler mit dem Satzgenerator (https://programmingwiki.de/Sprache). Es sollen neben männlichen und weiblichen Formen auch Plural-Subjekte zur Sprache hinzugefügt werden. Dazu muss die Definition von "Satz" entsprechend ergänzt und neue Platzhalter hinzugefügt werden.

Anschliessend kann in Partnerarbeit ein eigener Satzgenerator für ein ausgewähltes Thema erfunden werden (z.B. Verbote im Schulhaus, Glückskeks-Sprüche, Zitate).

Eine Figur der Filmgeschichte ist durch ihre Art zu sprechen bekannt geworden. Yoda aus der Star Wars Saga verwendet die Satzglieder in einer anderen Reihenfolge: Objekt, Subjekt, Prädikat. Durch eine einfache Umstellung der Satzdefinition im Programm kann ein Yoda-Generator erstellt werden.

Durch die Nutzung des Computerprogramms wird beiläufig eine eindeutige, formale Beschreibung der gewünschten Sprache erarbeitet.

Ziel eines fächerverbindenden Unterrichts von Deutsch und Medien und Informatik ist es, die im Sprachunterricht behandelte Satzglieder und Satzgliedstellung auch aus dem Blickwinkel der Informatik zu betrachten. Mit einem spracherzeugenden Computerprogramm kann eine zusätzliche, intrinsische Motivation für die Auseinandersetzung mit grammatikalischen Regeln der Muttersprache oder einer Fremdsprache geschaffen werden. Weiterführend kann die Computerlinguistik und ihre Anwendung in Chat-Bots, computergenerierten Texten und der automatisierten Textanalyse, Rechtschreibprüfung usw. und ihrer Grenzen thematisiert werden (siehe auch Abschnitt Computerlinguistik).

Syntaxdiagramme

Die theoretische Informatik beschäftigt sich als Teilgebiet der Informatik u.a. mit der Frage, wie man eine Sprache mathematisch so präzise beschreiben kann, dass ein Computer entscheiden kann, ob ein eingegebenes Wort (bzw. Satz) zur Sprache gehört - d.h. syntaktisch korrekt ist. Wenn beim Programmieren der Compiler eine Fehlermeldung ausgibt, dass auf Zeile 27 wohl ein Semikolon vergessen wurde, dann tut er dies nicht, um den Entwickler zu ärgern oder zur Sorgfältigkeit zu ermahnen. Vielmehr kann er ohne das Semikolon nicht mehr sicher entscheiden, was der Programmierer wohl gemeint hat. Entfernen wir in einem deutschen Text alle Satzzeichen, ergibt sich in einigen Fällen ebenso eine ganz andere Bedeutung.

Je nach Umfang und Komplexität lassen sich formale Sprachen mit unterschiedlichen Mitteln beschreiben. Eine einfache Darstellung für formale Sprachen sind sogenannte Syntax-Diagramme (auch Railroad-Diagramme genannt). Der Name Railroad entspringt der Idee, wie eine Eisenbahn von einem Startpunkt (typischerweise ganz links) den Schienen entlang zum Ende (ganz rechts) zu fahren. Im folgenden Syntax-Diagramm wird die Sprache der digitalen 24h Uhrzeiten (0:00 bis 23:59) beschrieben. Jede gültige Uhrzeit lässt sich mit dem Finger entlang der Linien bilden. Damit haben sie starke Ähnlichkeiten mit Flussdiagrammen, die im Informatikunterricht auf Primarstufe ebenfalls thematisiert werden können.

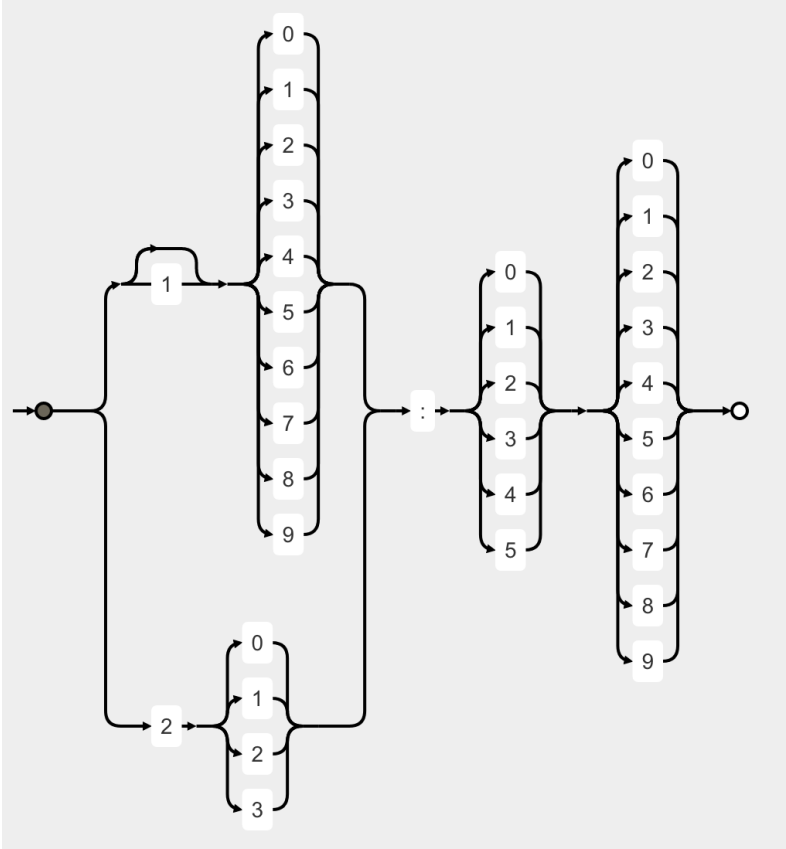


Abbildung 2: Syntaxdiagramm für Uhrzeiten

Die zuvor entwickelte Hans-Sprache lässt sich ebenfalls mit einem Railroad-Diagramm darstellen. Die Diagramme werden schnell gross und unübersichtlich. Zudem werden gewisse Dinge wiederholt. Die Darstellung wird mit wachsender Komplexität der Sprache immer ineffizienter (siehe Abb. 3).

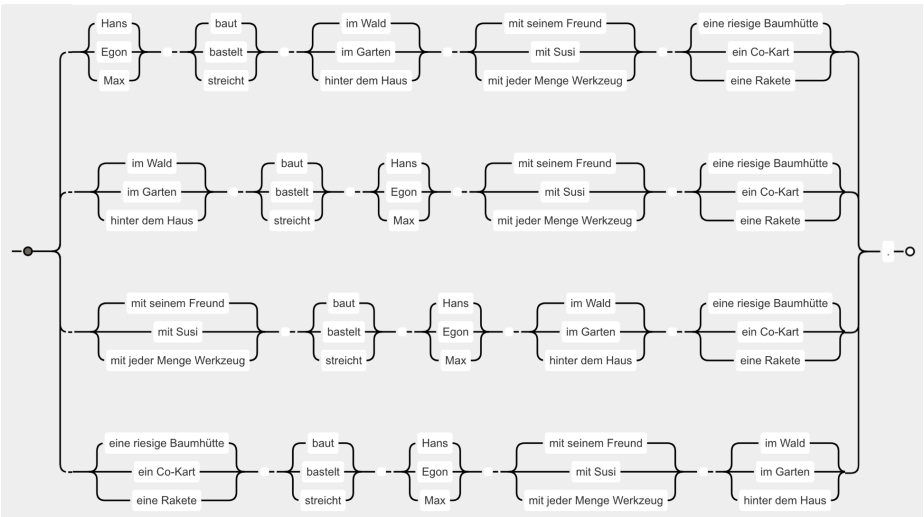


Abbildung 3: Railroad-Diagramm für die “Hans-Sprache”

Um die Darstellung zu vereinfachen, können mehrere Syntaxdiagramm mit Platzhaltern zusammengesetzt werden. Für das Beispiel der Hans-Sprache sieht das wie in Abb. 4 aus. Jeder Platzhalter (Rechteck) wird mit einem eigenen Syntaxdiagramm beschrieben. Beim Ablaufen des Diagramms, muss das

Teildiagramm jeweils gedanklich für den Platzhalter eingesetzt werden, um wieder das Railroad-Diagramm zu erhalten.

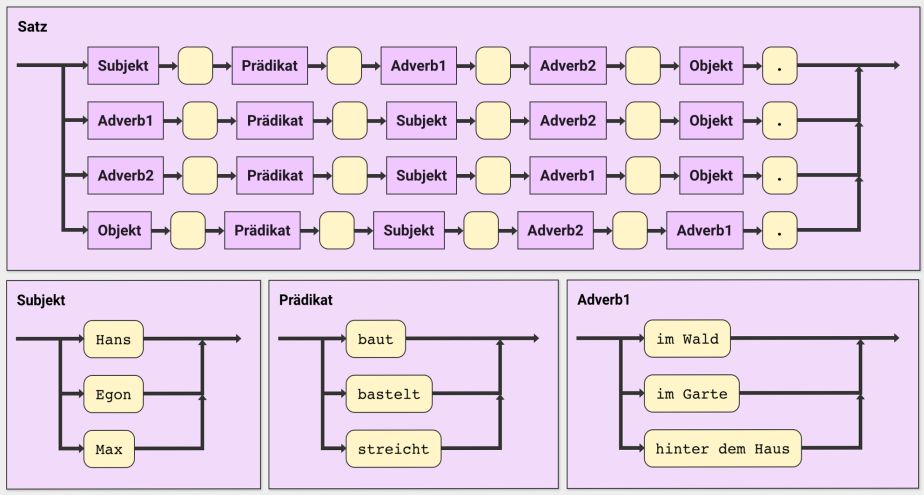


Abbildung 4: Ausschnitt zusammengesetztes Syntaxdiagramm mit Platzhaltern

Umsetzung in der Volksschule:

Mit einem Railroad-Diagramm für Uhrzeiten oder Autokennzeichen kann die formale Beschreibung der Syntax einer Sprache eingeführt werden. Anhand einiger Eingabebeispiele kann die Leistungsfähigkeit des Diagramms überprüft werden. Die Schülerinnen und Schüler erhalten zum Beispiel das Diagramm zu den Uhrzeiten und prüfen Eingabe wie 3:33, 23:66 und 03:10. Eine systematische Darstellung aller zur Sprache gehöriger Eingaben bezieht auch Randfälle ein, die schnell vernachlässigt werden. So gehört 00:12 nicht zu der durch das Railroad-Diagramm beschriebenen Sprache, 0:12 hingegen schon.

Die Schülerinnen und Schüler erstellen ein eigenes Railroad-Diagramm für eine einfache Sprache (z.B. Postleitzahlen). Auf Grund der Grösse eignet sich die Darstellung kaum für komplexere Sprachen. Anschliessend lesen sie für eine komplexere Sprache ein zusammengesetztes Syntaxdiagramm mit Platzhaltern und können die beiden Darstellungen (Railroad / Syntaxdiagramm) gedanklich ineinander überführen.

Lehrplan 21:

MI.2.2.c: können Abläufe mit Schleifen und Verzweigungen aus ihrer Umwelt erkennen, beschreiben und strukturiert darstellen (z.B. mittels Flussdiagrammen).

Die Informatik unterscheidet verschiedene Typen von formalen Sprachen. Noam Chomsky beschrieb 1956 in der nach ihm benannten Chomsky-Hierarchie vier Sprachklassen, die wie in einem Schalenmodell immer höhere Anforderungen an den Entscheidungsalgorithmus stellen, ob eine Eingabe zur Sprache gehört oder nicht. Für die allgemeinste und anspruchsvollste Stufe (Typ 0) kann ein Computer zwar erkennen, ob eine Eingabe syntaktisch korrekt ist, er kann jedoch nicht immer sagen, dass sie es nicht ist. Das klingt etwas paradox - stellt man sich die

Arbeitsweise des Computers vor, würde er in einem solche Fall ewig weiterrechnen, aber nie zu einer Entscheidung kommen. Für interessierte Leser lohnt sich ein Blick in die Berechenbarkeitstheorie. Als Bettlektüre wäre *Das Affenpuzzle: und weitere bad news aus der Computerwelt* von David Harel zu empfehlen.

Mit Syntaxdiagrammen lassen sich nur Sprachen vom Typ 2 (kontextfreie Sprachen) und Typ 3 (reguläre Sprachen) darstellen. In der praktischen Informatik sind diese Sprachklassen von besonderer Bedeutung, da sie mit effizienten Algorithmen verarbeitet werden können. Die allermeisten Programmiersprachen gehören deshalb zum Typ 2. Wichtig bleibt festzuhalten, dass mit den hier eingeführten Syntaxdiagrammen nur einen Teil aller definierbaren Sprachen beschrieben werden kann. Computerlinguisten streiten darum, zu welchem Typ natürliche Sprachen gehören. Huybregts 1984 und Shieber 1985 führen einen formalen Beweis, dass Schweizerdeutsch nicht kontextfrei, und somit zum Typ 1 gehören müsste. Dies ist insofern relevant, als dass es für Maschinen deutlich schwieriger wäre, natürliche Sprache zu verarbeiten, wenn sie nicht von den effizienten Entscheidungsalgorithmen für Typ 2 Gebrauch machen könnten (mehr zum Thema im Abschnitt Computerlinguistik). Gleichzeitig würde auch unser Beschreibungsmittel Syntaxdiagramm nicht mehr ausreichen, um natürliche Sprachen vollständig zu beschreiben.

Syntaxdiagramme sind wie Verkehrsschilder oder Flussdiagramme selbst eine formale Sprache mit Symbolen (Syntax) und einer zugeordneten Bedeutung (Semantik). Um formale Sprachen zu beschreiben, bedienen wir uns anderer formaler Sprachen zur Sprachbeschreibung. Auch Beschreibungssprachen haben Grenzen und ermöglichen nur Sprachen bis zu einem bestimmten Typ der Chomsky-Hierarchie zu beschreiben. Aus diesem Grund verwendet die theoretische Informatik unterschiedliche Beschreibungsmittel für unterschiedliche Sprachklassen.

Ziel eines fächerverbindenden Unterrichts von Deutsch und Medien und Informatik ist es, die Verwandtschaft der grammatikalischen Regeln natürlicher Sprachen und formaler Sprachen aufzuzeigen. Die formale Beschreibung einer Sprache gelingt durch geeignete Beschreibungsmittel wie einem Syntaxdiagramm.

Formale Grammatik

Gezeichnete Syntaxdiagramme sind sehr anschaulich für Menschen. Für Computer sind bildliche Darstellungen aber weniger geeignet. Eine sehr gebräuchliche textuelle Sprachenbeschreibungssprache ist die formale Grammatik. Eine übliche Schreibweise für Grammatiken ist die nach ihren Erfindern benannte Backus-Naur-Form (kurz BNF). Die Bildungsregeln für die Sprache der Uhrzeiten lassen sich damit wie folgt aufschreiben:

```
P = {

  Uhrzeit -> Stunden : Minute1 Minute2

  Stunden -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
… | 23

  Minute1 -> 0 | 1 | 2 | 3 | 4 | 5

  Minute2 -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

}
```

Eine formale Grammatik wird als Tupel mit `G = (N,T,P,s)` angegeben. Sie besteht aus Platzhaltern, sogenannten Nichtterminalen (`N`), die sich als Menge angeben lassen. Im Beispiel: `N = {Uhrzeit, Stunden, Minute1, Minute2}`. Alle Zeichen die in der Sprache tatsächlich vorkommen werden Terminale (`T`) genannt und ebenfalls als Menge notiert: `T = {:,0,1,2,3,4,5,6,7,8,9,10,11,12,...,23}`. Der wichtigste Teil der Grammatik sind die obene gezeigten Produktionsregeln (`P`), weshalb man in der Praxis häufig nur diesen Teil der Grammatik angibt. Das kleine `s` bestimmt ein Nichtterminal als Start- oder Spitzensymbol, bei welchem mit der Ersetzung der Platzhalter begonnen werden muss - in diesem Fall `s = Uhrzeit`. Häufig verabredet man, dass automatisch das erste definierte Nichtterminal in `P` als Startsymbol verwendet wird. Damit lassen sich alle anderen Angaben von `P` ableiten und wir können im Folgenden auf die Angabe des vollständigen Tupels verzichten.

Bei den Produktionsregeln (`P`) wird jeweils ein Nichtterminal auf der linken Seite durch einen Pfeil `->` von einer möglichen Ersetzung auf der rechten Seite getrennt. Die Ersetzung kann aus beliebig vielen Nichtterminalen und Terminalen bestehen. Eine Besonderheit ist der senkrechte Strich `|` der als ODER gelesen wird und eine platzsparende Schreibweise ermöglicht. Man könnte auch jeweils alle Varianten einzeln aufschreiben.

Das Nichtterminal `Minute1` hat demnach 6 mögliche Ersetzungen:

```
  Minute1 -> 0 | 1 | 2 | 3 | 4 | 5
```

entspricht:

```
Minute1 -> 0      Minute1 -> 1      Minute1 -> 2

Minute1 -> 3      Minute1 -> 4      Minute1 -> 5
```

Diese Schreibweise entspricht jener aus den ersten Experimenten mit Sprachgeneratoren. Wir haben bei der Hans-Sprache bereits mit Produktionsregeln einer formalen Grammatik gearbeitet:

```
  Satz -> Subjekt Prädikat Adverb1 Adverb2 Objekt .
  Satz -> Adverb1 Prädikat Subjekt Adverb2 Objekt .
  Satz -> Adverb2 Prädikat Subjekt Adverb1 Objekt .
  Satz -> Objekt Prädikat Subjekt Adverb2 Adverb1 .
```

```
Subjekt -> Hans

Prädikat -> baut

Adverb1 -> im Wald

Adverb2 -> mit seinem Freund

Objekt -> eine riesige Baumhütte
```

Computer können mit Hilfe einer solchen formalen Grammatik und einem Algorithmus entscheiden, ob eine Eingabe zur definierten Sprache gehört, also syntaktisch korrekt ist oder nicht. Wie kann man sich nun die Arbeit eines solchen Entscheidungsalgorithmus vorstellen? Angenommen Sie müssten selbst entscheiden, ob eine bestimmte Eingabe zu einer gegebenen Sprache gehört. Versuchen Sie es am nachfolgenden Beispiel. Gegeben sind die Produktionsregeln zu folgender Grammatik. Versuchen Sie für die im Kästchen rechts gezeigte Eingabe zu entscheiden, ob diese syntaktisch korrekt ist.

```
Song     -> Notes

Notes    -> Note

Notes    -> Note Notes

Note     -> Key - Duration

Note     -> Pause - Duration

Key      -> KeyName Octave

KeyName  -> C | D | E | F | G | H | A

Octave   -> 0 | 1 | 2 | 3

Duration -> 1 | 2 | 4 | 8 | 16 | 32

Pause    -> P
```

Hänschen Klein in der Musik-Sprache:

```
G0-4 E0-4 E0-2
F0-4 D0-4 D0-2

C0-4 D0-4 E0-4
F0-4 G0-4 G0-4
G0-2

G0-4 E0-4 E0-2
F0-4 D0-4 D0-2

C0-4 E0-4 G0-4
G0-4 C0-2
```

Betrachten wir zur Vereinfachung nur die erste Note `G0-4`. Gehört diese Eingabe zur beschriebenen Musik-Sprache? Wir beginnen die Ersetzung der Platzhalter wie verabredet mit dem ersten Nichtterminal aus `P`:

```
Song
```

Wir ersetzen das Nichtterminal mit einer passenden rechten Regelseite. Die ersten beiden Ersetzungsschritte sehen dann so aus:

```
Song => Notes => Note
```

Mit einem => wird eine Ersetzung angezeigt, wobei wie bei einem Gleichheitszeichen in der Mathematik eine äquivalente Darstellung beschrieben wird. Vergleichbar mit 12 = 6 + 6 = 3 + + 3 + 3 + 3. Im zweiten Schritt hätten wir uns auch für eine andere Variante entscheiden können:

```
Song => Notes => Note Notes
```

Ein Computer kann alle Varianten durchprobieren und prüfen, ob ein Weg zur Lösung führt. Wir verwenden die Methode des "scharfen Hinsehens" und sehen, dass für `G0-4` die erste Variante wohl genügen wird. Erneut haben wir für Note eine Wahlmöglichkeit. Da die Eingabe nicht mit einem `P` wie in `Pause` beginnt entscheiden wir uns intuitiv für die Ersetzung:

`Song => Notes => Note => Key - Duration`

Nun haben wir erstmals mehrere Platzhalter, die es zu Ersetzen gilt. Wir beginnen mit dem am weitesten linksstehenden Nichtterminal `Key` und wenden erneut eine passende Regel an.

`Song => Notes => Note => Key - Duration => KeyName Octave - Duration`

Wichtig ist hier, dass der Rest unverändert übernommen wird. Das Minus und `Duration` bleiben unverändert stehen. Wieder wählen wir das am weitesten linksstehende Nichtterminal `KeyName` aus und wenden eine passende Regel an:

`... => KeyName Octave - Duration => G Octave - Duration`

Diesmal haben wir die Regel ausgewählt, die zu einem passenden Terminal `G` geführt hat, welches im Eingabewort an der ersten Stelle vorkommt. Wir ersetzen das nächste Nichtterminal `Octave` mit einem passenden Terminal:

`... => G Octave - Duration => G 0 - Duration`

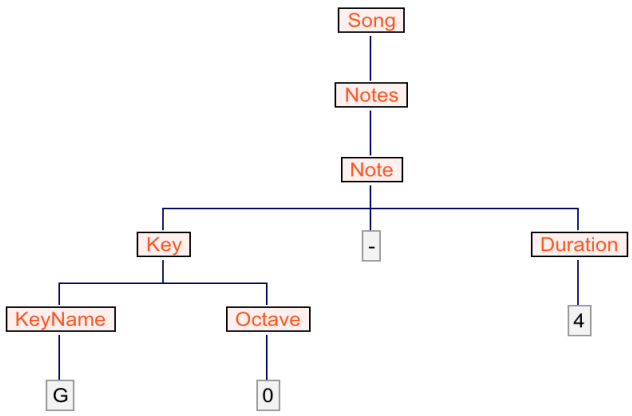
Es verbleibt noch die letzte Ersetzung für `Duration`:

`... => G 0 - Duration => G 0 - 4`

Damit haben wir die Eingabe `G0-4` vom Startsymbol `Song` aus rekonstruiert. Da uns diese Ableitung gelungen ist, können wir mit Sicherheit sagen, dass die Eingabe zur Sprache gehört und damit syntaktisch korrekt ist.

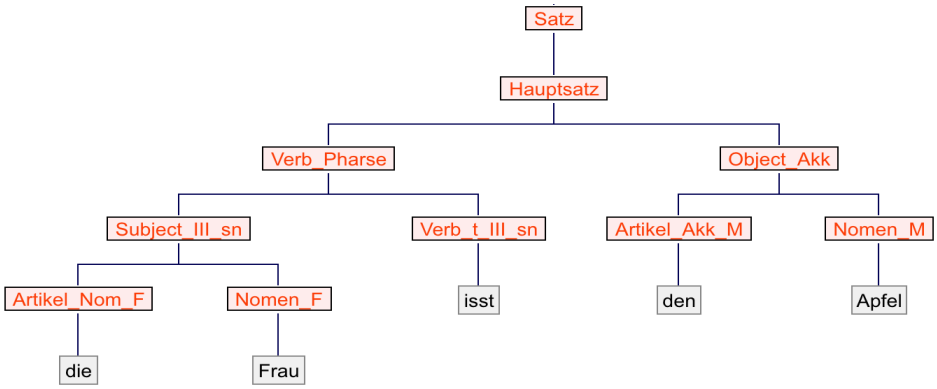
Ein Computer muss im schlimmsten Fall alle möglichen Ersetzungen ausprobieren und mit der Eingabe vergleichen. Der dafür nötige Algorithmus entspricht dem eben per Hand durchgeführten Ableitungsprozess.

Als Alternative zur langen Kettendarstellung mit den => (auch als Satzformenfolge bezeichnet), verwendet man häufig eine graphische Darstellung des Ableitungsprozesses - einen Ableitungsbaum.



Mit Hilfe des Werkzeugs FLACI können für beliebige Eingaben Ableitungsbäume wie dieser generiert werden. Auf: https://flaci.com/G3 bzw. https://flaci.com/Ghjkuzfia kann dies selbst ausprobiert werden.

Mit einer deutlich umfangreicheren formalen Grammatik für einen Ausschnitt der deutschen Sprache, ergibt die Ableitung der Eingabe "Die Frau isst den Apfel" einen deutlich komplexer Ableitungsbaum:



Bereits an den hier verwendeten Bezeichnungen der Platzhalter (Nichtterminale) kann man gewisse Bezüge zur uns bekannten deutschen Grammatik erahnen. Die zugehörige formale Grammatik füllt bereits mehrere A4 Seiten.

Umsetzung in der Volksschule:

Die formale Grammatik in BNF wurde bereits bei den Sprachgeneratoren eingesetzt und angewendet, jedoch nicht selbst thematisiert. Für die Volksschulstufe sind formale Grammatik und die Ableitung ein anspruchsvolles Thema und eigenen sich damit frühestens für den Zyklus 3 oder für die Sekundarstufe II.

Mit dem Werkzeug FLACI.com können Grammatiken als Syntaxdiagramme oder in Textform notiert und anschliessend beliebige Eingaben abgeleitet werden. Ebenso können Zufallswörter der Sprache gebildet werden, Ableitungsbäume betrachtet und der Ableitungsprozess Schritt für Schritt nachvollzogen werden.

Ziel eines fächerverbindenden Unterrichts von Deutsch und Medien und Informatik ist es, den Entscheidungsprozess, ob eine Eingabe (Satz) grammatikalisch korrekt ist, selbst zu analysieren und zu formalisieren. Die Ableitung mit Hilfe einer formalen Grammatik kann mit einem Algorithmus beschrieben und so dem Computer als Aufgabe übertragen werden.

Abstrakte Automaten

Ein weiteres Beschreibungsmodell für formale Sprachen, welches sich stärker an der Arbeitsweise einer Maschine orientiert, ist der abstrakte Automat. Dieser beschreibt vereinfacht die Arbeitsweise eines digitalen Rechners mit Zuständen und Zustandsübergängen. Die Reduktion eines Computers auf ein abstraktes Modell mit ganz wenige Bestandteilen und primitiven Funktionen ist eine wichtige Basis für andere Bereiche der theoretischen Informatik wie die Komplexitäts- und Berechenbarkeitstheorie. "Abstrakt" deshalb, da sie jegliche praxisrelevante Rahmenbedingungen (Platzbedarf, Kosten, Umgebungstemperaturen, Störungsquellen usw.) ignorieren und davon abstrahieren.

Ein Automat ist eine formale Beschreibung von Zuständen, Ein- bzw. Ausgaben und teilweise eines Speichers (in Form eines unendlich langen Bands oder Stapels) von dem bzw. in den geschrieben und gelesen werden kann. Man unterscheidet verschieden Automatentypen, die wiederum unterschiedliche Sprachklassen der Chomsky-Hierarchie beschreiben können. Der allgemeinste Automatentyp ist die nach ihrem Erfinder benannte Turingmaschine für Typ 0 Sprachen. Eine Turingmaschine hat eine Steuereinheit, welche auf einem potentiell unendlich langem Band Zeichen liest bzw. schreibt. Dabei kann der Lese- und Schreibkopf immer nur genau ein Zeichen auf dem Band gleichzeitig verarbeiten (siehe Abb. 5). Die Steuereinheit legt dabei fest, wie sich die Maschine verhalten soll.

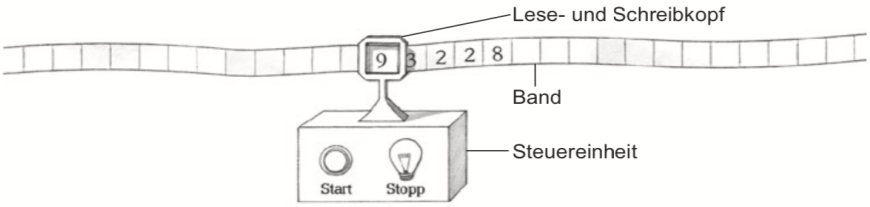
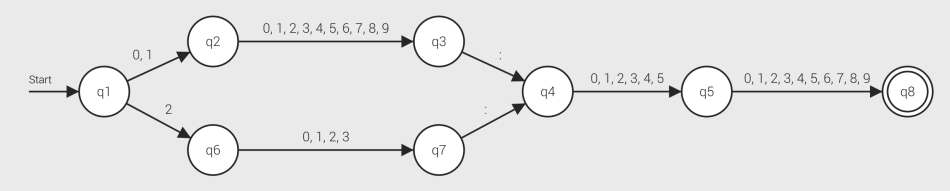


Abbildung 5: Darstellung einer Turing-Maschine
Aus: Dem Computer ins Hirn geschaut von Eckart Zitzler, Springer 2017, S.17

So vereinfacht dieser Automat auch ist, entspricht die Turing-Maschine der Leistungsfähigkeit aller heutiger Computer, Smartphones und Supercomputer. D.h. alles was ein heutiger Computer berechnen kann, lässt sich auch von einer Turing-Maschine erledigen - auch wenn dies extrem ineffizient wäre. Die Turing-Maschine bleibt ein theoretisches Modell, auch wenn zur Veranschaulichung die Maschine durchaus auch physisch konstruiert wurde. Der Kauf von Turing-Maschinen im lokalen Elektronik-Fachmarkt ist dennoch keine Option.

Zur Beantwortung der Frage, ob eine Eingabe zu einer Sprache gehört, ist nur eine Ausgabe Ja oder Nein nötig. Man spricht bei dieser einfachen Form von Automaten von Akzeptoren, da sie selbst keine Ausgabe generieren. Die folgende Abbildung zeigt den Übergangsgraph eines Automaten für die Sprache der Uhrzeiten der Digitaluhr.



Die Kreise stellen Zustände dar. Der Automat befindet sich zu jedem Zeitpunkt in genau einem Arbeitszustand (hier q1 bis q8). Für die Eingabe "23:15" durchläuft der Automat die folgende Zustandsfolge:

Dazu wird beim mit "Start" gekennzeichneten Zustand q1 begonnen und das erste Zeichen des Eingabeworts gelesen. Von "23:15" ist das die "2". Nach dem Graphen muss es mit einer "2" vom Zustand q1 zum Zustand q6 weiter gehen. Mit der verbleibenden Eingabe wird die Arbeit fortgesetzt. Endet die Verarbeitung in einem doppelt eingekreisten Zustand, wird das Ergebnis "akzeptiert" in allen anderen Fällen "nicht akzeptiert" lauten. Ist für ein bestimmtes Zeichen keine Verbindung im Diagramm vorhanden, wird die Abarbeitung gestoppt und ebenfalls ein "nicht akzeptiert" ausgegeben.

Umsetzung in der Volksschule:

Am Beispiel der Automaten kann die allgemeine Arbeitsweise von Computern thematisiert werden. Die Verknüpfung mit Sprache und der Entscheidung, ob ein Wort zur Sprache gehört, ist für die Volksschule aber wohl sehr anspruchsvoll. Es kann an einem Beispiel für eine einfache formale Sprache die Arbeitsweise des Automaten mit einem Werkzeug wie FLACI.com veranschaulicht und diskutiert werden. Im Lehrmittel Connected 2 (S. 30) des LMVZ werden Automaten als Arbeitsmodell von Computern eingeführt und voraussichtlich auf den höheren Stufen in Connected 3 / 4 intensiver thematisiert.

Lehrplan 21:

MI.2.2.e: verstehen, dass ein Computer nur vordefinierte Anweisungen ausführen kann und dass ein Programm eine Abfolge von solchen Anweisungen ist.

Ein Akzeptor-Automat lässt sich vergleichsweise einfach in eine Programmiersprache überführen. Im folgenden JavaScript-Quelltext wurde für jeden Zustand eine gleichnamige Funktion verwendet, welche als Übergabeparameter die verbleibende Resteingabe entgegennimmt:

```
function q1 (rest){
  if(rest.length == 0) return false; // keine Zeichen mehr
  if(rest[0] == "0" || rest[0] == "1")
    return q2 (rest.slice(1));
  if(rest[0] == "2")
    return q6 (rest.slice(1));
  return false;
}
...
function q8 (rest){
  if(rest.length == 0) return true; // keine Zeichen mehr
  return false;
}
// Start des Automaten mit q1 und dem Eingabewort
// liefert true, wenn das Eingabewort zur Sprache gehört
q1(["2","3",":","1","5"]);
```

Übung: Probieren Sie das Beispiel unter https://programmingwiki.de/Automat aus und testen Sie den Automaten mit verschiedenen Eingaben. Versuchen Sie das Programm nachzuvollziehen und mit dem Zustandsgraphen zu vergleichen.

Damit haben wir gezeigt, dass sich das abstrakte Modell auch ganz konkret in einem Computerprogramm implementieren und ausführen lässt. Das Programm kann entscheiden, ob eine Eingabe zur Sprache der Uhrzeiten gehört oder nicht. Solche Eingabeprüfungen findet man überall, wo Menschen Informationen eingeben müssen. Zum Beispiel bei Formularen auf Webseiten, wo gültige Emailadresse, Telefonnummer oder Kreditkartennummer eingetragen werden müssen. Hat der Entwickler eine entsprechende Prüfung vorgesehen, wird die Seite die Eingabe ablehnen, wenn sie nicht der erwarteten Sprache gültiger Emailadressen oder Kreditkartennummern entspricht.

Mit FLACI kann unter https://flaci.com/Ajiqnuivz mit dem abstrakten Automaten für Uhrzeiten experimentiert werden, ohne sich mit der konkreten Implementation in einer Programmiersprache befassen zu müssen (Abb. 6).

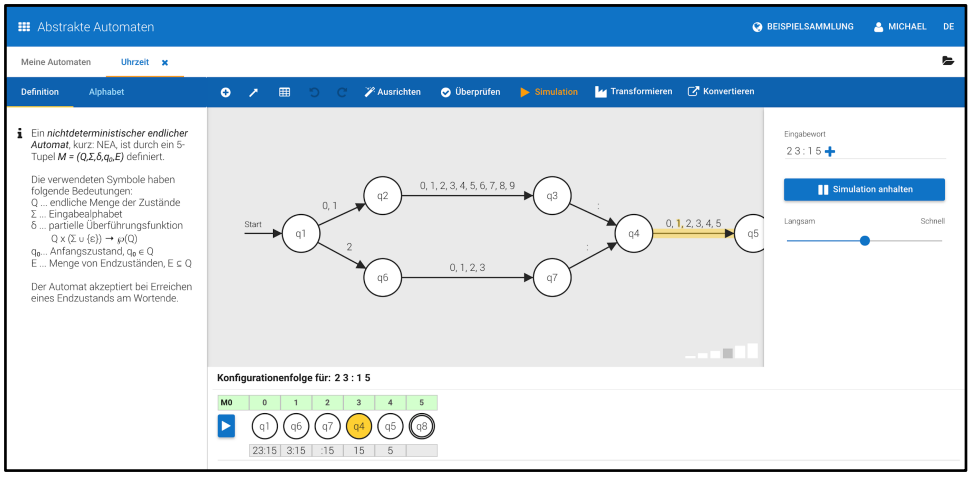


Abbildung 6: Arbeitsweise des Automaten simuliert in FLACI

Ziel eines fächerverbindenden Unterrichts von Deutsch und Medien und Informatik ist es, Sprache auch aus dem Blickwinkel der Informatik zu betrachten und Sprachen formal zu beschreiben. Syntaxdiagrammen, formalen Grammatik und abstrakte Automaten sind unterschiedlich leistungsfähige, formale Beschreibungsmittel für Sprachen. Diese dienen als Grundlage für die automatisierte Verarbeitung von Sprache durch den Computer.

Computerlinguistik

In der Computerlinguistik geht es um die maschinelle Verarbeitung natürlicher Sprache. Problemstellungen wie Spracherkennung oder auch die semantische Analyse von Texten sind mit Blick auf Assistenzsysteme wie Siri und Alexa heute sehr gefragt und dank immer schnellerer Hardware sind sie auch realisierbar und finanzierbar geworden. Im Lehrplan 21 werden im Teilbereich *Informatiksysteme* unter anderem Suchmaschinen und deren Funktionsweise betrachtet. Suchmaschinen haben viel mit der automatischen Analyse von natürlicher Sprache bis hin zu einer semantischen Interpretation zu tun.

Auf Grund der Komplexität und Mehrdeutigkeit natürlicher Sprachen scheitert man daran, eine formale Grammatik aufzustellen, die sich effizient mit dem Computer verarbeiten lässt. Heute setzt man deshalb für viele Aufgaben Algorithmen aus dem Bereich maschinelles Lernen (künstliche Intelligenz) ein. Mit Hilfe von millionenfachen Sprachbeispielen kann dem Computer ein ähnlich heuristisches Analyseverhalten antrainiert werden, wie auch wir Menschen innerhalb kürzester Zeit im Sprachgebrauch über die grammatikalische Korrektheit von komplexen Sätzen urteilen können (als Sprachgefühl bezeichnet).

Ein Beispiel für diese Art von maschineller Sprachanalyse sind aktuelle Übersetzungsdienste wie Google Translator oder Deepl.com welche mit Hilfe neuronaler Netze eine Textanalyse und Übersetzung in eine andere Sprache vornehmen können. Dazu wurden viele Millionen Beispiele menschlicher Übersetzungen verwendet, um von diesen zu lernen. Man spricht von überwachtem Lernen, da bei den vorgelegten Beispielen die zu gewünschte Lösung bereits bekannt ist und das System darauf optimiert werden kann, möglichst geringe Abweichungen zur Musterlösung zu liefern. Das Google Multilingual Neural Machine Translation

System bekam beispielsweise unzählige Texte verschiedener Sprachen und deren Übersetzungen in Englisch und umgekehrt. Interessant bei diesem Ansatz ist es, dass auch sogenannte Zero-Shot-Übersetzungen durch das System möglich wurden. Damit ist gemeint, dass eine Übersetzung direkt von Japanisch nach Koreanisch möglich ist, obwohl dem System nie direkte Übersetzungen in diese beiden Sprachen zum Erlernen vorgelegt wurden (https://ai.googleblog.com/2016/11/zero-shot-translation-with-googles.html). Die Entwickler des Systems gehen davon aus, dass im neuronalen Netzwerk eine Art Meta-Sprache entstanden ist, die zur Übersetzung verwendet wird. Damit wird deutlich, dass selbst die Entwickler nur noch bedingt nachvollziehen können, wie genau das erlernte Wissen in einem neuronalen Netzwerk abgebildet und angewendet wird. Ähnlich wie wir grammatikalische Fehler in unserer Muttersprache zwar schnell erkennen aber nicht immer gut begründen oder anderen erklären können.

Das Beispiel zeigt, dass sich Alternativen zu den bisher eingeführten Beschreibungswerkzeuge für Sprachen (Syntaxdiagramme, formale Grammatiken, abstrakte Automaten) für natürliche Sprachen finden lassen, die mit Wahrscheinlichkeiten statt mathematisch präzisen Definitionen für Sprachen arbeiten.

Es gibt heute vielfältige Systeme, die in der Lage sind, Texte zu analysieren und relevante Informationen herauszulesen und für eine Weiterbearbeitung aufzubereiten. So gibt es im Netz einige Dienste, die automatisierte Textzusammenfassungen erstellen, um Zeit beim Lesen zu sparen. Die Textbeiträge auf Facebook oder Gmail werden analysiert, um beispielsweise passende Werbung für den Nutzenden auszuwählen, unerwünschte Nachrichten herauszufiltern usw. Jahresberichte börsennotierter Unternehmen werden heute mit Hilfe von Algorithmen automatisiert gelesen und mit den jeweiligen Aktienkursverläufen in der Unternehmensgeschichte abgeglichen. Stark vereinfacht: welche Informationen im Jahresbericht korrelieren mit einem darauffolgenden Kursanstieg, womit sich Prognosen zum weiteren Kursverlauf erstellt lassen. Neben der Analyse von geschriebenem Text ist zunehmend auch die gesprochene Sprache mit Speech-Recognition ein grosses Thema. Die Sprachassistenten wie Siri, Alexa oder Cortana müssen zusätzlich in möglichst kurzer Zeit die Eingabe der Nutzenden analysieren und darauf reagieren. Die Datenmenge ist deutlich grösser als bei geschriebenem Text, die Analyseverfahren mit maschinellem Lernen sind aber vergleichbar aufgebaut. Bisher war es bei mobilen Geräten meist unabdingbar, dass für die Analyse von Audiodaten oder für Übersetzungen eine Internetverbindung und die deutlich stärkere Rechenleistung von Serversystemen nötig war. Inzwischen können beide Aufgaben (mit teilweise etwas schlechterer Qualität) von der aktuellen Hardware bereits vom Gerät verarbeitet werden. Dies ist aus Datenschutzperspektive eine positive Entwicklung, da die Sprachdaten das Gerät nicht mehr verlassen müssen und so auch nicht von Unternehmen oder Geheimdiensten mitgeschnitten werden können.

Umsetzung in der Volksschule:

Anhand von Beispielen können die Leistungsfähigkeit aber eben auch die Grenzen von automatisierten Textanalysesystemen exploriert werden. Dazu kann ein englischer, koreanischer und ein polnischer Wikipedia-Artikel automatisch mit einem Übersetzungsdienst wie deepl.com ins Deutsche übersetzt werden. Obwohl die Qualität in den vergangenen Jahren stark zugenommen hat, entstehen immer noch genügend zu diskutierende Fehler.

In älteren Word-Versionen gab es im Menü Extras eine AutoZusammenfassung von Texten, welche jedoch in den aktuellen Versionen nicht mehr angeboten wird. Ein entsprechendes online Werkzeug kann exemplarisch gezeigt und die Ergebnisse gemeinsam diskutiert werden, um auch hier über die Grenzen der Algorithmen zu sprechen.

Mit Scratch lassen sich ebenfalls einzelne Wörter oder kurze Sätze in verschiedenste Sprachen automatisiert übersetzen (basierend auf den Google Übersetzungsdienst). In der Kombination mit weiteren Scratch-Blöcken lassen sich so einfache Vokabeltrainer im Unterricht selbst herstellen.

Ziel eines fächerverbindenden Unterrichts von Deutsch und Medien und Informatik ist es, das Bewusstsein zu schärfen, dass Maschinen von uns geschriebene oder gesprochene Sprache etwa in sozialen Medien analysieren und auswerten. Sprachanalyse und Spracheingabe basierend auf heuristischen Verfahren deren Ergebnisse nicht immer dem entsprechen, was wir erwarten.

Textgeneratoren und Sprachsynthese

Der umgekehrte Weg, digitale Texte mit dem Computer zu erzeugen oder vorlesen zu lassen, ist ebenfalls ein komplexes Thema in der Informatik. Bereits 1966 entwickelte Joseph Weizenbaum das Computerprogramm ELIZA, welches man als Vorläufer eines heutigen Chat-Bots verstehen kann. Man konnte sich mit ELIZA schriftlich in natürlicher Sprache über ein Terminal unterhalten und das System lieferte menschlich wirkende Antworten. Das System analysierte die Eingabe des Benutzers und generierte daraufhin eine Rückfrage nach einem einfachen Schema.

Heute werden Algorithmen eingesetzt, um datenlastige Texte wie Sportberichte für Zeitungen vollautomatisiert zu generieren. Auch in der Schule kennen wir inzwischen Systeme, die für Zeugniseinträge gewisse Formulierungen automatisiert generieren. Börsenberichte werden inzwischen häufig von einem Computer geschrieben, um den jeweils komplexen juristischen Ansprüchen gerecht zu werden und keine Fehler zu begehen. Auch in der Schweiz wurde bereits mit automatisch generierten Texten in Zeitungen experimentiert. Der Tamedia Textroboter Tobi konnte 2018 in der online Ausgabe des Tages-Anzeigers zu 526 verschiedenen Gemeinden des Kantons Bern und Zürich einen individuellen Abstimmungsbericht aus Textbausteinen generieren. Der Leser konnte durch die Angabe seiner Postleitzahl und seiner eigenen Wahlentscheidung einen für ihn zugeschnittenen Bericht lesen. Technisch wird dieser ganz ähnlich gearbeitet haben wie die Experimente zu Sprachgeneratoren in den ersten Kapiteln dieses Textes. Aktuell

liegen die Grenzen dieser Technologie in der Erstellung längerer, zusammenhängender Texte - das Generieren eines ganzen Romans gelingt deshalb noch nicht.

Ähnlich wie bei der Textanalyse von Audioeingaben ist auch der umgekehrte Weg, geschriebenen Text vom Computer vorlesen zu lassen, noch immer eine grosse Herausforderung für die Informatik. Auch hier wird inzwischen stark auf maschinelles Lernen und heuristische Verfahren gesetzt, um möglichst natürlich klingende Sprachsynthese zu erreichen (Google: https://google.github.io/tacotron/publications/tacotron2/index.html, Forscher von Facebook: https://audio-samples.github.io/). In den nächsten Jahren sind spürbare Qualitätsverbesserungen durch die immer leistungsfähigere Hardware zu erwarten. Sogenannte Deep-Fakes, bei denen sowohl die Stimme als auch das Videobild berühmter Persönlichkeiten automatisiert aus zuvor gesammelten Beispieldaten berechnet werden kann, wirft neue gesellschaftliche Fragen auf.

Abschliessend sei hier noch auf den Turing-Test zur Beurteilung künstlicher Intelligenz hingewiesen. Bei diesem Test, den Alan Turing bereits im Jahr 1950 formulierte, geht es darum zu erkennen, ob es sich bei zwei Kommunikationspartnern jeweils um eine Maschine oder einen Menschen handelt. Wäre das Publikum auch nach intensiver Befragung nicht mehr in der Lage, den Menschen von der Maschine zu unterscheiden, wäre der Turing-Test bestanden. Die Befragung kann dabei in Textform (z.B. über einen Chat) erfolgen. Dennoch konnte bislang noch kein System entwickelt werden, welches diesen Test bestehen konnte. Ob es in Zukunft möglich sein wird, bleibt ungewiss. Die Formalisierung von Sprachen als verbindendes Element zwischen Menschen und Computern werden aber wohl immer ein zentraler Aspekt der Informatik bleiben.

Umsetzung in der Volksschule:

Es lohnt sich mit den Schülerinnen und Schülern über die stetig fortschreitende Automatisierung in der Informationsgesellschaft zu sprechen und ihre Folgen, Vor- und Nachteile zu diskutieren. Das Schreiben von journalistischen Texten wurde lange Zeit als Beispiel für "was Computer eben nicht können" verwendet. Ebenso das Komponieren von Musik oder die Gestaltung von Kunst werden zunehmend auch mit Algorithmen nachgebildet, die immer besser darin werden, menschliches Verhalten nachzuahmen.

Ein eindrückliches Beispiel der Automatisierung ist die in PowerPoint ab Version 2016 integrierte "Folien Schnellstarter"-Funktion. Beim Erstellen einer neuen Präsentation sucht PowerPoint vollautomatisch zu einem beliebigen Thema Informationen und Bilder aus dem Internet zusammen und generiert einen PowerPoint-Foliensatz zum Ausbauen.

Mit einem Werkzeug wie Scratch lassen sich Sprachein- und -ausgabe, einfach Satzgeneratoren oder eine Chat-Bot selbst erstellen. Am konkreten Beispiel kann man so auf der einen Seite die Leistungsfähigkeit der heutigen Technik erkunden und gleichzeitig die Grenzen und Herausforderungen bei der Mensch-Maschinen-Interaktion durch Sprache erfahrbar machen. Selbst ein einfacher Chat-Bot wird

schnell ein komplexes Werk, möchte man nur auf einige Fragen mit passenden Antworten reagieren.

Lehrplan 21:

MI.1.1.f: können Chancen und Risiken der zunehmenden Durchdringung des Alltags durch Medien und Informatik beschreiben (z.B. Globalisierung, Automatisierung, veränderte Berufswelt, ungleiche Möglichkeiten zum Zugang zu Information und Technologie).

Ziel eines fächerverbindenden Unterrichts von Deutsch und Medien und Informatik ist es, die gesellschaftlichen Auswirkungen der zunehmenden Automatisierung zu diskutieren. Die natürliche Sprache wird als Domäne des Menschen wahrgenommen - wenn Computer zunehmen wie Menschen kommunizieren und reagieren stellen sich neue Fragen und Herausforderungen.