Машини на Тюринг с добавки

Иво Стратев

14 ноември 2020 г.

Многолентови машини на Тюринг

Дефиниция

Нека $k \in \mathbb{N}_+$. $\langle Q,\ \Sigma,\ \Gamma,\ \delta,\ s,\ H\rangle$ наричаме **k-лентова машина на Тюринг**, ако

- Q е крайно множество. Множеството Q е множеството на контролните състояния на машината.
- Σ е крайна азбука, такава че $\Sigma \cap \{\triangleright, \sqcup, \leftarrow, \rightarrow, \downarrow\} = \emptyset$. Азбуката Σ е входната азбука на машината.
- Γ е крайна множество, такова че $\Sigma \cap \Gamma = \emptyset$ и $\Gamma \cap \{\triangleright, \sqcup, \leftarrow, \rightarrow, \downarrow\} = \emptyset$. Множеството Γ е множеството на спомагателните символи за машината.
- $s \in Q$. s е началното състояние.
- H е крайно множество, такова че $Q \cap H = \emptyset$. Множеството H е множеството на стоп състоянията.
- $\delta : Q \times (\Sigma \cup \Gamma \cup \{\triangleright, \sqcup\})^k \rightarrow H \cup (Q \times (\Sigma \cup \Gamma \cup \{\leftarrow, \rightarrow, \downarrow\})^k)$. δ е контролната функция. която задава програмата на машината и δ има свойството

$$
\begin{aligned}
&(\forall p, q \in Q)(\forall x \in seq(k, \Sigma \cup \Gamma \cup \{\triangleright \sqcup\}))(\forall c \in seq(k, \Sigma \cup \Gamma \cup \{\leftarrow, \rightarrow, \downarrow\}))\\
&(\delta(p, \langle x(1), x(2), \ldots, x(k)\rangle) = \langle q, \langle c(1), c(2), \ldots, c(k)\rangle\rangle \implies\\
&(\forall i \in \{1, 2, \ldots, k\})(x(i) = \triangleright \implies c(i) = \rightarrow)).
\end{aligned}
$$

Една k-лентова машина на Тюринг представлява устройство, което има k глави (курсора), всяка (всеки) от които може да бъде прикачен към отделна лента, но контролът на всичките k глави е синхронизиран и се извършва еднозначно от състоянието на машината и символите, които главите виждат едновременно (в един такт). Особеност е, че считаме първата лента за входна, тоест тя бива инциализирана с входната дума, а останалите ленти в най-общият случай биват инициализирани с $\rhd\sqcup^{t_i}$.

Понеже k може да е по-голямо от 1 добавяме нова команда $\downarrow$, която да бъде интерпретирана като: не прави нищо за съответната лента.

Дадените дефиниции от секцията за еднолентови машинит на Тюринг могат да бъдат обобщени за произволно k, но няма да го правим, защото в сила е следната важна теорема.

Теорема

Работата на всяка многолентова машина на Тюринг може да бъде симулирана от еднолентова машина на Тюринг. Тоест броят на лентите е без значение от гледна точка на изчислителна сила. Наличието на повече от една лента същност е само за удобство.

Пример

Нека Σ е крайна азбука. Нека $L = \{\omega\#\omega \mid \omega \in \Sigma^*\}$.

Ще докажем, че L е разрешим от двулентова машина на Тюринг.

Идеята ни ще е, че ако искаме да разпознаем $\alpha\#\alpha$, то можем да си мислим, че първата лента е инициализирана с дума от вида $\alpha\#\beta$. Тогава трябва да копираме на втората лента α. След това ще ни остане само да проверим дали $\alpha = \beta$ използвайки и двете ленти.

Правим следнта забележка, след като става дума за (полу)разрешимост то и двете ленти при инициализация се заделя допълнителна памет от една празна клетка в началото. Тоест ако γ е входната дума, то първата лента се инициализира с $\rhd \sqcup \gamma$, а втората с $\rhd\sqcup$.

Започваме от преходите, с които да копираме думата преди $\#$.

$$\begin{aligned}
\delta(start, \langle\sqcup,\sqcup\rangle) &= \langle copy, \langle\rightarrow,\rightarrow\rangle\rangle\\
(\forall x \in \Sigma)(\delta(copy, \langle x,\sqcup\rangle) &= \langle move, \langle\downarrow, x\rangle\rangle)\\
(\forall x \in \Sigma)(\delta(move, \langle x,x\rangle) &= \langle copy, \langle\rightarrow,\rightarrow\rangle\rangle)
\end{aligned}$$

Ако при копирането достигнем до $\#$ значи трябва да оставим глава на първата лента на място, а на втората да върнем една позиция на ляво и след това да преминем през всички копирани букви и да започнем проверката на уловието

$\alpha = \beta$.

$$
\begin{aligned}
\delta(\mathtt{copy}, \langle \#, \sqcup \rangle) &= \langle \mathtt{rewind}, \langle \downarrow, \leftarrow \rangle \rangle \\
(\forall x \in \Sigma)(\delta(\mathtt{rewind}, \langle \#, x \rangle) &= \langle \mathtt{rewind}, \langle \downarrow, \leftarrow \rangle \rangle) \\
\delta(\mathtt{rewind}, \langle \#, \sqcup \rangle) &= \langle \mathtt{match}, \langle \rightarrow, \rightarrow \rangle \rangle \\
(\forall x \in \Sigma)(\delta(\mathtt{match}, \langle x, x \rangle) &= \langle \mathtt{match}, \langle \rightarrow, \rightarrow \rangle \rangle) \\
\delta(\mathtt{match}, \langle \sqcup, \sqcup \rangle) &= \mathtt{yes}
\end{aligned}
$$

Нека разгледаме как работи ДМТ описана от горните преходи, като всички липсващи са $\mathtt{no}$.

Първо ако $\omega = \varepsilon$.

$$
\begin{aligned}
\delta^{10}(\mathtt{start}, \langle \triangleright\underline{\sqcup}\#, 2 \rangle, \langle \triangleright\underline{\sqcup}, 2 \rangle) = &\quad (\text{by } \delta(\mathtt{start}, \langle \sqcup, \sqcup \rangle) = \langle \mathtt{copy}, \langle \rightarrow, \rightarrow \rangle \rangle) \\
\delta^{9}(\mathtt{copy}, \langle \triangleright \sqcup \underline{\#}, 3 \rangle, \langle \triangleright \sqcup \_, 3 \rangle) = &\quad (\text{by } \delta(\mathtt{copy}, \langle \#, \sqcup \rangle) = \langle \mathtt{rewind}, \langle \downarrow, \leftarrow \rangle \rangle) \\
\delta^{8}(\mathtt{rewind}, \langle \triangleright \sqcup \underline{\#}, 3 \rangle, \langle \triangleright\underline{\sqcup}, 2 \rangle) = &\quad (\text{by } \delta(\mathtt{rewind}, \langle \#, \sqcup \rangle) = \langle \mathtt{match}, \langle \rightarrow, \rightarrow \rangle \rangle) \\
\delta^{7}(\mathtt{match}, \langle \triangleright \sqcup \#\_, 4 \rangle, \langle \triangleright \sqcup \_, 3 \rangle) = &\quad (\text{by } \delta(\mathtt{match}, \langle \sqcup, \sqcup \rangle) = \mathtt{yes}) \\
&\quad \langle \mathtt{yes}, \triangleright \sqcup \# \rangle
\end{aligned}
$$

Сега за $\omega = \mathtt{ab}$

$$
\begin{aligned}
\delta^{33}(\mathtt{start}, \langle \triangleright\underline{\sqcup}\mathtt{ab}\#\mathtt{ab}, 2 \rangle, \langle \triangleright\underline{\sqcup}, 2 \rangle) = &\quad (\text{by } \delta(\mathtt{start}, \langle \sqcup, \sqcup \rangle) = \langle \mathtt{copy}, \langle \rightarrow, \rightarrow \rangle \rangle) \\
\delta^{32}(\mathtt{copy}, \langle \triangleright \sqcup \underline{\mathtt{a}}\mathtt{b}\#\mathtt{ab}, 3 \rangle, \langle \triangleright \sqcup \_, 3 \rangle) = &\quad (\text{by } \delta(\mathtt{copy}, \langle \mathtt{a}, \sqcup \rangle) = \langle \mathtt{move}, \langle \downarrow, \mathtt{a} \rangle \rangle) \\
\delta^{31}(\mathtt{move}, \langle \triangleright \sqcup \underline{\mathtt{a}}\mathtt{b}\#\mathtt{ab}, 3 \rangle, \langle \triangleright \sqcup \underline{\mathtt{a}}, 3 \rangle) = &\quad (\text{by } \delta(\mathtt{move}, \langle \mathtt{a}, \mathtt{a} \rangle) = \langle \mathtt{copy}, \langle \rightarrow, \rightarrow \rangle \rangle) \\
\delta^{30}(\mathtt{copy}, \langle \triangleright \sqcup \mathtt{a}\underline{\mathtt{b}}\#\mathtt{ab}, 4 \rangle, \langle \triangleright \sqcup \mathtt{a}\_, 4 \rangle) = &\quad (\text{by } \delta(\mathtt{copy}, \langle \mathtt{b}, \sqcup \rangle) = \langle \mathtt{move}, \langle \downarrow, \mathtt{b} \rangle \rangle) \\
\delta^{29}(\mathtt{move}, \langle \triangleright \sqcup \mathtt{a}\underline{\mathtt{b}}\#\mathtt{ab}, 4 \rangle, \langle \triangleright \sqcup \mathtt{a}\underline{\mathtt{b}}, 4 \rangle) = &\quad (\text{by } \delta(\mathtt{move}, \langle \mathtt{b}, \mathtt{b} \rangle) = \langle \mathtt{copy}, \langle \rightarrow, \rightarrow \rangle \rangle) \\
\delta^{28}(\mathtt{copy}, \langle \triangleright \sqcup \mathtt{ab}\underline{\#}\mathtt{ab}, 5 \rangle, \langle \triangleright \sqcup \mathtt{ab}\_, 5 \rangle) = &\quad (\text{by } \delta(\mathtt{copy}, \langle \#, \sqcup \rangle) = \langle \mathtt{rewind}, \langle \downarrow, \leftarrow \rangle \rangle) \\
\delta^{27}(\mathtt{rewind}, \langle \triangleright \sqcup \mathtt{ab}\underline{\#}\mathtt{ab}, 5 \rangle, \langle \triangleright \sqcup \mathtt{a}\underline{\mathtt{b}}, 4 \rangle) = &\quad (\text{by } \delta(\mathtt{rewind}, \langle \#, \mathtt{b} \rangle) = \langle \mathtt{rewind}, \langle \downarrow, \leftarrow \rangle \rangle) \\
\delta^{26}(\mathtt{rewind}, \langle \triangleright \sqcup \mathtt{ab}\underline{\#}\mathtt{ab}, 5 \rangle, \langle \triangleright \sqcup \underline{\mathtt{a}}\mathtt{b}, 3 \rangle) = &\quad (\text{by } \delta(\mathtt{rewind}, \langle \#, \mathtt{a} \rangle) = \langle \mathtt{rewind}, \langle \downarrow, \leftarrow \rangle \rangle) \\
\delta^{25}(\mathtt{rewind}, \langle \triangleright \sqcup \mathtt{ab}\underline{\#}\mathtt{ab}, 5 \rangle, \langle \triangleright\underline{\sqcup}\mathtt{ab}, 2 \rangle) = &\quad (\text{by } \delta(\mathtt{rewind}, \langle \#, \sqcup \rangle) = \langle \mathtt{match}, \langle \rightarrow, \rightarrow \rangle \rangle) \\
\delta^{24}(\mathtt{match}, \langle \triangleright \sqcup \mathtt{ab}\#\underline{\mathtt{a}}\mathtt{b}, 6 \rangle, \langle \triangleright \sqcup \underline{\mathtt{a}}\mathtt{b}, 3 \rangle) = &\quad (\text{by } \delta(\mathtt{match}, \langle \mathtt{a}, \mathtt{a} \rangle) = \langle \mathtt{match}, \langle \rightarrow, \rightarrow \rangle \rangle) \\
\delta^{23}(\mathtt{match}, \langle \triangleright \sqcup \mathtt{ab}\#\mathtt{a}\underline{\mathtt{b}}, 7 \rangle, \langle \triangleright \sqcup \mathtt{a}\underline{\mathtt{b}}, 4 \rangle) = &\quad (\text{by } \delta(\mathtt{match}, \langle \mathtt{b}, \mathtt{b} \rangle) = \langle \mathtt{match}\langle \rightarrow, \rightarrow \rangle \rangle) \\
\delta^{22}(\mathtt{match}, \langle \triangleright \sqcup \mathtt{ab}\#\mathtt{ab}\_, 8 \rangle, \langle \triangleright \sqcup \mathtt{ab}\_, 5 \rangle) = &\quad (\text{by } \delta(\mathtt{match}, \langle \sqcup, \sqcup \rangle) = \mathtt{yes}) \\
&\quad \langle \mathtt{yes}, \triangleright \sqcup \mathtt{ab}\#\mathtt{ab} \rangle
\end{aligned}
$$

Нека разгледаме и един отрицателен пример, например само за думата $\mathtt{ab}$, която не е от L.

$$
\begin{aligned}
\delta^{33}(\mathtt{start}, \langle \triangleright\underline{\sqcup}ab, 2\rangle, \langle \triangleright\underline{\sqcup}, 2\rangle) = &\quad (\text{by } \delta(\mathtt{start}, \langle \sqcup, \sqcup\rangle) = \langle \mathtt{copy}, \langle \rightarrow, \rightarrow\rangle\rangle) \\
\delta^{32}(\mathtt{copy}, \langle \triangleright \sqcup \underline{a}b, 3\rangle, \langle \triangleright \sqcup \_, 3\rangle) = &\quad (\text{by } \delta(\mathtt{copy}, \langle a, \sqcup\rangle) = \langle \mathtt{move}, \langle \downarrow, a\rangle\rangle) \\
\delta^{31}(\mathtt{move}, \langle \triangleright \sqcup \underline{a}b, 3\rangle, \langle \triangleright \sqcup \underline{a}, 3\rangle) = &\quad (\text{by } \delta(\mathtt{move}, \langle a, a\rangle) = \langle \mathtt{copy}, \langle \rightarrow, \rightarrow\rangle\rangle) \\
\delta^{30}(\mathtt{copy}, \langle \triangleright \sqcup a\underline{b}, 4\rangle, \langle \triangleright \sqcup a\_, 4\rangle) = &\quad (\text{by } \delta(\mathtt{copy}, \langle b, \sqcup\rangle) = \langle \mathtt{move}, \langle \downarrow, b\rangle\rangle) \\
\delta^{29}(\mathtt{move}, \langle \triangleright \sqcup a\underline{b}, 4\rangle, \langle \triangleright \sqcup a\underline{b}, 4\rangle) = &\quad (\text{by } \delta(\mathtt{move}, \langle b, b\rangle) = \langle \mathtt{copy}, \langle \rightarrow, \rightarrow\rangle\rangle) \\
\delta^{28}(\mathtt{copy}, \langle \triangleright \sqcup ab\_, 5\rangle, \langle \triangleright \sqcup ab\_, 5\rangle) = &\quad (\text{by } \delta(\mathtt{copy}, \langle \sqcup, \sqcup\rangle) = \mathtt{no}) \\
&\quad \langle \mathtt{no}, \triangleright \sqcup ab\rangle
\end{aligned}
$$

Задачи за упражнение

Задача 1.

Ако $n = |\Sigma|$ намерете $|Q|$.

Задача 2.

Ако $n = |\Sigma|$ определете процентът на полезни преходи. Пресметнете приблизителната стойност за $n = 10$.

Задача 3.

Ако $\alpha \in \Sigma$ и $n = |\alpha|$ пресметнете за колко на брой стъпки построената ДМТ приема думата $\alpha\#\alpha$.

Задача 4.

Покажате ДМТ, която изчислява функцията $f : \mathbb{N}^2 \rightarrow \mathbb{N}$, действаща по правилото $f(n, t) = 2^{\mathtt{Log}(n)+1}.t$.

Недетерминирани машини на Тюринг

Дефиниция за еднолентова недетерминирана машина на Тюринг

$\langle Q,\ \Sigma,\ \Gamma,\ \Delta,\ s,\ H\rangle$ наричаме **еднолентова недетерминирана машина на Тюринг**, ако

- Q е крайно множество. Множеството Q е множеството на контролните състояния на машината.
- Σ е крайна азбука, такава че $\Sigma \cap \{\triangleright, \sqcup, \leftarrow, \rightarrow\} = \emptyset$. Азбуката Σ е входната азбука на машината.
- Γ е крайна множество, такова че $\Sigma \cap \Gamma = \emptyset$ и $\Gamma \cap \{\triangleright, \sqcup, \leftarrow, \rightarrow\} = \emptyset$. Множеството Γ е множеството на спомагателните символи за машината.
- $s \in Q$. s е началното състояние.
- H е крайно множество, такова че $Q \cap H = \emptyset$. Множеството H е множеството на стоп състоянията.
- $\Delta \subseteq (Q \times (\Sigma \cup \Gamma \cup \{\triangleright, \sqcup\})) \times (H \cup (Q \times (\Sigma \cup \Gamma \cup \{\leftarrow, \rightarrow\})))$. Δ е релацията описваща възможните преходи на машината и има свойството

$$(\forall p, q \in Q)(\forall x \in \Sigma \cup \Gamma \cup \{\leftarrow, \rightarrow\})(\langle\langle p, \triangleright\rangle, \langle q, x\rangle\rangle \in \Delta \implies x = \rightarrow)$$

Преминаваме към описанието на работата на една ЕНМТ.
Нека $\mathcal{N} = \langle Q,\ \Sigma,\ \Gamma,\ \Delta,\ s,\ H\rangle$ е ЕНМТ.
Конфигурация на машината $\mathcal{N}$ ще наричаме всеки елемент на множеството

$$(Q \cup H) \times \mathbb{N}_+ \times (\{\triangleright\} \cdot (\Sigma \cup \Gamma \cup \{\sqcup\})^*)$$

Множеството на началните конфигурации е $\{s\} \times \{2\} \times (\{\triangleright\sqcup\} \cdot \Sigma^*)$,
а $H \times \mathbb{N}_+ \times (\{\triangleright\} \cdot (\Sigma \cup \Gamma \cup \{\sqcup\})^*)$ е множеството на заключителните конфигурации.
Нека $\Lambda = \{\triangleright\} \cdot (\Sigma \cup \Gamma \cup \{\sqcup\})^*$. Дефинираме бинарна релация $singleStep_\Delta$ между $Q \times \mathbb{N}_+ \times \Lambda$ и $(Q \cup H) \times \mathbb{N}_+ \times \Lambda$.

$$\begin{aligned} singleStep_\Delta = &\\ \{\langle\langle q, i, \alpha\rangle, \langle h, i, \alpha\rangle\rangle \mid q \in Q, i \in \mathbb{N}_+, \alpha \in \Lambda, h \in H \mathbin{\&} \langle\langle q, \alpha[i]\rangle, h\rangle \in \Delta\} &\\ \cup \{\langle\langle q, i, \alpha\rangle, \langle p, i+1, \alpha\rangle\rangle \mid q, p \in Q, i \in \mathbb{N}_+, \alpha \in \Lambda \mathbin{\&} \langle\langle q, \alpha[i]\rangle, \langle p, \rightarrow\rangle\rangle \in \Delta\} &\\ \cup \{\langle\langle q, i, \alpha\rangle, \langle p, i-1, \alpha\rangle\rangle \mid q, p \in Q, i \in \mathbb{N} \setminus \{0, 1\}, \alpha \in \Lambda \mathbin{\&} \langle\langle q, \alpha[i]\rangle, \langle p, \leftarrow\rangle\rangle \in \Delta\} &\\ \cup \{\langle\langle q, i, \alpha\rangle, \langle p, i, \alpha[i, x]\rangle\rangle \mid q, p \in Q, i \in \mathbb{N}_+, x \in \Sigma \cup \Gamma, \alpha \in \Lambda \mathbin{\&} \langle\langle q, \alpha[i]\rangle, \langle p, x\rangle\rangle \in \Delta\} & \end{aligned}$$

След като имаме релацията $\mathsf{singleStep}_\Delta$, която описва една стъпка по машината $\mathcal{N}$ чрез рекурсия дефинираме редица от бинарни релации step^n_Δ в $(Q \cup H) \times \mathbb{N}_+ \times \Lambda$, които удовлетворяват рекуретната връзка

$$\mathsf{step}^0_\Delta = \mathsf{Id}_{(Q\cup H)\times\mathbb{N}_+\times\Lambda}$$

$$\mathsf{step}^{n+1}_\Delta = \{\langle\langle q, i, \alpha\rangle, \langle p, j, \beta\rangle\rangle \mid q, p \in Q \cup H \ \&\ i, j \in \mathbb{N}_+ \ \&\ \alpha, \beta \in \Lambda \ \&\ (\exists t \in Q \cup H) \\ (\exists k \in \mathbb{N}_+)(\exists \gamma \in \Lambda)(\langle q, i, \alpha\rangle \ \mathsf{singleStep}_\Delta \ \langle t, k, \gamma\rangle \ \&\ \langle t, k, \gamma\rangle \ \mathsf{step}^n_\Delta \ \langle p, j, \beta\rangle)\}$$

Полагаме $\mathsf{step}^*_\Delta = \bigcup_{n\in\mathbb{N}} \mathsf{step}^n_\Delta$. Така step^*_Δ е рефлексивното и транзитивно затваряне на $\mathsf{singleStep}_\Delta$ в множеството $(Q \cup H) \times \mathbb{N}_+ \times \Lambda$.

Така езикът върху, който машината $\mathcal{N}$ завършва е

$$\{\omega \in \Sigma^* \mid (\exists k \in \mathbb{N}_+)(\exists \gamma \in \Lambda)(\exists h \in H)(\langle s, 2, \triangleright \sqcup \omega\rangle \ \mathsf{step}^*_\Delta \ \langle h, k, \gamma\rangle)\}$$

Ако разпишем какво значи две конфигурации да са в релацията step^*_Δ и вземем предвид, че $s \notin H$, то получаваме езика

$$\{\omega \in \Sigma^* \mid (\exists k \in \mathbb{N}_+)(\exists \gamma \in \Lambda)(\exists h \in H)(\exists n \in \mathbb{N}_+)(\langle s, 2, \triangleright \sqcup \omega\rangle \ \mathsf{step}^n_\Delta \ \langle h, k, \gamma\rangle)\}$$

Вече сме готови да кажем какво значи една ЕНМТ да разрешава език.

Разрешимост от ЕНМТ

Нека Σ е крайна азбука и нека L е език над Σ.
Нека $\mathcal{N} = \langle Q,\ \Sigma,\ \Gamma,\ \Delta,\ \mathsf{start},\ \{\mathsf{yes}, \mathsf{no}\}\rangle$ е ЕНМТ.
Казваме, че $\mathcal{N}$ разрешава L, ако езикът върху, който завършва $\mathcal{N}$ е Σ^* и

$$(\forall \omega \in \Sigma^*)(\omega \in L \iff (\exists k \in \mathbb{N}_+)(\exists \gamma \in \{\triangleright\}\cdot(\Sigma\cup\Gamma\cup\{\sqcup\})^*)(\langle s, 2, \triangleright\sqcup\omega\rangle \ \mathsf{step}^*_\Delta \ \langle \mathsf{yes}, k, \gamma\rangle))$$

Един език е разрешим от ЕНМТ, ако има ЕНМТ, която го разрешава.

Пример

Нека Σ е крайна азбука и нека $L = \{\omega\omega \mid \omega \in \Sigma\}$. Ще покажем, че L е разрешим от ЕНМТ. Ако $\omega = \alpha.x$ идеята ще ни е следната от $\triangleright\underline{\sqcup}\alpha.x\alpha.x$ да достигнем до $\triangleright \sqcup \alpha.x\alpha.x\_$, след което да запомним x, да я изтием като запишем \$ и да започнем да се връщаме назад като преминаваме през буквите различни от x, а в ситуация на достигане на x позволяваме да преминем през нея или да презапишем \$. Тоест ще ползваме недетерминираността за да отгатнем средата на $\omega\omega$. Така независимо каква е α успешно ще достигнем до $\triangleright \sqcup \alpha.\underline{\$}\alpha.\$$ и ще продължим напълно детерминистично. Тоест детерминистично запомняме и маркираме последната буква в активната част на лентата, след което недетерминистично я match-ваме и след това отново продължаваме детерминистично да match-ваме останалите букви.

Ще строим множеството от контролни състояния и релацията на преходите Δ като вместо $\langle X, Y\rangle \in \Delta$ ще пишем $X \mathrel{\Delta} Y$ и всички преходи, които не изброим отново ще са $\mathfrak{no}$ преходи с цел разрешимост, а не полуразрешимост.

Първо да съобразим, че е възможно $\omega = \varepsilon$ и значи имаме преходи

$$\begin{aligned}
\langle \mathsf{start}, \sqcup\rangle &\mathrel{\Delta} \langle \mathsf{letter}, \rightarrow\rangle\\
\langle \mathsf{letter}, \sqcup\rangle &\mathrel{\Delta} \mathsf{yes}\\
(\forall x \in \Sigma)(\langle \mathsf{letter}, x\rangle &\mathrel{\Delta} \langle \mathsf{seek}, \rightarrow\rangle)\\
(\forall x \in \Sigma)(\langle \mathsf{seek}, x\rangle &\mathrel{\Delta} \langle \mathsf{seek}, \rightarrow\rangle)\\
\langle \mathsf{seek}, \sqcup\rangle &\mathrel{\Delta} \langle \mathsf{read}, \leftarrow\rangle\\
(\forall x \in \Sigma)(\langle \mathsf{read}, x\rangle &\mathrel{\Delta} \langle \mathsf{find}_x, \$\rangle)\\
(\forall x \in \Sigma)(\langle \mathsf{find}_x, \$\rangle &\mathrel{\Delta} \langle \mathsf{find}_x, \leftarrow\rangle)\\
(\forall x \in \Sigma)(\forall y \in \Sigma^*)(\langle \mathsf{find}_x, y\rangle &\mathrel{\Delta} \langle \mathsf{find}_x, \leftarrow\rangle)\\
(\forall x \in \Sigma)(\langle \mathsf{find}_x, x\rangle &\mathrel{\Delta} \langle \mathsf{skip}, \$\rangle)\\
\langle \mathsf{skip}, \$\rangle &\mathrel{\Delta} \langle \mathsf{skip}, \rightarrow\rangle
\end{aligned}$$

Но е възможно $\omega = x$, така ще сме достигнали до $\triangleright \sqcup \$\$\_$ това значи, че трябва да се върнем наляво преминавайки само през \$ и да достигнем до $\sqcup$ и да приемем думата. Разбира се може да достигнем и до ситуация $\triangleright \sqcup \beta\underline{y}\$\$$, в която да отхвърлим, но както казахме отхвърлящите преходи няма да ги изброяваме.

$$\begin{aligned}
\langle \mathsf{skip}, \sqcup\rangle &\mathrel{\Delta} \langle \mathsf{finish}, \leftarrow\rangle\\
\langle \mathsf{finish}, \$\rangle &\mathrel{\Delta} \langle \mathsf{finish}, \leftarrow\rangle\\
\langle \mathsf{finish}, \sqcup\rangle &\mathrel{\Delta} \mathsf{yes}
\end{aligned}$$

Все още не сме описали напълно всички преходи, защото като отгатнем средата започваме детерминистичен matching.

$$\begin{aligned}
(\forall x \in \Sigma)(\langle \mathsf{skip}, x\rangle &\mathrel{\Delta} \langle \mathsf{seek}, \rightarrow\rangle)\\
\langle \mathsf{seek}, \$\rangle &\mathrel{\Delta} \langle \mathsf{match}, \leftarrow\rangle\\
(\forall x \in \Sigma)(\langle \mathsf{match}, x\rangle &\mathrel{\Delta} \langle \mathsf{skip}_x, \$\rangle)\\
(\forall x \in \Sigma)(\langle \mathsf{skip}_x, \$\rangle &\mathrel{\Delta} \langle \mathsf{memory}_x, \leftarrow\rangle)\\
(\forall x \in \Sigma)(\forall y \in \Sigma^*)(\langle \mathsf{memory}_x, y\rangle &\mathrel{\Delta} \langle \mathsf{memory}_x, \leftarrow\rangle)\\
(\forall x \in \Sigma)(\langle \mathsf{memory}_x, \$\rangle &\mathrel{\Delta} \langle \mathsf{match}_x, \leftarrow\rangle)\\
(\forall x \in \Sigma)(\langle \mathsf{match}_x, \$\rangle &\mathrel{\Delta} \langle \mathsf{match}_x, \leftarrow\rangle)\\
(\forall x \in \Sigma)(\langle \mathsf{match}_x, x\rangle &\mathrel{\Delta} \langle \mathsf{skip}, \$\rangle)
\end{aligned}$$

Нека разгледаме какво ще бъде изчислението при $\omega = \mathsf{abb}$. Вместо step^*_Δ ще пишем само $\vdash$.

$$
\begin{gathered}
\langle \mathsf{start},\ 2,\ \vartriangleright\underline{\sqcup}\mathsf{abbabb}\rangle \vdash \langle \mathsf{letter},\ 3,\ \vartriangleright\sqcup\underline{\mathsf{a}}\mathsf{bbabb}\rangle \vdash \\
\langle \mathsf{seek},\ 4,\ \vartriangleright\sqcup\mathsf{a}\underline{\mathsf{b}}\mathsf{babb}\rangle \vdash \langle \mathsf{seek},\ 5,\ \vartriangleright\sqcup\mathsf{ab}\underline{\mathsf{b}}\mathsf{abb}\rangle \vdash \\
\langle \mathsf{seek},\ 6,\ \vartriangleright\sqcup\mathsf{abb}\underline{\mathsf{a}}\mathsf{bb}\rangle \vdash \langle \mathsf{seek},\ 7,\ \vartriangleright\sqcup\mathsf{abba}\underline{\mathsf{b}}\mathsf{b}\rangle \vdash \\
\langle \mathsf{seek},\ 8,\ \vartriangleright\sqcup\mathsf{abbab}\underline{\mathsf{b}}\rangle \vdash \langle \mathsf{seek},\ 9,\ \vartriangleright\sqcup\mathsf{abbabb}\underline{\ }\rangle \vdash \\
\langle \mathsf{read},\ 8,\ \vartriangleright\sqcup\mathsf{abbab}\underline{\mathsf{b}}\rangle \vdash \langle \mathsf{find}_{\mathsf{b}},\ 8,\ \vartriangleright\sqcup\mathsf{abbab}\underline{\$}\rangle \vdash \\
\langle \mathsf{find}_{\mathsf{b}},\ 7,\ \vartriangleright\sqcup\mathsf{abba}\underline{\mathsf{b}}\$\rangle \vdash \langle \mathsf{find}_{\mathsf{b}},\ 6,\ \vartriangleright\sqcup\mathsf{abb}\underline{\mathsf{a}}\mathsf{b}\$\rangle \vdash \\
\langle \mathsf{find}_{\mathsf{b}},\ 5,\ \vartriangleright\sqcup\mathsf{ab}\underline{\mathsf{b}}\mathsf{ab}\$\rangle \vdash \langle \mathsf{skip},\ 5,\ \vartriangleright\sqcup\mathsf{ab}\underline{\$}\mathsf{ab}\$\rangle \vdash \\
\langle \mathsf{skip},\ 6,\ \vartriangleright\sqcup\mathsf{ab}\$\underline{\mathsf{a}}\mathsf{b}\$\rangle \vdash \langle \mathsf{seek},\ 7,\ \vartriangleright\sqcup\mathsf{ab}\$\mathsf{a}\underline{\mathsf{b}}\$\rangle \vdash \\
\langle \mathsf{seek},\ 8,\ \vartriangleright\sqcup\mathsf{ab}\$\mathsf{ab}\underline{\$}\rangle \vdash \langle \mathsf{match},\ 7,\ \vartriangleright\sqcup\mathsf{ab}\$\mathsf{a}\underline{\mathsf{b}}\$\rangle \vdash \\
\langle \mathsf{skip}_{\mathsf{b}},\ 7,\ \vartriangleright\sqcup\mathsf{ab}\$\mathsf{a}\underline{\$}\$\rangle \vdash \langle \mathsf{memory}_{\mathsf{b}},\ 6,\ \vartriangleright\sqcup\mathsf{ab}\$\underline{\mathsf{a}}\$\$\rangle \vdash \\
\langle \mathsf{memory}_{\mathsf{b}},\ 5,\ \vartriangleright\sqcup\mathsf{ab}\underline{\$}\mathsf{a}\$\rangle \vdash \langle \mathsf{match}_{\mathsf{b}},\ 4,\ \vartriangleright\sqcup\mathsf{a}\underline{\mathsf{b}}\$\mathsf{a}\$\$\rangle \vdash \\
\langle \mathsf{skip},\ 4,\ \vartriangleright\sqcup\mathsf{a}\underline{\$}\$\mathsf{a}\$\$\rangle \vdash \langle \mathsf{skip},\ 5,\ \vartriangleright\sqcup\mathsf{a}\$\underline{\$}\mathsf{a}\$\$\rangle \vdash \\
\langle \mathsf{skip},\ 6,\ \vartriangleright\sqcup\mathsf{a}\$\$\underline{\mathsf{a}}\$\$\rangle \vdash \langle \mathsf{seek},\ 7,\ \vartriangleright\sqcup\mathsf{a}\$\$\mathsf{a}\underline{\$}\$\rangle \vdash \\
\langle \mathsf{match},\ 6,\ \vartriangleright\sqcup\mathsf{a}\$\$\underline{\mathsf{a}}\$\$\rangle \vdash \langle \mathsf{skip}_{\mathsf{a}},\ 6,\ \vartriangleright\sqcup\mathsf{a}\$\$\underline{\$}\$\$\rangle \vdash \\
\langle \mathsf{memory}_{\mathsf{a}},\ 5,\ \vartriangleright\sqcup\mathsf{a}\$\underline{\$}\$\$\$\rangle \vdash \langle \mathsf{match}_{\mathsf{a}},\ 4,\ \vartriangleright\sqcup\mathsf{a}\$\underline{\$}\$\$\$\rangle \vdash \\
\langle \mathsf{match}_{\mathsf{a}},\ 3,\ \vartriangleright\sqcup\underline{\mathsf{a}}\$\$\$\$\$\rangle \vdash \langle \mathsf{skip},\ 3,\ \vartriangleright\sqcup\underline{\$}\$\$\$\$\$\rangle \vdash \\
\langle \mathsf{skip},\ 4,\ \vartriangleright\sqcup\$\underline{\$}\$\$\$\$\rangle \vdash \langle \mathsf{skip},\ 5,\ \vartriangleright\sqcup\$\$\underline{\$}\$\$\$\rangle \vdash \cdots \vdash \\
\langle \mathsf{skip},\ 9,\ \vartriangleright\sqcup\$\$\$\$\$\$\underline{\ }\rangle \vdash \langle \mathsf{finish},\ 8,\ \vartriangleright\sqcup\$\$\$\$\$\underline{\$}\rangle \vdash \cdots \vdash \\
\langle \mathsf{finish},\ 2,\ \vartriangleright\underline{\sqcup}\$\$\$\$\$\$\rangle \vdash \langle \mathsf{yes},\ 2,\ \vartriangleright\underline{\sqcup}\$\$\$\$\$\$\rangle
\end{gathered}
$$