**Analýza rekurzivním sestupem** ([anglicky](https://cs.wikipedia.org/wiki/Angli%C4%8Dtina) *recursive descent parser*) je metoda [syntaktické analýzy](https://cs.wikipedia.org/wiki/Syntaktick%C3%A1_anal%C3%BDza) [shora dolů](https://cs.wikipedia.org/wiki/Syntaktick%C3%A1_anal%C3%BDza_shora_dol%C5%AF),

[**https://cs.wikipedia.org/wiki/LL\_syntaktick%C3%BD\_analyz%C3%A1tor**](https://cs.wikipedia.org/wiki/LL_syntaktick%C3%BD_analyz%C3%A1tor)

neterminály (proměnné), terminály (konstanty)

**terminální symbol** je znak vstupu, který není nijak přepisován.

**Neterminální symbol** je znak, který slouží k rozvinutí (a kontrole) pravidel gramatiky.

V programu se rekurzivně volají pravidla,

Rekurzivní sestup

<https://www.algoritmy.net/article/100/Konstrukce-prekladace>

Rekurzivní sestup je způsob, jakým dochází ke zpracování vstupu. Gramatika definuje množinu přepisovacích pravidel, která se skládá z *terminálních* a *neterminálních*symbolů. Terminální symbol je znak vstupu, který není nijak přepisován. Neterminální symbol je znak, který slouží k rozvinutí (a kontrole) pravidel gramatiky.

Na počátku zpracování obsahuje zásobník automatu jeden neterminální znak, říkejme mu E. Aby mohl automat zkontrolovat validitu vstupu, tak tento znak musí být schopen rozvinout na celý řetězec vstupu (tzn. na všechny odpovídající terminální symboly). V každém kroku se proto automat podívá na vrchol zásobníku, je-li tam neterminální symbol, tak jej správným způsobem rozvine (buď pomocí LL1 gramatiky nebo jiného orákula). Je-li na vrcholu zásobníku terminální symbol, který rozvinout nejde, tak se podívá na vstup. Pokud je na vstupu stejný symbol, tak oba symboly srovná (odstraní), v opačném případě havaruje. V okamžiku, kdy již na vstupu není žádný znak a zásobník je prázdný, je ověřena syntaktická správnost vstupu.

Příklad

Mějme gramatiku, která zpracovává jednoduchý aritmetický výraz, který se skládá z operací sčítání a násobení (s obvyklými prioritami), čísel a závorek. Terminálními symboly pro nás proto jsou znaky *číslo*, *+*, *\**, *(*, *)*.

Níže uvedená gramatika kontroluje správnost libovolného výrazu. Pro přehlednost není ([LL1](https://www.algoritmy.net/article/69/LL1-gramatika)), proto při přechodech musíme dát na svoji intuici.

1)\\; E \\rightarrow T + E  
2)\\; E \\rightarrow T   
3)\\; T \\rightarrow S * T   
4)\\; T \\rightarrow S  
5)\\; S \\rightarrow číslo  
6)\\; S \\rightarrow (E)

když je neterminál postupně se rozvijí

terminál jsou ty malá

když je terminál, bere další

Udělat faktorizace u 2 pravidel

U a

Začíná a

a’ =

D’=d

D’=

Err znamená, že při analýze došlo k chybě, a překladač by měl oznámit syntax-error.

Vlastně vytýkání

Rekurzivní sestup

vlastně

syntaktická analýza – vše je syntakticky správně

jedná se o LL(1) gramatiku?

**LL(1) gramatika**

LL1 gramatika je postavena tak, aby se v žádném okamžiku nedostala do situace, ve které by mohlo jít zpracování vstupu více cestami (tj. zpracování je **deterministické**).

Zásobníkový automat gramatiky je vždy schopen rozhodnout na základě **svého stavu** a **prvního nezpracovaného znaku**, do kterého dalšího stavu má přejít.

Každá pravá strana začíná terminálním symbolem

Pokud se levé strany shodují, potom každá pravá strana začíná jiným terminálním symbolem.