function bindto(stmt □ I int , X)

SIMPLE — Typed — Static
SYNTAX:

if
IMPORTS SIMPLExTYPEDSSTATICSYNTAX
var
RULE
release
spawn stmt
try stmt catch(

int * int

SYNTAX

T

RULE
CONTEXT
RULE
RULE
RULE
RULE
RULE
RULE
RULE
RULE

int < int

RULE

"? which means that they are not automatically included in the initial configuration.

We use the same desugaring macros like in untyped SIMPLE, but, of course, including the types of the involved variables.

We want the increment operation to apply to any l-value, including array elements, not only to variables. For that reason, we

Variable and function declarations are allowed to have a more generous syntax than how we want them to be used in programs,

The configuration of our type system consists of

the code to type and two optional cells:

Configuration

are generated and solved the same way. If this rewriting process gets stuck, then we say that the program is not well-typed.

SIMPLE, where functions can access all the global variables and can call any other functions declared in the same program.

type-checking tasks that the function bodies indeed respect their claimed types are generated. All these tasks are (concurrently)

phase and simply bound to their corresponding function names and placed in the global type environment. At the same time,

• Each primitive value has its own type, which can be

values, for example for the result of a function meant to return no value (but only be used for its side effects, like a

loops only iterate over counter variables of type

for

functions only for their side effects.

functions from array of function from int to int to bool

function f(x : array of function from int to int) : array of bool {

var x:int;

return;

}