Na minulém cvičení jsme si ukázali, že nedeterministické konečné automaty (NFA) můžou být exponenciálně úspornější než deterministické (DFA), a to na jazycích $L_n = \{w \in \{a,b\}^* :$ n-tý znak slova w od konce je $a\}$, kde $n \in \mathbb{N}^+$. Připomeňme si úvahu, která k tomu vedla (pro názornost nechť $n = 5$): Uvážíme-li např. slova $u = aaaba$ a $v = aabba$, pak přiřetězením slova $w = ab$ dostaneme $uw = aaabaab \in L_5$, ale $vw = aabbaab \notin L_5$. Tudíž DFA rozhodující jazyk L_5 musí po přečtení slova uw skončit v akceptujícím stavu, zatímco po přečtení vw musí skončit v neakceptujícím stavu; zejména po uw musí skončit v jiném stavu než po vw, a proto i po u musí skončit v jiném stavu než po v. Podobnou úvahu můžeme provést pro libovolná dvě slova délky 5, z čehož plyne, že dotyčný DFA musí mít aspoň 2^5 stavů.

Ještě jednou si zopakujme to podstatné: Jak jsme vlastně dosvědčili, že u a v nemůžou skončit ve stejném stavu? Existovalo totiž slovo w, které bylo „pravým doplňovačem do jazyka L“ pro slovo u, ale ne pro slovo v. Řečeno obměněně: pokud u i v skončí ve stejném stavu, pak musí mít stejnou „množinu pravých doplňovačů do jazyka L“. Této množině se odborně říká *pravý kontext (daného slova) vzhledem k jazyku L*; formálně tedy za pravý kontext považujeme zobrazení $PK_L\colon \Sigma^* \to \mathcal{P}(\Sigma^*)$ definované předpisem $PK_L(u) = \{w \in \Sigma^* : uw \in L\}$. Jádro tohoto zobrazení (tj. relace ekvivalence na Σ^*, kde spolu kamarádí ta slova, která mají stejný pravý kontext), se nazývá *prefixová ekvivalence jazyka L* a standardně se značí $\sim_L$. Naše pozorování tedy nyní můžeme formulovat tak, že pokud DFA pro jazyk L skončí po přečtení u i v ve stejném stavu, pak určitě platí $u \sim_L v$.

Tato úvaha nám hned dává další možnost (vedle pumping lemmatu), jak o nějakém jazyku L dokázat, že není regulární — stačí najít nekonečně mnoho slov, z nichž žádná dvě nejsou vzhledem k L prefixově ekvivalentní. V DFA pro L by pak tato slova musela skončit v navzájem různých stavech, což není možné, protože DFA smí mít jen konečně mnoho stavů.

To, co jsme nyní odvodili, je jednou (a tou nejčastěji využívanou) z implikací tzv. *Myhillovy-Nerodovy věty*. Důkaz neregularity např. jazyka $L = \{w \in \Sigma^* : \#_a(w) = \#_b(w)\}$ je na necelé dva řádky:

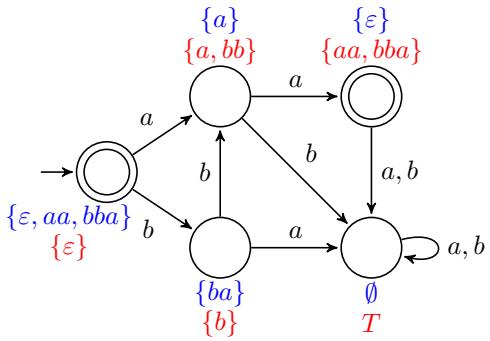
Uvažme množinu $M = \{a^n \mid n \in \mathbb{N}\}$. Pro lib. různá $i, j \in \mathbb{N}$ platí $a^i b^i \in L$, ale $a^j b^i \notin L$, tedy $a^i \not\sim_L a^j$. Proto $|\Sigma^*/{\sim_L}| \geq |M| = \aleph_0$, a tedy dle MN-věty L není regulární.

(Pro jistotu to vysvětlím ještě krok pro kroku: Nejprve vymyslíme těch nekonečně mnoho slov, která nejsou prefixově ekvivalentní. Dokážeme, že tomu tak skutečně je — zde slovo b^i je „pravým doplňovačem do L“ pro slovo a^i, ale ne pro slovo a^j (pro $j \neq i$). Jelikož žádná dvě slova z M nejsou prefixově ekvivalentní, pak tříd rozkladu Σ^* dle $\sim_L$ je aspoň tolik, kolik je prvků M, tedy (spočetně) nekonečně mnoho (tomu počtu tříd rozkladu se říká *index* dotyčné relace ekvivalence). No a teď už stačí se jen odvolat na MN-větu.)

Nyní se zkusme zabývat obrácenou otázkou: když máme jazyk L, jehož prefixová ekvivalence $\sim_L$ má konečný index, zvládneme z ní zkonstruovat DFA pro L? Zkusme to pro názornost třeba na jazyku $L = \{\varepsilon, aa, bba\}$. Nejprve napočítáme pravé kontexty jednotlivých slov:

u	$PK_L(u)$
ε	$\{\varepsilon, aa, bba\}$
a	$\{a\}$
b	$\{ba\}$
aa	$\{\varepsilon\}$
ab	$\emptyset$
ba	$\emptyset$
bb	$\{a\}$
bba	$\{\varepsilon\}$
všechna ostatní slova délky ≥ 3	$\emptyset$

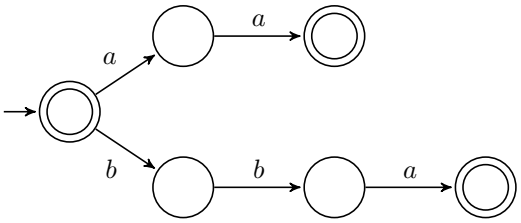
Platí tedy $\Sigma^*/{\sim_L} = \{\{\varepsilon\}, \{a, bb\}, \{b\}, \{aa, bba\}, T\}$, kde T obsahuje všechna doposud nevyjmenovaná slova, tj. ab, ba a všechna slova délky ≥ 3 až na bba. Víme, že slova s různými pravými kontexty musí skončit v různých stavech, ale zatím nemáme důvod se domnívat, že by slova se stejným pravým kontextem nemohla skončit ve stejném stavu — zkusme tedy zvolit stavy automatu tak, aby odpovídaly přesně jednotlivým pravým kontextům, tj. jednotlivým třídám rozkladu (a tedy i přesně tomu, která slova v dotyčném stavu skončí). Jak zvolíme počáteční stav? V něm bychom měli skončit po přečtení prázdného slova, bude to tedy třída obsahující ε. Které stavy prohlásíme za akceptující? Podle toho, zda příslušný pravý kontext obsahuje ε (tj. zda příslušná třída obsahuje slova ležící v L, nebo mimo L). A jak povedeme hrany? Ze třídy obsahující slovo $u \in \Sigma^*$ pod písmenem $c \in \Sigma$ do třídy obsahující slovo uc. Dostaneme tak následující automat (u každého stavu je modře napsán příslušný pravý kontext a červeně množina slov, která v daném stavu skončí, tj. příslušná třída rozkladu $\Sigma^*/{\sim_L}$):



Samozřejmě se nyní nabízí otázka: A máme jistotu, že je ta konstrukce hran korektní? Co kdyby v nějaké třídě ležela slova u, v taková, že uc by leželo v jiné třídě než vc? Pro názornost si můžeme představit třeba relaci ekvivalence $\sim$ na $\{a\}^*$ definovanou takto: pro lib. $u, v \in \{a\}^*$ platí $u \sim v$, právě když $3 \,|\, |u| \Leftrightarrow 3 \,|\, |v|$. Platí tedy $\{a\}^*/{\sim} = \{\{\varepsilon, a^3, a^6, a^9, \dots\}, \{a, a^2, a^4, a^5, a^7, a^8, \dots\}\}$. Kdybychom z této relace ekvivalence chtěli vytvořit DFA, pak nastane problém: Z první třídy rozkladu (stavu automatu) povede hrana zcela jistě do druhé třídy, ale odpověď na to, kam má vést hrana z druhé třídy, nyní není jednoznačná: pokud si jako reprezentanta z této třídy vytáhneme slovo a, pak bychom chtěli vést hranu opět do této třídy (neboť tam leží a^2), ale pokud si vezmeme jako reprezentanta a^2, pak vidíme, že bychom hranu měli vést do první třídy (neboť tam leží a^3).

Uvedená transformace z relace ekvivalence na automat tedy nefunguje vždy — vidíme, že aby fungovala, je potřeba, aby platilo, že kdykoli spolu kamarádí slova u, v, pak spolu kamarádí i uc a vc pro lib. $c \in \Sigma$ (a v důsledku toho i uw s vw pro lib. $w \in \Sigma^*$). Relace ekvivalence, která splňuje tuto vlastnost, se nazývá *pravá kongruence*. (Pokud jste doposud slovo „kongruence“ znali jako synonymum pro ekvivalenci na celých čísel podle zbytkových tříd, pak věžte, že to není náhoda. Z jistého algebraického pohledu se totiž jedná o tentýž koncept, ale o tom vám povykládám až osobně…) Můžete si rozmyslet (popř. nahlédnout do skript), že prefixová ekvivalence libovolného jazyka je vždycky pravou kongruencí, takže pro ni tato konstrukce fungovat bude.

Už na začátku jsme si rozmysleli, že pokud má $\sim_L$ nekonečný index, pak L není regulární, a nyní jsme si ukázali, že pokud má $\sim_L$ konečný index, pak ji umíme „přeložit“ na DFA rozhodující L, tedy L je regulární. (Koneckonců ta konstrukce by fungovala úplně stejně i pro neregulární L, akorát bychom dostali deterministický **ne**konečný automat, což z informatického hlediska není příliš užitečné, ale z algebraického hlediska je to rozhodně zajímavé.) Kdybychom nyní zapomněli na původní $\sim_L$, uměli bychom ji nyní vyčíst z obdrženého DFA? Ano, jednoduše — kamarádí spolu ta slova, která skončí ve stejném stavu. Takto samozřejmě můžeme z libovolného DFA $\mathcal{A}$ vytvořit jeho „stavovou ekvivalenci“ $\sim_{\mathcal{A}}$ na Σ^*. Kdybychom měli pro výše uvažovaný jazyk L najít nějaký DFA, nejspíš bychom nakreslili tento:



Tento automat není totální, takže vyvstává otázka, co se slovy, která se nedočtou do konce. Jelikož to z algebracikého hlediska není příliš pěkné, tak se ve skriptech v této pasáži uvažují jen totální automaty (pozor na tuto past!). Po doplnění o „odpadkový“ stav tedy dotyčná stavová ekvivalence tohoto automatu vypadá takto: slova $\varepsilon, a, aa, b, bb, bba$ jsou každá samostatně ve své třídě a všechna zbylá slova tvoří jednu třídu. Je zřejmé, že stavová ekvivalence každého automatu je pravou kongruencí (pokud slova u, v skončí ve stejném stavu, tak i slova uc, vc skončí ve stejném stavu).

Zatím jsme si tedy ukázali, jak z prefixové ekvivalence (nějakého) jazyka L vyrobit automat a jak z (libovolného) automatu vyrobit pravou kongruenci, přičemž tyto dvě konstrukce jsou (v tomto směru) vzájemně inverzní, tj. pokud z $\sim_L$ vyrobíme automat a z něj pak pravou kongruenci, pak to bude přesně původní $\sim_L$. Také už jsme si rozmysleli, že ta první konstrukce funguje díky tomu, že prefixová ekvivalence libovolného jazyka je pravou kongruencí. Nyní se nabízí otázka: Nestačilo by pro tuto konstrukci začít s libovolnou pravou kongruencí (která nemusí být nutně prefixovou ekvivalencí nějakého jazyka)? Stačilo, až na jeden problém. Pravá kongruence je totiž jen (dostatečně pěkná) relace ekvivalence na slovech; není vztažená k žádnému konkrétnímu jazyku, takže z ní nelze vyčíst informaci o tom, které třídy rozkladu jakožto stavy výsledného automatu by

měly být akceptující, nicméně pomineme-li akceptující stavy, pak konstrukce automatu funguje pro libovolnou pravou kongruenci (přičemž pokud chceme dostat konečný automat, pak se samozřejmě bavíme jen o pravých kongruencích s konečným indexem).

Platí tedy, že konstrukce „pravá kongruence → automat“ a „automat → pravá kongruence“ jsou při uvedených omezeních vzájemně inverzní? Skoro. Ještě je tu jedna past, a to nedosažitelné stavy — ty totiž v automatu být můžou, ale ve výsledné pravé kongruenci se nijak neprojeví. Můžeme tedy shrnout, že pravé kongruence vzájemně jednoznačně odpovídají automatům bez označení akceptujících stavů, s totální přechodovou funkcí a bez nedosažitelných stavů.

A k čemu nám toto pozorování vlastně je? Díky prefixovým ekvivalencím jsme odvodili dobře použitelnou nutnou a postačující podmínku pro regulariu jazyka (MN-větu). A proč jsme se bavili o obecných pravých kongruencích? Tyto úvahy nám totiž ukážou, že struktura automatů rozhodujících daný jazyk je velmi specifická. Pro každý jazyk L totiž existuje „nejhezčí“ automat (odpovídající prefixové ekvivalenci pro L, a odborně zvaný *minimální*, i když pojem *nejmenší* by byl přesnější) a všechny ostatní automaty pro L jsou jen „zesložitěním“ toho nejhezčího. MN-větu lze tedy také vnímat tak, že jazyk je regulární, právě když ten nejhezčí automat pro něj má konečně mnoho stavů.

Proč tomu tak je? Nejprve si všimněme, že máme-li automat $\mathcal{A}$ pro jazyk L, pak stavová ekvivalence $\sim_{\mathcal{A}}$ i prefixová ekvivalence $\sim_L$ nejsou jen tak ledajaké pravé kongruence, ale mají jistý vztah k jazyku L: totiž z každé třídy rozkladu leží v L buď všechna slova, nebo žádné. (Této vlastnosti se odborně říká, že $\sim$ *respektuje* L, nebo také že $\sim$ *nasycuje* L; ve skriptech je pro totéž uváděna poněkud krkolomná formulace, že „L je sjednocením některých tříd rozkladu Σ^* podle $\sim$“.) U $\sim_{\mathcal{A}}$ je to podle toho, zda je příslušný stav akceptující; u $\sim_L$ je to podle toho, zda v příslušném pravém kontextu leží ε. Můžete si rozmyslet, že pokud u, v nejsou prefixově ekvivalentní podle L, tak spolu nemůžou kamarádit ani v žádné pravé kongruenci respektující L. Řečeno obměněně: pokud u, v spolu kamarádí v nějaké pravé kongruenci respektující L, pak spolu kamarádí i ve $\sim_L$. To tedy znamená, že $\sim_L$ není jen tak ledajaká pravá kongruence respektující L, ale je to dokonce (vzhledem k inkluzi) největší pravá kongruence respektující L. Jinými slovy: podíváme-li se na příslušné rozklady, pak každá pravá kongruence respektující L může vypadat jedině tak, že se v $\sim_L$ „roztrhnou“ některé třídy rozkladu (vizte výše zkoumaný jazyk $L = \{\varepsilon, aa, bba\}$). Pro automaty to tedy znamená, že každý automat pro L může vypadat jedině tak, že se v tom nejhezčím automatu pro L (vzniklém z $\sim_L$) „zbytečně“ rozdělí některé stavy (a popřípadě „pro radost“ doplní ještě nedosažitelné stavy).

Minimální automat má tedy zejména nejmenší možný počet stavů. (Nicméně mějme na zřeteli past — netotální přechodové funkce! Pokud je uvažujeme, pak se i z minimálního automatu, vzniklého z $\sim_L$ — pokud existují slova s pravým kontextem $\emptyset$ vzhledem k L — dá odpovídající „odpadkový“ stav vynechat.) Jeho postavení je ale ještě význačnější. Precizně řečeno: minimální automat pro L má tu vlastnost, že do něj existuje parciální surjektivní úplný homomorfizmus z libovolného automatu pro L, ale to vám vysvětlím až někdy příště… :-)