Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Übungsblatt 6

Vorlesung Theoretische Grundlagen der Informatik im WS 17/18

Ausgabe 10. Januar 2018
Abgabe 23. Januar 2018, 11:00 Uhr (im Kasten im UG von Gebäude 50.34)

Aufgabe 1

(1 + 2 = 3 Punkte)

Betrachten Sie folgende Sprachen:

$$L_1 = \{\mathtt{0}^n\mathtt{1}^{2n}\mathtt{2}^m \mid n, m \geq 0\}$$
$$L_2 = \{\mathtt{0}^n\mathtt{1}^m\mathtt{2}^{2m} \mid n, m \geq 0\}$$

(a) Zeigen Sie, dass L_1 und L_2 kontextfrei sind.

(b) Ist $L_1 \cap L_2$ kontextfrei? Beweisen Sie Ihre Antwort!

Lösung:

(a)
$$S \to S\mathtt{2} \mid T \qquad\qquad S \to \mathtt{0}S \mid T$$
$$T \to \mathtt{0}T\mathtt{11} \mid \varepsilon \qquad\qquad T \to \mathtt{1}T\mathtt{22} \mid \varepsilon$$

(b) Nein. Es gilt $L = L_1 \cap L_2 = \{\mathtt{0}^n\mathtt{1}^{2n}\mathtt{2}^{4n} \mid n \geq 0\}$. Sei n die vermeintliche Konstante aus dem Pumpinglemma. Wegen $|vwx| \leq n$ enthält vx keine $\mathtt{0}$ oder keine $\mathtt{2}$. Dann enthält uv^2wx^2y entweder zu wenig Nullen oder zu wenig Zweien, um in L zu liegen. Die Sprache L kann also nicht kontextfrei sein.

Aufgabe 2

(2 Punkte)

Gegeben ist folgende kontextfreie Grammatik $G = (\{a, b, c, d\}, \{A, B, C, D, E\}, A, R)$. Die Regelmenge R enthält folgende Regeln:

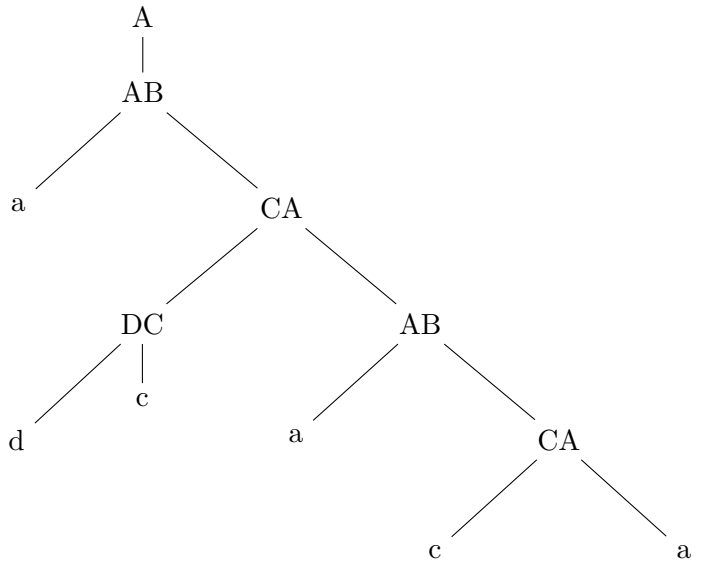
$$A \to AB \mid EC \mid a$$
$$B \to CA$$
$$C \to CC \mid DC \mid c$$
$$D \to d$$
$$E \to CA \mid DA$$

Überprüfen Sie mittels des aus der Vorlesung bekannten CYK-Algorithmus, ob das Wort $w = adcaca$ in $L(G)$ enthalten ist. Falls ja, geben Sie außerdem einen Syntaxbaum für die Ableitung von w an.

a	d	c	a	c	a

Lösung:

A					
	B, E				
A	A,E	B, E			
	B,E	A	A		
	C	B, E		B, E	
A	D	C	A	C	A
a	d	c	a	c	a



Aufgabe 3 (1+3+2 = 6 Punkte)

Sei eine Grammatik G durch $V = \{S, A, B\}, \Sigma = \{0, 1\}$ und folgende Regelmenge R gegeben:

$$S \to \mathtt{0}B \mid \mathtt{1}A$$
$$A \to \mathtt{0} \mid \mathtt{0}S \mid \mathtt{1}AA$$
$$B \to \mathtt{1} \mid \mathtt{1}S \mid \mathtt{0}BB$$

(a) Geben Sie eine zu G äquivalente kontextfreie Grammatik G' in Chomsky-Normalform an.

(b) Beweisen Sie[1], dass G die Sprache aller Wörter in $\{\mathtt{0}, \mathtt{1}\}^+$ erzeugt, bei denen die Anzahl der Nullen gleich der Anzahl an Einsen ist.

(c) Geben Sie eine Grammatik G'' an für die $L(G'')$ die Sprache aller Wörter in $\{\mathtt{0}, \mathtt{1}, \mathtt{2}\}^*$ für die die Anzahl der Nullen gleich die Anzahl der Einsen plus die Anzahl der Zweien ist.

Lösung:

(a) Wir führen zwei neue Variablen X und Y ein. Wir ersetzen die Regel $A \to \mathtt{1}AA$ durch $A \to \mathtt{1}X$ und die Regel $B \to \mathtt{0}BB$ durch $B \to \mathtt{0}Y$. Außerdem führen wir die folgenden neuen Regeln ein:

$$X \to AA$$
$$Y \to BB$$

Desweiteren werden zwei Variablen Y_0 und Y_1 eingeführt, welche jedes Auftreten von $\mathtt{0}$ beziehungsweise $\mathtt{1}$ in R ersetzen. Zusammen mit den neuen Regeln $Y_0 \to \mathtt{0}$ und $Y_1 \to \mathtt{1}$ erhalten wir eine zu G äquivalente Grammatik G' mit Variablen $V' = \{S, A, B, X, Y, Y_0, Y_1\}$ und folgenden

[1] Vergleiche Vorlesung 13, Folie 28.

Regeln

$$\begin{aligned}
S &\to Y_0B \mid Y_1A\\
A &\to Y_0 \mid Y_0S \mid Y_1X\\
B &\to Y_1 \mid Y_1S \mid Y_0Y\\
X &\to AA\\
Y &\to BB\\
Y_0 &\to \mathtt{0}\\
Y_1 &\to \mathtt{1}.
\end{aligned}$$

(b) Wir beweisen, dass für jedes Wort $w \in \{\mathtt{0},\mathtt{1},A,B,S\}^*$ mit $S \overset{*}{\to} w$ gilt $w_0 + w_A = w_1 + w_B$, wobei w_0, w_1, w_A und w_B die Anzahl der Nullen, Einsen, A's beziehungsweise B's in w angibt. Wir machen das per Induktion über $w_0 + w_1$, Wenn $w_0 + w_1 = 1$, dann gilt $w = S$ (weil jede Regel eine Null oder Eins einführt und keine Regel diese wieder löscht) und $w_0+w_A = 0 = w_1+w_B$. Sei also $w_0+w_1 \geq 1$. Dann gibt es ein Wort $v \in \{\mathtt{0},\mathtt{1},A,B,S\}^*$ mit $S \overset{*}{\to} v \to w$. Da jede Regel eine Null oder Eins einführt und keine Regel diese wieder löscht, gilt $v_0+v_1 < w_0+w_1$. Die Induktionsannahme für v ergilt also $v_0 + v_A = v_1 + v_B$.

Wir Unterscheiden nun welche Regel (ℓ, r) beim Übergang $v \to w$ angewendet wurde. Für jede Regel (ℓ, r) aus R gilt $\ell_0 = \ell_1 = 0$ und $\ell_A + r_1 + r_B = \ell_B + r_0 + r_A$. Damit gilt also

$$w_0 + w_A = v_0 + v_A - (\ell_0 + \ell_A) + (r_0 + r_A) = v_1 + v_B - (\ell_1 + \ell_B) + (r_1 + r_B) = w_1 + w_B.$$

Damit ist die Behauptung gezeigt und insbesondere gilt für jedes Wort $w \in L(G)$ dass $w_0 = w_1$.

Andererseits, sei $w \in \{\mathtt{0},\mathtt{1}\}^+$ ein beliebiges Wort mit $w_0 = w_1$.

(c) Analog zu der Grammatik G definieren wir $G'' = (V', \Sigma', R')$ mit $V' = \{S, A, B\}$, $\Sigma' = \{0, 1, 2\}$ und folgenden Regeln:

$$\begin{aligned}
S &\to \varepsilon \mid \mathtt{0}B \mid \mathtt{1}A \mid \mathtt{2}A,\\
A &\to \mathtt{0}S \mid \mathtt{1}AA \mid \mathtt{2}AA,\\
B &\to \mathtt{1}S \mid \mathtt{2}S \mid \mathtt{0}BB\}
\end{aligned}$$

Analog zu Aufgabenteil (b) gilt, dass wenn $S \overset{*}{\to} w$ für ein Wort $w \in (V' \cup \Sigma')^*$, so ist $w_A + w_0 = w_B + w_1 + w_2$. Entsprechend, wenn $w \in \{0, 1, 2\}^*$, dann $w_0 = w_1 + w_2$.

Außerdem lässt sich jedes Wort über $\{0, 1, 2\}$ mit so vielen Nullen wie Einsen und Zweien zusammen aus der Grammatik G' ableiten. Man beachte, dass wir die Regel $S \to \varepsilon$ brauchen, um das leere Wort zu erzeugen. Wegen dieser Regel werden Regeln der Form $A \to 0$ und $B \to 1$ beziehungsweise $B \to 2$ nicht benötigt.

Aufgabe 4 (4 Punkte)

Zeigen Sie[2], dass es zu jeder Chomsky-1-Grammatik eine äquivalente Chomsky-1-Grammatik gibt, bei der alle Regeln die Form

$$\begin{aligned}
A &\to C\\
A &\to CD\\
AB &\to CD\\
A &\to a\\
S &\to \varepsilon
\end{aligned}$$

[2]Vergleiche Vorlesung 13, Folie 23.

haben, wobei jeweils $A, B \in V, C, D \in V \setminus \{S\}$ und $a \in \Sigma$.

Lösung:
Sei $G = (\Sigma, V, S, R)$ eine beliebige Chomsky-1-Grammatik. Wir konstruieren aus G eine weitere Chomsky-1-Grammatik $G' = (\Sigma, V', S, R')$ mit $L(G') = L(G)$, so dass $R' \subset (V' \cup V'^2) \times (V' \cup V'^2 \cup \Sigma \cup \varepsilon)$. Das heißt, in G' haben alle Regeln die gewünschte Form. Unser Vorgehen ist analog zur Konstruktion der Chomsky-Normalform zu einer Chomsky-2-Grammatik wie sie in der Vorlesung vorgestellt wurde.

Schritt 1: Unser Ziel ist es, dass alle Regeln auf der rechten Seite nur Variablen oder nur ein Symbol aus Σ enthalten. Füge dazu für jedes Zeichen $a \in \Sigma$ eine neue Variable Y_a ein. Ersetze dann in allen rechten Seiten von R jedes Auftreten eines Zeichen $a \in \Sigma$ durch die entsprechende Variable Y_a. Füge außerdem für jedes $a \in \Sigma$ eine neue Regel $Y_a \to a$ hinzu.

Die so erzeugte Grammatik G_1 erfüllt $L(G_1) = L(G)$, da die einzige Regel mit Y_a auf der linken Seite die Regel $Y_a \to a$ ist, und somit Y_a und a als "gleichwertig" gesehen werden können. Außerdem ist G_1 eine Chomsky-1-Grammatik.

Schritt 2: Unser Ziel ist es, dass alle Regeln auf der linken Seite nicht mehr als zwei Variablen enthalten. Füge dazu für jede Regel (ℓ, r) mit $|\ell| \geq 3$ neue Variablen $C_1, \ldots, C_{|\ell|-2}$ ein. Sei o.B.d.A. (ℓ, r) die Regel $A_1 \ldots A_k \to B_1 \ldots B_t$ mit $t \geq k \geq 3$. Ersetze dann die Regel (ℓ, r) durch die Regeln

$$\begin{aligned} A_1 A_2 &\to A_2 C_1 \\ C_i A_{i+2} &\to A_{i+2} C_{i+1} \quad \text{für } i = 1, \ldots, k-3 \\ C_{k-2} A_k &\to B_1 \ldots B_t. \end{aligned}$$

Die so erzeugte Grammatik G_2 erfüllt wiederum $L(G_2) = L(G)$, da jede neu eingeführte Variable in nur einer Regel auf der linken Seite enthalten ist und deshalb die neuen Regeln zu der entsprechenden alten Regel "zurückverfolgt" werden können. Außerdem ist G_2 wiederum eine Chomsky-1-Grammatik.

Schritt 3: Unser letztes Ziel ist es, dass alle Regeln auf der rechten Seite nicht mehr als zwei Variablen enthalten. Füge dazu für jede Regel (ℓ, r) mit $|r| \geq 3$ neue Variablen $D_1, \ldots, D_{|r|-2}$ ein. Sei O.B.d.A. (ℓ, r) die Regel $\ell \to B_1 \ldots B_k$ mit $k \geq 3$. Ersetze dann die Regel (ℓ, r) durch die Regeln

$$\begin{aligned} \ell &\to B_1 D_1 \\ D_i &\to B_{i+1} D_{i+1} \quad \text{für } i = 1, \ldots, k-3 \\ D_{k-2} &\to B_{k-1} B_k. \end{aligned}$$

Nach den gleichen Überlegungen wie vorher erhalten wir, dass die so erzeugte Grammatik G' wiederum $L(G') = L(G)$ erfüllt. Außerdem ist G' eine Chomsky-1-Grammatik in der jede Regel einer der gewünschten Formen hat.

Aufgabe 5

(3 + 3 = 6 Punkte)

(a) Zeigen Sie, dass die Sprache

$$L = \{\langle G_1, G_2 \rangle \mid G_1, G_2 \text{ kontextfreie Grammatiken mit } L(G_1) \cap L(G_2) = \emptyset\}$$

unentscheidbar ist, indem Sie das PKP auf L reduzieren.

(b) Zeigen Sie, dass die Sprache

$$L = \{\langle G_1, G_2\rangle \mid G_1, G_2 \text{ rechtslineare Grammatiken mit } L(G_1) \cap L(G_2) = \emptyset\}$$

entscheidbar ist, indem Sie einen Algorithmus beschreiben, der L entscheidet.

Hinweis: Sie dürfen verwenden, dass die regulären Sprachen unter Schnitt abgeschlossen sind.

Lösung:

(a) Eine PKP-Instanz ist eine Menge $K = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. Seien G_1, G_2 kontextfreie Grammatiken mit folgenden Regeln

$$S \to x_1\ S\ y_1^R \mid x_2\ S\ y_2^R \mid \ldots \mid x_n\ S\ y_n^R \qquad S \to y_1\ S\ x_1^R \mid y_2\ S\ x_2^R \mid \ldots \mid y_n\ S\ x_n^R$$
$$S \to x_1\ \#\ y_1^R \mid x_2\ \#\ y_2^R \mid \ldots \mid x_n\ \#\ y_n^R \qquad S \to y_1\ \#\ x_1^R \mid y_2\ \#\ x_2^R \mid \ldots \mid y_n\ \#\ x_n^R$$

Dann hat K eine Lösung genau dann, wenn $L(G_1) \cap L(G_2)$ nicht leer ist. Aus der Unentscheidbarkeit des PKP folgt damit, dass auch L unentscheidbar ist.

(b) Aus der Vorlesung ist ein Verfahren bekannt, mit dem sich G_1, G_2 in deterministische endliche Automaten A_1, A_2 mit $L(A_1) = L(G_1), L(A_2) = L(G_2)$ transformieren lassen[3]. Die Automaten können geschnitten werden und man erhält einen DEA A mit $L(A) = L(G_1) \cap L(G_2)$. Nach Entfernen von unerreichbaren Zuständen aus A gilt $L(A) = \emptyset$ genau dann, wenn A keinen akzeptierenden Zustand hat.

Aufgabe 6

(2 + 4 = 6 Punkte)

Zeigen Sie, dass die kontextsensitiven Sprachen den Sprachen der Klasse $\mathsf{NTAPE}(n)$ entsprechen[4], indem Sie in zwei Schritten vorgehen.

(a) Sei L eine kontextsensitive Sprache. Dann existiert eine nichtdeterministische Turingmaschine M, die L mit linearem Platzbedarf akzeptiert.

(b) Sei M eine nichtdeterministische Turingmaschine die die Sprache $L(M)$ mit linearem Platzbedarf akzeptiert. Dann existiert eine kontextsensitive Grammatik G, so dass gilt $L(G) = L(M)$.

Lösung:
Sei L eine kontextsensitive Sprache, und G eine Typ-1-Grammatik, die L erzeugt. Zeige, dass L von einer nichtdeterministischen Turingmaschine T mit linearem Platzbedarf entschieden werden kann. Die Grammatik G hat konstante Größe und lässt sich damit in $\mathcal{O}(1)$ Platz auf das Band von T schreiben. Sei nun w eine Eingabe für T. Es gilt zu entscheiden, ob $w \in L$ ist. Im Falle $w = \varepsilon$ gilt $w \in L$ genau dann, wenn G die Ableitungsregel $S \to \varepsilon$ enthält. Andernfalls wird wie folgt vorgegangen. Enthält w ein Teilwort v, so dass G eine Regel $u \to v$ enthält, ist $u \to v$ eine *mögliche* Ableitung. Die NTM T sucht in jedem Schritt erst alle möglichen Ableitungen und wählt danach eine nichtdeterministisch aus. Dann wird v durch u ersetzt. Steht irgendwann nur noch S auf Band hält T und akzeptiert. Da T nur Ableitungen vornimmt, die durch G kodiert werden gilt

[3] Vergleiche Vorlesung 13, Folie 13ff.
[4] Vergleiche Vorlesung 13, Folie 18.

$w \in L$ falls T akzeptiert. Umgekehrt akzeptiert T falls $w \in L$ gilt, z.B. indem die Ableitungen, die von S zu w führen in umgekehrter Reihenfolge vorgenommen werden. Man beachte, dass die Eingabe beim umgekehrten Anwenden der Ableitungen immer kürzer wird. Zusammen mit der in $\mathcal{O}(1)$ kodierten Grammatik wird so also höchstens linear viel Platz benötigt.

Sei nun $L \in \mathsf{NTAPE}(n)$ und $T = (Q, \Sigma, \Gamma, s, \delta, F)$ eine NTM, die L akzeptiert. Wir konstruieren eine Grammatik G, die gerade die Sprache L erzeugt. Das Terminalalphabet von G sei gerade Σ. Die Idee ist es, die Kopfbewegungen und Zustandsübergänge von T als Ableitungen in G zu kodieren. Als erstes brauchen wir ein Band, das aus mindestens einem Bandsymbol besteht. Jedes Bandsymbol ist ein Tripel der Form $(Q \cup \{\bot\}) \times \Sigma \times \Gamma$. Dabei steht der erste Wert für den Zustand, bzw. $\bot$ falls der Lesekopf nicht auf diesem Zeichen steht. Der zweite Wert steht für das Eingabesymbol auf dem Eingabeband und der dritte Wert für das Bandsymbol auf dem Arbeitsband.

$$\begin{aligned} \forall \sigma \in \Sigma: \quad & S \;\to\; (s, \sigma, \sqcup) \mid (s, \sigma, \sqcup)\, B \\ \forall \sigma \in \Sigma: \quad & B \;\to\; BB \mid (\bot, \sigma, \sqcup) \end{aligned}$$

Man beachte, dass der Kopf von T zunächst im Zustand s auf dem linkesten Symbol der Eingabe steht. Außerdem lässt sich durch die obigen Regeln jedes Wort auf dem Eingabeband erzeugen. Die Idee ist es nun, dass eine solche Eingabe genau dann ableitbar sein soll, wenn M diese akzeptiert. Betrachte deshalb für alle $q \in Q \setminus F, \sigma, \sigma' \in \Sigma, \gamma, \gamma' \in \Gamma$ den Übergang $\delta(q, \gamma) = (\hat{q}, \hat{\gamma}, d)$ mit $d \in \{L, N, R\}$ und füge folgende Regeln hinzu:

$$\begin{aligned} (\bot, \sigma', \gamma')\ (q, \sigma, \gamma) &\;\to\; (\hat{q}, \sigma', \gamma')\ (\bot, \sigma, \hat{\gamma}) && \text{falls } d = L \\ (q, \sigma, \gamma) &\;\to\; (\hat{q}, \sigma, \hat{\gamma}) && \text{falls } d = N \\ (q, \sigma, \gamma)\ (\bot, \sigma', \gamma') &\;\to\; (\bot, \sigma, \hat{\gamma})\ (\hat{q}, \sigma', \gamma') && \text{falls } d = R \end{aligned}$$

Sobald ein akzeptierender Zustand $f \in F$ erreicht ist, muss das ursprüngliche Wort entpackt werden.

$$\begin{aligned} (f, \sigma, \gamma) &\;\to\; \sigma \\ (\bot, \sigma, \gamma)\ \sigma' &\;\to\; \sigma\ \sigma' \\ \sigma'\ (\bot, \sigma, \gamma) &\;\to\; \sigma'\ \sigma \end{aligned}$$

Ist $\varepsilon \in L$ fügen wir außerdem noch die Regel $S \to \varepsilon$ hinzu.

Aufgabe 7

(2 + 2 + 1 + (2) = 5 + (2) Punkte)

Sei $L = \{\mathtt{a}^n \mid n \text{ ist Primzahl}\}$.

(a) Zeigen Sie, dass der Typ von L höchstens Eins ist.

(b) Zeigen Sie, dass L von Typ Eins ist.

(c) Welchen maximalen Typ hat die Sprache L^*? Beweisen Sie Ihre Behauptung!

(d) Welchen Typ haben die Sprachen L^3 und L^6?

Bonusfrage für Interessierte, ohne Beweis. Bei dieser Aufgabe dürfen Sie gerne bekanntes (Un-)Wissen recherchieren und zitieren.

Lösung:

(a) Wir gehen ähnlich dazu vor, wie wir gezeigt haben, dass L nicht regulär ist. Der einzige Unterschied ist, dass beim Pumpinglemma für kontextfreie Sprachen die Zerlegung und das „Aufpumpen“ ein bisschen anders funktioniert.

Sei n eine Konstante wie vom Pumpinglemma gefordert und $z = uvwxy \in L$ ein Wort mit $p = |z| \geq n$, so dass $k = |vx| \geq 1$, $|vxw| \leq n$. Dann gilt für $i = p + 1$ dass uv^iwx^iy Länge $p + pk = p(k+1)$ hat, und also $uv^iwx^iy \notin L$ gilt. Damit ist L nicht kontextfrei bzw. höchstens kontextsensitiv.

(b) Mit einer Turingmaschine lässt sich auf naive Weise mit linearem Platzbedarf testen, ob eine gegebene Zahl n eine Primzahl ist, indem für alle Werte m bis $\sqrt(n)$ geprüft wird, dass m kein Teiler von n ist. Aus Aufgabe 6 (b) folgt dann direkt, dass es eine kontextsensitive Grammatik gibt, die L erzeugt. Damit hat L mindestens Typ Eins, und zusammen mit Teilaufgabe (a) ergibt sich, dass L genau Typ Eins hat.

(c) Es gilt $L^* = \Sigma^* \setminus \{\mathtt{a}\}$, denn jede Zahl außer Eins lässt sich als Summe von Primzahlen darstellen. Jede gerade Zahl lässt sich durch Aufsummieren von Zwei erzeugen, für jede ungerade Zahl größer gleich Drei kann man etwa einmal die Drei und dann ausreichend Zweien verwenden. Außerdem enthält L^* das leere Wort, womit nur das Wort $\mathtt{a}$ in L^* enthalten ist. Damit ist L^* eine koëndliche Sprache, für die wir auf Übungsblatt 2, Aufgabe 3 (b) Regularität gezeigt hatten.

(d) Laut der Goldbachvermutung lässt sich jede natürliche Zahl als Summe von höchstens drei Primzahlen darstellen. Damit hätte L^3 Typ 3. Die Goldbachvermutung ist allerdings eines der bekanntesten ungelösten Probleme der Mathematik. Der maximale Typ von L^3 ist also unbekannt!

Ein schwächeres Ergebnis stammt von Terence Tao[5], und zwar, dass jede ungerade Zahl größer als Eins die Summe von höchstens fünf Primzahlen ist. Jede Zahl lässt sich dann also Summe von höchstens sechs Primzahlen darstellen. Das heißt, dass L^6 Typ 3 hat.

[5]Siehe Artikel *Every Odd Number Greater Than 1 is the Sum of at Most Five Primes.*