

Syntax

Philipp Wendler

Zentralübung zur Vorlesung

„Einführung in die Informatik: Programmierung und Softwareentwicklung“

<https://www.sosy-lab.org/Teaching/2017-WS-InfoEinf/>

Organisatorisches

- Abgabe für 1. Übungsblatt nicht nötig
- Auf Übungsgruppen verteilen:
 - Freitag 12 Uhr und Freitag 14 Uhr haben noch viele Plätze
- Klausuranmeldung freigeschalten
 - Klausurbeginn um 16:30 Uhr
- Feiertage:
 - Übungen regulär am 27.10.2017 und 30.10.2017
 - Keine Vorlesung und Zentralübung am 01.11.2017
 - Keine Übungen am 03.11.2017 und 06.11.2017
 - Abgabe des 2. Übungsblatts regulär am 08.11.2017
- Hinweise zu Übungsabgaben auf Blatt beachten
 - Übungsabgaben nicht als Word-Dokument (→ PDF, .txt, etc.)

Organisatorisches



Wir werden **socrative** nutzen, um während der Zentralübung Multiple-Choice-Fragen zu stellen:

- Jeder Teilnehmer mit Internetzugang kann die Fragen live per App oder Browser beantworten (oder für sich auf dem Papier).
- Wir bekommen live Feedback.

Action required now:

1. Smartphone: installiere die App "Socrative Student" **oder** Laptop: öffne im Browser b.socrative.com/student
2. Betrete den Raum **InfoEinf1718**.
3. Beantworte die erste Frage sofort!

EBNF-Grammatik: Wozu?

Beschreibung von gültigen Worten (und Sätzen) einer Sprache

- Deutsche Sprache
- Bestimmte Teile der deutschen Sprache z.B. Palindrome, Uhrzeit
- Programmiersprache
- ...

Multiple-Choice-Frage



Was kann mit einer EBNF-Grammatik **nicht** beschrieben werden?

- a) syntaktisch korrekte Worte
- b) sinnvolle Worte
- c) syntaktisch korrekte Sätze
- d) sinnvolle Sätze

EBNF-Grammatik: Wie?

- **Terminalsymbole:** „Buchstaben“ in der Sprache
- **Nichtterminalsymbole:** Hilfsbegriffe, um mehrere „Buchstaben“ oder „Buchstabenfolgen“ der Sprache zusammenzufassen
- **Regeln** der Form $A = \text{Ausdruck}$
 - A: Nichtterminalsymbol
 - Ausdruck:
Nichtterminal- und Terminalsymbole verbunden durch Operatoren
 - $E1 E2$
 - $E1 \mid E2$
 - $[E1]$ (= ein $E1$ oder kein $E1$)
 - $\{E1\}$ (= beliebig viele $E1$, auch kein $E1$)
- **Startsymbol**

EBNF-Grammatik: Beispiel für Uhrzeiten (I)

- Eine Uhrzeit wird angegeben durch Angabe der Stunde, gefolgt von einem Doppelpunkt und der Angabe der Minuten.
- ACHTUNG: Korrekte Uhrzeiten gehen von 00:00 bis 23:59

Nichtterminalzeichen

Terminalzeichen

EBNF-Grammatik

`Uhrzeit` = Stunde `":"` Minute
Stunde = `"00"` | `"01"` | ... | `"23"`
Minute = `"00"` | `"01"` | ... | `"59"`

■■ Doppeltes Vorkommen ■■

EBNF-Grammatik: Beispiel für Uhrzeiten (II)

- Eine Uhrzeit wird angegeben durch Angabe der Stunde, gefolgt von einem Doppelpunkt und der Angabe der Minuten.
- **ACHTUNG:** Korrekte Uhrzeiten gehen von 00:00 bis 23:59

EBNF-Grammatik

Uhrzeit = Stunde ":" Minute

Stunde = ("0" NullBisNeun) | ("1" NullBisNeun) |
("2" NullBisDrei)

Minute = NullBisFünf NullBisNeun

NullBisDrei = "0" | "1" | "2" | "3"

NullBisFünf = NullBisDrei | "4" | "5"

NullBisNeun = NullBisFünf | "6" | "7" | "8" | "9"

Multiple-Choice-Frage



Wozu werden EBNF-Grammatiken **im Computer** verwendet?

- a) Bilden syntaktisch korrekter Programme
- b) Überprüfen syntaktisch korrekter Programme
- c) Ausführen syntaktisch korrekter Programme

EBNF-Grammatik - Ableitung: Allgemein

- Beginne mit Startsymbol
- Wiederhole, bis gewünschtes Wort “da steht”:
 - Ersetze ein oder mehrere Nichtterminalsymbole durch die rechte Seite ihrer Regel **oder**
 - Führe einen oder mehrere Operatoren aus

- Beginne mit Startsymbol
- Wiederhole, bis gewünschtes Wort “da steht”
 - Ersetze ein oder mehrere Nichtterminalsymbole durch die rechte Seite ihrer Regel **und**
 - Führe **anschließend** einen oder mehrere Operatoren aus, falls möglich

Lange Ableitung

Kurze Ableitung

Beispiel: Ableitung einer korrekten Uhrzeit: 17:49 (lang)

Uhrzeit

```

Regel
-> Stunde ":" Minute
Regel
-> (("0" NullBisNeun) | ("1" NullBisNeun) |
    ("2" NullBisDrei)) ":" Minute
Op|> "1" NullBisNeun ":" Minute
Regel
-> "1" (NullBisFünf | "6" | "7" | "8" | "9") ":" Minute
Op|> "1" "7" ":" Minute
Regel
-> "1" "7" ":" NullBisFünf NullBisNeun
Regel
-> "1" "7" ":" (NullBisDrei | "4" | "5") NullBisNeun
Op|> "1" "7" ":" "4" NullBisNeun
Regel
-> "1" "7" ":" "4" (NullBisFünf | "6" | "7" | "8" | "9")
Op|> "1" "7" ":" "4" "9"

```

Multiple-Choice-Frage



```
Uhrzeit = Stunde ":" Minute
Stunde = ("0" NullBisNeun) | ("1" NullbisNeun) |
         ("2" NullBisDrei)
Minute = NullBisFünf NullBisNeun
NullBisDrei = "0" | "1" | "2" | "3"
NullBisFünf = NullBisDrei | "4" | "5"
NullBisNeun = NullBisFünf | "6" | "7" | "8" | "9"
```

Was ist ein Schritt in einer kurzen Ableitung?

- a) Stunde -> ("0" NullBisNeun) | ("1" NullbisNeun) | ("2" NullBisDrei)
- b) Stunde -> "1" NullbisNeun
- c) Stunde -> "1" "7"

Beispiel: Ableitung einer korrekten Uhrzeit: 17:49 (kurz)

```

Uhrzeit  = Stunde ":" Minute
Stunde   = ("0" NullBisNeun) | ("1" NullbisNeun) |
          ("2" NullBisDrei)
Minute  = NullBisFünf NullBisNeun
NullBisDrei = "0" | "1" | "2" | "3"
NullBisFünf = NullBisDrei | "4" | "5"
NullBisNeun = NullBisFünf | "6" | "7" | "8" | "9"

```

Uhrzeit

```

Regel  ----> Stunde ":" Minute
Regel, Op | --> "1" NullBisNeun ":" NullBisFünf NullBisNeun
Regel, Op | --> "1" "7" ":" "4" "9"

```

Beispiel: Ableitung einer inkorrekten Uhrzeit: 33:33

Uhrzeit = Stunde ":" Minute

Stunde = ("0" NullBisNeun) | ("1" NullbisNeun) |
("2" NullBisDrei)

Minute = NullBisFünf NullBisNeun

NullBisDrei = "0" | "1" | "2" | "3"

NullBisFünf = NullBisDrei | "4" | "5"

NullBisNeun = NullBisFünf | "6" | "7" | "8" | "9"

Uhrzeit

-> Stunde ":" Minute

-> ???



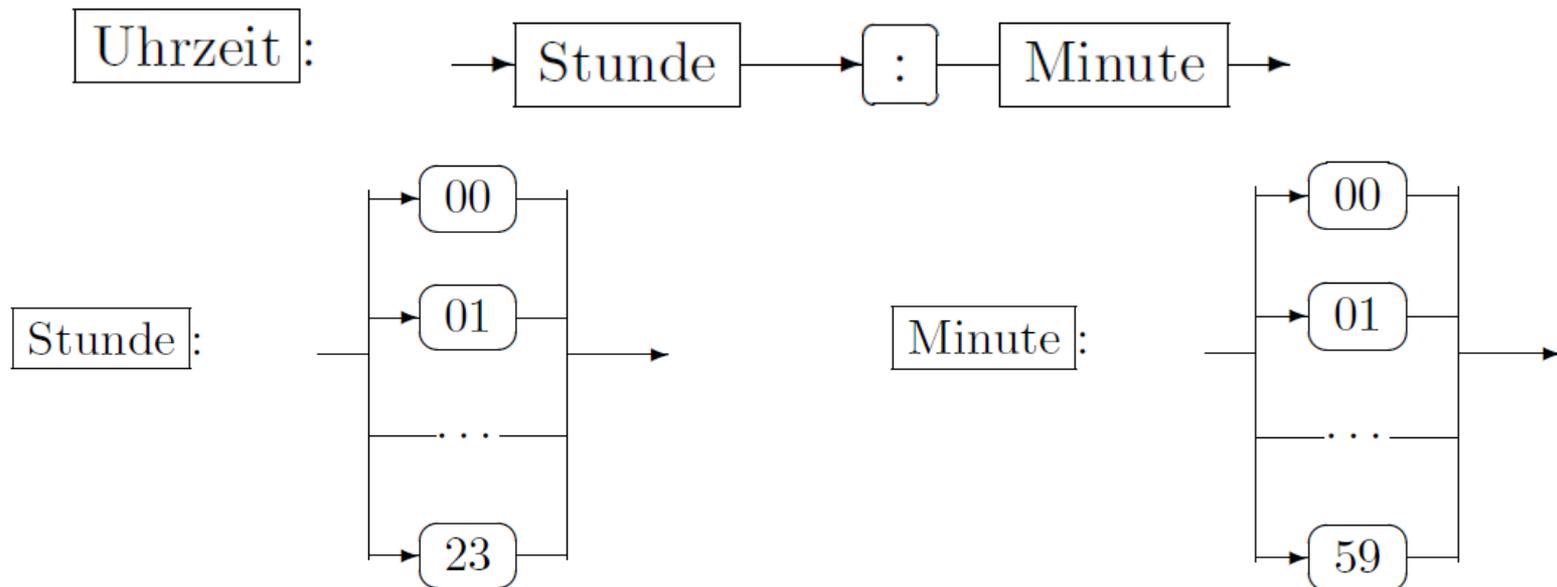
es gibt keine Regel, um "3" an
erster Stelle der Stunde abzuleiten,
d.h. 33:33 nicht ableitbar
d.h. 33:33 ist keine korrekte Uhrzeit

EBNF-Grammatik – Syntaxdiagramm für Uhrzeiten

Uhrzeit = Stunde ":" Minute

Stunde = "00" | "01" | ... | "23"

Minute = "00" | "01" | ... | "59"



Multiple-Choice-Frage



Wie viel hast du nach der Zentralübung verstanden?

- a) ...gar nichts: 0%
- b) ...die Idee, aber die Beispiele sind mir unklar: <50%
- c) ...die Idee und die Beispiele, aber ich könnte es nicht selbst: < 75%
- d) ...ich weiß, wie ich eine Aufgabe lösen müsste: > 75%