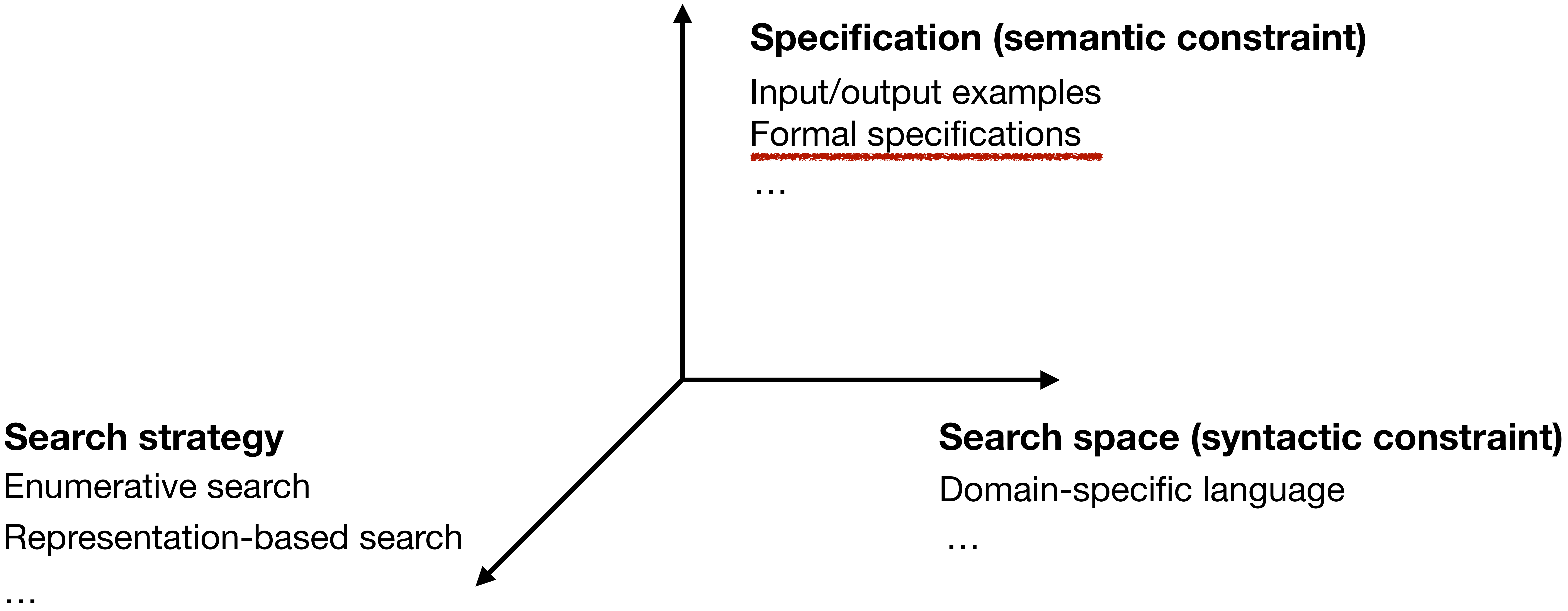
Advanced Software Security

7. Program Synthesis and Verification

Kihong Heo

KAIST

Dimensions in Program Synthesis



Specification (semantic constraint)

Input/output examples
Formal specifications
…

Search strategy

Enumerative search
Representation-based search
…

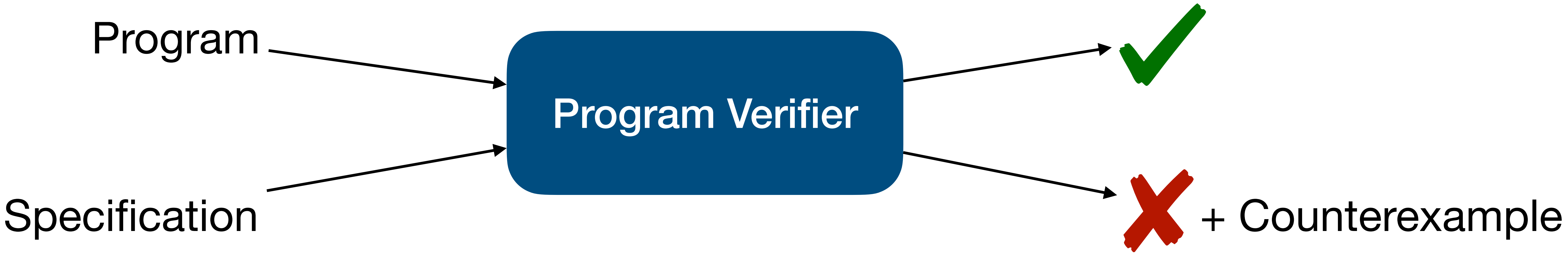
Search space (syntactic constraint)

Domain-specific language
…

Functional Synthesis

- Goal: find a function that satisfies a formal specification (i.e., logical formula)
  - Pre-condition: a predicate that all valid inputs to a function must satisfy
  - Post-condition: a post-condition that all outputs must satisfy
- Question: how can make a functional synthesizer using an inductive synthesizer?

(Hint)



Example

Specification

Find a function $f(x)$ where $\forall x, y \,.\, f(x,y) \geq x \wedge f(x,y) \geq y \wedge \big(f(x,y) = x \vee f(x,y) = y\big)$

Grammar

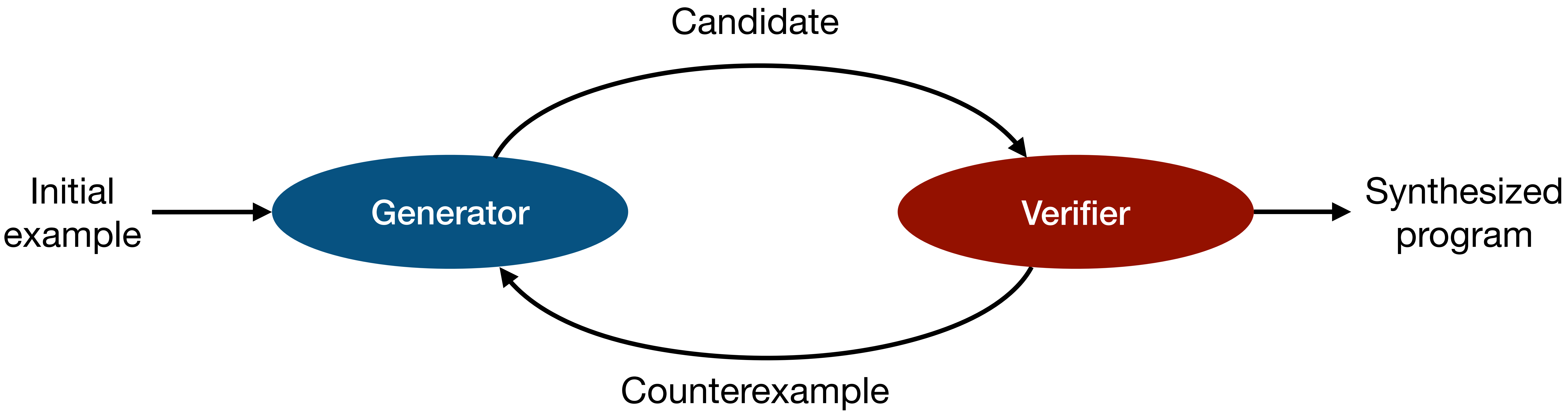
$$S \rightarrow x \mid y \mid S + S \mid S - S \mid \texttt{if}\ B\ S\ S$$
$$B \rightarrow S \leq S \mid S = S$$

Example

```
if(x >= y) x y
```

CEGIS

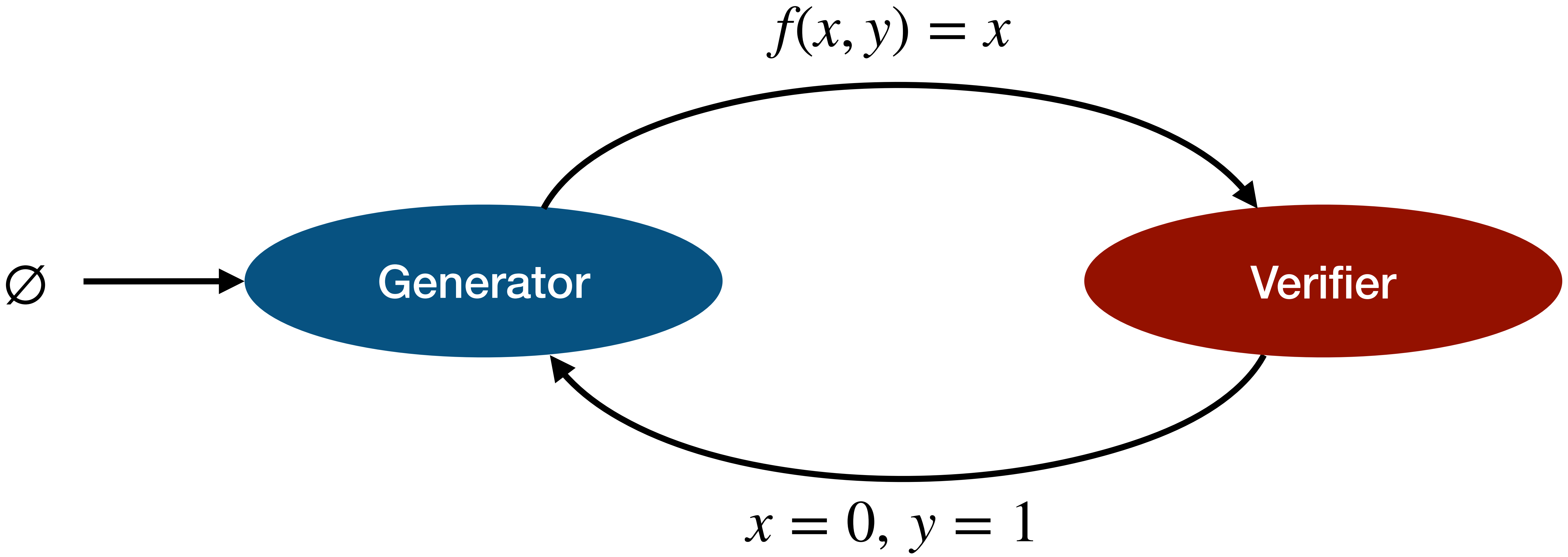
- CounterExample-Guided Inductive Synthesis
- A framework that enables us to use inductive synthesizers for functional synthesis
  - Generator: generate a candidate program (inductive synthesizer)
  - Verifier: check whether the candidate satisfies the specification (program verifier)



Example: CEGIS + Bottom-up Search

Specification

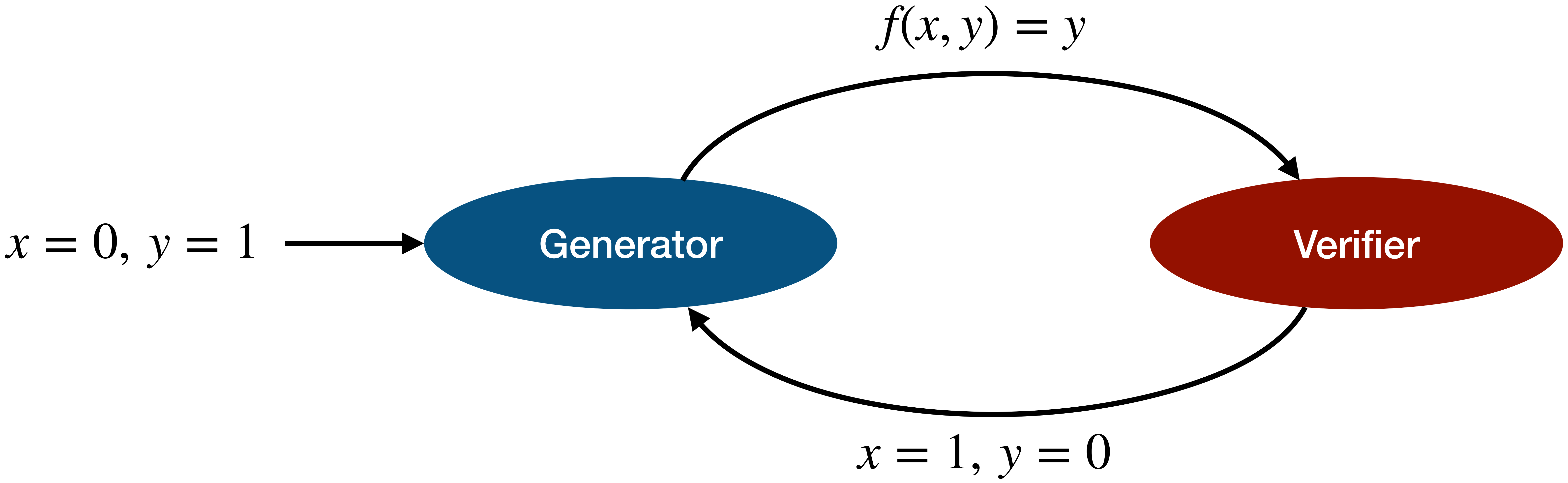
Find a function $f(x)$ where $\forall x, y \,.\, f(x,y) \geq x \wedge f(x,y) \geq y \wedge \big(f(x,y) = x \vee f(x,y) = y\big)$



Example: CEGIS + Bottom-up Search

Specification

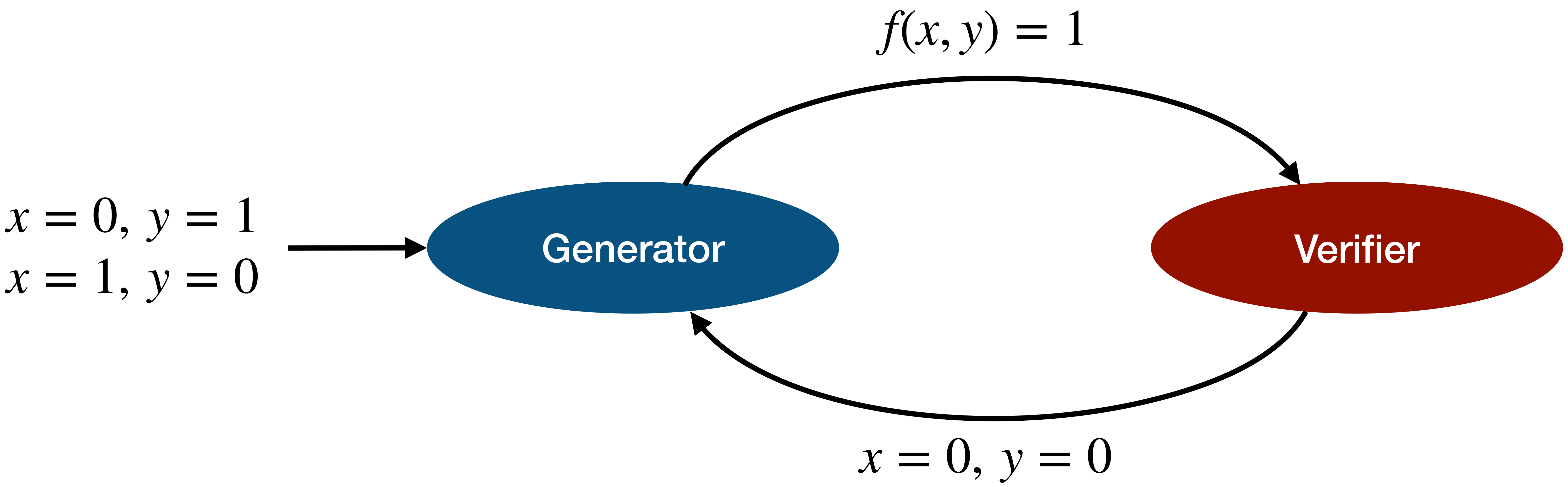
Find a function $f(x)$ where $\forall x, y \,.\, f(x,y) \geq x \wedge f(x,y) \geq y \wedge \big(f(x,y) = x \vee f(x,y) = y\big)$



Example: CEGIS + Bottom-up Search

Specification

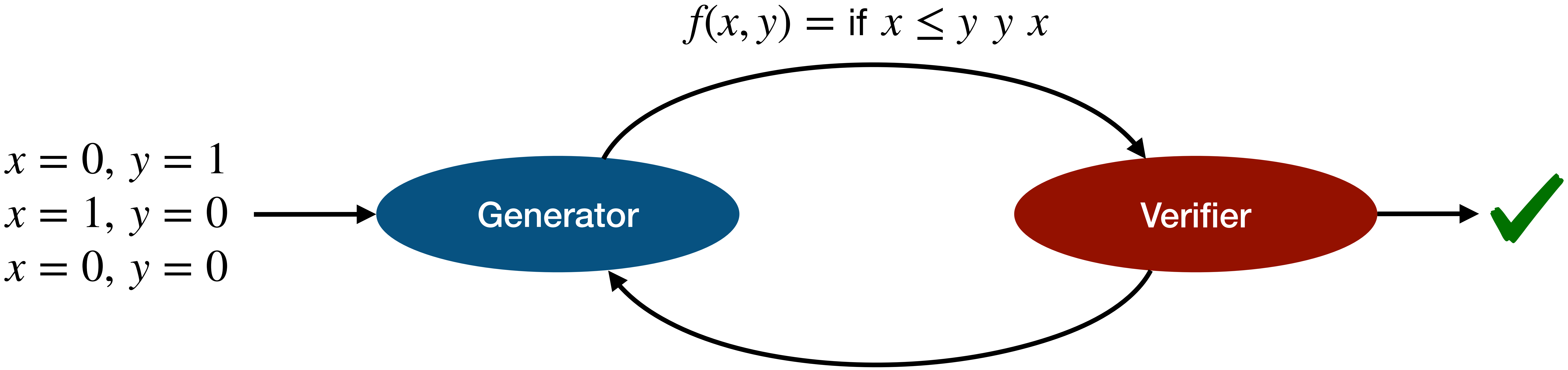
Find a function $f(x)$ where $\forall x, y\,.\, f(x,y) \geq x \land f(x,y) \geq y \land \big(f(x,y) = x \lor f(x,y) = y\big)$



Example: CEGIS + Bottom-up Search

Specification

Find a function $f(x)$ where $\forall x, y \,.\, f(x,y) \geq x \wedge f(x,y) \geq y \wedge \big(f(x,y) = x \vee f(x,y) = y\big)$



Program Verification via Theorem Proving

- Program verification: check whether a program satisfies a specification or not
- Theorem: “This program satisfies the specification”
- Proving: the correctness is verified if the theorem is proved
    - Otherwise, a counterexample is given
- Automated theorem provers: Z3, CVC, etc
- Interactive theorem provers (proof assistant): Coq, Isabelle/HOL, etc

A Real-World Usage of Theorem Provers

1. *N* students take IS593.
2. Each student will pick up and present one paper among *M* papers in the list provided by Prof. H.
3. Each student has different preferences on papers.
   (1) Each student has a ranking in his/her mind
   (2) If a student is assigned to the rank-1 paper, +5 happy pts
   (3) rank-2: +3 pts, rank-3: +1 pts, otherwise: 0 pts
4. Prof. H. is so kind that he wants to maximize the sum of happiness of all students.

How to solve this problem?
What is the assignment that makes everybody happy?

Solving

- $x_{ij} \in \{0,\ 1\}$: student *i* is assigned to paper *j*

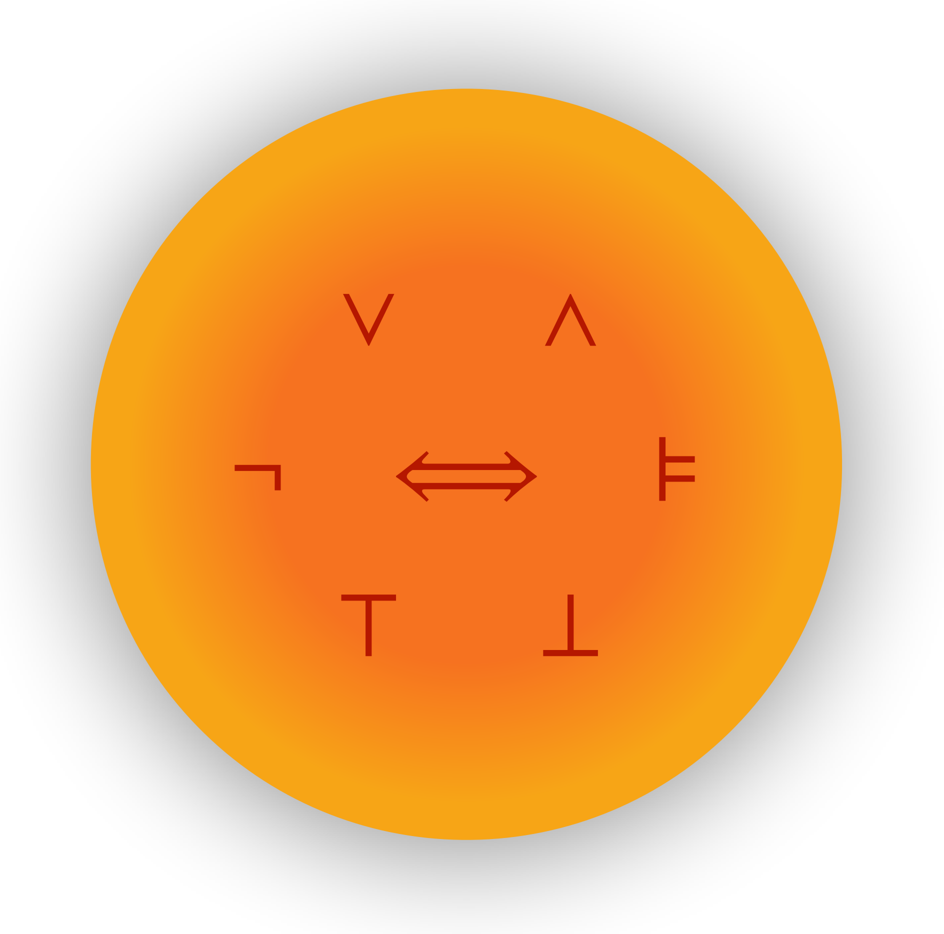
- $w_{ij} \in \{0,1,3,5\}$: happy points of student *i* on paper *j*

$$\forall i\,.\ \sum_{j=0}^{m-1} x_{ij} = 1 \qquad\qquad \forall j\,.\ 0 \leq \sum_{i=0}^{n-1} x_{ij} \leq 1$$

(One paper per student) (Maximum one student per paper)

Goal: find x that maximizes $\displaystyle\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} x_{ij} w_{ij}$

Dragon Ball Z3



- A Wish-granting Theorem Prover for Everyone (in our class)
  - Used to assign papers to you folks
  - A reference code for Z3
- Open-source: https://github.com/prosyslab-classroom/dragon-ball-z3