

FJA L^AT_EX nápověda

Pozn.: Některé příkazy vyžadují externí balíčky, které najdete v poznámkách. Pokud ale budete používat šablonu `ib102.cls`, tak tam jsou balíčky již vloženy. Sazbu domácích úkolů můžete dělat na www.overleaf.com, kde najdete i obsáhlejší návod na L^AT_EX.

Základy L^AT_EXu

ODSTAVCE A ŘÁDKOVÁNÍ	
nový paragraf	prázdný řádek
zalomení řádku	<code>\\</code>
zalomení strany	<code>\newpage</code>
FORMÁTOVÁNÍ TEXTU	
tučně	<code>\textbf{text}</code>
kurzíva	<code>\textit{text}</code>
podtržení	<code>\underline{text}</code>
VÝČTY	
nečíslovaný výčet	<code>\begin{itemize}</code> <code>\item první položka</code> <code>\item druhá položka</code> <code>\end{itemize}</code>
číslovaný výčet	<code>\begin{enumerate}</code> <code>\item první položka</code> <code>\item druhá položka</code> <code>\end{enumerate}</code>
OBRÁZKY	
vložení obrázku	<code>\includegraphics{ cesta k obrázku }</code> ¹

Sazba matematiky

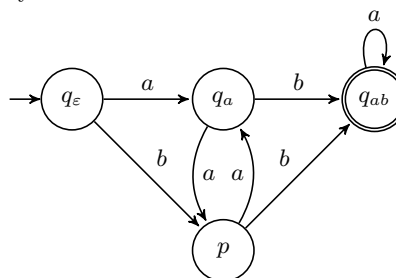
MATEMATICKÉ MÓDY	
matematika v textu	<code>\$ matematika \$</code>
rovnice na řádek	<code>\[rovnice \]</code>
INDEXY	
horní index	<code>a^n</code> <code>a^{n+m}</code>
dolní index	<code>a_n</code> <code>a_{n+m}</code>
ŘECKÉ ZNAKY	
$\alpha \beta \gamma \rho \sigma \delta \varepsilon \Sigma \Gamma$	<code>\alpha</code> <code>\beta</code> <code>\gamma</code> <code>\rho</code> <code>\sigma</code> <code>\delta</code> <code>\varepsilon</code> <code>\Sigma</code> <code>\Gamma</code>
BINÁRNÍ OPERÁTORY	
$\times \otimes \oplus \cup \cap$	<code>\times</code> <code>\otimes</code> <code>\oplus</code> <code>\cup</code> <code>\cap</code>
LOGICKÉ OPERÁTORY	
$\wedge \vee \neg \implies \iff$	<code>\and</code> <code>\lor</code> <code>\not</code> <code>\implies</code> <code>\Longleftarrow</code>
RELAČNÍ OPERÁTORY	
$\subset \supset \subseteq \supseteq$	<code>\subset</code> <code>\supset</code> <code>\subseteq</code> <code>\supseteq</code>
JAZYKY A ABECEDA	
$\{a, b, c\}$	<code>\set{a, b, c}</code> ³
\emptyset	<code>\emptyset</code>
\cdot	<code>\cdot</code>
$\text{co}\{a\}$	<code>\co\set{a}</code>
$\{a^n b^n \mid n \in \mathbb{N}_0\}$	<code>\set{ a^n b^n \mid</code> <code>n \in \mathbb{N}_0 }</code>

¹Vyžaduje balíček `graphicx`.

OSTATNÍ	
\in	<code>\in</code>
důkaz	<code>\begin{proof}</code> <code>... \end{proof}</code>
\mathbb{N}	<code>\mathbb{N}</code>
\mathcal{L}	<code>\mathcal{L}</code>
w neobsahuje a	<code>w \text{ neobsahuje } a</code>

Sazba konečných automatů

Příklad sazby konečného automatu:



```

\begin{tikzpicture} [>=latex, >=stealth', auto, node
  <- distance=2cm, semithick, initial text=]
  \node[state, initial] (q) {$q_{\varepsilon}$};
  \node[state] (qa) [right of = q] {$q_a$};
  \node[state, accepting] (qab) [right of = qa]
  <- {$q_{ab}$};
  \node[state] (p) [below of = qa] {$p$};

  \path[->, shorten >=1pt]
    (q) edge node {$a$} (qa)
    (qa) edge node {$b$} (qab)
    (q) edge node {$b$} (p)
    (p) edge [bend right] node {$a$} (qa)
    (qa) edge [bend right] node {$a$} (p)
    (p) edge node {$b$} (qab)
    (qab) edge [loop above] node {$a$} (qab)
  ;
\end{tikzpicture}

```

Automaty sazíme v prostředí `tikzpicture` – balíček `TikZ` a jeho knihovny `calc`, `arrows`, `automata`, `positioning`. Stav automatu definujeme pomocí `\node[atributy]` (jméno) [pozice] {popisek}, kde počáteční stavy označíme pomocí atributu `initial` a akceptující stavy pomocí `accepting`. Pozici definujeme vůči již nadefinovanému stavu. Pozice může být `right`, `left`, `above`, `below` nebo jejich kombinace. Přejechy automatu definujeme pomocí (počáteční stav) `edge [zakřivení] node popisek (koncový stav)`. Definice přechodu začínáme pomocí `\path` a zakončujeme středníkem. Pro přechod do sebe sama musíme nastavit zakřivení na `loop` a směr, odkud má smyčka vycházet.

²Definováno ve FJA stylu, jinak `\varepsilon`.

³Definováno ve FJA stylu, jinak `\{a, b, c\}`

Tabulky

	\triangleright	\sqcup	a
q_0	(q_0, \triangleright, R)	(q_0, \sqcup, R)	(q_0, a, L)
q_A	(q_A, \triangleright, R)	(q_A, \sqcup, R)	(q_0, a, L)

$$\begin{aligned} x &= a^k, \\ y &= a^l, \\ z &= a^{n-k-l}b^n \end{aligned}$$

```
\begin{tabular}{| c | c c c |}
\hline
& \rhd & \sqcup & a \\
q_0 & (q_0, \rhd, R) & (q_0, \sqcup, R) & (q_0, a, L) \\
q_A & (q_A, \rhd, R) & (q_A, \sqcup, R) & (q_0, a, L) \\
\hline
\end{tabular}
```

```
\begin{align*}
x &= a^k, \\
y &= a^l, \\
z &= a^{n-k-l}b^n \\
\end{align*}
```

$$\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_\varepsilon, \perp, \{q_{acc}\})$$

Pro sazbu tabulek se využívá prostředí `tabular`, kterému definujete jako parametry počty sloupců a jejich zarovnání a případně oddělovací čáry mezi sloupci. Z příkladu `{ | c | c c c | }` popisuje 4 vycentrované sloupce. Místo `c` se může použít `l` nebo `r` pro zarovnání celého sloupce doleva nebo doprava. Příkaz `\hline` definuje horizontální čáru. Tabulku zapisujeme po řádcích tak, že jednotlivé buňky oddělujeme pomocí `&` a řádky pomocí dvou zpětných lomítek.

```
\[ \mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_\varepsilon, \perp, \{q_{acc}\}) \\
\rightarrow q_\varepsilon \text{ \varepsilon, } \bot, \text{ \set{ } } \]
```

$$\delta(q_i, a, \perp) = \{(q_i, A\perp)\}$$

$$\delta(q_i, a, A) = \{(q_i, AA)\}$$

Sazba gramatik

$$\begin{aligned} P &= \{S \rightarrow \varepsilon \mid A, \\ &\quad A \rightarrow a \mid aA \mid bA, \\ &\quad B \rightarrow b \mid aB \mid bB\}. \end{aligned}$$

```
\begin{align*}
P &= \set{S \rightarrow \varepsilon \mid A, \\ &\quad A \rightarrow a \mid aA \mid bA, \\ &\quad B \rightarrow b \mid aB \mid bB}. \\
\end{align*}
```

```
\begin{align*}
\delta(q_i, a, \bot) &= \set{ (q_i, A\bot) } \\
\delta(q_i, a, A) &= \set{ (q_i, AA) } \\
\end{align*}
```

$$(q, \text{automat}, S) \stackrel{\varepsilon}{\vdash} (q, \text{automat}, XSY) \vdash$$

```
\[ (q, \text{automat}, S) \turnstile{\varepsilon} (q, \\
\rightarrow \text{automat}, XSY) \turnstile{} \]
```

Prostředí `align2` slouží pro zarovnání matematických výrazů podle operátorů. Zarovnává se v místě určeném `&`.

Užitečné příklady

$$(co-\{b\}^+ \cdot \{\varepsilon\}^+) \setminus \{b\}^+$$

```
\(\co{\set{ b }^+} \cdot \set{ \varepsilon }^+ \\
\rightarrow \setminus \set{ b }^+\)
```

$$L = \left\{ a^i b^j c^k \mid i, j, k \geq 0, \text{ pokud } i = 1, \text{ pak } j = k \right\}$$

```
\[L = \left\{ a^i b^j c^k \mid i, j, k \geq 0 \text{ \textit{pokud } } i=1 \text{ \textit{pak } } j=k \right\}\]
```

²Nachází se v balíku `amsmath`.

Kde vzít L^AT_EX, jak ho editovat?

Buď nainstalovat (na Linuxu z distribučního balíku, na Windows můžete nainstalovat například [texlive](http://www.tug.org/texlive/acquire-netinstall.html) z www.tug.org/texlive/acquire-netinstall.html), nebo používat www.overleaf.com.

Pokud máte lokální instalaci, k editování můžete používat buď libovolný textový editor a překládat na příkazové řádce příkazem `pdflatex`, nebo můžete využít specializovaný editor, například [Texmaker](http://www.xmlmath.net/texmaker/index.html) (www.xmlmath.net/texmaker/index.html). Každopádně budete potřebovat mít soubor se šablonou (`ib102.cls`) ve stejné složce jako vaše řešení.

V případě dotazů použijte příslušné diskusní fórum.