Mothra Operators: (Examples from \cite{WongMathur1995})

AAR array reference for array reference replacement
A(I) = B(J) + 3 => A(I) = A(I) + 3
** N/A

ABS absolute value insertion
P = X* Y => P = zpush(x) + y
** N/A

ACR array reference for constant replacement
FOUND = 0 => FOUND = A(I)
** N/A

✓ AOR arithmetic operator replacement
A = B + C => A = B-C
** Math → limited

ASR array reference for scalar variable replacement
P = X * A(I) => P = X * A(A(I))
** N/A

CAR constant for array reference replacement
P = X * A(I) => P = X * 0
** ~~Non_VOID_METHOD_CALL~~
** ~~CONSTRUCTOR_CALLS~~
** INLINE_CONST

CNR comparable array name replacement
A(I) = B(J) + 4 => A(I) = A(J) + 4
** N/A

CRP constant replacement
DO 10 I=1, N => DO 10 I=1, N, 2
** ~~Non_VOID_METHOD_CALL~~
** ~~CONSTRUCTOR_CALLS~~
** INLINE_CONST

CSR constant for scalar variable replacement
P = X * A(I) => P = X * A(1)
** Non_VOID_METHOD_CALL
** CONSTRUCTOR_CALLS
** INLINE_CONST

DER DO statement alterations
DO 10 I=1, N => onetrip 10 I=1, N

/* Wrong
** Non_VOID_METHOD_CALL
** CONSTRUCTOR_CALLS
** INLINE_CONST
** RETURN_VALS
*/

✓ DSA DATA statement alterations
DATA x/0/ => DATA x/1/
** INLINE_CONST
** EXPERIMENTAL_MEMBER_VARIABLE

GLR GOTO label replacement

α α GOTO 20 => GOTO 10
** NEGATE_CONDITIONALS -- to some extent

α LCR logical connector replacement
X && Y => X || Y
** N/A

✓ ROR relational operator replacement
X > Y => X >= Y
** CONDITIONAL_BOUNDARY
** NEGATE_CONDITIONAL

α RSR RETURN statement replacement
P = X* A(I) => RETURN
** N/A

α SAN statement analysis
Each statement is replaced by trap
** N/A

✓ SAR scalar variable for array reference replacement
P = X * A(I) => P = X * Y
** N/A

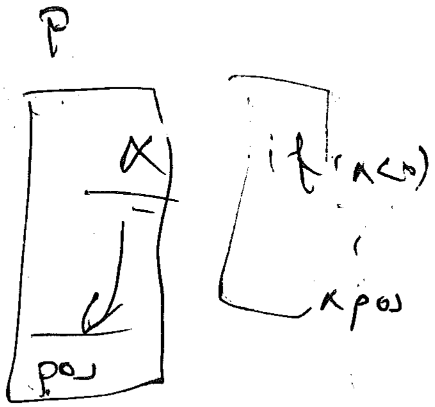
α SCR scalar for constant replacement
X = 5 => X = P
** N/A

✓ SDL statement deletion
Each statement deleted
** ~~VOID_METHOD_CALLS~~ , SDL

✓ SRC source constant replacement
X = 5 => X = 7
** INLINE_CONSTS
** EXPERIMENTAL_MEMBER_VARIABLE

✓ SVR scalar variable replacement
P = X * A(I) => P = Y * A(I)
Kap
** N/A
Inc ~~Inc~~ / ~~Dec~~

✓ UOI unary operator insertion
X = 5 => X = -1
** INVERT_NEGS



Mathur \cite{Mathur1991}:
2-Selective (Removing SVR and ASR from Mothra operators.)

Offutt et al. \cite{Offutt1993}:
4-Selective: Removing SVR, AST, CSR and SCR from Mothra
6-Selective: Removing ASR, CSR, SCR, SVR, ARC, and ACT from Mothra

Wong and Mathur ~~\cite~~{WongMathur1995}:
~~ABS~~ and ROR ✓
Sampling

** ~~Invert Negs~~

RetStaDel
Deletes return Statement
** Not sure but:

ArgDel
Argument Deletion
** N/A

✓ ArgLogNeg
Insert Logical Negation on Argument
** INVERET_NEGS

✓ OAAN
Arithmetic Operator Mutation
** MATH

OABN
Arithmetic by Bitwise Operator
** N/A

OAEA
Arithmetic Assignment by Plain Assignment
** ?

OALN
Arithmetic Operator by Logical Operator
** N/A

✓ OBBN
Bitwise Operator Mutation
** ~~Not Available~~ Math

OBNG
Bitwise Negation
** Not available

OBSN
Bitwise Operator by Shift Operator
** ~~Not available~~ ~~Math~~

✓ OCNG
Logical Context Negation
** Negate CONDITIONALS

OCOR
Cast Operator by Cast Operator
** N/A

✓ Oido
Increment/Decrement Mutation
** INCREMENTS

OLAN
Logical Operator by Arithmetic Operator
** N/A

OLBN
Logical Operator by Bitwise Operator
** N/A

Offut et al. \cite{offut:1996}

~~ABS~~, UOI ✓ ~~LCR~~, AOR, ROR ✓

Javalanche uses a subset of above: Negate Jump Condition ✓, Omit Method ✓ Call, Replace Arithmetic Operator ✓ Math, and Replace ~~Numerical~~ ✓ Constant

Barbosa et al. \cite{Barbosa:2001}

Selected 10 out of 77 Proteum Operators Nil

Namin and Andrews \cite{Namin:2006}

From Elimination-Based

- I-IndVarAriNeg
  - Inserts Arithmetic Negation at Non Interface Variables
- I-IndVarLogNeg
  - Inserts Logical Negation at Non Interface Variables
- II-ArgRepReq
  - Argument Replacement by Required Constants
- u-OALN
  - Arithmetic Operator by Logical Operator
- u-OASN
  - Arithmetic Operator by Shift Operator
- u-OCNG
  - Logical Context Negation
- u-OLLN
  - Logical Operator Mutation

From Cluster Analysis

- I-DirVarRepCon
  - Replaces Interface Variables by Used Constants
  - **
- I-IndVarAriNeg
  - Inserts Arithmetic Negation at Non Interface Variables
- I-RetStaDel
  - Deletes Return Statement
- II-ArgIncDec
  - Argument Increment and Decrement
- u-OABN
  - Arithmetic by Bitwise Operator
- u-OALN
  - Arithmetic Operator by Logical Operator
- u-OASN
  - Arithmetic Operator by Shift Operator
- u-OCNG
  - Logical Context Negation
- u-OLNG
  - Logical Negation
- u-ORSN
  - Relational Operator by Shift Operator
- u-SRSR
  - return Replacement
- u-STRP
  - Trap on Statement Execution

Namin et al. \cite{SiamiNamin:2008}

- IndVarAriNeg
  - Inserts Arithmetic Negation at Non Interface Variables
  - ** Invert-Negs
- α IndVarBitNeg
  - Inserts Bit Negation at Non Interface Variables

OLLN

Logical Operator Mutation
** MATH

OLNG

Logical Negation
** NEGATE_CONDITIONALS

OLSN

Logical Operator by Shift Operator
** N/A

ORSN

Relational Operator by Shift Operator
** ?

SGLR

goto Label Replacement
** N/A

SMTC

n-trip continue
** N/A

SMVB

Move Brace Up and Down
** N/A

SSWM

switch Statement Mutation
** RMSWITCH
~~** EXPERIMENTAL_SWITCH~~

STRI

Trap on if Condition
** REMOVE_CONDITIONAL
** DELETE_CONDITIONAL

SWDD

while Replacement by do-while
** N/A

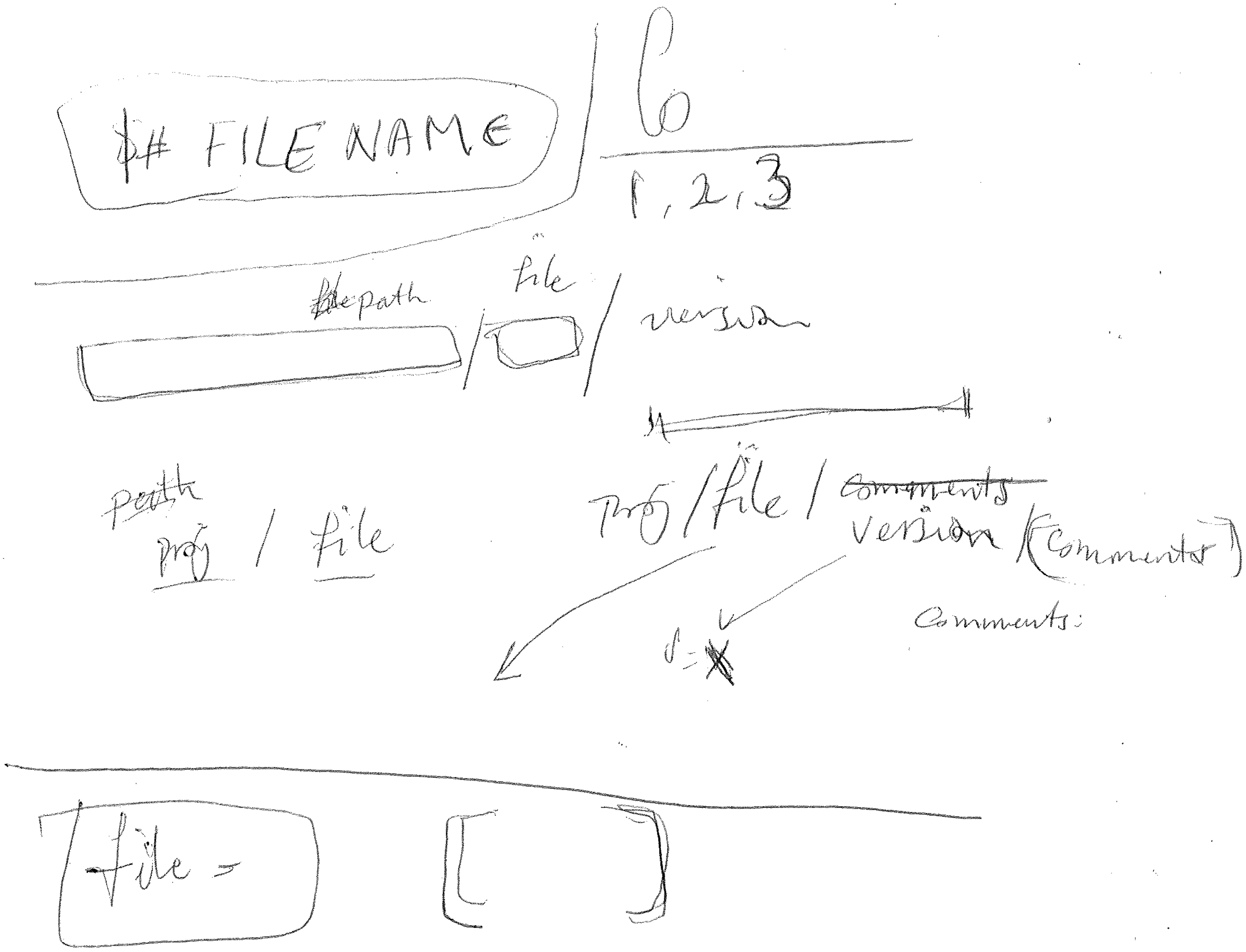
VGPR

Mutate Global Pointer References
** N/A

Deng and Offutt \cite{DengOffutt:2013}: SDL

SDL is good enough

1) All possible cases: Every possible case must be considered.
2) Boolean conditions: Most control structures have at least one Boolean condition, which should be deleted.
3) Inner statements: Statements inside control structures must be deleted.
4) Nested control structures: Nested control structures must be treated recursively.


$# FILE NAME
6
1, 2, 3
~~file~~path
file
version
~~path~~
proj / file
Proj / file / ~~comments~~
version (comments)
Comments:
v = X
file =