<http://mj.ucw.cz/vyuka/ads2/>

haystack, needle

DF: ∑ alphabet (set of symbols)

∑⋆  words (strings)

α, β ∈ ∑ ⋆ x,y ∈ ∑

|α | length

ϵ empty string

α β , α ⋆ β concate

∀α αϵ = ϵα = α

α [i] i∈ {0,…,|α|-1}

α [I”j] α[i]α[i+1] …α[j-1]}

 α [:j] prefix

 α[i:] suffix

α[:] = α

α[i:j]=|epsilon j≤I

Input: haystack δ∈∑⋆ (S=|δ|)

Needle γ∈∑⋆(N=|γ|)

Ouput: list of positions

Observation: if α is the state fot σ and α` is the state for σx then either α` = ϵ or α` = α``x for α`` suffix of α

Precalc: for every prefix of γ: the longest proper suffix which is also a preix of ϵ

**Knuth, Marris, Pratt** algorithm

Search automation

KMP Step(s,x):

1. While there is no forward edge then go back while γ [s]≠x & s≠0: s<-B(s)
2. If we can take forwad edge, do it if γ[s]=x: s<-s+1
3. Return s

KMP Search(σ):

1. S <- 0
2. For lal x∈σ
3. S<-KMPStep(s,x)
4. If s=N: report an occurrence

<http://www.kiv.zcu.cz/~mautner/Pt/Zpracovani_retezcu_I.pdf> #page 40

bootstraping - operate itself

**Aho, Corasick**

hlední více jehel v jedné kupce sena

vytvoření písmenkového stromu – pomocí automatu

a poté podle KMP

ac search:

ac build:

vytvořit strom

udělat zpětné hrany

vytvořit zkratky

udělat frontu

pro každé *c* udělat zkratky a zpětné hrany

**Rabin–Karp algorithm**

**Ford-Fulkerson algorithm**