HA-Lösung

TA-Lösung

Einführung in die theoretische Informatik – Aufgabenblatt 8

Beachten Sie: Soweit nicht explizit angegeben, sind Ergebnisse stets zu begründen!

Hausaufgaben: Abgabe bis zum 15.06.2016 (Mittwoch) um 12:00

Aufgabe 8.1 CYK-Algorithmus 2P+2P

Wir betrachten die Grammatik $G = (\{S, T, U, A, B, C\}, \{a, b, c\}, P, S)$ in CNF mit den folgenden Produktionen P:

$$S \to TS \mid CT \mid a \qquad A \to a$$
$$T \to AU \mid TT \mid c \qquad B \to b$$
$$U \to SB \mid AB \qquad C \to c$$

(a) Bestimmen Sie mit dem CYK-Algorithmus, ob $ccaab \in L(G)$ und $aabcc \in L(G)$. Geben Sie dabei auch die berechneten Tabellen an.

(b) Beschreiben Sie eine Erweiterung des CYK-Algorithmus, mit welcher für ein gegebens $w \in L(G)$ alle Ableitungsbäume bzgl. G berechnet werden können, und wenden Sie dieses Verfahren auf die Wörter aus (a) an.

Lösung:

(a) Nach dem CYK-Algorithmus ergeben sich folgende Berechnungstabelle:

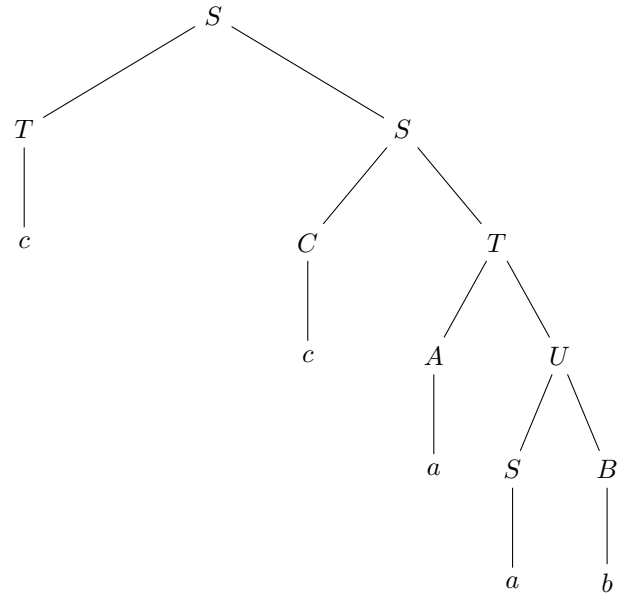
$_{15}\ S, T$				
$_{14}$	$_{25}\ S, T$			
$_{13}\ S$	$_{24}$	$_{35}\ T$		
$_{12}\ S, T$	$_{23}\ S$	$_{34}$	$_{45}\ U$	
$_{11}\ C, T$	$_{22}\ C, T$	$_{33}\ S, A$	$_{44}\ S, A$	$_{55}\ B$
c	c	a	a	b

$_{15}\ S, T$				
$_{14}\ T$	$_{25}$			
$_{13}\ T$	$_{24}$	$_{35}$		
$_{12}$	$_{23}\ U$	$_{34}$	$_{45}\ S, T$	
$_{11}\ S, A$	$_{22}\ S, A$	$_{33}\ B$	$_{44}\ C, T$	$_{55}\ C, T$
a	a	b	c	c

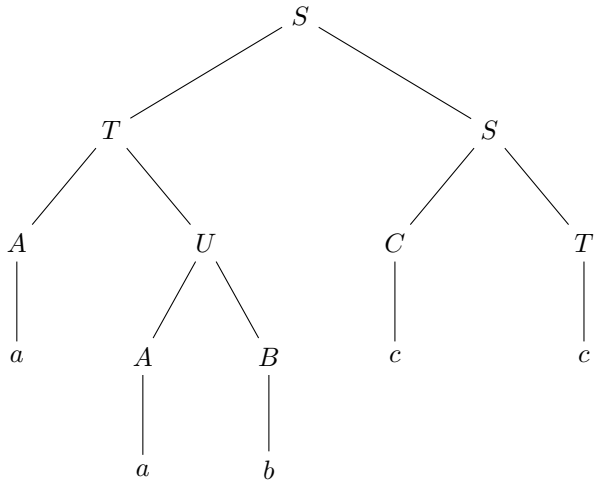
Also ist $ccaab \in L(G)$ und $aabcc \in L(G)$.

(b) Bei der Berechnung von V_{ij} annotiert man die Elemente $X \in V_{ij}$ mit den verwendeten Regeln und dem Index. Im Basisfall $X \in V_{ii}$ werden alle Variablen der Produktion $X \to w_i$ annotiert. Sei $X \to YZ$ und sei $Y \in V_{ik}$ und $Z \in V_{(k+1)j}$. Dann wird $X \in V_{ij}$ mit $(X \to YZ, k)$ annotiert.

$_{15}\ (T \to TT, 1), (T \to TT, 2), (S \to TS, 1), (S \to CT, 1)$				
$_{14}$	$_{25}\ (T \to TT, 2), (S \to CT, 2)$			
$_{13}\ (S \to TS, 2), (S \to TS, 1)$	$_{24}$	$_{35}\ (T \to AU, 3)$		
$_{12}\ (T \to TT, 1), (S \to CT, 1)$	$_{23}\ (S \to TS, 2)$	$_{34}$	$_{45}\ (U \to SB, 4), (U \to AB, 4)$	
$_{11}\ (C \to c), (T \to c)$	$_{22}\ (C \to c), (T \to c)$	$_{33}\ (S \to a), (A \to a)$	$_{44}\ (S \to a), (A \to a)$	$_{55}\ (B \to b)$
c	c	a	a	b



15 $(T \to TT, 4)$, $(T \to TT, 3)$, $(S \to TS, 3)$				
14 $(T \to TT, 3)$	25			
13 $(T \to AU, 1)$	24	35		
12	23 $(U \to AB, 2)$, $(U \to SB, 2)$	34	45 $(T \to TT, 4)$, $(S \to CT, 4)$	
11 $(S \to a)$, $(A \to a)$	22 $(S \to a)$, $(A \to a)$	33 $(B \to b)$	44 $(C \to c)$, $(T \to c)$	55 $(C \to c)$, $(T \to c)$
a	a	b	c	c



Aufgabe 8.2 **Infix-Abschluss** **3P+1P**

Wir betrachten eine Zerlegung $z = uvw$ für das Wort z. Man bezeichnet u dann als *Präfix*, w als *Suffix* und v als *Infix*. Der Infixabschluss einer Sprache L ist dann definiert als $L_{\text{infix}} = \{v \mid \exists u, w \in \Sigma^*.\ uvw \in L\}$.

(a) Geben Sie eine allgemeine Konstruktion an, die aus einer CFG G in CNF mit $L = L(G)$ eine Grammatik G' mit $L(G') = L_{\text{infix}}$ erzeugt.

(b) Wenden Sie Ihr Verfahren auf die Grammatik aus HA8.1 an.

Lösung:

(a) Sei $G = (V, \Sigma, P, S)$ CFG in CNF mit $L = L(G)$ und $L_{\text{infix}} = \{v \in \Sigma^* \mid \exists u, w \in \Sigma^* \colon uvw \in L(G)\}$.

Idee: Es gilt: $v \in L_{\text{infix}}$ gdw. es gibt einen Ableitungsbaum zu uvw bzgl. G für $u, w \in \Sigma^*$.

Wir schneiden jetzt einen rechten und einen linken Teil vom Ableitungsbaum zu uvw ab. Hierzu konstruieren wir $G' = (V \uplus V' \uplus \{S''\}, \Sigma, P \uplus P', \overline{S})$ wie folgt:

- $V' = \{{}^\bullet X^\bullet, X^\bullet, {}^\bullet X \mid X \in V\}$
- $P' = \{X^\bullet \to Y Z^\bullet, X^\bullet \to Y^\bullet, {}^\bullet X \to {}^\bullet Y Z, {}^\bullet X \to {}^\bullet Z, {}^\bullet X^\bullet \to {}^\bullet Y Z^\bullet, {}^\bullet X^\bullet \to {}^\bullet Y^\bullet, {}^\bullet X^\bullet \to {}^\bullet Z^\bullet \mid X \to Y Z \in P\} \cup \{{}^\bullet X^\bullet \to a, X^\bullet \to a, {}^\bullet X \to a \mid X \to a \in P\} \cup \{\overline{S} \to S \mid S^\bullet \mid {}^\bullet S \mid {}^\bullet S^\bullet \mid \varepsilon\}$

(b)

$$\overline{S} \to S \mid S^\bullet \mid {}^\bullet S \mid {}^\bullet S^\bullet \mid \varepsilon$$

$$\begin{array}{ll} S \to TS \mid CT \mid a & A \to a \\ T \to AU \mid TT \mid c & B \to b \\ U \to SB \mid AB & C \to c \end{array}$$

$$\begin{array}{ll} S^\bullet \to TS^\bullet \mid CT^\bullet \mid T^\bullet \mid C^\bullet \mid a & A^\bullet \to a \\ T^\bullet \to AU^\bullet \mid TT^\bullet \mid A^\bullet \mid T^\bullet \mid c & B^\bullet \to b \\ U^\bullet \to SB^\bullet \mid AB^\bullet \mid S^\bullet \mid A^\bullet & C^\bullet \to c \end{array}$$

$$\begin{array}{ll} {}^\bullet S \to {}^\bullet TS \mid {}^\bullet CT \mid {}^\bullet S \mid {}^\bullet T \mid a & {}^\bullet A \to a \\ {}^\bullet T \to {}^\bullet AU \mid {}^\bullet TT \mid {}^\bullet U \mid {}^\bullet T \mid c & {}^\bullet B \to b \\ {}^\bullet U \to {}^\bullet SB \mid {}^\bullet AB \mid {}^\bullet B & {}^\bullet C \to c \end{array}$$

$$\begin{array}{ll} {}^\bullet S^\bullet \to {}^\bullet TS^\bullet \mid {}^\bullet CT^\bullet \mid {}^\bullet S^\bullet \mid {}^\bullet T^\bullet \mid {}^\bullet C^\bullet \mid a & {}^\bullet A^\bullet \to a \\ {}^\bullet T^\bullet \to {}^\bullet AU^\bullet \mid {}^\bullet TT^\bullet \mid {}^\bullet U^\bullet \mid {}^\bullet T^\bullet \mid {}^\bullet A^\bullet \mid c & {}^\bullet B^\bullet \to b \\ {}^\bullet U^\bullet \to {}^\bullet SB^\bullet \mid {}^\bullet AB^\bullet \mid {}^\bullet B^\bullet \mid {}^\bullet S^\bullet \mid {}^\bullet A^\bullet & {}^\bullet C^\bullet \to c \end{array}$$

Aufgabe 8.3 **Pushdown Automaten / Kellerautomaten** **2P+2P+2P**

Konstruieren Sie für die folgenden Sprachen jeweils einen Kellerautomaten. Der Automat soll mit **leerem Stack** akzeptieren. Geben Sie zusätzlich für jeden Automaten jeweils ein nicht-leeres Wort w mit akzeptierendem Lauf an.

(a) $L_1 = \{a^n b^{3n} \mid n \geq 0\}$

(b) $L_2 = \{a^n b^m \in \{a,b\}^* \mid n \leq m \leq 2n\}$

(c) $L_3 = \{w \in \{a,b\}^* \mid 2\,|w|_a = 3\,|w|_b\}$

Lösung:

(a) $qX \xrightarrow{a} qBBB \quad qX \xrightarrow{\varepsilon} q\varepsilon \quad qB \xrightarrow{a} qBBBB \quad qB \xrightarrow{b} p\varepsilon \quad pB \xrightarrow{b} p\varepsilon$

$(q, abbb, X) \to (q, bbb, BBB) \to (p, bb, BB) \to (p, b, B) \to (p, \varepsilon, \varepsilon)$

(b) Idee: Für jedes a lege nichtdeterministisch entweder ein oder zwei b auf den Stack und überprüfe dann, ob die geratene Anzahl von bs mit der gegebenen übereinstimmt.

$$qX \xrightarrow{a} qB \quad qX \xrightarrow{a} qBB \quad qX \xrightarrow{\varepsilon} q\varepsilon$$
$$qB \xrightarrow{a} qBB \quad qB \xrightarrow{a} qBBB \quad qB \xrightarrow{b} p\varepsilon \quad pB \xrightarrow{b} p\varepsilon$$

$(q, aabbb, X) \to (q, abbb, BB) \to (q, bbb, BBB) \to (p, bb, BB) \to (p, b, B) \to (p, \varepsilon, \varepsilon)$

(c) Idee: Verwende Stack als (unären) Zähler und benutze explizites Bottom-Symbol, um auf 0 zu testen. Für jedes a zähle um 2 (codiert als XX) hoch und für jedes b ziehe 3 (codiert als YYY) ab.

$$q\bot \xrightarrow{a} qXX\bot \qquad qX \xrightarrow{a} qXXX \qquad qY \xrightarrow{a} p^+\varepsilon$$
$$q\bot \xrightarrow{b} qYYY\bot \qquad qY \xrightarrow{b} qYYYY \qquad qX \xrightarrow{b} p^{--}\varepsilon$$
$$p^+\bot \xrightarrow{\varepsilon} qX\bot \qquad p^+Y \xrightarrow{\varepsilon} q\varepsilon$$
$$p^{--}\bot \xrightarrow{\varepsilon} qYY\bot \qquad p^{--}X \xrightarrow{\varepsilon} p^-\varepsilon$$
$$p^-\bot \xrightarrow{\varepsilon} qY\bot \qquad p^-X \xrightarrow{\varepsilon} q\varepsilon$$
$$q\bot \xrightarrow{\varepsilon} q\varepsilon$$

$(q, abbaa, \bot) \to (q, bbaa, XX\bot) \to (p^{--}, baa, X\bot) \to (p^-, baa, \bot) \to (q, baa, Y\bot) \to (q, aa, YYYY\bot) \to (p^+, a, YYY\bot) \to (q, a, YY\bot) \to (p^+, \varepsilon, Y\bot) \to (q, \varepsilon, \bot) \to (q, \varepsilon, \varepsilon)$

Aufgabe 8.4 **Ogdens Lemma** **2P**

Zeigen Sie mit Hilfe des Lemmas von Ogden (TA7.3), dass die folgende Sprache nicht kontextfrei ist:

$$L = \{a^i b^j c^j \mid i \neq j\}$$

Lösung: Sei L kontextfrei. Dann gilt Ogdens Lemma für L. Sei p entsprechend dem Lemma für L gewählt. Wir wählen $z = a^{p+p!}b^p c^p \in L$, wobei genau c^p markiert sei.

Damit sind in vwx stets höchstens p Zeichen markiert. Nach Ogdens Lemma enthält vx mindestens ein c.

Wir unterscheiden zwei mögliche Fälle für vx:

- $|vx|_b \neq |vx|_c$: Dann gibt es aber eine Ungleichgewicht: $\left|uv^2wx^2y\right|_b \neq \left|uv^2wx^2y\right|_c$ und somit $uv^2wx^2y \notin L$.
- $|vx|_b = |vx|_c$: Angenommen es gibt ein $k > 0$ mit $v = b^k$ und $x = c^k$. Falls dies nicht der Fall wäre, können wir sofort einen Widerspruch mit $i = 2$ erzeugen. Wir wählen nun $i = 1 + \frac{p!}{k} \in \mathbb{N}$ zum aufpumpen. Dann gilt

  $$\left|uv^iwx^iy\right|_a = p + p! = (p-k) + (1 + \frac{p!}{k})k = \left|uv^iwx^iy\right|_b = \left|uv^iwx^iy\right|_c$$

  und somit $uv^iwx^iy \notin L$.

Da jeder Fall einen Widerspruch erzeugt, ist L nicht kontextfrei.

Tutoraufgaben: Besprechung in KW24

Erinnerung: Wir bezeichnen mit $L_\varepsilon(A)$, die Sprache die von einem PDA A mit leerem Stack akzeptiert wird. Weiterhin bezeichnen wir mit $L_F(A)$, die Sprache die von einem PDA A mit Endzuständen akzeptiert wird.

Notation von PDA-Regeln: Anstatt der in den Folien verwendeten Schreibweise $(q, YZ) \in \delta(p, a, X)$ für die Ersetzungsregeln eines PDA, schreibt man alternativ $pX \xrightarrow{a} qYZ$ $(p, q \in Q,\ X, Y, Z \in \Gamma,\ a \in \Sigma)$ oder stellt diese entsprechend als Graph mit Knotenmenge $Q\Gamma^{\leq 2}$ dar, wobei die Kante (pX, qYZ) dann mit a beschriftet ist.

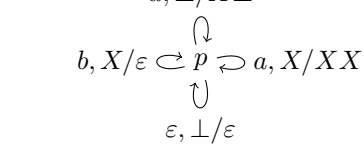
Für den PDA

$$\delta(p, a, \bot) = \{(p, X\bot)\} \qquad \delta(p, a, X) = \{(p, XX)\} \qquad \delta(p, b, X) = \{(p, \varepsilon)\} \qquad \delta(p, \varepsilon, \bot) = \{(p, \varepsilon)\}$$

schreibt man daher alternativ:

$$p\bot \xrightarrow{a} pX\bot \qquad pX \xrightarrow{a} pXX \qquad pX \xrightarrow{b} p \qquad p\bot \xrightarrow{\varepsilon} p$$

oder der stellt diesen entsprechend als Graph mit Knotenmenge Q dar, wobei die Kante (p, q) dann mit "$a, X/YZ$" beschriftet ist (siehe *Hopcroft at al. „Introduction to Automata Theory", Kapitel 6*):



Aufgabe 8.1 Deterministische PDAs

In der Vorlesung haben Sie Sie Lemma 3.65 ohne Beweis gesehen:

Sei $L \subseteq \Sigma^*$. Dann sind äquivalent:

(a) Es gibt einen DPDA D mit $L_\varepsilon(D) = L$

(b) Es gibt einen DPDA D' mit $L_F(D') = L$ **und** kein Wort aus L ist ein echter Präfix von einem anderen Wort aus L.

Zeigen Sie diese Äquivalenz.

Lösung:

- Sei D mit $L_\varepsilon(D) = L$. Erweitere D um explizites Bottom-Symbol $\bot$ mit $q\bot \xrightarrow{\varepsilon} q_F$ für alle $q \in Q$ und q_F neuer und einziger Endzustand. Der so erhaltene PDA ist noch deterministisch mit $L_F(D') = L_\varepsilon(D)$.

  Seien $u, uv \in L$. Da D deterministisch, muss D nach Lesen von u stets in derselben Konfiguration sein, insbesondere der Stack somit leer, womit uv nur für $v = \varepsilon$ akzeptiert werden kann.

- Sei (1) D mit $L_F(D) = L$ und (2) kein Wort aus L ist ein echter Präfix eines weiteren Worts aus L.

  Wieder analog zu allgemeinen PDAs: Erweitere D so zu D', dass beim ersten Erreichen einer Konfiguration mit Endzustand einfach der Stack deterministisch geleert wird, der Automat somit keine weitere Rechnung ausführen kann. Offensichtlich gilt dann: $L_F(D) \supseteq L_\varepsilon(D')$.

  Sei $w \in L_F(D) \setminus L_\varepsilon(D')$. Dann muss die eindeutige akzeptierende Berechnung von D auf w einen Endzustand mindestens zweimal besuchen und schließlich in einer solchen Konfiguration enden. Der Fall, dass man nach dem ersten Besuch einer Endkonfiguration nur noch ε liest, kann nicht sein, da dann das gelesene Wort w auch von D' noch akzeptiert wird. Daher gibt es aber auch einen echten Präfix von w, der von D akzeptiert und damit in L liegt. Widerspruch zu (2). Es kann somit kein solches w geben.

Aufgabe 8.2 Boolesche Ausdrücke

Wir betrachten folgende Grammatik für boolesche Ausdrücke über den *booleschen Variablen* x, y (welche somit Terminale der Grammatik sind):

$$S \to (S \wedge S) \mid (S \vee S) \mid \neg S \mid x \mid y$$

Konstruieren Sie einen DPDA D mit $L(G) = L_\varepsilon(D)$.

Lösung:

Die Übersetzung von CFG in PDA nach VL würde zu einem nichtdeterministischen Raten, welche der beiden Relgen $S \to (S \vee S) \mid (S \wedge S)$ angewendet werden soll, führen und somit nicht einen DPDA ergeben.

In diesem Fall kann man die Fälle zusammenführen zu $S \to (S X_{\mathsf{op}} S)$ mit $X_{\mathsf{op}} \to \vee \mid \wedge$. Daher muss man keine Ableitung raten. Einführen von Hilfssymbolen, um die PDA-Regeln auf die Form "$Q\Gamma \xrightarrow{\Sigma} Q\Gamma^{\leq 2}$" zu bringen, ergibt die folgenden Regeln:

$$qS \xrightarrow{(} qSX_{\mathsf{op}} \qquad qS \xrightarrow{\neg} qS \qquad qS \xrightarrow{x,y} q\varepsilon \qquad qX_{\mathsf{op}} \xrightarrow{\wedge,\vee} qSX_{\mathsf{cl}} \qquad qX_{\mathsf{cl}} \xrightarrow{)} q\varepsilon$$

Der DPDA lässt sich dann direkt als *recursive descent parser* lesen, wobei der Stack des DPDA gerade dem Call-Stack entspricht, z.B. in Python:

```
class RDParser:
    def _next_symbol(self):
        if self.i >= len(self.w):
            raise RuntimeError()
        self.c = self.w[self.i]
        self.i += 1

    def _Xcl(self):
        self._next_symbol()
        if self.c == ')': # qX_cl -)-> qε
            return
        raise RuntimeError()

    def _Xop(self):
        self._next_symbol()
        if self.c in ['∧', '∨']: # qX_op -∧,∨-> qSX_cl
            self._S()
            self._Xcl()
        else:
            raise RuntimeError()

    def _S(self):
        self._next_symbol()
        if self.c == '(': # qS -(-> qSX_op
            self._S()
            self._Xop()
            return
        elif self.c == '¬': # qS -¬-> qS
            self._S()
            return
        elif self.c in ['x','y']: # qS -x,y-> qε
            return
        raise RuntimeError()

    def parse(self,w):
        self.w = w
        self.i = 0
        try:
            self._S()
        except RuntimeError:
            print("Pasrse error at '%s' (position %s) in '%s'." % (self.c,self.i-1,self.w))
            return False
        return self.i == len(self.w)
```

Aufgabe 8.3 CFG $\longleftrightarrow$ PDA

Wir üben die Übersetzung zwischen CFG und PDA:

(a) Überführen Sie folgende CFG G (Startsymbol S) zunächst in CNF[1] und dann in einen PDA A mit $L_\varepsilon(A) = L(G) \setminus \{\varepsilon\}$:

$$S \to SS \mid aSb \mid bSa \mid \varepsilon$$

(b) Übersetzen Sie folgenden PDA A (Startkonfiguration qX) in eine CFG G mit $L_\varepsilon(A) = L(G)$:

$$qX \xrightarrow{l} qX[YXZ] \qquad q[YXZ] \xrightarrow{\varepsilon} pY[XZ] \qquad q[XZ] \xrightarrow{\varepsilon} qXZ \qquad qX \xrightarrow{n} qX \qquad qX \xrightarrow{x,y} q\varepsilon \qquad pY \xrightarrow{a,o} q\varepsilon \qquad qZ \xrightarrow{r} q\varepsilon$$

Notation: $pX \xrightarrow{a} qYZ$ steht kurz für $(q, YZ) \in \delta(p, a, X)$.

Lösung:

(a)

$$\begin{array}{lcl} S & \to & X_a X_{SX_b} \mid X_b X_{SX_a} \mid SS \\ X_{SX_b} & \to & SX_b \mid b \\ X_b & \to & b \\ X_{SX_a} & \to & a \mid SX_a \\ X_a & \to & a \end{array}$$

[1] Wenden Sie die Regeln in der Reihenfolge (3) (4) (1) (2) an, um eine kompaktere Grammatik zu erhalten.

$$
\begin{array}{lcl}
qX_{SX_b} & \to & qSX_b \\
qX_{SX_b} & \xrightarrow{b} & q \\
qS & \to & qX_aX_{SX_b} \mid qSS \mid qX_bX_{SX_a} \\
qX_a & \xrightarrow{a} & q \\
qX_{SX_a} & \to & qSX_a \\
qX_{SX_a} & \xrightarrow{a} & q \\
qX_b & \xrightarrow{b} & q
\end{array}
$$

(b)

$$
\begin{array}{lcl}
X_{q,[YXZ],p} & \to & X_{p,Y,q}X_{q,[XZ],p} \mid X_{p,Y,p}X_{p,[XZ],p} \\
X_{p,Y,q} & \to & a \mid o \\
X_{q,X,p} & \to & lX_{q,X,p}X_{p,[YXZ],p} \mid nX_{q,X,p} \mid lX_{q,X,q}X_{q,[YXZ],p} \\
X_{q,X,q} & \to & nX_{q,X,q} \mid y \mid x \mid lX_{q,X,p}X_{p,[YXZ],q} \mid lX_{q,X,q}X_{q,[YXZ],q} \\
X_{q,Z,q} & \to & r \\
X_{q,[XZ],q} & \to & X_{q,X,q}X_{q,Z,q} \mid X_{q,X,p}X_{p,Z,q} \\
X_{q,[XZ],p} & \to & X_{q,X,q}X_{q,Z,p} \mid X_{q,X,p}X_{p,Z,p} \\
X_{q,[YXZ],q} & \to & X_{p,Y,q}X_{q,[XZ],q} \mid X_{p,Y,p}X_{p,[XZ],q} \\
S & \to & X_{q,X,p} \mid X_{q,X,q}
\end{array}
$$