<https://cs.wikipedia.org/wiki/Transformace_na_LL(1)>

<https://docslide.com.br/documents/principy-prekladacu-56b2dd60ab199.html>

**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>

**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.

Transformace na LL(1)

**Transformace na LL(1)** gramatiku je postup jak gramatiku upravit na gramatiku LL(1). To je taková gramatika, která nemá konflikty v [rozkladové tabulce](https://cs.wikipedia.org/w/index.php?title=Rozkladov%C3%A1_tabulka&action=edit&redlink=1). Ne všechny gramatiky jdou převést na LL(1), takže aplikace této transformace nemusí vést k požadovanému výsledku.

**Obsah**

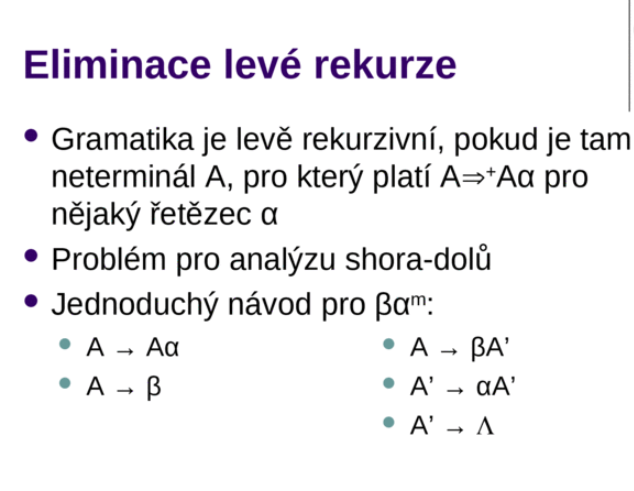
  [skrýt]

* [1Pravidla transformace](https://cs.wikipedia.org/wiki/Transformace_na_LL(1)#Pravidla_transformace)
  + [1.11. odstranění levé rekurze](https://cs.wikipedia.org/wiki/Transformace_na_LL(1)#1._odstran%C4%9Bn%C3%AD_lev%C3%A9_rekurze)
  + [1.22. levá faktorizace](https://cs.wikipedia.org/wiki/Transformace_na_LL(1)#2._lev%C3%A1_faktorizace)
  + [1.33. levá rohová substituce / extrakce pravého kontextu (dosazení)](https://cs.wikipedia.org/wiki/Transformace_na_LL(1)#3._lev%C3%A1_rohov%C3%A1_substituce_/_extrakce_prav%C3%A9ho_kontextu_(dosazen%C3%AD))
  + [1.44. Pohlcení Terminálu (řetězce)](https://cs.wikipedia.org/wiki/Transformace_na_LL(1)#4._Pohlcen%C3%AD_Termin%C3%A1lu_(%C5%99et%C4%9Bzce))

Pravidla transformace[[editovat](https://cs.wikipedia.org/w/index.php?title=Transformace_na_LL(1)&veaction=edit&section=1) | [editovat zdroj](https://cs.wikipedia.org/w/index.php?title=Transformace_na_LL(1)&action=edit&section=1)]

Pro transformaci se používají čtyři pravidla, každé pravidlo odstraňuje nějaký konflikt v [rozkladové tabulce](https://cs.wikipedia.org/w/index.php?title=Rozkladov%C3%A1_tabulka&action=edit&redlink=1)

1. **odstranění levé rekurze**



Levá rekurze se pozná tak, že pravidlo začínající Neterminálem má tento Neterminál na prvním místě na pravé straně pravidla.  
Mějme následující gramatická pravidla (Neterminály: {A,B} Terminály {a,b,c}) :

|  |  |  |
| --- | --- | --- |
| **pravidlo** | **first** | **poznámka** |
| A->AaB | b,c | zde je levá rekurze |
| A->b | b |  |
| A->c | c |  |
| B->c | c |  |
| B->ac | a |  |

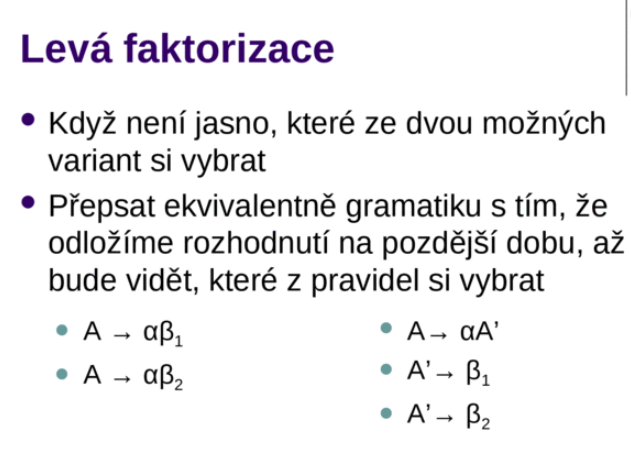
Postup:  
I. V pravidle, kde se vyskytuje levá rekurze vyjmu Neterminál na první pozici v pravé straně pravidla (v tomto případě A) a na konec pravidla dám nový Neterminál (v tomto případě A') a na začátek pravidla místo (A) dám (A')  
II. Ostatní pravidla začínající tímto Neterminálem (A) opíšu a nakonec jim přidám nový Neterminál (A')  
III. Doplním epsilon pravidlo pro nový Neterminál (A')

Upravená gramatická pravidla (Neterminály: {A,A',B} Terminály {a,b,c,epsilon}) :

|  |  |  |
| --- | --- | --- |
| **pravidlo** | **first** | **poznámka** |
| A'->aBA' | b,c | I. odebral se Neterminál A a nakonec se přidal Neterminál A' |
| A->bA' | b | II. přidal se neterminál A' |
| A->cA' | c | II. přidal se neterminál A' |
| B->c | c |  |
| B->ac | a |  |
| A'->epsilon | epsilon | III. nové epsilon pravidlo |



1. **levá faktorizace**



**Tímto** pravidlem se odstraňuje konflikt first-first  
Mějme následující gramatická pravidla (Neterminál: {A} Terminály {a,b,c}) :

|  |  |  |
| --- | --- | --- |
| **pravidlo** | **first** | **poznámka** |
| A->ab | a |  |
| A->ac | a |  |

Postup:  
Z konfliktních pravidel utvořím nové ponecháním konfliktního Terminálu (v tomto případě a) a **přidáním nového Neterminálu** (v tomto případě A')  
Pro nový Neterminál (A') vytvořím nová pravidla z původních konfliktních, tím že z nich odstraním konfliktní Terminál (a)

Upravená gramatická pravidla (Neterminály: {A,A'} Terminály {a,b,c}) :

|  |  |  |
| --- | --- | --- |
| **pravidlo** | **first** | **poznámka** |
| A->aA' | a | použil se konfliktní Terminál a přidal se Neterminál A' |
| A'->b | b | odstraněn konfliktní Terminál a |
| A'->c | c | odstraněn konfliktní Terminál a |

1. **levá rohová substituce / extrakce pravého kontextu (dosazení)**

Mějme následující gramatická pravidla (Neterminál: {A,B} Terminály {a,b,c,d}) :

|  |  |  |
| --- | --- | --- |
| **pravidlo** | **first** | **poznámka** |
| A->ab | a |  |
| A->Bc | a,c | zde se musí upravit |
| B->ab | a |  |
| B->cd | c |  |

Postup:  
V nevyhovujícím pravidle nahradíme Neterminál na pravé straně (v tomto případě B) pravou stranou pravidel pro pravidlo B (v tomto případě ab, cd).

Upravená gramatická pravidla (Neterminály: {A,B} Terminály {a,b,c,d}) :

|  |  |  |
| --- | --- | --- |
| **pravidlo** | **first** | **poznámka** |
| A->ab | a |  |
| A->abc | a | náhrada Neterminálu B pravou stranou z B->ab |
| A->cdc | c | náhrada Neterminálu B pravou stranou z B->cd |
| B->ab | a |  |
| B->cd | c |  |

Pro danou gramatiku jsou již pravidla pro B nedostupná a tudíž se mohou kompletně vynechat. Po aplikaci pravidla nám vznikl konflikt first-first a ten odstraníme aplikací 2. pravidla

Gramatická pravidla po odstranění first-first konfliktu (Neterminály: {A,A'} Terminály {a,b,c,d,epsilon}) :

|  |  |  |
| --- | --- | --- |
| **pravidlo** | **first** | **poznámka** |
| A->abA' | a | použili se konfliktní Terminály ab a přidal se Neterminál A' |
| A->cdc | c |  |
| A'->epsilon | epsilon | odstranění konfliktních Terminálů ab |
| A'->c | c | odstranění konfliktních Terminálů ab |

**4. Pohlcení Terminálu (řetězce)**

Odstraňuje first-follow konflikt

Mějme následující gramatická pravidla (Neterminál: {A,B} Terminály {a,c,d,epsilon}) :

|  |  |  |  |
| --- | --- | --- | --- |
| **pravidlo** | **first** | **follow** | **poznámka** |
| A->aBc | a |  | odtud se dostalo c do follow B |
| B->epsilon | epsilon | c |  |
| B->cd | c |  |  |

Postup:  
V pravidle odkud se dostal konfliktní Terminál (v tomto případě c) do follow sloučím s předcházejícím Neterminálem (v tomto případě B) tuto sloučeninu označím jako nový Neterminál (v tomto případě [Bc])  
Pro nový Neterminál vytvořím pravidla ze stávajících pravidel pro Neterminál (B) a na jejich konec přidám Terminál (c)

Upravená gramatická pravidla (Neterminály: {A,B,[Bc]} Terminály {a,c,d,epsilon}) :

|  |  |  |  |
| --- | --- | --- | --- |
| **pravidlo** | **first** | **follow** | **poznámka** |
| A->a[Bc] | a |  | sloučeni Neterminálu B s Terminálem c v nový Neterminál |
| B->epsilon | epsilon | epsilon |  |
| B->cd | c |  |  |
| [Bc]->c | c |  | pravidlo B->epsilon doplněné o Terminál c |
| [Bc]->cdc | c |  | pravidlo B->cd doplněné o Terminál c |

Pro danou gramatiku jsou již pravidla pro B nedostupná a tudíž se mohou kompletně vynechat. Po aplikaci pravidla nám vznikl konflikt first-first a ten odstraníme aplikací 2. pravidla

|  |  |  |  |
| --- | --- | --- | --- |
| **pravidlo** | **first** | **follow** | **poznámka** |
| A->a[Bc] | a |  |  |
| [Bc]->cB' | c |  | použil se konfliktní Terminál c přidal se Neterminál B' |
| B'->epsilon | epsilon | epsilon | odstranění konfliktního Terminálu c |
| B'->dc | d |  | odstranění konfliktního Terminálu c |

[Kategorie](https://cs.wikipedia.org/wiki/N%C3%A1pov%C4%9Bda:Kategorie):

* [Formální jazyky](https://cs.wikipedia.org/wiki/Kategorie:Form%C3%A1ln%C3%AD_jazyky)

Původní gramatika 1, problém – není jasno, kterou z více variant vybrat

S -> AB

A -> ah | abAc | aBc

B -> CDE

C -> d | e

D -> fD | f

E -> B | g

Upravená gramatika 1 pomocí levé faktorizace

S -> AB

A -> aA'

A' -> h | bAc | Bc

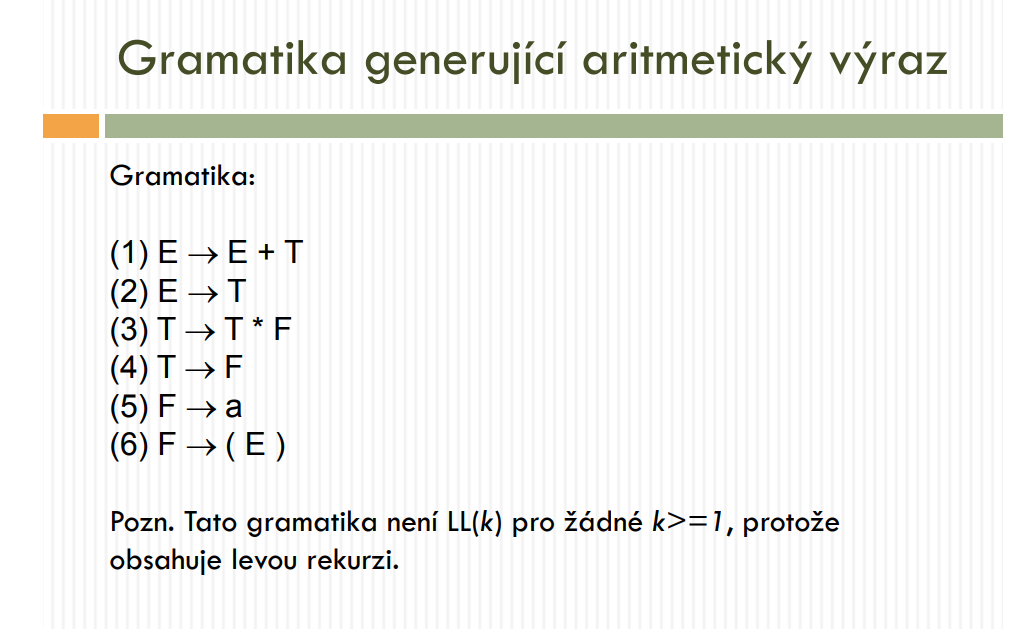
B -> CDE

C -> d| e

D -> fD'

D' -> fD' | epsilon

E -> B | g





https://edux.fit.cvut.cz/oppa/BI-PJP/prednasky/pjp4.pdf

Původní gramatika 2

S -> id = E

E -> E + E

E -> E \* E

E -> ( E )

E -> a

E -> b

Upravená gramatika 2/1 upravím – levá faktorizace

S -> id = E

E -> E + T

E -> T

T -> T \* F

T -> F

F -> (E)

F -> a

F -> b

Upravená gramatika 2/2 odstranění levé rekurze

S -> id = E

E -> TE’

E’ -> +TE’

E’-> e

T -> FT’

T’ -> \*FT’

T’ -> e

F -> ( E )

F ->a

F -> b