to be compiled

```
(def a (if 1
           (if true (setv c 2)
                (setv d 4))
           3))
```

```
compile =====
compile-atom ======
compile-f:  <function compile_expression at 0x10d49c500>
atom:  (u'def' u'a' (u'if' 1L (u'if' u'True' (u'setv' u'c' 2L) (u'setv' u'd' 4L)) 3L))


compile-atom ======
compile-f:  <function compile_def_expression at 0x10d49c578>
atom:  (u'def' u'a' (u'if' 1L (u'if' u'True' (u'setv' u'c' 2L) (u'setv' u'd' 4L)) 3L))
```

compile-def-expression doing
(setv result (.compile self result)) where result is the (if 1 (if true (setv c 2) (setv d 4)) 3)) form

```
compile =====
compile-atom ======
compile-f:  <function compile_expression at 0x10d49c500>
atom:  (u'if' 1L (u'if' u'True' (u'setv' u'c' 2L) (u'setv' u'd' 4L)) 3L)


compile-atom ======
compile-f:  <function compile_if at 0x10d49c488>
atom:  (u'if' 1L (u'if' u'True' (u'setv' u'c' 2L) (u'setv' u'd' 4L)) 3L)
```

compile-if doing
[condition (.compile self (.pop expression 0))]

```
compile =====
compile-atom ======
compile-f:  <function compile_integer at 0x10d49c6e0>
atom:  1


atom result 1:  {'nodes': [1L], 'value': 1L}
```

compile-atom doing (+ (Result) ret))
Because result mlast `Number{1] is not a instance of Result
add result mlast `Number{1} to (Result) to convert it to a Result object
The messages display twice. First time it's adding the raw Number class to (Result), second time it's adding the converted Result object)

```
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'nodes': [1L], 'value': 1L}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': <class Number 1 at 0x10d3f5c90>}
result after addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 1 at 0x10d3f5c90>}


compile =====
compile-atom ======
compile-f:  <function compile_expression at 0x10d49c500>
atom:  (u'if' u'True' (u'setv' u'c' 2L) (u'setv' u'd' 4L))
```

compile-if doing
[body (.compile self (.pop expression 0))]
remember the if expression is (if 1 (if true (setv c 2) (set d 4)) 3))
where body is (if true (setv c 2) (setv d 4)) while the orel is just 3

```
compile-atom ======
compile-f:  <function compile_if at 0x10d49c488>
atom:  (u'if' u'True' (u'setv' u'c' 2L) (u'setv' u'd' 4L))


compile =====
compile-atom ======
compile-f:  <function compile_symbol at 0x10d49c848>
atom:  True


atom result 1:  {'nodes': [u'True']}
```

compile-if doing
[condition (.compile self (.pop expression 0))]
for (if true (setv c 2) (setv d 4))

```
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'nodes': [u'True']}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': <class Id True at 0x10d3f5510>}
result after addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id True at 0x10d3f5510>}


compile =====
compile-atom ======
compile-f:  <function compile_expression at 0x10d49c500>
atom:  (u'setv' u'c' 2L)
```

compile-if doing
[body (.compile self (.pop expression 0))]
remember now the if expression is (if true (setv c 2) (setv d 4))
where body is (setv c 2) while the orel is (setv d 4)

```
compile-atom ======
compile-f:  <function compile_setv_expression at 0x10d49c5f0>
atom:  (u'setv' u'c' 2L)
```

compile-setv doing (setv result (.compile self result))

```
compile =====
compile-atom ======
compile-f:  <function compile_integer at 0x10d49c6e0>
atom:  2


atom result 1:  {'nodes': [2L], 'value': 2L}


result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'nodes': [2L], 'value': 2L}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': <class Number 2 at 0x10d3f5150>}
result after addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 2 at 0x10d3f5150>}


compile =====
compile-atom ======
compile-f:  <function compile_symbol at 0x10d49c848>
```

compile-setv doing (setv ld-name (.compile self name))
the setv expression is (setv c 2)

```
atom:  c


atom result 1:  {'nodes': [u'c']}


result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'nodes': [u'c']}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': <class Id c at 0x10d3f53d0>}
result after addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id c at 0x10d3f53d0>}
```

compile-atom converting Id c to result object

compile-setv doing (+= result (Set (mlast.Id [st-name]) [result.force-expr])).
where the whole expresion is (setv c 2)
result is compiled result of Number 2
notice after addition, expr becomes None. How does this work in long time?

```
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 2 at 0x10d3f5150>}
result to be added: {'nodes': [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]]}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 2 at 0x10d3f5150>}
result to be added: {'stmts': [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result after addition: {'stmts': [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}


result before addition: {'stmts': [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id c at 0x10d3f53d0>}
result after addition: {'stmts': [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id c at 0x10d3f53d0>}
```

compile-setv doing (+= result ld-name)
the setv expression is (setv c 2) so this returned the assigend value

```
atom result 1:  {'stmts': [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id c at 0x10d3f53d0>}
```

compile-if finished [body (.compile self (.pop expression 0))]
where the whole if expression is (if true (setv c 2) (setv d 4))

```
atom result 1:  {'stmts': [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id c at 0x10d3f53d0>}


compile =====
compile-atom ======
compile-f:  <function compile_expression at 0x10d49c500>
atom:  (u'setv' u'd' 4L)
```

compile-if doing orel part
where the whole if expression is (if true (setv c 2) (setv d
4))

```
compile-atom ======
compile-f:  <function compile_setv_expression at 0x10d49c5f0>
atom:  (u'setv' u'd' 4L)


compile =====
compile-atom ======
compile-f:  <function compile_integer at 0x10d49c6e0>
atom:  4


atom result 1:  {'nodes': [4L], 'value': 4L}


result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'nodes': [4L], 'value': 4L}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': <class Number 4 at 0x10d3f5e10>}
result after addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 4 at 0x10d3f5e10>}


compile =====
compile-atom ======
compile-f:  <function compile_symbol at 0x10d49c848>
atom:  d


atom result 1:  {'nodes': [u'd']}


result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'nodes': [u'd']}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': <class Id d at 0x10d3f59d0>}
result after addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id d at 0x10d3f59d0>}
```

compile-atom converting Id d to result object

```
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 4 at 0x10d3f5e10>}
result to be added: {'nodes': [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]]}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 4 at 0x10d3f5e10>}
result to be added: {'stmts': [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result after addition: {'stmts': [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}
```

compile-setv doing (+= result ld-name)
where the setv expression is (setv d 4)

```
result before addition: {'stmts': [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id d at 0x10d3f59d0>}
result after addition: {'stmts': [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id d at 0x10d3f59d0>}


atom result 1:  {'stmts': [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id d at 0x10d3f59d0>}
```

compile-if finished orel part
where the whole if expression is (if true (setv c 2) (setv d 4

```
atom result 1:  {'stmts': [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id d at 0x10d3f59d0>}


result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'nodes': [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []]}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
```

compile-if doing (setv ret (+ (Result) (mlast.Local [var]) ret))
where var is generated anonymous variable

but how body’s result is stored in expr? Rember the body
is a setv expression. In compile-assign we explicitly store
the result of body in the “name” and append the “name” to the
result of compiling in the following line:
(+= result ld-name)

compile-if doing (setv ret (+ (Result) (mlast.Local [var]) ret))
where var is generated anonymous variable

```
result to be added: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result after addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}
```

compile-if doing (setv ret (+ (Result) (mlast.Local [var])
ret))

```
result before addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id True at 0x10d3f5510>}
result after addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id True at 0x10d3f5510>}
```

compile-if doing (+= body (mlast.Set [var] [body.force-expr]))

```
result before addition: {'stmts': [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id c at 0x10d3f53d0>}
result to be added: {'nodes': [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]]}
result before addition: {'stmts': [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id c at 0x10d3f53d0>}
result to be added: {'stmts': [<class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result after addition: {'stmts': [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], 'temp_vars':
[], '_Result__used_expr': True, '_expr': None}
```

so after compile, the result of body is stored in expr,
and now is being stored in the temp-variable hua_anon?

compile-if doing (+= orel (mlast.Set [var] [orel.force-expr]))

```
result before addition: {'stmts': [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id d at 0x10d3f59d0>}
result to be added: {'nodes': [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]]}
result before addition: {'stmts': [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id d at 0x10d3f59d0>}
result to be added: {'stmts': [<class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result after addition: {'stmts': [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>], 'temp_vars':
[], '_Result__used_expr': True, '_expr': None}
```

compile-if doing (+= ret (mlast.If ret.force-expr body.stmts orel.stmts))

```
result before addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id True at 0x10d3f5510>}
result to be added: {'nodes': [<class Id True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]]
at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]]}
result before addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id True at 0x10d3f5510>}
result to be added: {'stmts': [<class If nodes: [<class Id True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c
at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at
0x10d3f5950>], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result after addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at
0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id
_hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}
```

compile-if doing (+= ret (apply Result [] {"expr" expr-name "temp_vars" [expr-name var]})). If’s expr is the anonymous variable introduced at the beginning. why temp-vars is done this way?

```
result before addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at
0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id
_hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [<class Id _hua_anon_var_1 at 0x10d3f5850>, <class Id _hua_anon_var_1 at 0x10d3f5690>], '_Result__used_expr': False, '_expr': <class Id _hua_anon_var_1 at 0x10d3f5850>}
result after addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at
0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id
_hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>], 'temp_vars': [<class Id _hua_anon_var_1 at 0x10d3f5850>, <class Id _hua_anon_var_1 at 0x10d3f5690>], '_Result__used_expr': True, '_expr': <class Id _hua_anon_var_1 at 0x10d3f5850>}


atom result 1:  {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at
0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id
_hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>], 'temp_vars': [<class Id _hua_anon_var_1 at 0x10d3f5850>, <class Id _hua_anon_var_1 at 0x10d3f5690>], '_Result__used_expr': True, '_expr': <class Id _hua_anon_var_1 at 0x10d3f5850>}


atom result 1:  {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at
0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id
_hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>], 'temp_vars': [<class Id _hua_anon_var_1 at 0x10d3f5850>, <class Id _hua_anon_var_1 at 0x10d3f5690>], '_Result__used_expr': True, '_expr': <class Id _hua_anon_var_1 at 0x10d3f5850>}
```

compile-if finished [body (.compile self (.pop expression 0))]
remember the if expression is (if 1 (if true (setv c 2) (set d 4)) 3))

```
compile =====
compile-atom ======
compile-f:  <function compile_integer at 0x10d49c6e0>
atom:  3
```

compile-if doing [orel (.compile self (.pop expression 0))]
remember the if expression is (if 1 (if true (setv c 2) (set d 4)) 3))

```
atom result 1:  {'nodes': [3L], 'value': 3L}


result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'nodes': [3L], 'value': 3L}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': <class Number 3 at 0x10d3f5610>}
result after addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 3 at 0x10d3f5610>}
```

compile-if doing (setv ret (+ (Result) (mlast.Local [var]) ret))
remember the if expression is (if 1 (if true (setv c 2) (set d 4)) 3))

```
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'nodes': [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], []]}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], []] at 0x10d3f5fd0>], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result after addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], []] at 0x10d3f5fd0>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}
```

compile-if doing (setv ret (+ (Result) (mlast.Local [var]) ret))
remember the if expression is (if 1 (if true (setv c 2) (set d 4)) 3))

```
result before addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], []] at 0x10d3f5fd0>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 1 at 0x10d3f5c90>}
result after addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], []] at 0x10d3f5fd0>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 1 at 0x10d3f5c90>}


result before addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at
0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id
_hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>], 'temp_vars': [<class Id _hua_anon_var_1 at 0x10d3f5850>, <class Id _hua_anon_var_1 at 0x10d3f5690>], '_Result__used_expr': True, '_expr': <class Id _hua_anon_var_1 at 0x10d3f5850>}
result to be added: {'nodes': [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Id _hua_anon_var_1 at 0x10d3f5850>]]}
result before addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at
0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id
_hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>], 'temp_vars': [<class Id _hua_anon_var_1 at 0x10d3f5850>, <class Id _hua_anon_var_1 at 0x10d3f5690>], '_Result__used_expr': True, '_expr': <class Id _hua_anon_var_1 at 0x10d3f5850>}
result to be added: {'stmts': [<class Set nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Id _hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result after addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at
0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id
_hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>, <class Set nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Id _hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}
```

compile-if doing (+= body (mlast.Set [var] [body.force-expr]))
remember the if expression is (if 1 (if true (setv c 2) (set d 4)) 3))

compile-if doing (+= orel (mlast.Set [var] [orel.force-expr]))

```
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 3 at 0x10d3f5610>}
result to be added: {'nodes': [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]]}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 3 at 0x10d3f5610>}
result to be added: {'stmts': [<class Set nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result after addition: {'stmts': [<class Set nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}
```

compile-if doing (+= ret (mlast.If ret.force-expr body.stmts orel.stmts))
remember the if expression is (if 1 (if true (setv c 2) (set d 4)) 3))

```
result before addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], []] at 0x10d3f5fd0>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 1 at 0x10d3f5c90>}
result to be added: {'nodes': [<class Number 1 at 0x10d3f5c90>, [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class
Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>,
<class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>, <class Set nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Id _hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>], [<class Set nodes: [[<class Id
_hua_anon_var_2 at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>]]}
result before addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], []] at 0x10d3f5fd0>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Number 1 at 0x10d3f5c90>}
result to be added: {'stmts': [<class If nodes: [<class Number 1 at 0x10d3f5c90>, [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at
0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at
0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>, <class Set nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Id _hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>],
[<class Set nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>]] at 0x10d3f5f50>], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result after addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], []] at 0x10d3f5fd0>, <class If nodes: [<class Number 1 at 0x10d3f5c90>, [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id
True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes:
[<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>, <class Set nodes: [[<class Id _hua_anon_var_2 at
0x10d3f5d90>], [<class Id _hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>], [<class Set nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>]] at 0x10d3f5f50>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}


result before addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], []] at 0x10d3f5fd0>, <class If nodes: [<class Number 1 at 0x10d3f5c90>, [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id
True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes:
[<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>, <class Set nodes: [[<class Id _hua_anon_var_2 at
0x10d3f5d90>], [<class Id _hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>], [<class Set nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>]] at 0x10d3f5f50>], 'temp_vars': [], '_Result__used_expr': True, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [<class Id _hua_anon_var_2 at 0x10d3f5710>, <class Id _hua_anon_var_2 at 0x10d3f5d90>], '_Result__used_expr': False, '_expr': <class Id _hua_anon_var_2 at 0x10d3f5710>}
result after addition: {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], []] at 0x10d3f5fd0>, <class If nodes: [<class Number 1 at 0x10d3f5c90>, [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id
True at 0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes:
[<class Id [<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>, <class Set nodes: [[<class Id _hua_anon_var_2 at
0x10d3f5d90>], [<class Id _hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>], [<class Set nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>]] at 0x10d3f5f50>], 'temp_vars': [<class Id _hua_anon_var_2 at 0x10d3f5710>, <class Id
_hua_anon_var_2 at 0x10d3f5d90>], '_Result__used_expr': True, '_expr': <class Id _hua_anon_var_2 at 0x10d3f5710>}


atom result 1:  {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], []] at 0x10d3f5fd0>, <class If nodes: [<class Number 1 at 0x10d3f5c90>, [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at
0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id
[<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>, <class Set nodes: [[<class Id _hua_anon_var_2 at
0x10d3f5d90>], [<class Id _hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>], [<class Set nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>]] at 0x10d3f5f50>], 'temp_vars': [<class Id _hua_anon_var_2 at 0x10d3f5710>, <class Id
_hua_anon_var_2 at 0x10d3f5d90>], '_Result__used_expr': True, '_expr': <class Id _hua_anon_var_2 at 0x10d3f5710>}


atom result 1:  {'stmts': [<class Local nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], []] at 0x10d3f5fd0>, <class If nodes: [<class Number 1 at 0x10d3f5c90>, [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at
0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id
[<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>, <class Set nodes: [[<class Id _hua_anon_var_2 at
0x10d3f5d90>], [<class Id _hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>], [<class Set nodes: [[<class Id _hua_anon_var_2 at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>]] at 0x10d3f5f50>], 'temp_vars': [<class Id _hua_anon_var_2 at 0x10d3f5710>, <class Id
_hua_anon_var_2 at 0x10d3f5d90>], '_Result__used_expr': True, '_expr': <class Id _hua_anon_var_2 at 0x10d3f5710>}
```

compile-if doing final step
remember the if expression is (if 1 (if true (setv c 2) (set d 4)) 3))

```
compile =====
compile-atom ======
compile-f:  <function compile_symbol at 0x10d49c848>
atom:  a
```

compile-assign (setv ld-name (.compile self name))

```
atom result 1:  {'nodes': [u'a']}
```

In the between process, we rename the temp-vars _hua_anon_var_2 to name instead of assign the result to name. Compare the temp-vars before and after

```
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'nodes': [u'a']}
result before addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': None}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': False, '_expr': <class Id a at 0x10d3f5cd0>}
result after addition: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id a at 0x10d3f5cd0>}
```

compile-assign (+= result ld-name)

```
result before addition: {'stmts': [<class Local nodes: [[<class Id a at 0x10d3f5d90>], []] at 0x10d3f5fd0>, <class If nodes: [<class Number 1 at 0x10d3f5c90>, [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at
0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id
[<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>, <class Set nodes: [[<class Id a at 0x10d3f5d90>], [<class
Id _hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>], [<class Set nodes: [[<class Id a at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>]] at 0x10d3f5f50>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id a at 0x10d3f5710>}
result to be added: {'stmts': [], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id a at 0x10d3f5cd0>}
result after addition: {'stmts': [<class Local nodes: [[<class Id a at 0x10d3f5d90>], []] at 0x10d3f5fd0>, <class If nodes: [<class Number 1 at 0x10d3f5c90>, [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at
0x10d3f5510>, [<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id
[<class Id d at 0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>, <class Set nodes: [[<class Id a at 0x10d3f5d90>], [<class
Id _hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>], [<class Set nodes: [[<class Id a at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>]] at 0x10d3f5f50>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id a at 0x10d3f5cd0>}


atom result 1:  {'stmts': [<class Local nodes: [[<class Id a at 0x10d3f5d90>], []] at 0x10d3f5fd0>, <class If nodes: [<class Number 1 at 0x10d3f5c90>, [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at 0x10d3f5510>,
[<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at
0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>, <class Set nodes: [[<class Id a at 0x10d3f5d90>], [<class Id
_hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>], [<class Set nodes: [[<class Id a at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>]] at 0x10d3f5f50>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id a at 0x10d3f5cd0>}


atom result 1:  {'stmts': [<class Local nodes: [[<class Id a at 0x10d3f5d90>], []] at 0x10d3f5fd0>, <class If nodes: [<class Number 1 at 0x10d3f5c90>, [<class Local nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], []] at 0x10d3f5990>, <class If nodes: [<class Id True at 0x10d3f5510>,
[<class Set nodes: [<class Id [<class Id c at 0x10d3f53d0>] at 0x10d3f5ed0>, [<class Number 2 at 0x10d3f5150>]] at 0x10d3f5750>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id c at 0x10d3f53d0>]] at 0x10d3f5790>], [<class Set nodes: [<class Id [<class Id d at
0x10d3f59d0>] at 0x10d3f5e90>, [<class Number 4 at 0x10d3f5e10>]] at 0x10d3f5bd0>, <class Set nodes: [[<class Id _hua_anon_var_1 at 0x10d3f5690>], [<class Id d at 0x10d3f59d0>]] at 0x10d3f5b90>]] at 0x10d3f5950>, <class Set nodes: [[<class Id a at 0x10d3f5d90>], [<class Id
_hua_anon_var_1 at 0x10d3f5850>]] at 0x10d3f5650>], [<class Set nodes: [[<class Id a at 0x10d3f5d90>], [<class Number 3 at 0x10d3f5610>]] at 0x10d3f5190>]] at 0x10d3f5f50>], 'temp_vars': [], '_Result__used_expr': True, '_expr': <class Id a at 0x10d3f5cd0>}


local a

if 1 then
   local _hua_anon_var_1
   if True then
```

```
        c = 2
        _hua_anon_var_1 = c
    else
        d = 4
        _hua_anon_var_1 = d
    end
    a = _hua_anon_var_1
else
    a = 3
end
```