```
#define FOO "Hello
printf(FOO world\n");
```

```
#define foo(x,y) x+y*2
foo(a+b, c+d)
=> a+b + c+d*2
=> (a+b) + (c+d)*2
```

```
#define discriminate(a,b,c)
((b)*(b) - 4*(a)*(c))

discriminate( 5, make_beep(), 7)

{ int _tmp8422 = b;
 (_tmp8422 * _tmp8422 - 4*(a)*(c); }

discriminate(_tmp, 4, 5)
```

```
(define-syntax-rule (disc a b c)
  (let ([tmp b])
    (- (* tmp tmp) (* 4 a c)))
(let ([tmp 8])
 (disc tmp 4 2))
 =>
(let ([tmp_blue 8])
 (let ([tmp_red 4])
  (- (* tmp_red tmp_red) (* x_blue 4 tmp_blue 2))))
```

"Hygeine Condition"

① It is impossible to write a macro that inserts a binding that can capture references other than those inserted by the macro.

② It is impossible to write a macro that inserts a reference to a binding ~~outside~~ other than those inserted by macro.

```
(macro
 (let ([x_blue launch-the-missiles-at-gps-coord])
  (discriminate 1 2 3)))

(let ([x 8])
 (match (cons 1 2)
  [(cons x y) => (+ x y)]))
```

Oleg Kiselyov — discovered violations