Uzáverové vlastnosti kontextových jazykov

Veta

$\mathcal{L}_{CS}$ je uzavretá na zjednotenie.

Predpoklady: Ak $L_1, L_2 \in \mathcal{L}_{CS}$ potom musia existovať kontextové G_1, G_2 také, že $L(G_1) = L_1$ a $L(G_2) = L_2$. Nech $G_1 = (N_1, T_1, P_1, S_1), G_2 = (N_2, T_2, P_2, S_2) N1$ a N_2 sú disjunktné T_1 a T_2 sú disjunktné.

Konštrukcia: novú gramatiku zostrojíme pomocou pridaných nových pravidiel P_{New}:

$S_0 \to S_1$

$S_0 \to S_2$

$G = (N, T, P, S) = (N_1 \cup N_2 \cup S_0, T_1 \cup T_2, P_1 \cup P_2 \cup P_{New}, S_0)$

Overenie: pridali sme nové pravidlá, ktoré sú kontextové, nová G je kontextová, taká, že L(G) = L a $L \in \mathcal{L}_{CS}$ Trieda $\mathcal{L}_{CS}$ je uzavretá vzhľadom na operáciu $\cup$.

Vytvoríme nový neterminál S0:

Konštrukcia: novú gramatiku zostrojíme pomocou pridaných nových pravidiel P_{New}:

$\boxed{S_0 \to S_1}$

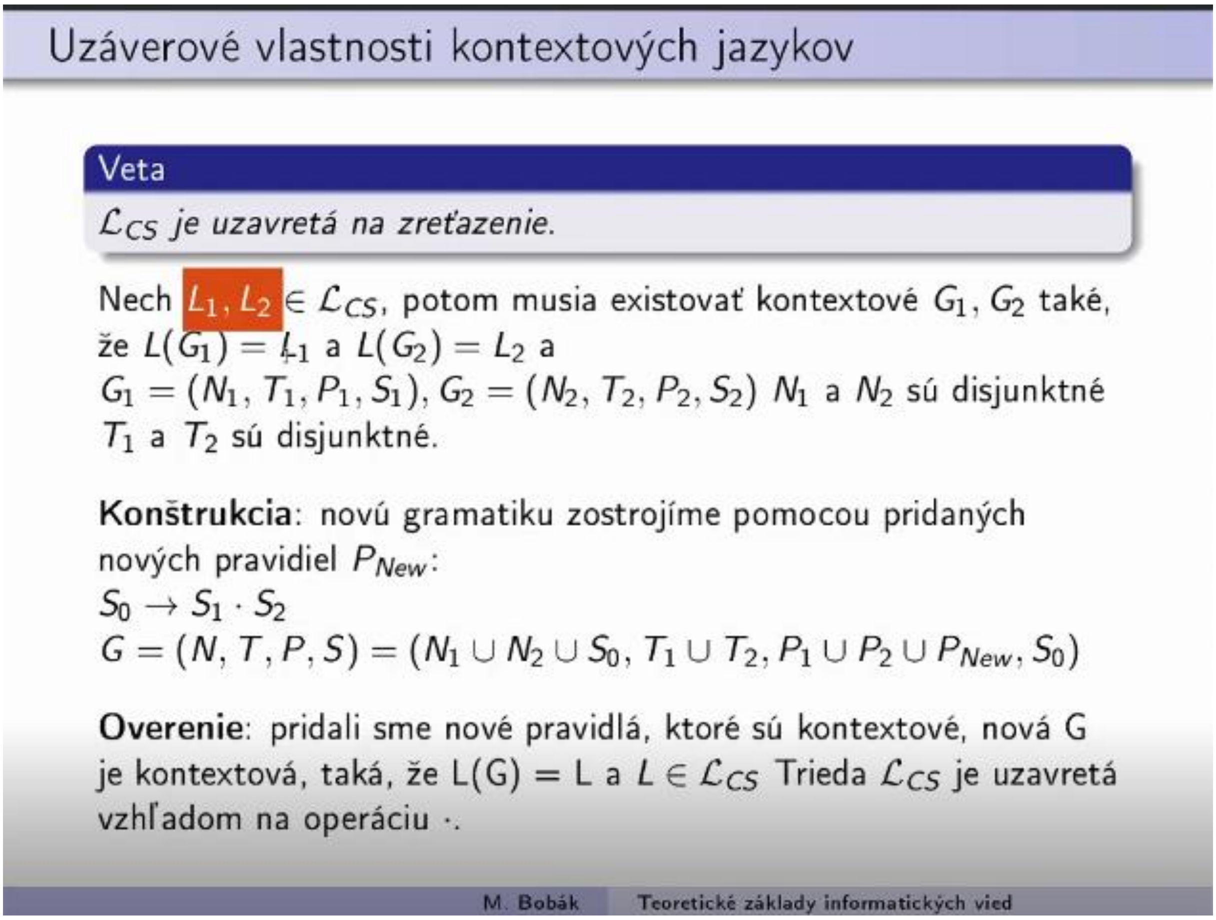
$\boxed{S_0 \to S_2}$

S0 vieme prepísať na S1 z G1 alebo S2 z G2

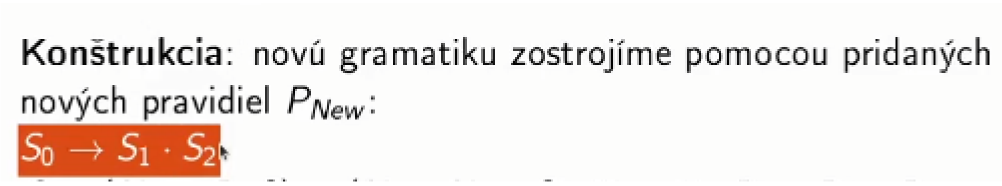
Následne zjednotíme všetky neterminály a pridáme S0, t.j. nový počiatočný neterminál (N). Pridáme terminály (T) a pravidlá (P), vrátane PNew, a nakoniec S0.

$$G = (N, T, P, S) = (N_1 \cup N_2 \cup S_0, T_1 \cup T_2, P_1 \cup P_2 \cup P_{New}, S_0)$$

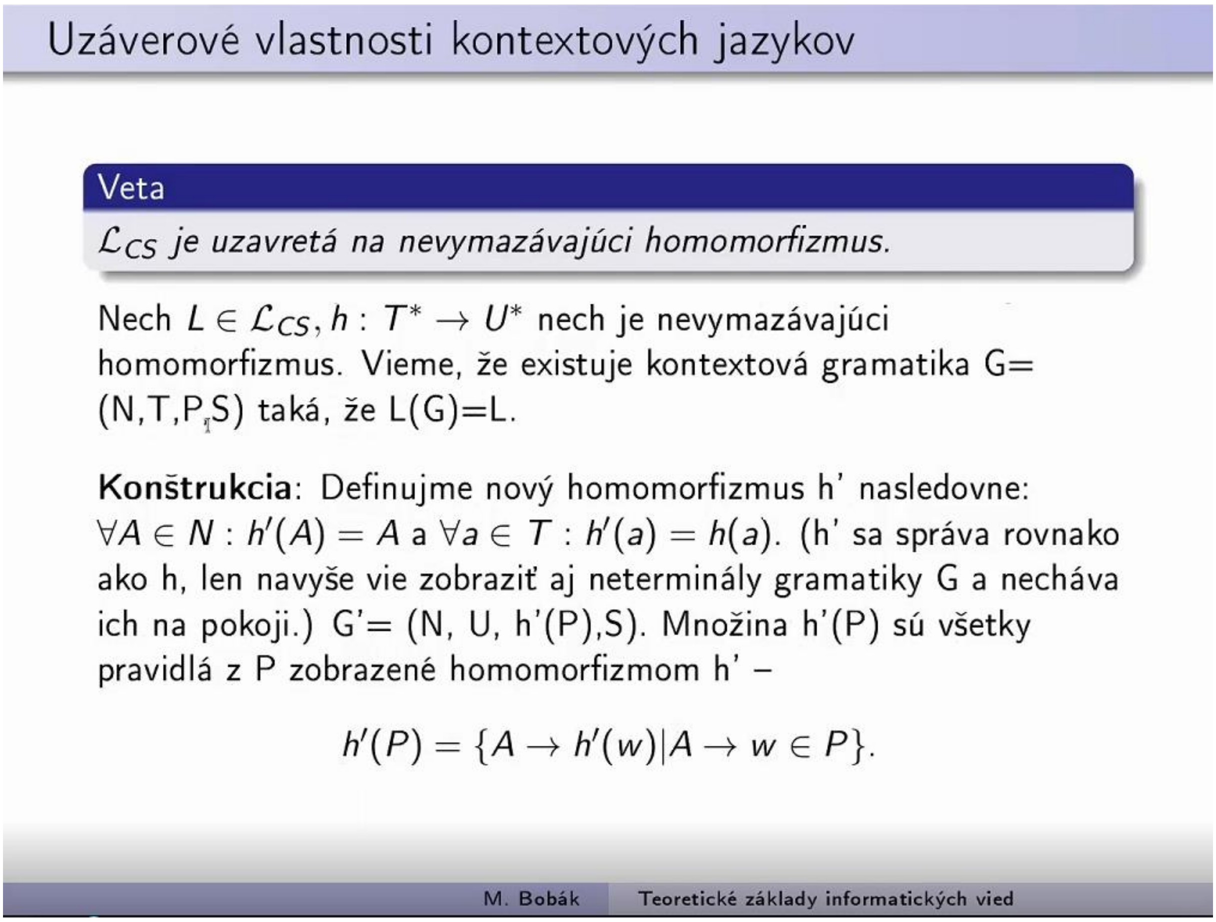
Predpokladáme že G1 a G2 sú kontextové, keď aj nová gramatika G je kontextová, tak je trieda LCS uzavretá na zjednotenie.



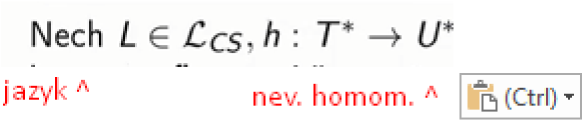
Oproti zjednoteniu, kde sme si mohli vybrať, či chceme vytvoriť slovo z gramatiky G1 alebo G2 (to bolo to S0->S1 a S0->S2), tuná počiatočné neterminály vstupných gramatík zreťazíme.



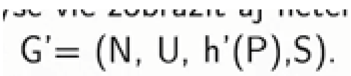
A opäť, pokiaľ po zreťazení vznikne gramatika ktorá je kontextová, tak je veta uzavretá na zreťazenie.



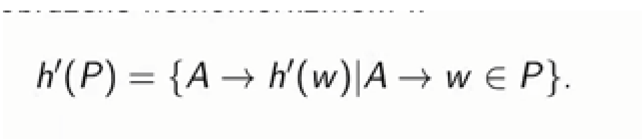
Vstupom je jazyk a nevymazávajúci homomorfizmus



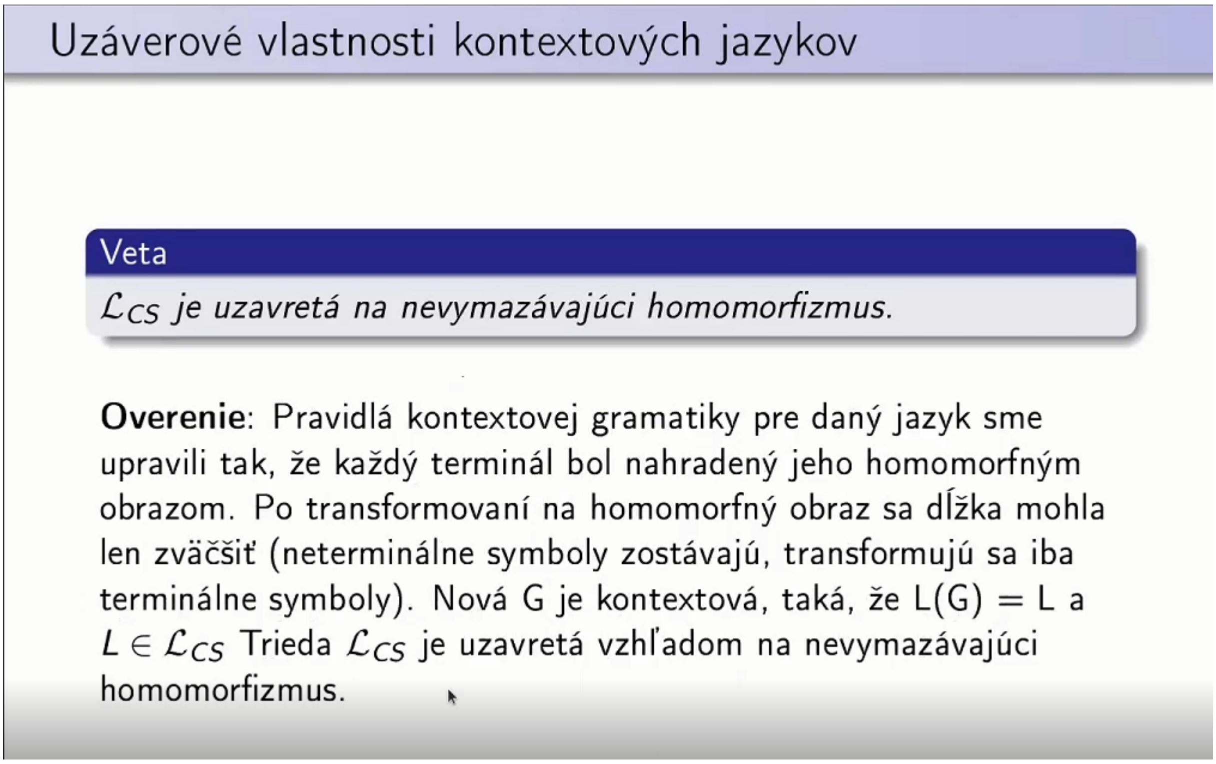
Zadefinujeme si homom. h‘ taký, že neterminály sa zobrazia na tie isté neterminály (h‘(A)=A) a terminálne symboli h‘ sa zobrazia na tie isté, ako pre h, t.j. h‘(a)=h(a). Teda h‘ necháva neterminály napokoji a terminály zobrazí na h.



Teda budeme mať tie isté neterminály, terminály, na pravidlá (P) aplikujeme homomorfizmus (h‘) a ostáva nám počiatočný neterminál S.



Toto vlastne hovorí, že keď v pôvodnej gramatike sa A->w (A sa zobrazí na w), tak v novej gramatike sa A zobrazí na A->h‘(w)



Homomorfizmus je nevymazávajúci preto, lebo pôvodná gramatika nám ostala taká istá, iba sa prípadne mohla zväčšiť, a stále nám ostal kontextový jazyk.

Poznámka: podľa mňa tento homom. vysvetlil strašne nejasne tak je možné že som to nepochopil správne.

Uzáverové vlastnosti kontextových jazykov

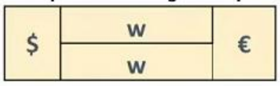
Veta

$\mathcal{L}_{CS}$ je uzavretá na prienik.

Nech $L_1, L_2 \in \mathcal{L}_{CS}$, potom musia existovať lineárne ohraničené automaty A_1, A_2 také, že $L(A_1) = L_1$ a $L(A_2) = L_2$.

Konštrukcia:

Dvojpáskový LOA rozpoznávajúci prienik:



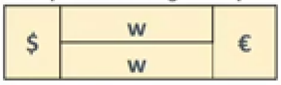
Akceptovanie:

A_1,LOA_1	A	N	A	N
A_2, LOA_2	A	A	N	N
A, LOA	**A**	**N**	**N**	**N**

Overenie: Nový automat A je LOA bez zacyklenia, rozpoznávajúci $L = L_1 \cap L_2$, z toho vyplýva $L \in \mathcal{L}_{CS}$. Trieda $\mathcal{L}_{CS}$ je uzavretá vzhľadom na operáciu $\cap$.

Toto treba ukazovať cez automaty. (pripomienka: LOA [lineárne ohraničený automat] je v podstate to isté, ako turingáč, len nemá nekonečnú pásku)

Dvojpáskový LOA rozpoznávajúci prienik:



Na páske dostaneme vstupné slovo, na krajoch máme zarážky.

Prejde binárna operácia pre dvojicu jazykov. Máme kontextové jazyky L1 a L2, pre ktoré existuje automat:

$$L(A_1) = L_1 \text{ a } L(A_2) = L_2.$$

Znova tento obrázok, je dvojpáskový, tak na prvom „poschodí“ nám bežý jeden automat, na druhom druhý automat. Sledujeme teraz, či budú automaty akceptovať dané slovo. Oba automaty musia slovo akceptovať (A), inak dostaneme N.

Akceptovanie:

A_1,LOA$_1$	A	N	A	N
A_2, LOA$_2$	A	A	N	N
A, LOA	**A**	**N**	**N**	**N**

Pokiaľ nám slovo akceptujú oba automaty, tak je jazyk uzavretý na prienik.

Uzáverové vlastnosti kontextových jazykov

Veta

$\mathcal{L}_{CS}$ je uzavretá na doplnok.

Nech $L \in \mathcal{L}_{CS}$, potom musí existovať LOA A bez zacyklenia taký, že $L(A) = L$.

Konštrukcia:
Immerman – Szelepcenyiho veta, 2. LOA problém
Idea – strom konfigurácií výpočtu – bez nekonečných vetiev – výmena akceptujúcich stavov za neakceptujúce

Overenie: A' je LOA bez zacyklenia, rozpoznávajúci L^C, z toho vyplýva $L^C \in \mathcal{L}_{CS}$. Trieda $\mathcal{L}_{CS}$ je uzavretá vzhľadom na komplement.

Toto je že vraj veľmi komplikovaný dôkaz, tak nešiel do detailu. Jednoducho pokiaľ LOA A' rozpozná doplnok jazyka L, tak je to uzavreté na doplnok/komplement.

Uzáverové vlastnosti kontextových jazykov

Veta

$\mathcal{L}_{CS}$ ***nie*** *je uzavretá na homomorfizmus*[a].

[a] aj vymazávací

Kontrapríklad:

G_{cs}		*po aplikovaní homomorfizmu*	
aB -> ab	h(a)=a	aB -> a	frázové pravidlo
bB -> bb	h(b)=ε	B -> ε	regulárne pravidlo
bD-> bd	h(d)=c	D-> c	regulárne pravidlo

Zdroj: Daniela Chudá: Teoretické základy informatických vied, FIIT STU, 2020.

Tu iba jednoducho aplikuješ homomorfizmus na tie pravidlá vpravo

h(a)=a

h(b)=ε

h(d)=c

Keď tento homom. aplikuješ napr. na to prvé pravidlo, dostaneš aB->a.

po aplikovaní homomorfizmu

aB -> a	frázové pravidlo
B -> ε	regulárne pravidlo
D-> c	regulárne pravidlo

Keď ti vznikne niečo, čo nie je kontextové pravidlo, máš kontrapríklad. Môžeš si určiť akýkoľvek homomorfizmus, ale musí ti vyjsť pravidlo, ktoré nie je kontextové.

Uzáverové vlastnosti rekurzívne vyčísliteľných jazykov

Veta

Trieda rekurzívne vyčísliteľných jazykov $\mathcal{L}_{RE}$ je uzavretá vzhľadom na operácie:

- ***prieniku***
- ***zjednotenia***
- ***zreťazenia***
- *iterácie*
- *reverz*
- *homomorfizmus (aj vymazávajúci)*

Veta

Trieda rekurzívne vyčísliteľných jazykov $\mathcal{L}_{RE}$ **nie** *je uzavretá na komplement.*

Iba prvé tri sme si ukázali, tie posledné tri sme si hovorili iba že existujú.

Veta

Trieda rekurzívne vyčísliteľných jazykov $\mathcal{L}_{RE}$ **nie** *je uzavretá na komplement.*

Toto iba je že neexistuje algoritmus ktorým môžeme toto dokázať.

Uzáverové vlastnosti rekurzívne vyčísliteľných jazykov

Veta

$\mathcal{L}_{RE}$ je uzavretá na prienik.

Nech $L_1, L_2 \in \mathcal{L}_{RE}$, potom musia existovať TS (môžu sa cykliť) A_1, A_2 také, že $L(A_1) = L_1$ a $L(A_2) = L_2$.

Konštrukcia:

Dvojpáskový TS rozpoznávajúci prienik:

...	w	...
...	w	...

Akceptovanie:

A_1,TS_1	A	A	A	N	N	N	C	C	C
A_2, TS_2	A	N	C	A	N	C	A	N	C
A, TS	**A**	**N**	**C**	**N**	**N**	**C**	**C**	**C**	**C**

Overenie: Nový automat A je TS rozpoznávajúci $L_1 \cap L_2$. Trieda $\mathcal{L}_{RE}$ je uzavretá vzhľadom na operáciu $\cap$.

Existujú dva TS, ktoré modelujú dané LRE.

Nech $L_1, L_2 \in \mathcal{L}_{RE}$, potom musia existovať TS (môžu sa cykliť) A_1, A_2 také, že $L(A_1) = L_1$ a $L(A_2) = L_2$.

Dvojpásková stroj, prvé poschodie rozpoznáva prvé slovo, druhé poschodie druhé slovo. Sledujeme, či stroj akceptuje slovo, alebo či sa zacyklí.

Dvojpáskový TS rozpoznávajúci prienik:

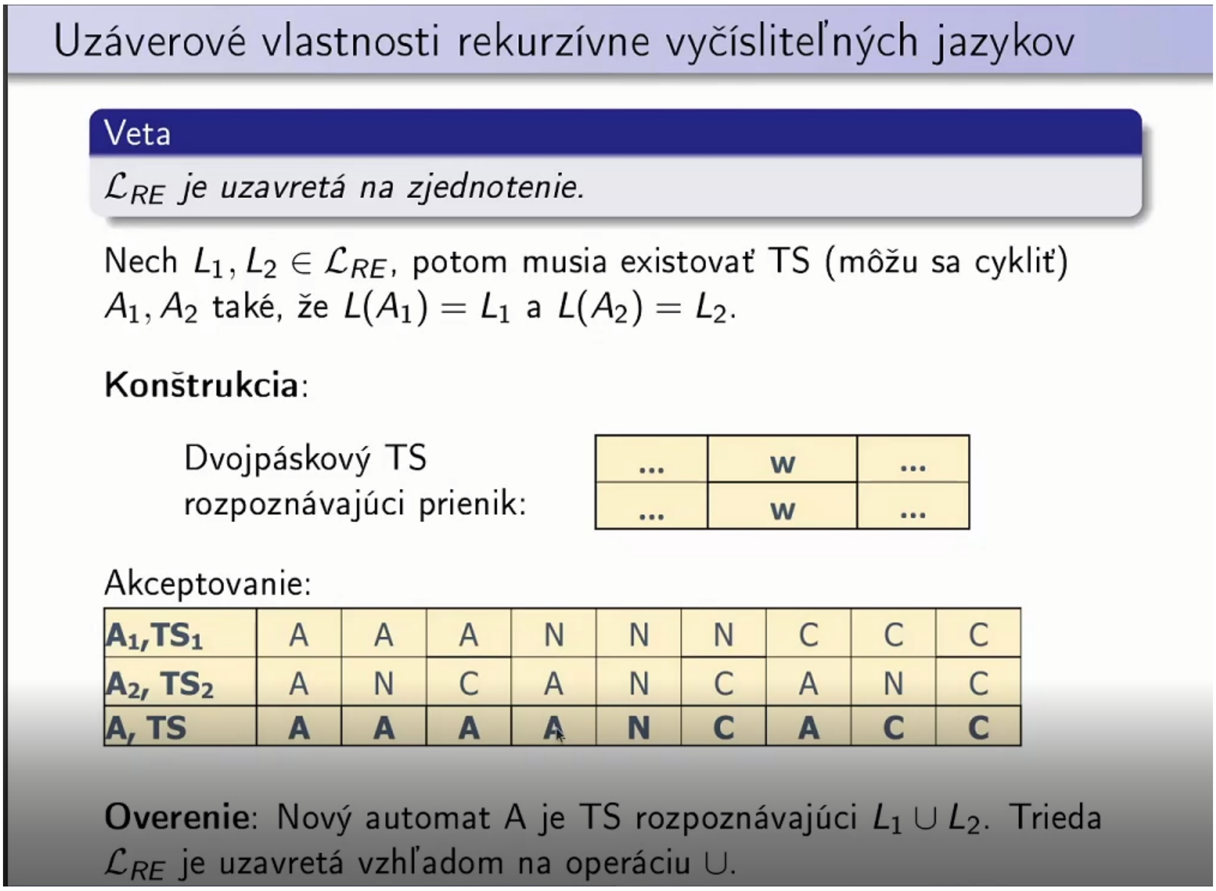
...	w	...
...	w	...

Akceptovanie:

A_1,TS_1	A	A	A	N	N	N	C	C	C
A_2, TS_2	A	N	C	A	N	C	A	N	C
A, TS	**A**	**N**	**C**	**N**	**N**	**C**	**C**	**C**	**C**

Musíme mať obe akceptované aby sme dostali A, inak dostaneme N alebo C (cyklí sa)

Pokiaľ automat rozpozná slovo, máme uzavreté na prienik.



Uzáverové vlastnosti rekurzívne vyčísliteľných jazykov

Veta

$\mathcal{L}_{RE}$ je uzavretá na zjednotenie.

Nech $L_1, L_2 \in \mathcal{L}_{RE}$, potom musia existovať TS (môžu sa cykliť) A_1, A_2 také, že $L(A_1) = L_1$ a $L(A_2) = L_2$.

Konštrukcia:

Dvojpáskový TS rozpoznávajúci prienik:

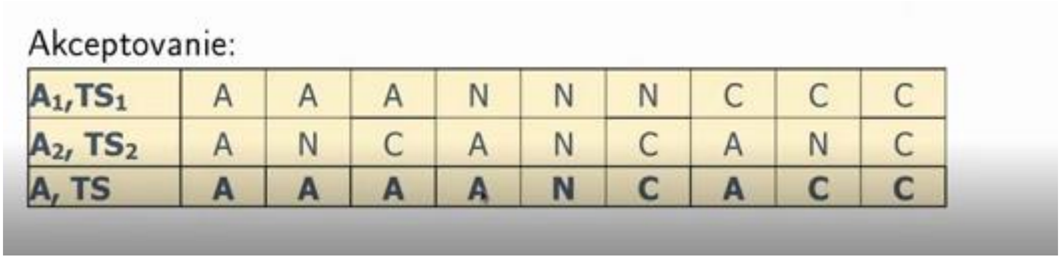
...	w	...
...	w	...

Akceptovanie:

A_1,TS_1	A	A	A	N	N	N	C	C	C
A_2, TS_2	A	N	C	A	N	C	A	N	C
A, TS	**A**	**A**	**A**	**A**	**N**	**C**	**A**	**C**	**C**

Overenie: Nový automat A je TS rozpoznávajúci $L_1 \cup L_2$. Trieda $\mathcal{L}_{RE}$ je uzavretá vzhľadom na operáciu $\cup$.

Pre zjednotenie je to to isté, ako pre prienik, ale trochu inak akceptujeme slová.



Akceptovanie:

A_1,TS_1	A	A	A	N	N	N	C	C	C
A_2, TS_2	A	N	C	A	N	C	A	N	C
A, TS	**A**	**A**	**A**	**A**	**N**	**C**	**A**	**C**	**C**

Stačí nám jedlo slovo akceptované aby sme dostali A. N dostaneme iba keď máme dve neakceptované slová. Keď máme C/N, N/C alebo C/C tak sa automat zacyklí.

Keď automat nám akceptuje slovo, máme uzavreté na zjednotenie.

Uzáverové vlastnosti rekurzívne vyčísliteľných jazykov

Veta

$\mathcal{L}_{RE}$ je uzavretá na zreťazenie.

Nech $L_1, L_2 \in \mathcal{L}_{RE}$, potom musia existovať TS (môžu sa cykliť) A_1, A_2 také, že $L(A_1) = L_1$ a $L(A_2) = L_2$.

Konštrukcia:
TS rozpoznávajúci zreťazenie bude jednopáskový.
Nedeterministicky rozhodnem, ktorá časť slova patrí L_1 a ktorá L_2, postupne simulujem A_1, A_2.
Akceptovanie: Akceptujem, len keď A_1 akceptuje a následne A_2 akceptuje.

Overenie: Nový automat A je TS rozpoznávajúci $L_1 \cdot L_2$. Trieda $\mathcal{L}_{RE}$ je uzavretá vzhľadom na operáciu $\cdot$.

Veta

$\mathcal{L}_{RE}$ je uzavretá na zreťazenie.

Nech $L_1, L_2 \in \mathcal{L}_{RE}$, potom musia existovať TS (môžu sa cykliť) A_1, A_2 také, že $L(A_1) = L_1$ a $L(A_2) = L_2$.

TS ktorý nám modeluje dané jazyky (už by si mal vedieť 😉)

TS rozpoznávajúci zreťazenie bude jednopáskový.

Už stačí iba jednopáskový (halleluyah)

Nedeterministicky rozhodnem, ktorá časť slova patrí L_1 a ktorá L_2, postupne simulujem A_1, A_2.

Doslova si tipneš tipos, ktorá časť je z ktorého jazyka a na tej časti simuluješ príslušný TS.

Akceptovanie: Akceptujem, len keď A_1 akceptuje a následne A_2 akceptuje.

Easy, pokiaľ A1 a A2 sú akceptované, tak je to uzavreté na zreťazenie

Uzáverové vlastnosti tried jazykov z Chomského hierarchie

Trieda jazykov	$\cup$	$\cdot$	$\cap$	$*$	$+$	c	R	h	h bez ϵ
$\mathcal{R}$	A	A	A	A	A	A	A	A	A
$\mathcal{L}_{CF}$	A	A	N	A	A	N	A	A	A
$\mathcal{L}_{CS}$	A	A	A		A	A	A	N	A
$\mathcal{L}_{RE}$	A	A	A	A	A	N	A	A	A

Zdroj: Daniela Chudá:Teoretické základy informatických vied, FIIT STU, 2020.

Sprava doľava:

Zjednotenie – zreťazenie – prienik – iterácia – kladná iterácia – komplement – reverz – homomorfizmus – nevymazávajúci homomorfizmus.

Táto tabuľka ti v podstate hovorí, že na ktoré operácie sú daná jazyky uzatvorené. Napr. LCF (bezkontextové) nie sú uzatvorené na prienik a komplement (máš tam N).

LCS (kontextové): nie sú uzavreté na homomorfizmus

LRE (rekurzívne): nie sú uzavreté na komplement.

Regulárne sú OK na všetko.