دانشگاه صنعتی شریف

دانشکده‌ی مهندسی کامپیوتر

درس نظریه‌ی زبان‌ها و ماشین‌ها

سوالات نمونه

پاسخ‌نامه‌ی مجموعه‌ی ۹ : زبان‌های مستقل از متن - بخش ۲

استاد: دکتر علی موقر

تیم دستیاران درس - نیم‌سال دوم ۰۲ - ۰۱

۳۱ اردیبهشت ۱۴۰۲

۱. ۱. الف)

$$G_1 = (\{S, A, B\}, \{0,1\}, R_1, S)$$

$$R_1:\ \ S \to 0S|1A$$

$$A \to 0A|1B$$

$$B \to 0B|1B|1$$

ب)

$$G_2 = (\{S\}, \{0,1\}, R_2, S)$$

$$R_2:\ \ S \to 0S0|0S1|1S0|1S1|0$$

۱. ۲.

$$G = (\{S, X, Y, A, C\}, \{a, b, c\}, R, S)$$

$$R:\ \ S \to \mathrm{XC|AY}$$

$$X \to \mathrm{aXb|\epsilon}$$

$$C \to \mathrm{cC|\epsilon}$$

$$A \to \mathrm{aA|\epsilon}$$

$$Y \to \mathrm{bYc|\epsilon}$$

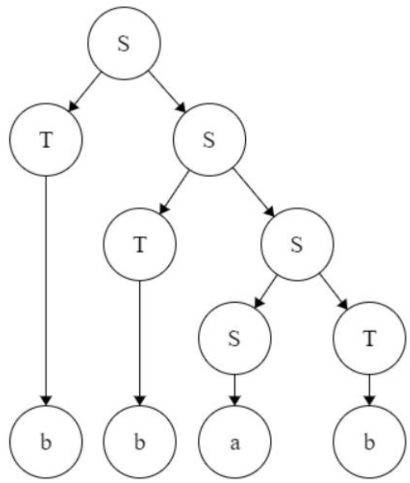
رشته abc را در نظر بگیرید. دو اشتقاق از چپ مختلف به صورت زیر برای آن نوشته می‌شوند:

$$S \to \mathrm{XC} \to \mathrm{aXbC} \to \mathrm{abC} \to \mathrm{abcC} \to \mathrm{abc}$$

$$S \to \mathrm{AY} \to \mathrm{aAY} \to \mathrm{aY} \to \mathrm{abYc} \to \mathrm{abc}$$

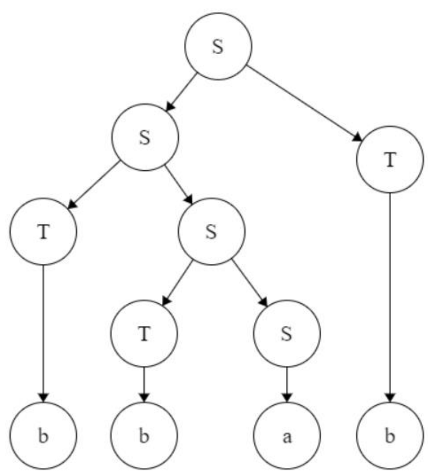
پس این گرامر مبهم هست.

۱. ۳. الف)



$$S \rightarrow TS \rightarrow TTS \rightarrow TTST \rightarrow TTSb \rightarrow TTab \rightarrow Tbab \rightarrow bbab$$

ب)



$$S \rightarrow ST \rightarrow TST \rightarrow bST \rightarrow bTST \rightarrow bbST \rightarrow bbaT \rightarrow bbab$$

ج) می‌دانیم اگر گرامری حداقل دو اشتقاق از چپ مختلف برای یک رشته داشته باشد، دارای ابهام است. اشتقاق به چپ نوشته شده در قسمت ب و اشتقاق به چپ زیر ابهام این گرامر را نشان می‌دهند: (درخت این اشتقاق در قسمت الف آمده)

$$S \rightarrow TS \rightarrow bS \rightarrow bTS \rightarrow bbS \rightarrow bbST \rightarrow bbaT \rightarrow bbab$$

د)

$$G = (\{S, T\}, \{a, b\}, R, S)$$

$$R: \ S \to bS|aT$$

$$T \to bT|\epsilon$$

۱. ۴. ابتدا start variable جدیدی می‌سازیم:

$$S_0 \to S$$

$$S \to ASB$$

$$A \to aASA|a|\epsilon$$

$$B \to SbS|A|bb$$

سپس ϵ − rule ها را حذف می‌کنیم:

$$\mathrm{S}_0 \to S$$

$$S \to ASB|SB|AS|S$$

$$A \to aASA|a|aSA|aAS|aS$$

$$B \to SbS|A|bb$$

سپس unit rule ها را حذف می‌کنیم:

$$\mathrm{S}_0 \to ASB|SB|AS$$

$$S \to ASB|SB|AS$$

$$A \to aASA|a|aSA|aAS|aS$$

$$B \to SbS|bb|aASA|a|aSA|aAS|aS$$

حال با اضافه کردن چند متغیر و rule جدید به فرم نرمال چامسکی می‌رسیم:

$$S_0 \to CB|SB|AS$$
$$S \to CB|SB|AS$$
$$A \to FA|a|GA|DC|DS$$
$$B \to HS|EE|FA|a|GA|DC|DS$$
$$C \to AS$$
$$D \to a$$
$$E \to b$$
$$F \to DC$$
$$G \to DS$$
$$H \to SE$$

البته زبان معادل این گرامر تهی است و یک گرامر ساده با یک رول $S \to AB$ نیز معادل آن است.

۱. ۵. می‌دانیم که هر زبانی که توسط یک گرامر منظم راست گرد توصیف شود، منظم است پس کافی است نشان دهیم که اگر این گراف فاقد دور باشد، یک گرامر G_r منظم راست گرد معادل گرامر G وجود دارد.

یک رول دلخواه گرامر G را در نظر می‌گیریم. اگر از فرم $A \to a$ باشد، این را به رول‌های گرامر G_r اضافه می‌کنیم. اگر از فرم $A \to BC$ باشد، تمام رول‌هایی که B در سمت چپ آن‌ها قرار دارند را بر روی عبارت BC اعمال می‌کنیم. حال تعدادی رول به فرم $A \to bC$ و $A \to DEC$ داریم. تمام رول‌های به فرم اول را به رول‌های گرامر G_r اضافه می‌کنیم. حال تمام رول‌هایی که D و E در سمت چپ آن‌ها قرار دارند را بر روی عبارت DEC اعمال می‌کنیم و به همین صورت ادامه می‌دهیم تا تمام رول‌ها right linear شوند.

اگر گراف G' دارای حلقه باشد، ممکن است پس از رسیدن به DEC و با اعمال چند رول دیگر، دوباره به یک رول برسیم که B در سمت راست آن قرار دارد. اما با توجه به این که G' فاقد دور است، پس از حذف هر متغیر، دیگر به آن برنخواهیم خورد و در نتیجه پس از اعمال کردن تعداد متناهی رول به عبارت BC به عبارتی به فرم $b_1 \dots b_n C$ خواهیم رسید که right linear است. اثبات با برهان خلف:

فرض کنید دنباله‌ای از رول‌های $r_1 \dots r_n$ وجود دارند که پس از اعمال شدن روی BC، باعث ظاهر شدن دوباره B در سمت راست معادله می‌شوند. متغیرهای سمت چپ این رول‌ها را با L_1 تا L_n نشان می‌دهیم. ($L_1 = B$)

با توجه به این که اعمال یک رول r_i ($1 < i \leq n$) باعث ظاهر شدن دوباره B شده است، می‌دانیم که از فرم $L_i \rightarrow XB$ یا $L_i \rightarrow BX$ می‌باشد و در نتیجه یال (L_i, B) در گراف وجود دارد. همچنین می‌دانیم اعمال یک رول r_j ($1 \leq j < i$) باعث ظاهر شدن L_i شده است. پس (L_j, L_i) در گراف وجود دارد. به همین صورت یک دنباله از یال‌ها به صورت $(B, L_{k_1}), (L_{k_1}, L_{k_2}) \dots (L_j, L_i)(L_i, B)$ پیدا شده و در نتیجه گراف دور خواهد داشت.

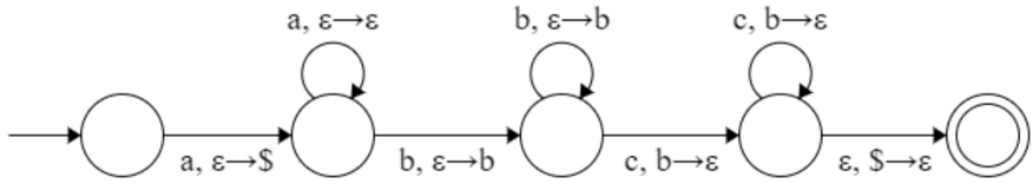
۲. ۱. دو زبان دلخواه مستقل از متن L و L' را در نظر می‌گیریم و زبان L_c را به صورت $L_c = LL'$ تعریف می‌کنیم. می‌دانیم که دو گرامر مستقل از متن مانند $G = (V, \Sigma, R, S)$ و $G' = (V', \Sigma', \mathrm{R}', S')$ وجود دارند که به ترتیب هر یک از این دو زبان را تشخیص می‌دهند. حال گرامر $G_c = (V_c, \Sigma_c, R_c, S_c)$ را برای تشخیص L_c را به صورت زیر تعریف می‌کنیم:

$$V_c = V \cup V' \quad \Sigma_c = \Sigma \cup \Sigma' \quad R_c = R \cup R' \cup \{S_c \rightarrow SS'\}$$

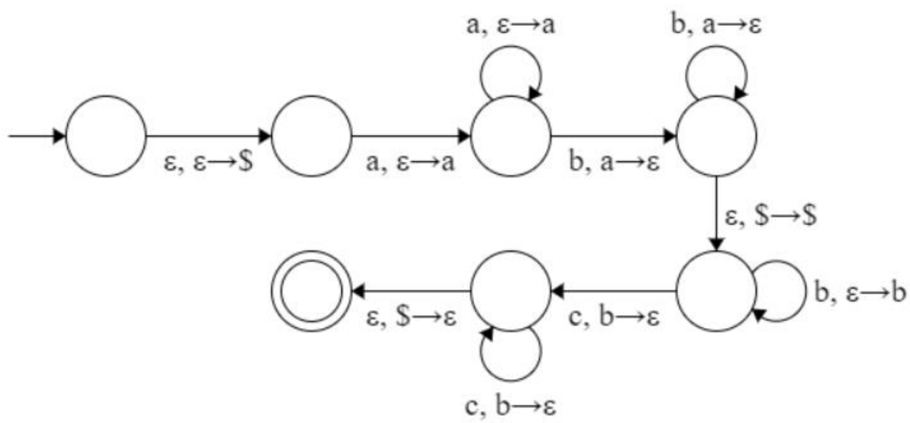
۲. ۲. گرامر $G = (V, \Sigma, R, S)$ را در نظر می‌گیریم که در فرم نرمال چامسکی است و زبان L را تشخیص می‌دهد. حال گرامر $G' = (V', \Sigma, R', S')$ را می‌سازیم که زبان $PREFIX(L)$ را تشخیص دهد:

برای هر متغیر مانند A در مجموعه V، یک متغیر A و یک متغیر A' در V' قرار می‌دهیم. مجموعه R' شامل تمام رول‌های مجموعه R است و به ازای هر رول به فرم $A \rightarrow XY$ دو رول به فرم‌های $A' \rightarrow XY'$ و $A' \rightarrow X'$ ، و به ازای هر رول به فرم $A \rightarrow a$ دو رول به فرم‌های $A' \rightarrow a$ و $A' \rightarrow \epsilon$ به R' اضافه می‌کنیم. این گرامر زبان $PREFIX(L)$ را تشخیص می‌دهد پس این یک زبان مستقل از متن است.

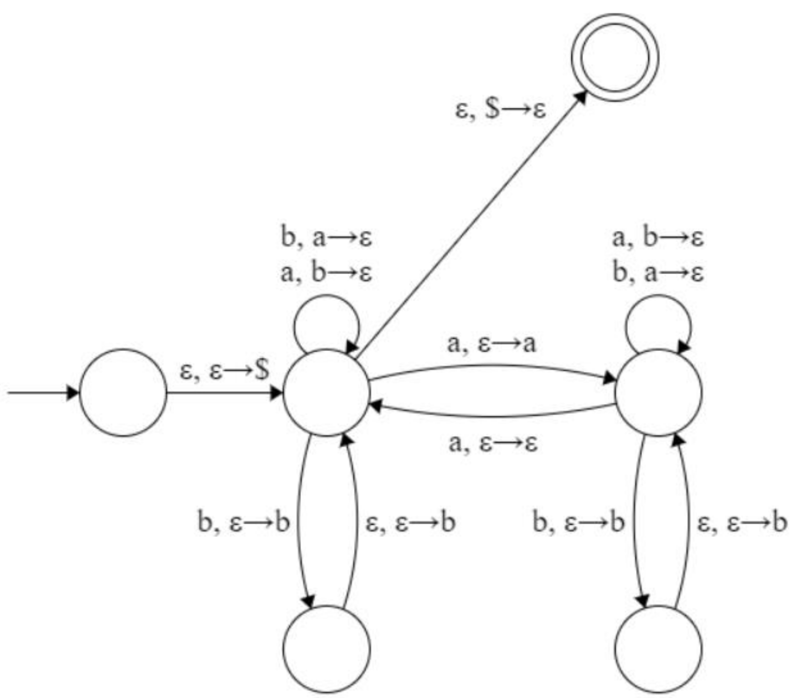
۳. ۱. الف)


a, ε→ε
b, ε→b
c, b→ε
a, ε→$
b, ε→b
c, b→ε
ε, $→ε

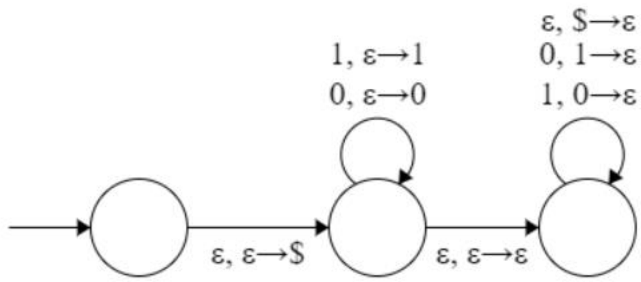
ب)


a, ε→a
b, a→ε
ε, ε→$
a, ε→a
b, a→ε
ε, $→$
b, ε→b
ε, $→ε
c, b→ε
c, b→ε

ج)


ε, $→ε
b, a→ε
a, b→ε
a, b→ε
b, a→ε
ε, ε→$
a, ε→a
a, ε→ε
b, ε→b
ε, ε→b
b, ε→b
ε, ε→b

۳. ۲. زبان گرامر داده شده شامل رشته‌های باینری حاصل از الحاق یک رشته باینری و معکوس $bitwise\ not$ آن است. ماشین پشته‌ای زیر نیمه اول عبارت را در استک ذخیره می‌کند و سپس حرف به حرف آن را خوانده و با not آن حرف در ورودی انطباق می‌دهد. واضح است که فقط هنگامی که رشته درست به صورت کامل خوانده شده باشد استک خالی می‌شود.



۳. ۳. زبان L را در نظر می‌گیریم که به ازای خالی شدن پشته ماشین پشته‌ای $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$ پذیرفته می‌شود. ماشین پشته‌ای $P' = \left(Q', \Sigma, \Gamma', \delta', q_s, \{q_f\}\right)$ را می‌سازیم که همین زبان را به ازای توقف در استیت نهایی بپذیرد. (z یک کاراکتر دلخواه است که عضو Γ نمی‌باشد)

$$Q' = Q \cup \{q_s, q_f\} \qquad \Gamma' = \Gamma \cup \{z\}$$

$$\delta'(q, a, A) = \begin{cases} \delta(q, a, A) & q \in Q \wedge A \neq z \\ (q_f, \epsilon) & a = \epsilon \wedge A = z \\ (q_0, z) & q = q_s \wedge a = \epsilon \wedge A = \epsilon \end{cases}$$

حال زبان L را در نظر می‌گیریم که به ازای توقف در استیت نهایی ماشین پشته‌ای $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$ پذیرفته می‌شود. ماشین پشته‌ای $P' = (Q', \Sigma, \Gamma', \delta', q_s, \emptyset)$ را می‌سازیم که همین زبان را به ازای خالی شدن پشته بپذیرد. (تمام استیت‌های نهایی به q' متصل شده و در این استیت استک آن‌ها خالی می‌شود)

$$Q' = Q \cup \{q_s, q'\} \qquad \Gamma' = \Gamma \cup \{z\}$$

$$\delta'(q, a, A) = \begin{cases} \delta(q, a, A) & q \in Q \wedge (q \notin F \vee a \neq \epsilon \vee A \neq \epsilon) \\ \delta(q, \epsilon, \epsilon) \cup \{(q', \epsilon)\} & q \in F \wedge a = \epsilon \wedge A = \epsilon \\ (q', \epsilon) & q = q' \wedge a = \epsilon \\ (q_0, z) & q = q_s \wedge a = \epsilon \wedge A = \epsilon \end{cases}$$

۴. ۱. زبان مستقل از متن $L = \{a^m b^n c^n dda^{3m} | n, m \in N\}$ با گرامر معادل زیر را در نظر می‌گیریم.

$$S \rightarrow aSaaa | aTddaaa$$

$$T \rightarrow bTc | bc$$

حال فرض می‌کنیم که $L_{1/2}$ مستقل از متن است (برهان خلف) و طول پمپ آن را p در نظر می‌گیریم. طبق لم تزریق، رشته $s = a^p b^p c^p d$ به فرم $s = uvxyz$ نوشته می‌شود به صورتی که شرط های زیر برقرار باشند:

$$\forall i \geq 0, \quad s' = uv^i xy^i z \in L_{1/2}$$

$$|vy| > 0$$

$$|vxy| \leq p$$

با در نظر گرفتن $i = 2$ یکی از حالات زیر رخ می‌دهد، با رد کردن تمام حالات مستقل از متن نبودن $L_{1/2}$ اثبات می‌شود: (فرض می‌کنیم v و y هر کدام فقط از یک نوع حرف تشکیل شده‌اند زیرا در غیر این صورت توالی حروف بهم می‌خورد و رد آن حالات بدیهی است.)

۱. $s' = a^{p+|vy|} b^p c^p d$ واضح است که با توجه به تعداد حروف، این رشته نمی‌تواند نیمه اول رشته‌ای در زبان L باشد.

۲. $s' = a^{p+|v|} b^{p+|y|} c^p d$ با توجه به شروط لم پمپاژ، حداقل یکی از $|y|$ یا $|v|$ ناصفر است. اگر $|v|$ ناصفر باشد، مانند قسمت قبل تعداد حروف از نیم رشته کم‌تر خواهد بود و اگر $|y|$ ناصفر باشد برابری تعداد b و c ها بهم می‌خورد.

۳. $s' = a^p b^{p+|vy|} c^p d$ برابری تعداد c و b ها بهم می‌خورد.

۴. $s' = a^p b^{p+|v|} c^{p+|y|} d$ با توجه به تعداد حروف طول این رشته بیش‌تر از نصف طول رشته در بر گیرنده آن در L است.

۵. $s' = a^p b^p c^{p+|vy|} d$ برابری تعداد c و b ها بهم می‌خورد.

۶. $s' = a^p b^p c^{p+|v|} d^2$ طول این رشته بیش‌تر از نصف طول رشته در بر گیرنده آن در L است.

۴. ۲. برای تمام بخش‌ها، اول فرض می‌کنیم که زبان داده شده مستقل از متن است و سپس با اعمال لم تزریق و برهان خلف خواسته مسئله را اثبات می‌کنیم. طبق لم تزریق هر رشته s عضو زبان مستقل از متن L به فرم $s = uvxyz$ نوشته می‌شود به صورتی که:

$$\forall i \geq 0, \quad s' = uv^ixy^iz \in L$$

$$|vy| > 0$$

$$|vxy| \leq p$$

الف) برای طول تزریق p، رشته $s = a^pb^{p+1}c^{p+2}$ را در نظر می‌گیریم. باتوجه به $|vxy| \leq p$ می‌توان گفت که برای حروف تشکیل دهنده vy حالات زیر را داریم:

- فقط حاوی a باشد. در این حالت با در نظر گرفتن $i = 2$ می‌دانیم که تعداد a ها بیش‌تر مساوی b ها شده و رشته حاصل عضو L نخواهد بود.
- فقط حاوی b باشد. در این حالت با در نظر گرفتن $i = 2$ می‌دانیم که تعداد b ها بیش‌تر مساوی c ها شده و رشته حاصل عضو L نخواهد بود.
- فقط حاوی c باشد. در این حالت با در نظر گرفتن $i = 0$ می‌دانیم که تعداد c ها کم‌تر مساوی b ها شده و رشته حاصل عضو L نخواهد بود.
- حاوی a و b باشد. در این حالت با در نظر گرفتن $i = 2$ می‌دانیم که تعداد b ها بیش‌تر مساوی c ها شده و رشته حاصل عضو L نخواهد بود.
- حاوی b و c باشد. در این حالت با در نظر گرفتن $i = 0$ می‌دانیم که تعداد b ها کم‌تر مساوی a ها شده و رشته حاصل عضو L نخواهد بود.

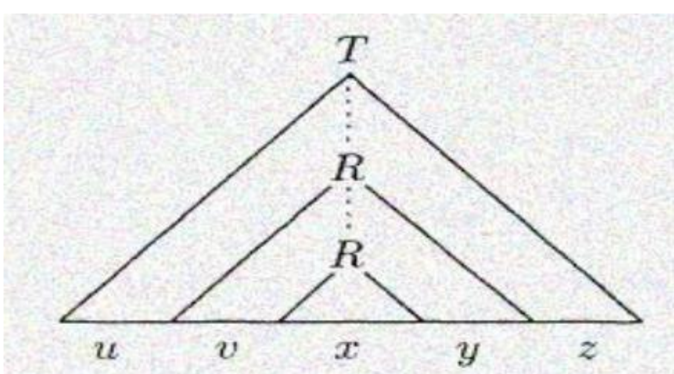
ب) برای طول تزریق p، رشته $s = a^pb^{2p}a^p$ را در نظر می‌گیریم. اگر $n_a(vy) \neq n_b(vy)$، با در نظر گرفتن $i = 2$ شرط $n_a(s') = n_b(s')$ برقرار نخواهد بود و s' عضو L نخواهد بود.

درحالت $n_a(vy) = n_b(vy)$ داریم $vxy = a^jb^k$ یا $vxy = b^ka^j$ $(0 < k, j < p)$ که در هردو حالت با در نظر گرفتن $i = 2$، رشته s' تقارن خود را در دست داده و $s' = s'^R$ برقرار نخواهد بود.

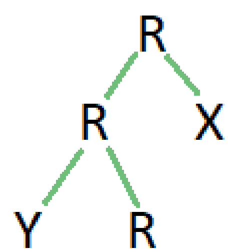
ج) برای طول تزریق p ، رشته $s = a^{(p+1)^2} b^{p+1}$ را در نظر می‌گیریم. با توجه به شروط لم پمپینگ، برای حروف تشکیل دهنده vy حالات زیر را داریم:

- فقط شامل a باشد. با در نظر گرفتن $i = 2$ داریم $s' = a^{(p+1)^2+|vy|} b^{p+1}$ و باتوجه به این که $|vy| \le p$ مشخص است که $(p+1)^2 + |vy|$ بر $p+1$ بخش پذیر نیست و در نتیجه s' عضو L نمی‌باشد.
- فقط شامل b باشد. با در نظر گرفتن $i = 2$ داریم $s' = a^{(p+1)^2} b^{p+1+|vy|}$ و با توجه به این که $|vy| \le p$ مشخص است که $(p+1)^2$ بر $p+1+|vy|$ بخش پذیر نیست و در نتیجه s' عضو L نمی‌باشد.
- شامل a و b باشد. فرض می‌کنیم که v فقط شامل a و y فقط شامل b می‌باشد چون در غیر این صورت یکی از این دو قسمت شامل هردو a و b خواهد بود و پمپ کردن آن توالی حروف را بهم می‌زند. با در نظر گرفتن $i = 2$ داریم $s' = a^{(p+1)^2+|v|} b^{p+1+|y|}$ با توجه به این که $|v|, |y| < p$ مشخص است که $(p+1)^2 + |v|$ بر $p+1+|y|$ بخش پذیر نیست و در نتیجه s' عضو L نمی‌باشد.

۴. ۳. گرامر G در فرم نرمال چامسکی که زبان مورد نظر را تشخیص می‌دهد را در نظر می‌گیریم. برای اثبات لم تزریق عادی، عمق parse tree برای رشته مورد نظر را حداقل $|V| + 1$ در نظر می‌گرفتیم تا مطمئن باشیم در یک مسیر از ریشه تا برگ، حداقل یکی از متغیرها (آن را با R نشان می‌دهیم) حداقل دوبار تکرار شده است.



مشخص است در صورتی که R دوم مستقیما فرزند R اول باشد، از بین v و y یکی تهی خواهد بود. اما گر عمق parse tree را حداقل $2|V|+1$ بگیریم (طول رشته حداقل برابر با $k = 2^{2|V|+1} + 1$ باشد)، می‌توان گفت یک متغیر مانند R وجود دارد که در مسیری از ریشه تا برگ حداقل سه بار تکرار شده است. حال R اول و R سوم را در نظر می‌گیریم و می‌دانیم بین آن‌ها حداقل یک متغیر دیگر وجود دارد. در ‘‘بدترین’’ حالت هر سه R پشت سر هم هستند و شکل آن بخش از درخت به صورت زیر خواهد بود:



باتوجه به این که در فرم نرمال چامسکی متغیرها (به جز S) nullable نیستند، می‌توان گفت X و Y هر دو بخش‌هایی از رشته را تولید می‌کنند و در نتیجه v و y تهی نخواهند بود.

می‌دانیم که ارتفاع سه‌تایی مورد نظر حداقل همان $2|V|+1$ خواهد بود و در نتیجه $|vxy| \leq k$.