# Změny v block

První dvě - pouze je přidána možnost specifikace datového typu. Specifikace není nutná. Pokud není uvedena, tak se proměnná považuje za číslo (TODO: bylo by pěkné umět rozeznat typ podle pravé strany a automaticky ho dpolnit, ale mi zní dost složitě, proto je to prostě číslo)

Poslední porovnání je specifikace parametrů do procedury. U parametrů může být dodán typ (TODO: tohle funguje jako nad, pokud typ není, tak to je prostě číslo)

# Změny ve statementu

1) přidání možnosti multi přiřazení (a = b = c = 5)

2) přidání možnosti parelelního přiřazení ( {a, b} = {1, 2})

3) else větev

4) ternární operátor

5) for cyklus – jednoduchý, neumí vlastní step

6) foreach – možná ani nebudem implementovat, ale takhle by je první návrh, problém je, že takhle nevíme, s kterou hodnou zrovna pracujem

7) return statement

# Změny ve factoru

Přidání hodnoty – porovnání stringů.

# Výsledná gramatika

Zamýšlené typy jsou int, float, string, bool (možná char?). Co by jazyk měl umožňovat je (seznam dole)

Co nevím jestli zvládnem je implementace pole a práce s ním (tedy foreach), ale to bude záviset I na bodech z GUI.

Seznam funkcí a vlastností:

integer variable and constant (whole number) (not sure if this isn't basically done by the language itself)

assignments defined in original PL/0

basic operations (+,-,\*,/,&,|,^,())(==, <=, <, >, >=)

loops - one is compulsory. Other loop types are for bonus points. For now lets try for, while, foreach (if we will have arrays)

simple condition without

subprogram definition function/ procedure and its call

bonus (1pt): else

bonus (1pt): other loop types

bonus (1pt): boolean and related operation

bonus (1pt): data type: real (with integer functionality) - aka. floating point

bonus (1pt): string with concat operator

bonus (1pt): switch

bonus (1pt): multiple assignment - "a = b = c = d = 3"

bonus (1pt): ternary operator - " (a < b) ? a : b;"

bonus (1pt): parallel assignment - "{a, b, c, d} = {1, 2, 3, 4}"

bonus (2pt): string comparison operator

bonus (2pt): subprogram parameters by value

bonus (2pt): subprogram return value

maybe bonus (2pt): array and related operations