Lake Symbols for Island Parsing

Katsumi Okuda, The University of Tokyo / Mitsubishi Electric Corporation

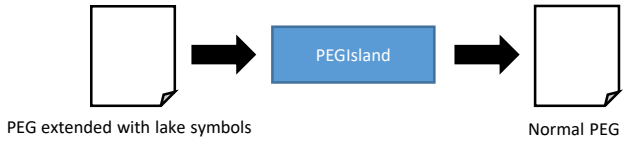
Shigeru Chiba, The University of Tokyo

Motivation

- Reduce the number of rules in grammar for parser generation
- Ease the writing of island grammar with lake symbols

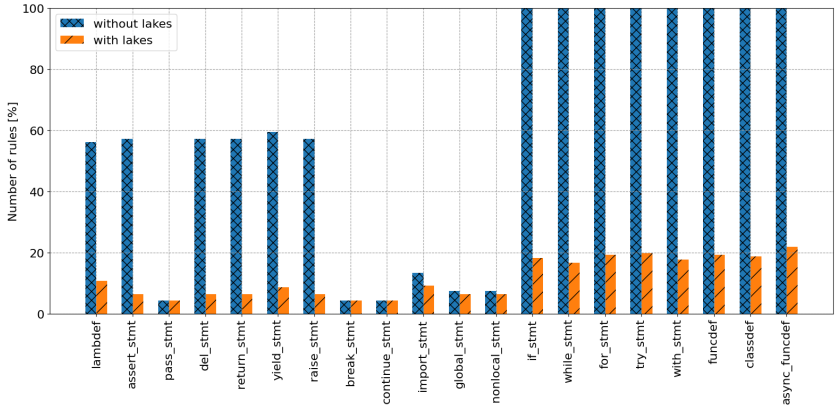
Approach

- Extend a parsing expression grammar (PEG) with lake symbols
- Convert an extended PEG into a normal PEG



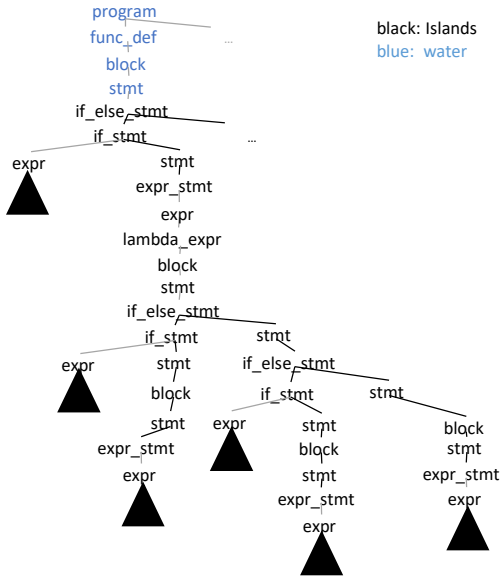
The numers of rules in grammars for Python

- With lake symbols, 89% of rules can be reduced for extracting a specific program construct on average compared to a complete parser



Island grammar without lakes

```
program        <- spacing (stmt / func_def)*
func_def       <- FUNCTION ID LPAREN ID? RPAREN block
program        <- spacing program_sea*
program_sea    <- if_else_stmt / program_water / .
if_else_stmt   <- if_stmt (ELSE stmt)?
if_stmt        <- IF LPAREN expr RPAREN stmt
stmt           <- block / if_else_stmt / exp_stmt
exp_stmt       <- expr SEMICOLON
block          <- LBRACE stmt* RBRACE
expr           <- rel_expr (ASSIGN rel_expr)*
rel_expr       <- add_expr ((EQ / GT / LT) add_expr)*
add_expr       <- mul_expr ((PLUS / MINUS) mul_expr)*
mul_expr       <- pri_expr ((MUL / DIV) pri_expr)*
pri_expr       <- LPAREN expr RPAREN / lambda_expr / funcall / NUMBER / STRING / ID
lambda_expr    <- LAMBDA LPAREN ID? RPAREN block
funcall        <- ID LPAREN expr? RPAREN
```



Island grammar with lake

```
program        <- spacing (stmt / func_def)*
func_def       <- FUNCTION ID LPAREN ID? RPAREN block
program        <- spacing <program_sea>*
<program_sea>  <- if_else_stmt / program_water
if_else_stmt   <- if_stmt (ELSE stmt)?
if_stmt        <- IF LPAREN expr RPAREN stmt
stmt           <- block / if_else_stmt / exp_stmt
exp_stmt       <- expr SEMICOLON
block          <- LBRACE stmt* RBRACE
expr           <- rel_expr (ASSIGN rel_expr)*
rel_expr       <- add_expr ((EQ / GT / LT) add_expr)*
add_expr       <- mul_expr ((PLUS / MINUS) mul_expr)*
mul_expr       <- pri_expr ((MUL / DIV) pri_expr)*
pri_expr       <- LPAREN expr RPAREN / lambda_expr / funcall / NUMBER / STRING
/ ID
lambda_expr    <- LAMBDA LPAREN ID? RPAREN block
funcall        <- ID LPAREN expr? RPAREN
expr           <- <expr_sea>+
<expr_sea>     <- if_else_stmt / LPAREN expr_sea* RPAREN / block / STRING
```

<lake_symbol>