دانشکده‌ی علوم ریاضی

نظریه‌ی زبان‌ها و اتوماتا ۲۱ آذرماه ۱۳۹۱

جلسه‌ی ۲۳: ویژگی‌های زبان‌های مستقل از متن

مدرّس: دکتر شهرام خزائی *نگارنده: شفیع قلی‌زاده*

۱ لم تزریق برای زبان‌های مستقل از متن

برای هر زبان مستقل از متن L، یک عدد طبیعی n وجود دارد به طوری که برای هر رشته‌ی z در L با طول حداقل n، می‌توان z را به صورت $z = uvwxy$ نوشت که شرایط زیر را داشته باشد:

۱) $|vwx| \leq n$

۲) $|vx| \geq 1$

۳) $\forall i \geq 0 \; ; \; uv^iwx^iy \in L$

از لم تزریق برای اثبات این نکته استفاده می‌شود که زبانی خاص مستقل از متن نیست.

مثال ۱ *زبان L را به صورت زیر تعریف می‌کنیم:*

$$L = \{0^n 1^n 2^n \mid n \geq 1\}$$

می‌خواهیم ثابت کنیم که این زبان مستقل از متن نیست.
فرض می‌کنیم که زبان مستقل از متن باشد. به ازای nای که در شرایط لم تزریق صدق می‌کند، رشته‌ی $z = 0^n 1^n 2^n$ را در نظر می‌گیریم.
به ازای هر u, v, w, x, y که $z = uvwxy$ و $|vwx| \leq n$ و رشته‌های x و v هر دو تهی نباشند، یکی از دو حالت زیر برقرار است:

۱) *vwx شامل ۲ نیست.*

۲) *vwx شامل ۰ نیست.*

طبق لم تزریق به ازای هر $i \geq 0$، رشته‌ی uv^iwx^iy در L هست، اگر در هر دو حالت قرار دهیم $i = 0$، در هر دو حالت به تناقض می‌رسیم.

اثبات لم تزریق

برهان. گرامری برای $L - \{\epsilon\}$ به فرم نرمال چامسکی با m متغیر در نظر می‌گیریم. ادعا می‌کنیم $n = ۲^m$ در لم تزریق صدق می‌کند.برای اثبات ابتدا لم زیر را ثابت می‌کنیم.

لم ۱ *هر درخت تجزیه با محصول z، که $|z| \geq ۲^m$، دارای مسیری با طول حداقل $(m + ۱)$ است.*

برهان. برای اثبات از استقرا روی m استفاده کرده و معادلاً ثابت می‌کنیم اگر مسیرهای درخت تجزیه دارای حداکثر طول m باشند، آنگاه $|z| \leq ۲^{m-۱}$.

پایه‌ی استقرا: به ازای $m = ۱$ به راحتی می‌توان نشان داد که $|z| = ۱$ و از آنجا که $۱ = ۲^۰ = ۲^{m-۱}$ حکم برقرار است.

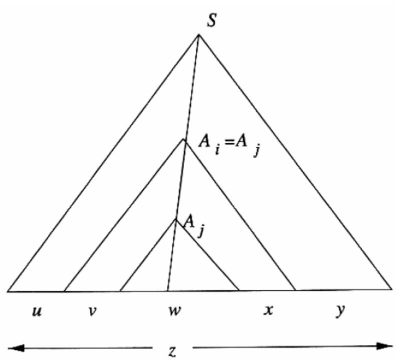
گام استقرا: فرض می‌کنیم طولانی‌ترین مسیر درخت تجزیه، طولی برابر m دارد ($m > ۱$). چون $m > ۱$ پس ریشه‌ی درخت از یک قانون تولید به فرم $A \to BC$ استفاده می‌کند، یعنی نمی‌توان درخت را با قانون تولیدی شروع کرد که از یک حرف پایانه استفاده می‌کند. دو زیردرختی را که ریشه‌ی آن‌ها B و C است، در نظر می‌گیریم. این دو زیردرخت نمی‌توانند مسیری با طول بیشتر از $m - ۱$ داشته باشند. در نتیجه با توجه به فرض استقرا، این دو زیردرخت محصولاتی با حداکثر طول $۲^{m-۲}$ دارند. بنابراین برای محاسبه‌ی محصول نهایی درخت داریم:

$$|z| \leq ۲^{m-۲} + ۲^{m-۲} = ۲^{m-۱}$$

بنابراین حکم استقرا ثابت می‌شود.

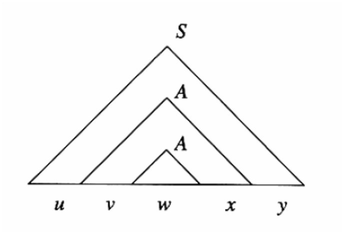
■

اکنون می‌توانیم به اثبات لم تزریق بازگردیم.
گرامر $L - \{\epsilon\}$ دارای m متغیر است که $n = ۲^m$. فرض می‌کنیم $z \in L$ و $|z| > n$. با توجه به لم۱ می‌دانیم درخت تجزیه z مسیری به طول حداقل $m + ۱$ دارد.
چون در زبان $L - \{\epsilon\}$، m متغیر به کار رفته است و طول مسیر z حداقل $m + ۱$ است، طبق اصل لانه‌ی کبوتری نتیجه می‌شود که حداقل یک متغیر دو بار تکرار شده است. متغیرهای موجود در مسیر را $A_۰$ و $A_۱$ و ... و A_k در نظر می‌گیریم. می‌دانیم که در $m + ۱$ متغیر آخر حداقل یک متغیر تکراری حضور دارد. فرض می‌کنیم $A_i = A_j$ که $k - m \leq i < j \leq k$.

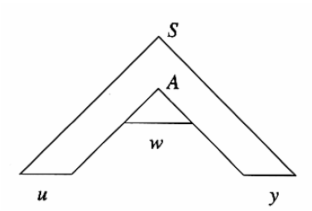


حال می‌توانیم درخت را مشابه شکل بالا تقسیم‌بندی کنیم. چنان‌که از شکل بالا پیداست، رشته w محصول زیردرخت به ریشه A_j است و رشته vwx محصول زیردرخت A_i است. با توجه به ساخت فرم نرمال چامسکی v و x هر دو

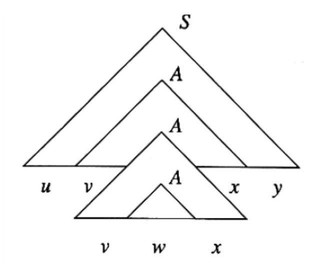
نمی‌توانند همزمان برابر ϵ باشند. نهایتاً این‌که رشته‌های u و y به ترتیب قسمت‌هایی از رشته z هستند که در سمت چپ و راست زیردرخت A_i حضور دارند.
اگر $A_i = A_j = A$ می‌توان درخت تجزیه جدیدی مانند شکل زیر را از روی درخت تجزیه اولیه ساخت. این درخت در واقع رشته uv^iwx^iy را به ازای $i = 1$ نمایش می‌دهد.



می‌توان زیردرخت با ریشه A_i را که محصول vwx دارد را با زیردرخت با ریشه A_j که محصول w دارد جایگزین کرد. درخت حاصل در شکل زیر نمایش داده شده است. این درخت در واقع رشته uv^iwx^iy را به ازای $i = 0$ ایجاد می‌کند.



در حالت دیگر می‌توان زیردرخت با ریشه‌ی A_j را که محصول w دارد را با زیردرخت با ریشه‌ی A_i که محصول vwx دارد جایگزین کرد. درخت حاصل در شکل زیر نمایش داده شده است. این درخت در واقع رشته uv^iwx^iy را به ازای $i = 2$ ایجاد می‌کند. با تکرار این عمل رشته uv^iwx^iy به ازای مقادیر بالاتر i نیز ساخته خواهد شد.



نهایتاً این‌که در کلیه‌ی حالات در نظر گرفته شده برای انتخاب A_i باید شرط $k - i \leq m$ برقرار باشد. می‌دانیم زیردرخت با ریشه A_i، محصول vwx دارد. چون طبق شرایط لم تزریق $|vwx| \leq n$ و $n = 2^m$، با استفاده از لم۱ می‌توان نشان داد که زیردرخت با ریشه‌ی A_i مسیری به طول بیش‌تر از $m + 1$ ندارد و بنابراین شرط $k - i \leq m$ برقرار خواهد بود. ■

۲ ویژگی‌های بستاری زبان‌های مستقل از متن

قضیه ۲ *زبان‌های مستقل از متن نسبت به اجتماع، الحاق، عمل ستاره، یکریختی، معکوس یکریختی و معکوس‌گیری بسته است.*

برهان. ابتدا اثبات را برای عملگر اجتماع انجام می‌دهیم. فرض می‌کنیم $L_۱$ و $L_۲$ زبان‌های مستقل از متن باشند و $G_۱$ و $G_۲$ گرامرهایی هستند که به ترتیب $L_۱$ و $L_۲$ را تولید می‌کنند.

$$G_۱ = (V_۱, T, P_۱, S_۱)$$

$$G_۲ = (V_۲, T, P_۲, S_۲)$$

در این‌جا بدون کاسته شدن از کلیت مسأله می‌توانیم فرض کنیم که $V_۱$ و $V_۲$ متغیر مشترکی ندارند و $S \notin V_۱ \cup V_۲$. گرامر G را به صورت زیر در نظر بگیرید.

$$G = (V, T, P, S)$$

$$V = V_۱ \cup V_۲ \cup \{S\}$$

$$P = P_۱ \cup P_۲ \cup \{S \rightarrow S_۱ \mid S_۲\}$$

یعنی یک متغیر جدید S به مجموعه متغیرهای گرامرها اضافه می‌کنیم و قانون تولید جدید $S \rightarrow S_۱ \mid S_۲$ را به مجموعه قوانین گرامرها می‌افزاییم. می‌توان به راحتی ثابت کرد که $L(G) = L_۱ \cup L_۲$.
برای عمل الحاق باید دو قانون تولید $S \rightarrow S_۱S_۲ \mid \epsilon$ را به مجموعه قوانین تولید گرامرها اضافه کرد.
برای عمل ستاره باید دو قانون تولید $S \rightarrow SS_۱|\epsilon$ را به مجموعه قوانین تولید گرامرها اضافه کرد.
برای عمل یکریختی فرض می‌کنیم L زبان مستقل از متن روی $\Sigma_۱$ باشد، $h : \Sigma_۱ \rightarrow \Sigma_۲^*$ یک یکریختی باشد. نشان می‌دهیم $h(L)$ نیز یک زبان مستقل از متن است.
فرض کنید $G = (V, \Sigma_۱, P, S)$ یک گرامر برای زبان L باشد. گرامر $G' = (V, \Sigma_۲, P', S)$ را در نظر بگیرید که قوانین تولید P' همان قوانین تولید P هستند که در بدنه آن‌ها هر حرف پایانه $a \in \Sigma_۱$ با رشته‌ی $h(a)$ جایگزین می‌شود. می‌توان نشان داد که $h(L) = L(G')$.
برای معکوس‌گیری فرض می‌کنیم L یک زبان مستقل از متن و G گرامری برای آن باشد و $L^R = \{w^R | w \in L\}$، در گرامر زبان به ازای هر قانون تولید $A \rightarrow \alpha$ قرار می‌دهیم $A \rightarrow \alpha^R$. با استقرا روی طول مشتقات G و G^R می‌توان به راحتی نشان داد که $L(G^R) = L^R$ و بنابراین تمام قوانین تولید در G^R معکوس قوانین تولید G هستند.
برای معکوس یکریختی فرض کنید L یک زبان مستقل از متن روی الفبای Θ و $h : \Sigma \rightarrow \Theta$ یک یکریختی باشد. نشان می‌دهیم $h^{-۱}(L)$ نیز مستقل از متن است. طبق تعریف داریم:

$$h^{-۱}(L) = \{w | h(w) \in L\}$$

فرض کنید P یک ماشین پشته‌ای[۱] برای L باشد:

$$P = (Q, \Theta, \Gamma, \delta, q_۰, Z_۰, F)$$

[۱] pushdown automata

ماشین پشته‌ای جدیدی به صورت زیر تعریف می‌کنیم:

$$P' = (Q', \Sigma, \Gamma, \delta', [q_0, \epsilon], Z_0, F')$$

در این‌جا Q' مجموعه تمام زوج مرتب‌های $[q, x]$ است که $q \in Q$ و x پیشوندی از $h(a)$ است به ازای یک $a \in \Sigma$.

$$F' = \{[q, \epsilon] | q \in F\}$$

به ازای هر $a \in \Sigma$ و $X \in \Gamma$ داریم:

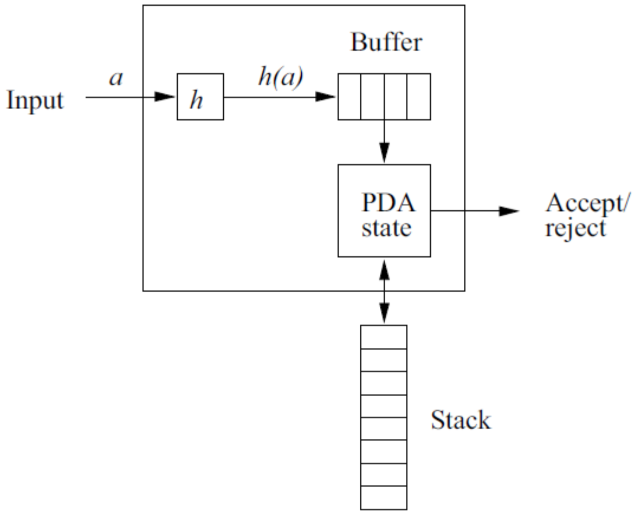
$$\delta'([q, \epsilon], a, X) = \{(a, h(a)), x\}$$

این قانون برای حالتی است که بافر تهی است، که در این صورت Q با خواندن حرف a از رشته‌ی ورودی، بدون تغییر دادن حالت، آن را با $h(a)$ پر می‌کند. اگر $\delta(q, a, x)$ شامل (p, δ) باشد، آنگاه $\delta'([q, ax], \epsilon, X)$ شامل $([p, x], \Delta)$ است. این قانون برای وقتی است که بافر تهی نیست، که در این صورت ماشین پشته‌ای P' با خواندن یک حرف از بافر (بدون خواندن حرفی از رشته ورودی) حرکت ماشین پشته‌ای P را شبیه‌سازی می‌کند.
برای کامل کردن اثبات باید نشان داد که دو عبارت زیر معادل هستند:

۱) $([q_0, \epsilon], w, Z_0) \vdash^*_{P'} ([q, \epsilon], \epsilon, \alpha)$

۲) $(q_0, h(w), Z_0) \vdash^*_{P} (q, \epsilon, \alpha)$

این بدان معنا است که P' هر رشته‌ی w را می‌پذیرد اگر و فقط اگر P رشته‌ی $h(w)$ را بپذیرد.



■

قضیه ۳ *زبان‌های مستقل از متن، تحت اشتراک، تفاضل و مکمل‌گیری بسته نیستند.*

برهان. در این مورد می‌توانیم برای اثبات به معرفی مثال نقض بسنده کنیم. زبان‌های L و M را به صورت زیر در نظر بگیرید.

$$M = \{۰^i ۱^n ۲^n | n \geq ۱ \ , \ i \geq ۱\} \qquad\qquad L = \{۰^n ۱^n ۲^i | n \geq ۱ \ , \ i \geq ۱\}$$

این زبان‌ها مستقل از متن هستند چون توسط گرامرهای زیر تولید می‌شوند:

$$\begin{array}{c}
L = \{۰^n ۱^n ۲^i | n \geq ۱ \ , \ i \geq ۱\} \\
S \rightarrow AB \\
A \rightarrow ۰A۱ \mid ۰۱ \\
B \rightarrow ۲B \mid ۲
\end{array}
\qquad\qquad
\begin{array}{c}
M = \{۰^i ۱^n ۲^n | n \geq ۱ \ , \ i \geq ۱\} \\
S \rightarrow AB \\
A \rightarrow ۰A \mid ۰ \\
B \rightarrow B۲ \mid ۲
\end{array}$$

زبان L مستقل از متن است   زبان M مستقل از متن است

اشتراک دو زبان مستقل از متن فوق، به صورت زیر است:

$$L \cap M = \{۰^n ۱^n ۲^n \mid n \geq ۱\}$$

با استفاده از لم تزریق به راحتی می‌توان نشان داد که اشتراک این دو زبان مستقل از متن نیست.
در مورد تفاضل نیز کافیست توجه کنیم: $L \cap M = L - (L - M)$. اگر تفاضل هر دو زبان مستقل از متن، خود مستقل از متن باشد $L - M$ مستقل از متن است و بنابراین $L - (L - M)$ هم مستقل از متن است. اما طبق تساوی، $L \cap M$ لزوماً مستقل از متن خواهد بود که متناقض است.
در مورد مکمل‌گیری: $L \cap M = \overline{\overline{L} \cup \overline{M}}$. اگر مکمل هر زبان مستقل از متن خود مستقل از متن باشد، $\overline{L}$ و $\overline{M}$ مستقل از متن خواهند بود و بنابراین $\overline{L} \cup \overline{M}$ مستقل از متن است و نهایتاً $\overline{\overline{L} \cup \overline{M}}$ هم مستقل از متن است. در حالی که طبق تساوی، $L \cup M$ لزوماً مستقل از متن خواهد بود که متناقض است. ■

قضیه ۴ *اگر L زبان مستقل از متن و R زبان منظم باشند $(L - R)$ و $(L \cap R)$ مستقل از متن هستند.*

برهان. کافی است یک ماشین پشته‌ای برای L در نظر بگیریم که $L(P) = L$ و یک اتوماتا برای R بسازیم که $L(A) = R$ باشد.

$$P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, Z_۰, F_P)$$
$$A = (Q_A, \Sigma, \delta_A, q_A, F_A)$$

حال یک ماشین پشته‌ای جدید به صورت تعریف می‌کنیم:

$$P' = (Q_P \times Q_A, \Sigma, \Gamma, \delta', [q_P, q_A], Z_۰, F_P \times F_A)$$

که به ازای هر $q \in Q_P$، $p \in Q_A$ و $X \in \Gamma$ داریم:

$$\delta([q,p], a, X) = \{([r, \hat{\delta}_A(p,a)], \gamma) \ : \ (r, \gamma) \in \delta_P(q, a, X)\}$$

ماشین پشته‌ای P' در واقع ماشین‌های P و A را به طور موازی شبیه‌سازی می‌کند.
برای کامل کردن اثبات باید نشان داد که دو عبارت زیر معادل هستند:

۱ $(q_P, w, Z_\circ) \vdash_P^* (q, \epsilon, \gamma)$

۲ $((q_p, q_A), w, z_\circ) \vdash_{P'}^* ((q, \hat{\delta}(q_A, w), \epsilon, \gamma)$

■

۳ ویژگی‌های تصمیمی زبان‌های مستقل از متن

مسائل زیر برای زبان‌های مستقل از متن تصمیم‌پذیر نیستند:

۱) آیا گرامر مستقل از متن G مبهم است؟

۲) آیا گرامر مستقل از متن G ذاتاً مبهم است؟

۳) آیا دو زبان مستقل از متن جدا از هم هستند؟

۴) آیا دو زبان مستقل از متن برابر هم هستند؟

۵) آیا یک گرامر مستقل از متن تمام زبان را پوشش می‌دهد؟

مسائل زیر برای زبان‌های مستقل از متن تصمیم‌پذیراند:

۱) آیا $L = \emptyset$؟
با استفاده از الگوریتم کشف متغیرهای مولد، می‌توان تهی بودن یک زبان را تعیین کرد. اگر متغیر شروع، یک متغیر مولد باشد، زبان ناتهی است.

۲) آیا L نامحدود است؟ ($|L| = \infty$)
برای تعیین نامتناهی بودن یک زبان مستقل از متن می‌توان از لم زیر استفاده کرد.

لم ۵ *اگر و فقط اگر گرامر زبان L با فرم نرمال چامسکی دارای m متغیر باشد و زبان دارای رشته‌ای با طول بین n و $(2n - 1)$ باشد که $n = 2^m$، زبان نامتناهی است.*

البته باید دقت کرد که لم فوق یک الگوریتم کارا پیشنهاد نمی‌دهد. اما الگوریتم کارایی نیز برای تعیین متناهی بودن یک زبان مستقل از متن وجود دارد که به عنوان تمرین به خواننده واگذار می‌شود.

۳) آیا رشته‌ی w در L هست؟
در این مورد از الگوریتم CYK استفاده می‌کنیم.
ورودی: $w = a_1 a_2 \ldots a_n$
گرامر G به فرم نرمال چامسکی را در نظر می‌گیریم. مجموعه‌های $X_{i,j}$ را که $1 \leq i \leq j \leq n$ به صورت زیر تعریف می‌کنیم:

$$X_{i,j} = \{A | A \stackrel{*}{\Rightarrow}_G a_i a_{i+1} \ldots a_j\}$$

$X_{i,j}$ مجموعه همه متغیرهایی است که می‌توانند $a_i \dots a_j$ را تولید کنند. هدف، محاسبه X_{ij}ها به صورت بازگشتی و با استفاده از برنامه‌سازی پویا[۲] است.
کافی است ابتدا مجموعه‌های زیر را بسازیم:

$$X_{۱,۱}\ ,\ X_{۲,۲}\ ,\ \dots\ ,\ X_{n,n}$$

$$X_{i,i} = \{A | A \to a_i \in P\}$$

سپس به طور بازگشتی مجموعه‌های زیر را سطر به سطر و با شروع از پایین سطر می‌سازیم:

$$\begin{array}{lllll} X_{۱,n} & & & & \\ \vdots & & & & \\ X_{۱,۲} & X_{۲,۳} & \dots & X_{n-۱,n} & \\ X_{۱,۱} & X_{۲,۲} & \dots & X_{n-۱,n-۱} & X_{n,n} \end{array}$$

برای محاسبه ی $X_{i,j}$ها به این صورت عمل می‌کنیم که اگر برای یک k $(i \le k \le j)$ و یک قانون تولید $A \to BC$ داشته باشیم $B \in X_{ik}$ و $C \in X_{k+۱,j}$ آن‌گاه متغیر A را به مجموعه‌ی $X_{i,j}$ اضافه می‌کنیم.

سرانجام پس از محاسبه مجموعه‌ی $X_{i,j}$ می‌توان تعیین کرد که آیا رشته‌ی w در L هست یا خیر. زیرا $w \in L$ اگر و فقط اگر متغیر شروع گرامر متعلق به مجموعه‌ی $X_{۱,n}$ باشد.

مثال ۲ *رشته‌ی $w = baaba$ و گرامر L به فرم نرمال چامسکی زیر را در نظر می‌گیریم:*

$$\begin{aligned} S &\to AB \mid BC \\ A &\to BA \mid a \\ B &\to CC \mid b \\ C &\to AB \mid a \end{aligned}$$

برای این‌که بدانیم آیا $w \in L$ از الگوریتم CYK استفاده می‌کنیم:

$$\begin{array}{|ccccc} \{S,A,C\} & & & & \\ \emptyset & \{S,A,C\} & & & \\ \emptyset & \{B\} & \{B\} & & \\ \{S,A\} & \{B\} & \{S,C\} & \{S,A\} & \\ \{B\} & \{A,C\} & \{A,C\} & \{B\} & \{A,C\} \\ \hline b & a & a & b & a \end{array}$$

در این مورد برای مثال $X_{۲,۴} = \{B\}$ و طریقه‌ی محاسبه آن با استفاده از الگوریتم بازگشتی به صورت زیر است. $X_{۲,۴}$ مجموعه متغیرهایی است که می‌توانند aab را بسازند و به نوبه‌ی خود می‌تواند به یکی از دو صورت زیر ساخته شود:

[۲] dynamic programming

- الحاق رشته b به رشته‌ی aa: رشته b را اعضای $\{B\} = X_{1,1}$ و رشته aa را اعضای $\{B\} = X_{2,3}$ می‌سازند. بنابراین با الحاق اعضای این دو مجموعه به هم باید به دنبال متغیرهایی باشیم که به صورت $I \rightarrow BB$ هستند. در این‌جا چنین متغیری وجود ندارد.

- الحاق رشته ab به رشته‌ی a: رشته ab را اعضای $\{S, C\} = X_{3,4}$ و رشته a را اعضای $\{A, C\} = X_{2,2}$ می‌سازند. بنابراین با الحاق اعضای این دو مجموعه به هم باید دنبال متغیرهایی باشیم که به یک از شکل‌های $I \rightarrow AS$، $I \rightarrow CC$، $I \rightarrow CS$، $I \rightarrow AC$ باشند. در این‌جا متغیر B در شرایط قانون آخر صدق می‌کند و بنابراین $B \in X_{2,4}$.

در پایان چون $S \in X_{1,n}$ بنابراین داریم: $w \in L$.