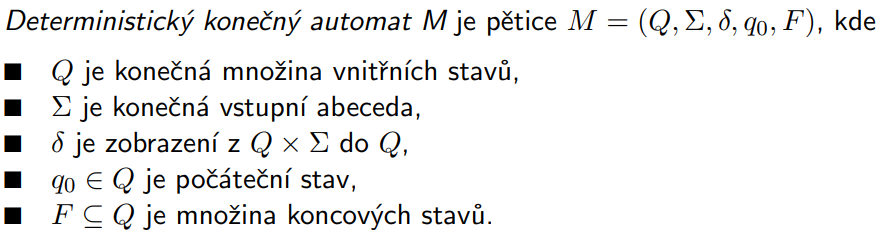
**BI-SPOL-2 Regulární jazyky: Deterministické a nedeterministické konečné automaty. Determinizace konečného automatu. Minimalizace deterministického konečného automatu. Operace s konečnými automaty. Regulární gramatiky, regulární výrazy, regulární rovnice**

BI-AAG

**Regulární jazyky**

* lze popsat regulární gramatikou, generovat deterministickým konečným automatem a popsat regulárním výrazem
* konečný jazyk je vždy regulární

**Deterministický konečný automat**



* **konfigurace**: (q, w) ∈ Q × Σ∗; počáteční (q0, w); koncová (q, ε)
* **přechod**: ⊢M je relace nad Q × Σ∗ taková, že *(q, w) ⊢M (p, w′)* právě tehdy, když *w = aw′* a *δ(q, a) = p* pro nějaké a ∈ Σ, w ∈ Σ∗. Prvek relace ⊢M nazveme přechodem.
* **přijímaný jazyk**: řetězec je přijat, jestli existuje nějaká posloupnost přechodů *⊢∗M* vedoucí do koncové konfigurace
* je úplně určený, pokud má definované zobrazení δ pro všechny dvojice stavů

**Nedeterministický konečný automat**

### 

* konfigurace, přechod i přijímaný jazyk stejně jako DKA
* může mít více přechodů se stejným znakem v jednom stavu
* NKA a DKA jsou ekvivalentní, pokud přijímají stejný jazyk

**Dosažitelný stav**

* pokud existuje posloupnost přechodů z počátečního stavu do daného stavu, je tento stav dosažitelný, jinak je nedosažitelný

**Užitečný stav**

* pokud existuje posloupnost přechodů z daného stavu do koncového stavu, je tento stav užitečný, jinak je zbytečný

**Determinizace KA**

* pro každý NKA existuje ekvivalentní DKA
* spojení více počátečních stavů do jednoho (např vstup A a D ⇒ nový poč. stav AD)
* vytváření nových stavů, jejichž přechody vzniknou sloučením vstupních symbolů (např. přechod 0 v A → B, D → C, takže nový stav BC a přechod AD → BC)

**Minimalizace DKA**

* (převod na DKA)
* odstranění nedosažitelných stavů
* odstranění zbytečných stavů
* redukce ekvivalentních stavů (prvně rozdělení na koncové a nekoncové stavy, sloučení ekvivalentních stavů a opakovat, dokud se to neustálí)

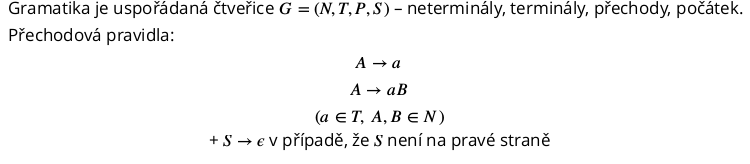
**Operace s KA**

* sjednocení: L(M) = L(M1) ∪ L(M2)
* průnik: L(M) = L(M1) ∩ L(M2)
* doplněk: vstup musí být úplně určený DKA; prohození koncových stavů

( L(M′) = Σ∗ \ L(M))

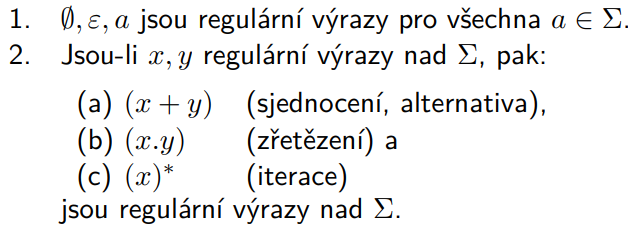
* součin (zřetězení): ke koncovému stavu M1 přidáme ε-přechod do počátečního stavu M2, L(M) = L(M1) . L(M2)
* iterace: přidá se nový počáteční stav s ε-přechodem do původního počátečního stavu (ten už není počáteční), L = L\*

**Regulární gramatiky**

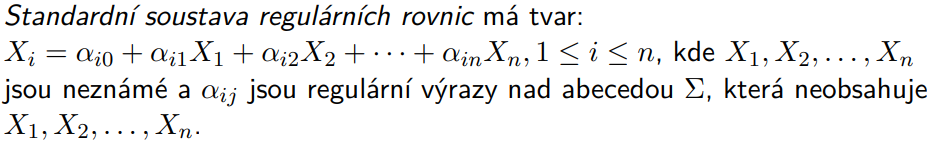


**Regulární výrazy**

* RV *V* nad abecedou Σ

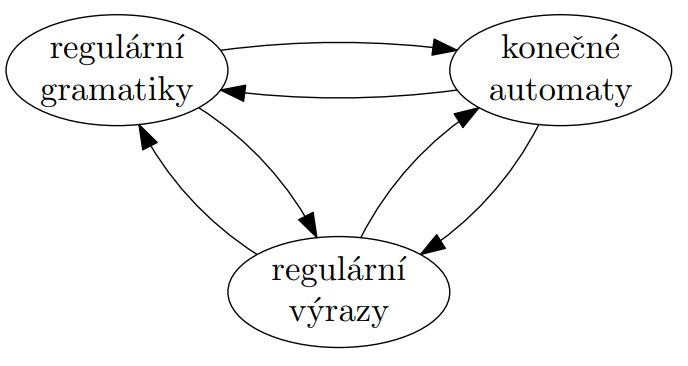


**Regulární rovnice**



* levá RR: x = xα + β , řešením je x = βα\* (α, β jsou známé)
* pravá RR: x = αx + β , řešením je x = α\*β (α, β jsou známé)

**Vztahy pro regulární jazyky**

****