# Rozšíření

## SIMPLE

Interpret podporuje podmínku if bez části else a u podmíněného příkazu a cyklů lze použít jeden příkaz místo složeného. Implementace podmínky if bez části else vnesla do LL gramatiky konflikt, který je popsán v kapitole o analyzátoru struktury programu. Bylo potřeba v LL gramatice rozdělit podmíněný příkaz if/else na příkaz if s nepovinnou částí else. Podpora samostatného příkazu místo složeného u podmíněného příkazu a cyklů byla vyřešena jednoduše nahrazením na odpovídajících místech v LL gramatice { STMTS } samostatným neterminálem STMT, ze kterého se { STMTS } může vygenerovat.

## WHILE

Interpret podporuje cykly typu while a do-while. Vzhledem k již implementovanému cyklu for, který je z těchto cyklů nejsložitější, nebyla implementace těchto dalších dvou cyklů problém. Oba nové cykly jsou si velmi podobné, liší se pouze umístěním podmínky, která při vyhodnocení false zapříčiní skok mimo cyklus. U obou cyklů se syntakticky vyskytují jednotlivé části v pořadí přesně tak, jak jsou vykonávány (na rozdíl od cyklu for, jehož inkrementační část nacházející se v hlavičce cyklu se vykonává až jako poslední), takže není potřeba generovat zbytečné skoky.

## BASE

Rozšíření *BASE* se týká pouze lexikálního analyzátoru a je řešeno opět konečným automatem. Jako první je nutné rozlišit soustavu (BIN, OCT, HEX) a dále načítat znaky odpovídající dané soustavě. Při escape sekvenci v řetězci je dále důležité kontrolovat počet znaků (BIN – 8, OCT – 3, HEX – 2) – při nedodržení počtu znaků je vrácena lexikální chyba. Načtená hodnota je potom převedena pomocí funkce *strtol* do desítkové soustavy.