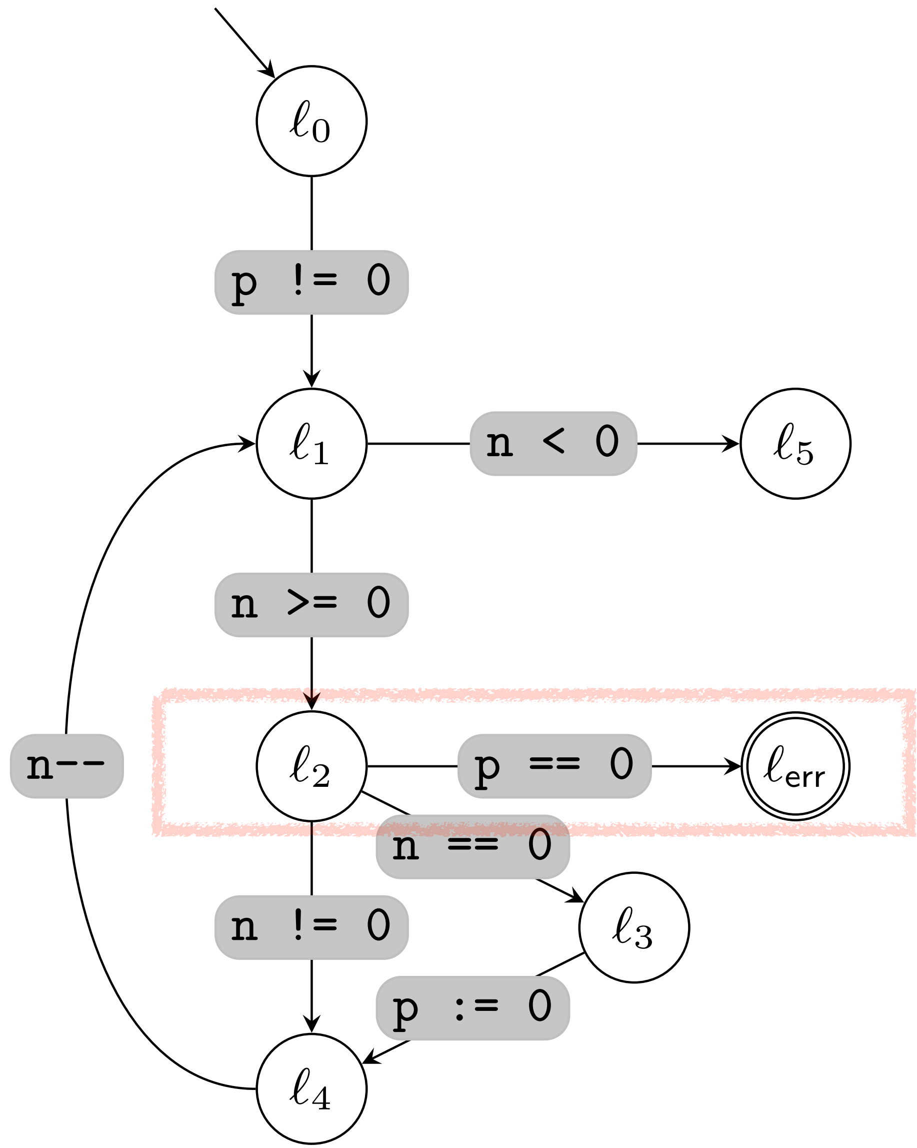
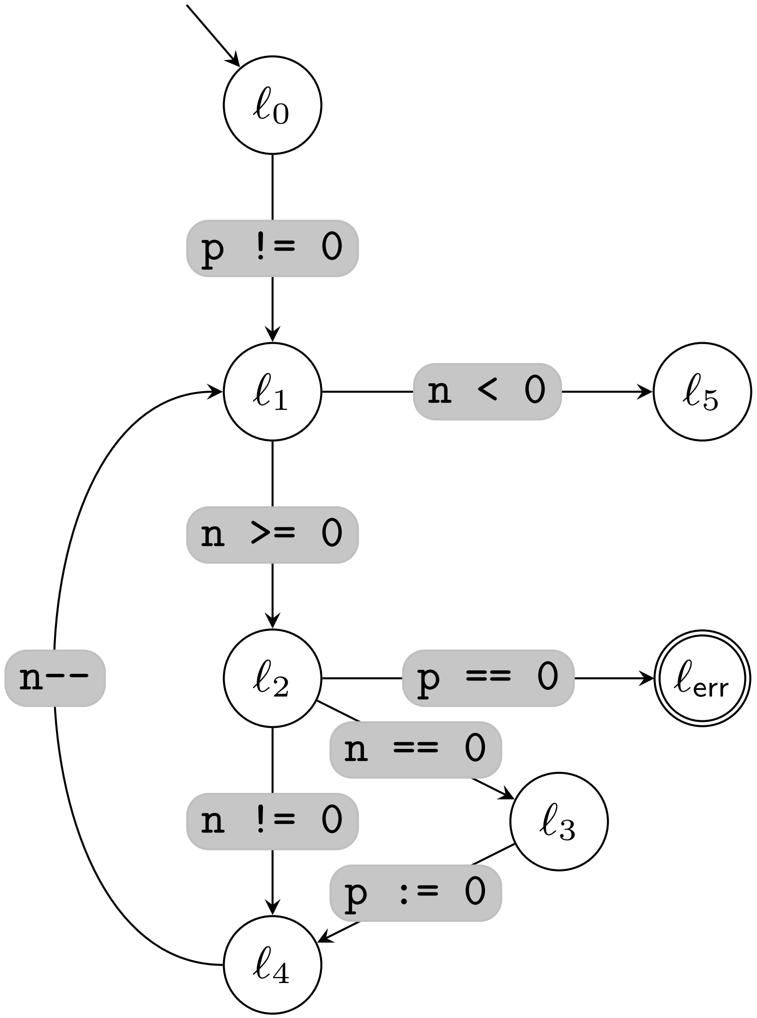
trace,
correct trace,
control flow trace,
program,
correct program

in formal setting based on automata
and regular languages

```
ℓ0: assume p != 0;

ℓ1: while(n >= 0)
    {
ℓ2:     assert p != 0;

        if(n == 0)
        {
ℓ3:         p := 0;
        }
ℓ4:     n--;
    }

ℓ5:
```



no execution violates assertion = no execution reaches error location



automaton

alphabet: {statements}

if program is correct
then
set of correct control flow traces is regular

if program is not correct
then
set of correct control flow traces is non-regular

set of correct traces is non-regular
set of execution traces is non-regular
set of correct execution traces is non-regular

fixed set of statements Σ

trace τ = sequence of statements = word over Σ

$$\tau = st_1 \ldots st_n$$

$$\{\text{traces}\} = \Sigma^{\star}$$

$$\{\text{correct traces}\} = \{\tau \in \Sigma^\star \mid \{\varphi_{\mathsf{pre}}\}\ \tau\ \{\varphi_{\mathsf{post}}\} \text{ is valid}\}$$

for trace $\tau = st_1 \ldots st_n$

$$\{\varphi\}\ \tau\ \{\psi\}$$

if

$$\{\varphi\}\ st_1\ \{\varphi_1\},\ \{\varphi_1\}\ st_2\ \{\varphi_2\},\ \ldots, \{\varphi_{n-1}\}\ st\ \{\psi\}$$

program = control flow graph

nodes (“locations”)
edges labeled by statements
initial location
exit locations

program = control flow graph = automaton

nodes ("locations")
edges labeled by statements
initial location
exit locations

$$\mathcal{P} = (\mathsf{Loc}, \delta, \ell_0, F)$$

$$\delta \subseteq \mathsf{Loc} \times \Sigma \times \mathsf{Loc}$$

$$\mathcal{P} = (\mathsf{Loc}, \delta, \ell_0, F)$$

$$\mathcal{L}(\mathcal{P}) \subseteq \Sigma^\star$$

$$\{\text{control flow traces}\} = \mathcal{L}(\mathcal{P})$$

correctness of program $\mathcal{P}$ via inclusion

$$\{\varphi_{\mathsf{pre}}\}\ \mathcal{P}\ \{\varphi_{\mathsf{post}}\}$$

if

$$\{\text{control flow traces}\} \subseteq \{\text{correct traces}\}$$

correctness of program $\mathcal{P}$ via inclusion

$$\{\varphi_{\mathsf{pre}}\}\ \mathcal{P}\ \{\varphi_{\mathsf{post}}\}$$

if

$$\{\text{control flow traces}\} \subseteq \{\text{correct traces}\}$$

$$\mathcal{L}(\mathcal{P}) \subseteq \{\tau \in \Sigma^\star \mid \{\varphi_{\mathsf{pre}}\}\ \tau\ \{\varphi_{\mathsf{post}}\} \text{ is valid}\}$$

program correct

if

$$\{\text{control flow traces}\} \subseteq \mathcal{L}(\mathcal{A}) \subseteq \{\text{correct traces}\}$$

program correct

if

$$\{\text{control flow traces}\} \subseteq \mathcal{L}(\mathcal{A}) \subseteq \{\text{correct traces}\}$$

proof rule:
find a regular subset of correct traces
that is large enough to contain all control flow traces

infeasible traces

```
x==0. x==1
```

```
x:=0. x==1
```

infeasibility $\implies$ correctness

trace τ infeasible: $\{true\}\ \tau\ \{false\}$

execution trace: feasible control flow trace

regular set of control flow traces =
abstraction of non-regular set of execution traces

$\boxed{\text{abstraction does not introduce incorrect traces}}$

if program is correct
then
set of correct control flow traces is regular

if program is correct
then
set of correct control flow traces is regular

because ...
if program is correct
then
set of correct control flow traces = set of all control flow traces

if program is correct
then
set of correct control flow traces is regular

if program is not correct
then
set of correct control flow traces is in general non-regular