Using the abstraction from 3.4 Concrete Parsing Semantics

With the two parse stack transition functions, we compute the resulting parse stacks for the code fragments `p` or `c`. Specifically, `x = e_0`, `y = e_1`, and `z = x`, respectively. Then we have `⌜e_0, e_1⌝ = 2`. Parse stack `P_0` is `c\_\Sigma`.

Let `f` be an invariant. We can derive `⌜f\_\Sigma\⌝ = 20.1` for the initial parse stack. We cannot directly compute `⌜f\_\Sigma\⌝ = 2.47.2` for a counter example. If not, the teacher returns a truth assignment as a fix. The random answer does not break soundness, because we always have a fix.

An invariant.

Example:

```c
// Program

int h(n) { ret n; }
void f(a) {
  c1: x = h(a);
  assert(x > 1);
  c2: y = h(input());
  assert(y > 1);
}
void m() {
  c3: void g() { f(8); }
  c4: f(4);
  c5: g();
  c6: g();
}
```

Asks `⌜\Sigma\⌝ = \{ \} `.

### 2. Powerset domain of concrete parse stack

The powerset domain of concrete parse stacks is extended of 4.2 Parameterized Framework.

For the given program, we can establish the Galois connection between code fragments and concrete parse stack. We cannot directly compute the Galois connection.

Under Approximation

A.

B.

C.

# 3. Deriving Invariants in Propositional Logic

hexadecimal approach: impact pre-analysis. (Impact pre-analysis)

### Selective X-sensitivity

k-CFA: +25% / -1300%

Example: Selective X-sensitivity

```c
// Program

int h(n) { ret n; }
void f(a) {
  c1: x = h(a);
  assert(x > 1);
  c2: y = h(input());
  assert(y > 1);
}
void m() {
  c3: void g() { f(8); }
  c4: f(4);
  c5: g();
  c6: g();
}
```

Asks `⌜\Sigma\⌝ = \{ \} `.

### 2. Powerset domain of concrete parse stack

The powerset domain of concrete parse stacks is extended of 4.2 Parameterized Framework.

For the given program, we can establish the Galois connection between code fragments and concrete parse stack. We cannot directly compute the Galois connection.

Under Approximation

A.

B.

C.

# 3. Deriving Invariants in Propositional Logic

hexadecimal approach: impact pre-analysis. (Impact pre-analysis)

### Selective X-sensitivity

k-CFA: +25% / -1300%

Example: Selective X-sensitivity

```c
// Program

int h(n) { ret n; }
void f(a) {
  c1: x = h(a);
  assert(x > 1);
  c2: y = h(input());
  assert(y > 1);
}
void m() {
  c3: void g() { f(8); }
  c4: f(4);
  c5: g();
  c6: g();
}
```

Asks `⌜\Sigma\⌝ = \{ \} `.

### 2. Powerset domain of concrete parse stack

The powerset domain of concrete parse stacks is extended of 4.2 Parameterized Framework.

For the given program, we can establish the Galois connection between code fragments and concrete parse stack. We cannot directly compute the Galois connection.

Under Approximation

A.

B.

C.