# Kapitola 6

# LL gramatiky

## 6.1 Definice LL(k) gramatik

**Definice 6.1.** Nechť $G = (N, \Sigma, P, S)$ je CFG, $k \geq 1$ je celé číslo. Definujme funkci $FIRST_k^G : (N \cup \Sigma)^+ \to \mathcal{P}(\{w \in \Sigma^* \mid |w| \leq k\})$ předpisem

$$FIRST_k^G(\alpha) = \{w \in \Sigma^* \mid (\alpha \Rightarrow^* w \wedge |w| \leq k) \vee (\alpha \Rightarrow^* wx \wedge |w| = k \wedge x \in \Sigma^*)\}$$

a funkci $FOLLOW_k^G : N \to \mathcal{P}(\{w \in \Sigma^* \mid |w| \leq k\})$ předpisem

$$FOLLOW_k^G(A) = \{w \in \Sigma^* \mid S \Rightarrow^* \gamma A \alpha,\ \ w \in FIRST_k^G(\alpha)\}.$$

Nechť dále $w = a_1 a_2 \dots a_n$ je libovolný řetěz. Pak klademe

$$k : w = \begin{cases} a_1 \dots a_k & k < n \\ w & k \geq n \end{cases}$$

**Poznámka 6.2.** Nechť relace $\Rightarrow_L$ značí levou derivaci, $\Rightarrow_L^*$ jako obvykle její transitivní a reflexivní uzávěr. Není těžké ukázat, že pokud bychom v definicích funkcí $FIRST$ a $FOLLOW$ použili $\Rightarrow_L^*$ namísto $\Rightarrow^*$, obdržíme tytéž množiny terminálních řetězů, tj. například platí: $FOLLOW_k^G(A) = \{w \in \Sigma^* \mid S \Rightarrow_L^* xA\alpha,\ \ w \in FIRST_k^G(\alpha)\}$. K tomu stačí indukcí ověřit následující dvě tvrzení (při obvyklém značení a $X, Y \in (N \cup \Sigma)^*$):

(1) $\{w \in \Sigma^* \mid \gamma \Rightarrow_L^n w\} = \{w \in \Sigma^* \mid \gamma \Rightarrow^n w\}$ a

(2) je-li $Y \Rightarrow^n \gamma X \beta \wedge \gamma \Rightarrow^* x \wedge \beta \Rightarrow^* y$, pak $Y \Rightarrow_L^n xX\alpha \wedge \alpha \Rightarrow^* y$ pro nějaké $\alpha$.

**Úmluva:** V dalším textu budeme i levé derivace značit symbolem $\Rightarrow$ resp. $\Rightarrow^*$, pokud nebude řečeno jinak.

**Definice 6.3.** Nechť $G = (N, \Sigma, P, S)$ je CFG, $k \geq 1$ je celé číslo. Řekneme, že $G$ je LL(k) gramatika, právě když pro libovolné dvě nejlevější derivace ($w \in \Sigma^*$)

$$(1)\quad S \Rightarrow^* wA\alpha \Rightarrow w\beta\alpha \Rightarrow^* wx \tag{6.1}$$

$$(2)\quad S \Rightarrow^* wA\alpha \Rightarrow w\gamma\alpha \Rightarrow^* wy \text{ , podmínka} \tag{6.2}$$

$$(3)\quad k : x = k : y \tag{6.3}$$

$$\text{implikuje rovnost } \beta = \gamma. \tag{6.4}$$

Řekneme, že gramatika $G$ je LL právě když je LL(k) pro nějaké $k \in \mathbb{N}$, jazyk $L$ je LL(k) právě když existuje LL(k) gramatika $G$ taková, že $L = L(G)$.

## 6.2 Vlastnosti LL gramatik

**Věta 6.4.** *Každá LL(k) gramatika je jednoznačná.*

**Důkaz:** Předpokládejme, že $G$ není jednoznačná. Pak existuje věta $u \in L(G)$, která má alespoň dvě různé levé derivace:

$$(1)\ S \Rightarrow^* wA\alpha \Rightarrow w\beta\alpha \Rightarrow^* wx = u$$
$$(2)\ S \Rightarrow^* wA\alpha \Rightarrow w\gamma\alpha \Rightarrow^* wy = u$$

kde $A \to \beta \mid \gamma$ jsou dvě různá pravidla. Pak $k : x = k : y =$, ale $\beta \neq \gamma$, a tedy $G$ není LL(k). □

**Věta 6.5.** *Je-li $G$ levorekurzivní, pak není LL(k) pro žádné $k$.*

**Důkaz:** Nechť $G = (N, \Sigma, P, S)$. Buď $A \in N$ levorekursivní neterminál, tj. $A \Rightarrow^* A\alpha$ pro nějaké $\alpha$. Jestliže $\alpha \Rightarrow^* \varepsilon$, pak $G$ není jednoznačná, a tedy ani LL(k).

Jestliže $\alpha \not\Rightarrow^* \varepsilon$, nechť $\alpha \Rightarrow^* v, v \in \Sigma^+$, a $A \Rightarrow^+ \beta \Rightarrow^* u$, kde $\beta \not\Rightarrow^* A\alpha$ (použití pravidla, které již nevede k levé rekurzi $A \Rightarrow^* A\alpha$). Pak existují levé derivace:

$$(1)\ S \Rightarrow^* wA\alpha' \Rightarrow^* wA\alpha^k\alpha' \Rightarrow^+ w\beta\alpha^k\alpha' \Rightarrow^* wuv^k\alpha'$$
$$(2)\ S \Rightarrow^* wA\alpha' \Rightarrow^* wA\alpha^k\alpha' \Rightarrow^+ wA\alpha^{k+1}\alpha' \Rightarrow^* wuv^{k+1}\alpha'\ ,$$

kde $k : uv^k = k : uv^{k+1}$. Současně však muselo být (viz kroky $\Rightarrow^+$) v jistém kroku derivace (1), konkrétně v části $A \Rightarrow^+ \beta$, použito nějakého pravidla $p_1$, které již nemůže vést na levou rekurzi; v odpovídajícím kroku derivace (2), konkrétně v části $A \Rightarrow^+ A\alpha$, ale bylo použito jiného pravidla $p_2$, které k levé rekuzi vede. Tedy $p_1 \neq p_2$ a $G$ není LL($k$). □

**Věta 6.6.** *Nechť $G = (N, \Sigma, P, S)$ je CFG. Pak $G$ je LL(k) právě když platí podmínka: Jsou-li $A \to \beta$ a $A \to \gamma$ dvě libovolná různá pravidla v $P$, pak pro všechny nejlevější větné formy $wA\alpha$ platí:*

$$FIRST_k(\beta\alpha) \cap FIRST_k(\gamma\alpha) = \emptyset.$$

**Důkaz:** Předpokládejme, že existují $w, A, \alpha, \beta, \gamma$ tak, jak uvedeno výše, ale $FIRST_k(\beta\alpha) \cap FIRST_k(\gamma\alpha) \neq \emptyset$. Nechť $x \in FIRST_k(\beta\alpha) \cap FIRST_k(\gamma\alpha)$. Odtud (a z předpokladu o $wA\alpha$ a z definice funkce *FIRST*) plyne existence dvou derivací

$$S \Rightarrow^* wA\alpha \Rightarrow w\beta\alpha \Rightarrow^* wxy$$
$$S \Rightarrow^* wA\alpha \Rightarrow w\gamma\alpha \Rightarrow^* wxz$$

a současně $k : xy = k : xz$, protože $x \in FIRST_k(\beta\alpha) \cap FIRST_k(\gamma\alpha)$ (je-li $|x| < k$, pak $y = \varepsilon = z$). Jelikož máme $\beta \neq \gamma$, pak $G$ není LL(k).

Naopak, předpokládejme, že $G$ není LL(k). To jest, existují dvě různé derivace

$$S \Rightarrow^* wA\alpha \Rightarrow w\beta\alpha \Rightarrow^* wx$$
$$S \Rightarrow^* wA\alpha \Rightarrow w\gamma\alpha \Rightarrow^* wy$$

takové, že $k : x = k : y$, ale $\beta \neq \gamma$. Tedy $A \to \beta$ a $A \to \gamma$ jsou dvě různá pravidla, ale množiny $FIRST_k(\beta\alpha)$ a $FIRST_k(\gamma\alpha)$ nejsou disjunktní – obě obsahují řetěz $k : x$. □

**Důsledek 6.7.** *Nechť $G = (N, \Sigma, P, S)$ je CFG bez $\varepsilon$-pravidel. Pak $G$ je LL(1) právě když $\forall A \in N$ a pro každá dvě různá $A$-pravidla $A \to \beta$, $A \to \gamma$ z $P$ platí*

$$FIRST_1(\beta) \cap FIRST_1(\gamma) = \emptyset.$$

**Důkaz:** Jelikož $\beta \not\Rightarrow^* \varepsilon$ a $\gamma \not\Rightarrow^* \varepsilon$, pak
$\forall \alpha.\ FIRST_1(\beta\alpha) \cap FIRST_1(\gamma\alpha) = FIRST_1(\beta) \cap FIRST_1(\gamma)$. □

**Věta 6.8.** *Nechť $G = (N, \Sigma, P, S)$ je CFG. $G$ je LL(1) gramatika právě když $\forall A \in N$ a pro každá dvě různá $A$-pravidla $A \to \beta$, $A \to \gamma$ z $P$ platí*

$$FIRST_1(\beta FOLLOW_1(A)) \cap FIRST_1(\gamma FOLLOW_1(A)) = \emptyset \tag{6.5}$$

**Důkaz:** Pro $\beta \neq \gamma$ provedeme rozbor po případech:

- je-li $\beta \Rightarrow^* \varepsilon$ a $\gamma \Rightarrow^* \varepsilon$, pak $G$ není jednoznačná. Tedy $G$ není LL($k$) a současně průnik (6.5) je neprázdný: je roven $FOLLOW_1(A)$.
- je-li $\beta \not\Rightarrow^* \varepsilon$ a $\gamma \not\Rightarrow^* \varepsilon$, pak tvrzení platí díky důsledku 6.7.
- je-li $\beta \not\Rightarrow^* \varepsilon$ a $\gamma \Rightarrow^* \varepsilon$, pak z věty 6.6 pro tento případ plyne, že $G$ není LL(1), právě když existuje levá větná forma $wA\alpha$ tavová, že $FIRST_1(\beta) \cap FIRST_1(\alpha) \neq \emptyset$, právě když $FIRST_1(\beta FOLLOW_1(A)) \cap FIRST_1(\gamma FOLLOW_1(A)) = \emptyset$, protože $FIRST_1(\alpha) \subseteq FOLLOW_1(A)$. Identicky pro případ $\beta \Rightarrow^* \varepsilon$ a $\gamma \not\Rightarrow^* \varepsilon$.

□

## 6.3 Syntaktická analýza LL(1) gramatik

Syntaktickou analýzu LL(k) gramatik lze provádět deterministickým zásobníkovým automatem automatem (DPDA). Požadujeme však, aby v případě, že vstupní slovo patří do jazyka, automat navíc poskytl informaci o struktuře věty (například její levé odvození, či derivační strom, resp. jednoznačné zakódování tohoto stromu). Proto automat rozšíříme o možnost zápisu výstupního symbolu na (přidanou) výstupní pásku Formální definici takového automatu s výstupem ponecháváme čtenáři.

Syntaktickou analýzu ukážeme nejprve pro LL(1) gramatiky, přičemž přímo vycházíme z tvrzení věty 6.8.

Nechť je dána LL(1) gramatika $G = (N, \Sigma, P, S)$, kde pravidla z $P$ jsou očíslována $i = 1, \ldots, card(P)$. Je-li $w \in L(G)$, pak *levým rozborem* $w$ nazveme posloupnost čísel pravidel použitých v levém odvození věty $w$.

DPDA $\mathcal{A}$ provádějící LL(1) syntaktickou analýzu vět z $L(G)$ má jeden stav, počáteční obsah zásobníku je $S\$$, kde $S$ je kořen gramatiky $G$ a \$ je symbol nevyskytující se v gramatice. Automat akceptuje prázdným zásobníkem. Označme $M$ přechodovou funkci [1] typu $M : (N \cup \Sigma \cup \{\$\}) \times (\Sigma \cup \varepsilon) \to \{<\alpha, i> \mid A \to \alpha \text{ je } i\text{-té pravidlo v } P\} \cup \{\text{odstraň, přijmi, chyba}\}$,
a je definována takto:

1. Protože automat má jen jeden stav, v definci přechodové funkce ho neuvádíme.

1. Je-li $A \to \alpha$ $i$-té pravidlo, klademe $M(A,a) = <\alpha, i>$ pro všechna $a \in$ ***FIRST***$_1(\alpha)$. Je-li též $\varepsilon \in$ ***FIRST***$_1(\alpha)$, pak $M(A,b) = <\alpha, i>$ pro všechna $b \in$ ***FOLLOW***$_1(A)$.
   V obou případech: je-li
2. $M(a,a) =$ odstraň, pro všechna $a \in \Sigma$ .
3. $M(\$, \varepsilon)$ = přijmi. Automat vymaže ze zásobníku symbol $ a akceptuje.
4. $M(x,a) =$ chyba, pro $x \in \Sigma, x \neq a$.

Uvedená přechodová funkce $M$ se též někdy nazývá $LL(1)$ *tabulkou* pro $G$, její část zkonstruovaná dle bodu 1 pak *redukovanou* $LL(1)$ tabulkou. Díky Větě 6.8 se snadno nahléne, že $M$ je přechodovou funkcí *deterministického* PDA (s výstupem), a to právě když $G$ je $LL(1)$; v opačném případě by $M(A,a)$ obsahovala dvě různé položky $<\alpha, i>$ a $<\beta, j>$ pro nějaká $A \in N, a \in \Sigma \cup \{\varepsilon\}$. Činnost automatu lze neformálně popsat takto:

1. Je-li na vrcholu zásobníku neterminál, řekněme $A$, pak automat má (v obou podpřípadech) udělat krok dle bodu 1 definice funkce $M$. Nechť první symbol ještě nezpracované části vstupu je $a$: automat provede $\varepsilon$-krok, nahradí $A$ na vrcholu zásobníku řetězem $\alpha$ a na výstup zapíše $i$, tj. číslo použitého pravidla $A \to \alpha$.
2. Je-li na vrcholu zásobníku terminál, řekněme $a$ a na vstupu je rovněž $a$, pak (tak jako u nedeterminininistické analýzy shora dolů) automat přečte ze vstupu $a$ a z vrcholu zásobníku $a$ odstraní.
3. Je-li na vrcholu zásobníku symbol $ (indikující „prázdný“ zásobník) a na vstupu je (již jen) $\varepsilon$ (indikující **eof** vstupního souboru), automat akceptuje (vymaže zásobník).
4. ve všech ostatních případech automat ukončí výpočet a neakceptuje.

Výše uvedené úvahy lze formalizovat takto: množinu konfigurací $K$ automatu $\mathcal{A}$ definujeme jako $(N \cup \Sigma \cup \{\$\})^* \times (\Sigma \cup \varepsilon)^* \times \{1, \ldots,$ ***card***$(P)\}^*$ reprezentující obsah zásobníku, dosud nepřečteno část vstupního slova a dosud vyprodukovaný výstup. Počáteční konfigurací pro vstupní slovo $w$ je $(S\$, w, \varepsilon)$. Na $K$ definujeme binární relaci (krok výpočtu) $\vdash$ takto:

1. $(ax, A\gamma, \pi) \vdash (ax, \alpha\gamma, \pi i)$ jestliže $M(A,a) = <\alpha, i>$.
2. $(ax, a\gamma, \pi) \vdash (x, \gamma, \pi)$ (pozn.: v tomto případě je $M(a,a) =$ odtraň).
3. $(\varepsilon, \$, \pi) \vdash (\varepsilon, \varepsilon, \pi)$ , kde $(\varepsilon, \varepsilon, \pi)$ je akceptující konfigurace a $\pi$ je levý rozbor $w \in L(G)$.
   Pro ostatní konfigurace není krok výpočtu definován. Případně je možné $K$ rozšířit o konfiguraci „chyba“ a definovat:
4. $(ax, X\gamma, \pi) \vdash$ chyba

Tvrzení obsažené v bodě 3 je třeba dokázat:

**Věta 6.9.** *Nechť $G = (N, \Sigma, P, S)$ je LL(1) gramatika, jejíž pravidla jsou očíslována $i = 1, \ldots,$ **card**$(P)$ a nechť $\mathcal{A}$ s přechodovou funkcí $M$ jsou takové, jak definováno výše. Pak platí: $(S\$, w, \varepsilon) \vdash^* (\varepsilon, \varepsilon, \pi) \iff w \in L(G)$ a $\pi$ je levý rozbor $w$.*

**Důkaz:** Idea důkazu: nechť $\mathcal{N}$ je PDA provádějící nedeterministickou syntaktickou analýzu vět z $L(G)$ zkonstruovaný dle lemmatu o nedeterministické syntaktické analýze shora dolů. Lze ověřit, že každý úspěšný výpočet automatu $\mathcal{N}$ lze simulovat výpočtem v $\mathcal{A}$ a též i obráceně, že ke každému akceptujícímu výpočtu v $\mathcal{A}$ existuje úspěšný výpočet automatu $\mathcal{N}$ (po případech dle definice přechodových funkcí automatů $\mathcal{N}$ a $\mathcal{A}$). Jelikož nahrazování neterminálů na vrcholu zásobníku odpovídá levé derivaci, je $\pi$ je levým rozborem $w$. □

## 6.4 SLL(k) gramatiky a jejich analýza

Pozorný čtenář si jistě položil otázku, zda tvrzení Věty 6.6 nejde z případu $k = 1$ zobecnit na $k > 1$. Ukážeme, že tomu tak není. Nejprve se zabývejme podmínkou (6.5) z Věty 6.6 zobecněnou pro $k \geq 1$.

**Věta 6.10.** *Pro libovolnou redukovanou CFG $G = (N, \Sigma, P, S)$ a libovolné $k \geq 1$ celé jsou následující dvě tvrzení* (6.6) *a* (6.7) *ekvivalentní:*

$$\left.\begin{array}{l} \forall A \in N.\ \forall A \to \beta \mid \gamma.\ \beta \neq \gamma : \\ FIRST_k(\beta FOLLOW_k(A)) \cap FIRST_k(\gamma FOLLOW_k(A)) = \emptyset \end{array}\right\} \quad (6.6)$$

$$\left.\begin{array}{lll} \textit{Pro libovolné dvě levé derivace} & (1) & S \Rightarrow^* wA\alpha \Rightarrow w\beta\alpha \Rightarrow^* wx \\ & (2) & S \Rightarrow^* w'A\alpha' \Rightarrow w'\gamma\alpha' \Rightarrow^* w'y \\ & (3) & \textit{podmínka } k : x = k : y \\ & & \textit{implikuje rovnost } \beta = \gamma. \end{array}\right\} \quad (6.7)$$

**Důkaz:** Negace tvrzení (6.6) je ekvivaletní s tvrzením:

$$\begin{array}{l} \exists A \in N.\ \exists A \to \beta \mid \gamma.\ \beta \neq \gamma : \\ \exists y \in FIRST_k(\beta FOLLOW_k(A)) \cap FIRST_k(\gamma FOLLOW_k(A)), \end{array}$$

které je ekvivaletní tvrzení:

$$\begin{array}{l} \exists A \in N.\ \exists A \to \beta \mid \gamma : \\ S \Rightarrow_L^* wA\delta_1, \quad y \in FIRST_k(\beta\delta_1), \\ S \Rightarrow_L^* w'A\delta_2, \ \ y \in FIRST_k(\gamma\delta_2) \text{ a } \beta \neq \gamma, \end{array}$$

což je ekvivalentní negaci trvrzení (6.7) – viz definice *FIRST*, *FOLLOW* a poznámka 6.2. □

Je tedy vidět, že každá gramatika splňující podmínku (6.7) je LL(k) gramatikou, ale obrácené tvrzení neplatí: lze ukázat (viz níže), že pro každé $k > 1$ existuje LL(k) gramatika taková, že nesplňuje podmínku (6.7). Má tedy smysl definovat tzv. SLL(k) gramatiky, a to (například) takto:

**Definice 6.11.** Nechť $G = (N, \Sigma, P, S)$ je redukovaná CFG, $k \geq 1$ je celé číslo. řekneme, že $G$ je SLL(k) gramatika, právě když pro libovolné dvě nejlevější derivace ($w \in \Sigma^*$)

$$\begin{array}{ll} (1) & S \Rightarrow^* wA\alpha \Rightarrow w\beta\alpha \Rightarrow^* wx \\ (2) & S \Rightarrow^* w'A\alpha' \Rightarrow w'\gamma\alpha' \Rightarrow^* w'y \ \text{, podmínka} \\ (3) & k : x = k : y \\ & \text{implikuje rovnost } \beta = \gamma. \end{array}$$

Řekneme, že gramatika $G$ je SLL právě když je SLL(k) pro nějaké $k \in \mathbb{N}$, jazyk $L$ je SLL(k) právě když existuje SLL(k) gramatika $G$ taková, že $L = L(G)$.

Je tedy vidět, že syntaktická analýza SLL(k) gramatik je přímočarým rozšířením syntaktické analýzy LL(1) gramatik. Detaily (zatím) ponecháváme čtenáři.

## 6.5 Příloha: algoritmy pro výpočet funkcí FIRST a FOLLOW

Nechť $\Sigma$ je abeceda, $L_1, L_2 \subseteq \Sigma^*, k \geq 1$. Definujeme funkci $\oplus_k : \Sigma^* \times \Sigma^* \longrightarrow \Sigma^*$ takto: $L_1 \oplus_k L_2 = \{w \mid \ w = k : xy \ \text{ pro nějaká } \ x \in L_1, y \in L_2\}$.

---

Je dána gramatika $G = (N, \Sigma, P, S)$ a řetězec $\alpha = Y_1 \cdot Y_2 \cdot \cdots \cdot Y_l$, kde $Y_x = N \cup \Sigma$.

1) $FI_k(x) = \{x\}$ pro $x \in \Sigma$
2) Výpočet $FI_k(x)$ pro $x \in N$:

   Nechť $N = \{X_1, X_2, \ldots, X_n\}$. Budeme počítat hodnotu $FI_k(X_i)$ současně pro všechny neterminály ($i = 1, \ldots, n$). Nechť všechna pravidla pro neterminál $X_i$ jsou tato:

   $$X_i \to Y_1^1 \ldots Y_{k_1}^1 \mid Y_1^2 \ldots Y_{k_2}^2 \mid \ldots \mid Y_1^j \ldots Y_{k_j}^j$$

   Potom

   $$\begin{aligned} FI_k(X_i) = {} & [\ FI_k(Y_1^1) \ \oplus_k \ FI_k(Y_2^1) \oplus_k \ \ldots \ \oplus_k \ FI_k(Y_{k_1}^1)\ ] \\ & \cup \ \ldots \ \cup \\ & [\ FI_k(Y_1^1) \ \oplus_k \ FI_k(Y_2^1) \oplus_k \ \ldots \ \oplus_k \ FI_k(Y_{k_1}^1)\ ]. \end{aligned}$$

   Hodnoty $FI_k(X_i)$ jsou pevnými body uvedené soustavy rekurzivních rovnic. Počáteční hodnoty jsou $FI_k(X_i) = \emptyset$.
3) $FIRST_k(\alpha) = FI_k(Y_1) \oplus_k FI_k(Y_2) \oplus_k \cdots \oplus_k FI_k(Y_l)$

---

Je dána gramatika $G = (N, \Sigma, P, S)$. Funkce $FO$ je definována pro $A \in N$.
Postupně počítáme hodnoty: $FO_1(A)$ pro všechny $A \in N$,

$$\begin{aligned} & FO_2(A) \text{ pro všechny } A \in N \\ & \vdots \\ & FO_k(A) \text{ pro všechny } A \in N \end{aligned}$$

Při výpočtu $FO_i(A)$ postupujeme následovně:

1) $FO_i(S) := \{\epsilon\}$ pro počáteční neterminál $S$.
   $FO_i(A) := \emptyset$ pro ostatní neterminály.
2) Pro každé pravidlo tvaru: $B \to \alpha A \beta \in P$, kde $\beta \neq \epsilon$

   $$FO_i(A) := FO_i(A) \cup [(FI_i(\beta) - \{\epsilon\}) \oplus_i FO_{i-1}(B)]$$

3) OPAKUJ
   Pro každé pravidlo tvaru: $B \to \alpha A \beta \in P$, kde $\beta = \epsilon$ nebo $\epsilon \in FI_1(\beta)$

   $$FO_i(A) := FO_i(A) \cup FO_i(B)$$

   Tak dlouho, dokud se nedosáhne pevného bodu.

---

## 6.6 Transformace gramatik do LL(1) tvaru

- odstranění levé rekurze
- levá substituce — odstranění konfliktu FIRST-FIRST
- pohlcení pravého kontextu — odstranění konfliktu FIRST-FOLLOW