#12: Constraint Solving

Sankha Narayan Guria

EECS 700: Introduction to Program Synthesis

Why do we care?

1. Synthesis is combinatorial search, and so is SAT/SMT
2. SAT/SMT solvers are really good these days
3. ???   ⟵——— this week
4. Profit!!!

Boolean SATisfiability

gin ∨ tonic

Solution:

minor ↦ T
gin ↦ F
tonic ↦ T

Satisfiability Modulo Theories

(gin ∨ tonic) ∧ (age < 21 ⇒ abv = 0) ∧ (age = 20)

In the United States, "gin" is defined as an alcoholic beverage of no less than 40% ABV...
Wikipedia

Satisfiability Modulo Theories

(gin ∨ tonic) ∧ (age < 21 ⇒ abv = 0) ∧ (age = 20) ∧ (gin ⇒ abv ≥ 40)

theory of Linear Integer Arithmetic

age ↦ 20
abv ↦ 0
gin ↦ F
tonic ↦ T

Popular Solvers

Microsoft

Stanford

SRI

JKU Linz, Austria

Z3

CVC5

Yices2

Boolector

SMT competition: http://smtcomp.sourceforge.net

```
.smt2                // SMTLib format

(declare-fun age () Int)
(declare-fun abv () Int)
```
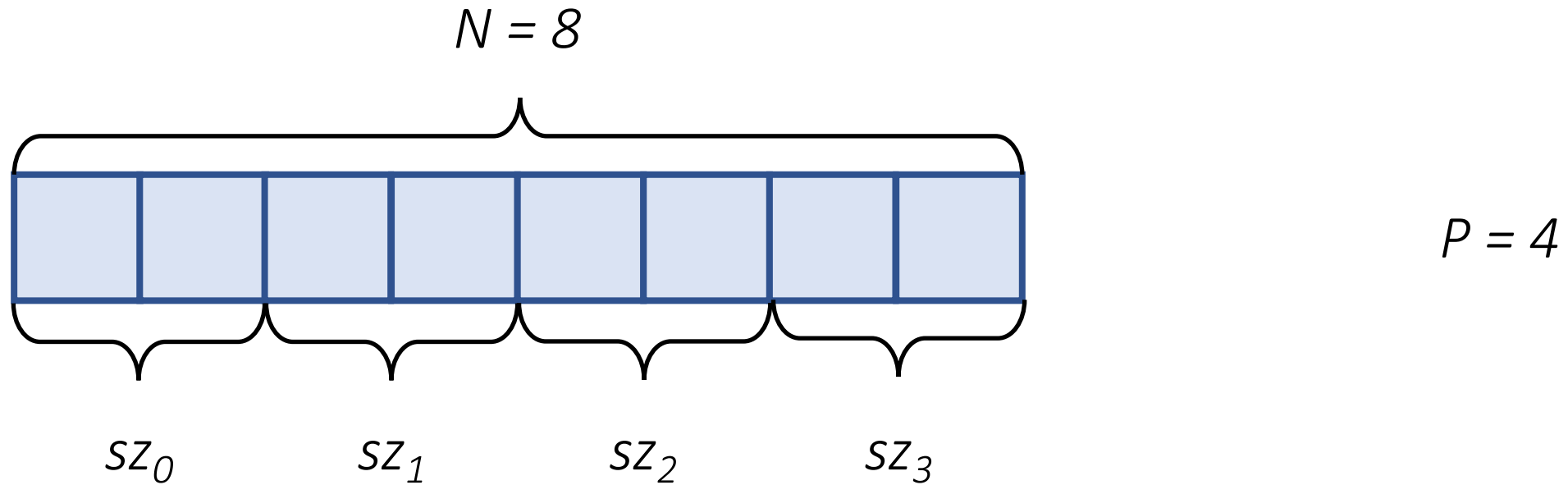
SMT-LIB

Uniform format for SMT problems understood by all solvers

```
(declare-fun age () Int)
(declare-fun abv () Int)
(declare-fun gin () Bool)
(declare-fun tonic () Bool)
(assert (or gin tonic))
(assert (implies (< age 21) (= abv 0)))
(assert (= age 20))
(assert (implies gin (>= abv 40)))
(check-sat)
(get-model)
```
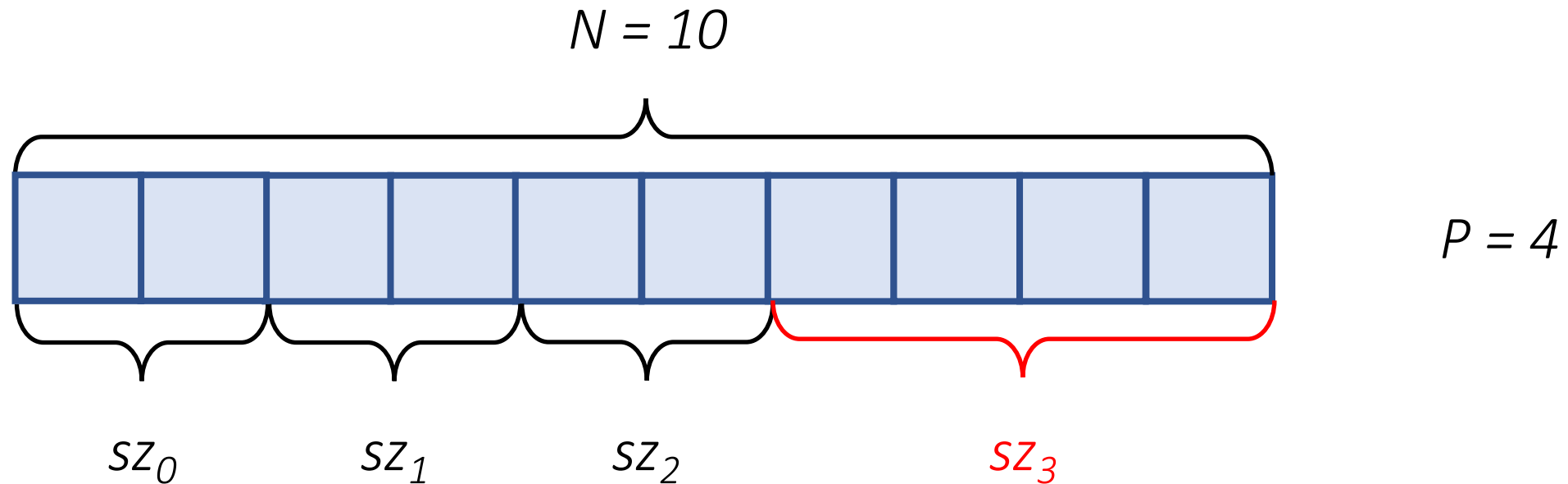
Problem: Array Partitioning
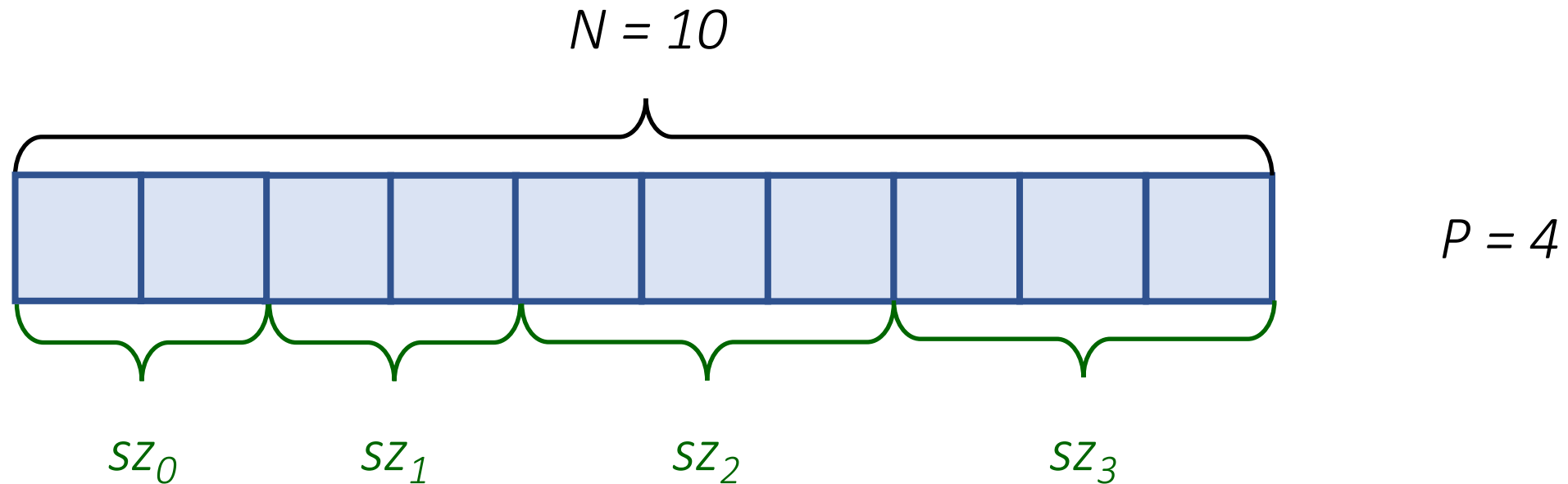
Partition an array of size N evenly into P sub-ranges

$$N = 8$$

$P = 4$

sz_0 sz_1 sz_2 sz_3

Problem: Array Partitioning

Partition an array of size N evenly into P sub-ranges

$$N = 10$$

$$P = 4$$

sz_0 sz_1 sz_2 sz_3

Problem: Array Partitioning

Partition an array of size *N* evenly into *P* sub-ranges

$N = 10$

$P = 4$

sz_0 sz_1 sz_2 sz_3

Can we always make them differ by at most 1?

to the rescue!

code: https://github.com/nadia-polikarpova/smt-talk

Let's generalize this into a program!

$N = 10$

$P = 4$

sz_0 sz_1 sz_2 sz_3

K $N - K$
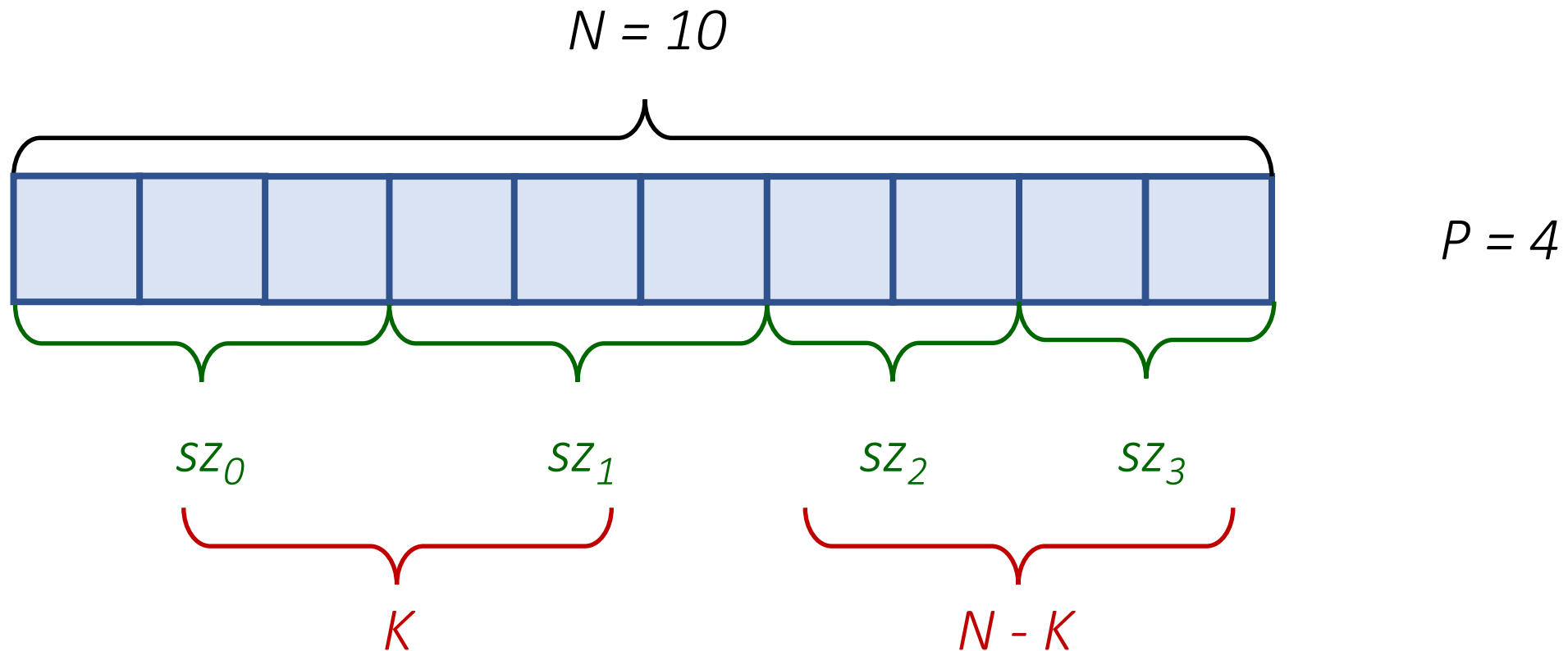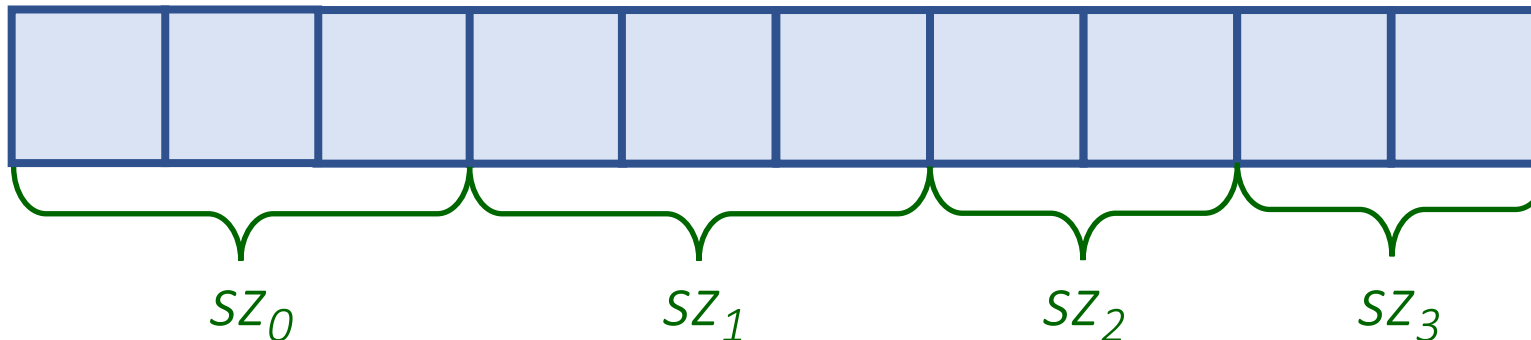
A program for partitioning

```
for i in range(P):
    if i < K:
        sz[i] = n/P + 1
    else:
        sz[i] = n/P
```

I want this program to work for all *n*!
What should my *K* be?

sz_0 sz_1 sz_2 sz_3

A program for partitioning

```
for i in range(P):
    if i < K:
        sz[i] = n/P + 1
    else:
        sz[i] = n/P
```

How do I prove that this program works for all *n*?

sz_0 sz_1 sz_2 sz_3
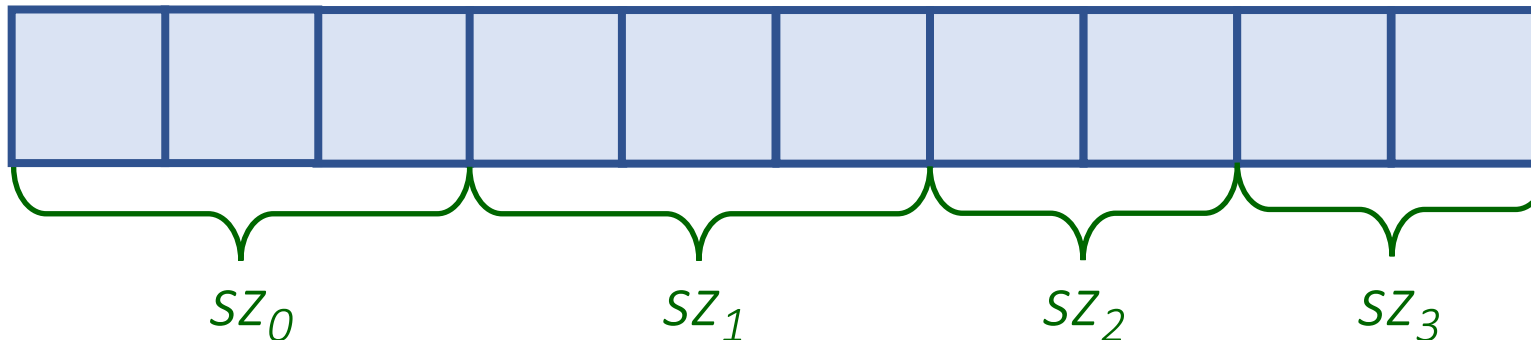
Verification with SMT

```
for i in range(P):
    if i < K:
        sz[i] = n/P + 1
    else:
        sz[i] = n/P
```

want: prove $\quad \forall n. spec(n)$

$\forall n. \sum sz = n$

have: solve $\quad \exists n. prop(n)$

idea: solve for counter-examples!

$\exists n. \neg spec(n)$

$\exists n. \sum sz \neq n$