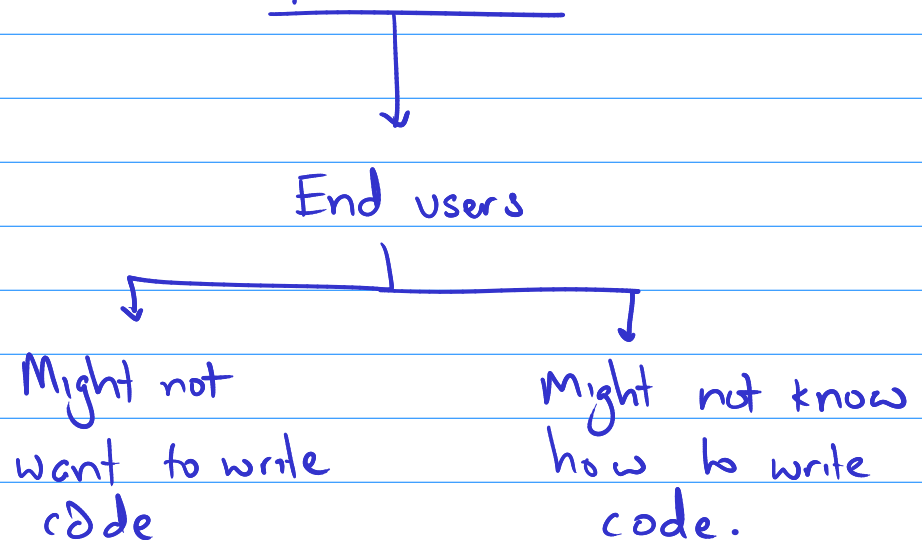


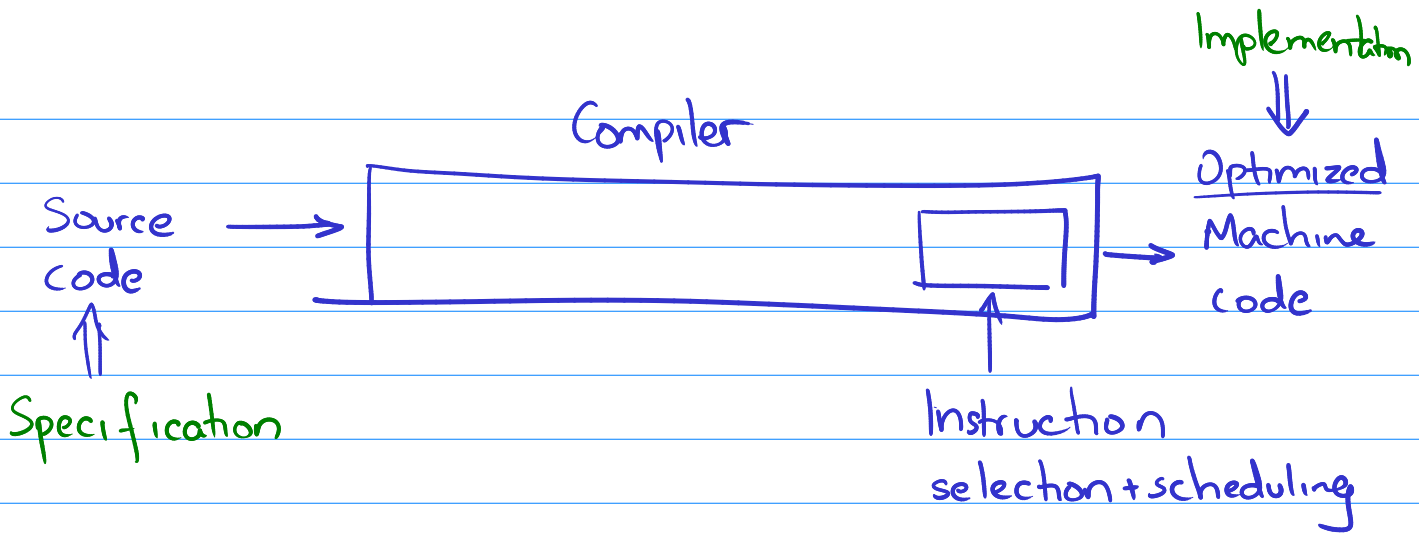
Lecture 12 Program Synthesis

Computer-Augmented Program Engineering

- How do we help the programmer in writing code?

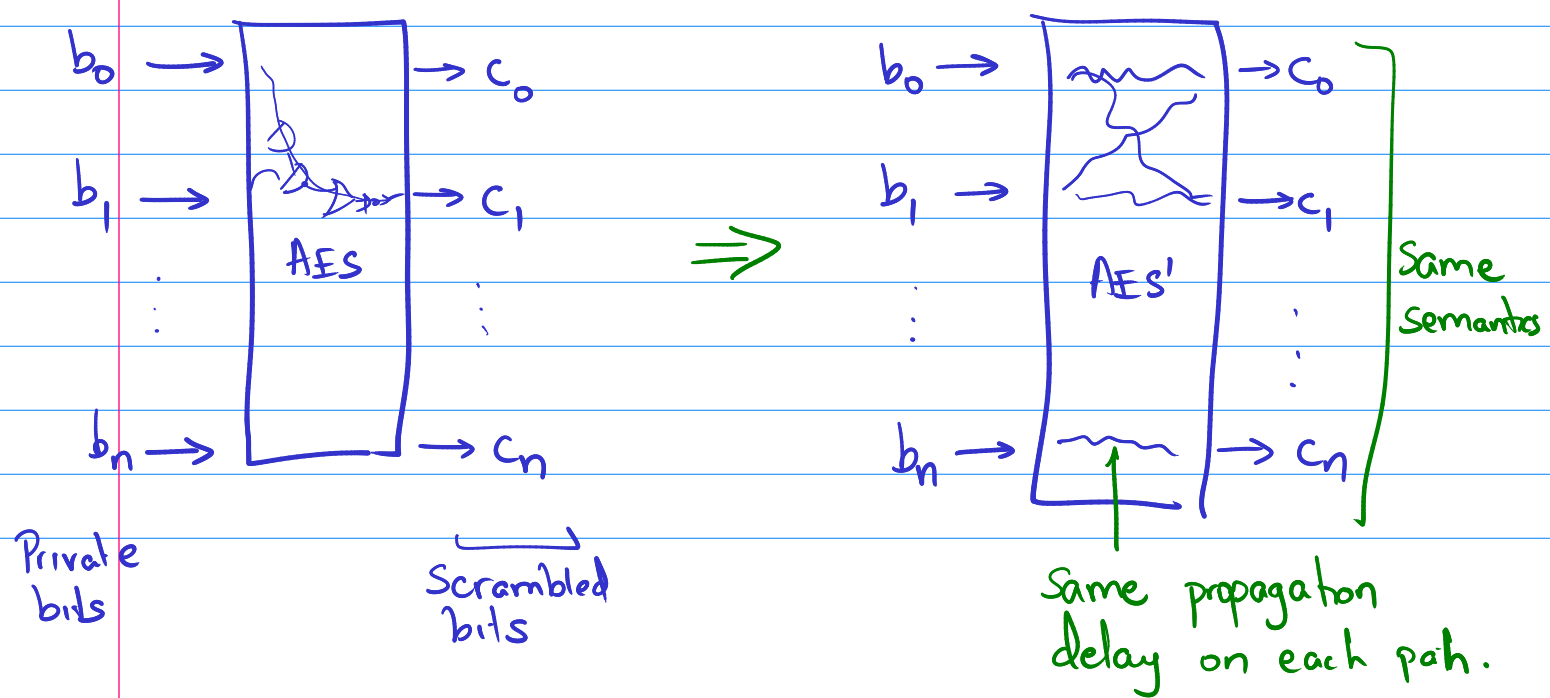


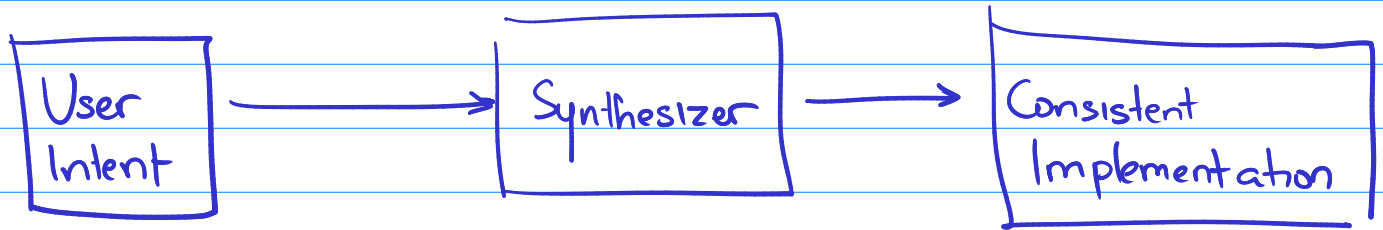
- Excel users, spreadsheet functions, VBScript macros
- Autograding for beginning programming courses
- Program repair



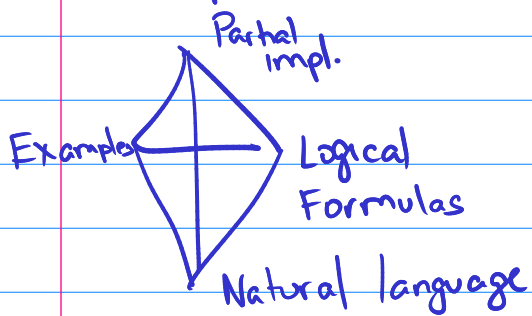
- Semantics have to be equivalent
- Final program must be extremely fast
- Compiler itself has to be fast.

- Can we automatically build secure systems?

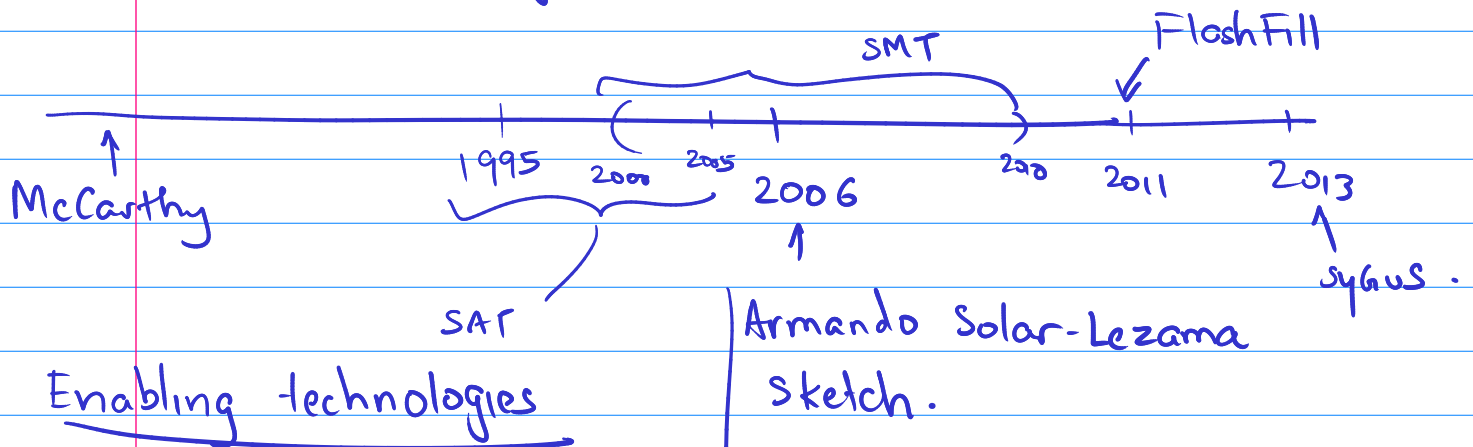




Multimodal
Specifications



Renaissance of Program Synthesis



- Mature constraint solvers
- Growth in CPU power
- Better algorithms
- Better HCI

```

int x = 0;
for (int i = 0; i < 22; i++) {
    x = x + a[i]
}

```

assert (x == a[0] + a[1]);

```

int x = input();
int y = x << 11;
assert (y == x + x);

```

$\exists c.$

$\forall x \in \mathbb{N}$

$\forall y \in \mathbb{N}. \underline{y = x \ll c}$

$\Rightarrow \underline{y = x + x. ?}$

$\exists c$

$\forall x \in \mathbb{N}. \exists y \in \mathbb{N}. y = x \ll c$

$\Rightarrow y = x + x ?$

$\exists c.$

$\forall x. x \ll c = x + x ?$

Yifei

```

int x = input();
int y = x << 3;
assert (y == x + x);

```

$$\forall x. x \ll 3 = x + x? \dots$$

$$\forall x. \exists y. [y = x \ll 3 \Rightarrow y = x + x?]$$

$$\forall x. \exists y. y = x \ll 3 \text{ and } y = x + x?$$

$$x-1$$

$$4$$

Proposed.

$$4 = 5 \ll 3 \Rightarrow 4 = 5 \times 5$$

$$\forall x \in \mathbb{N} (x-1 = x \ll 3) \Rightarrow (x-1 = x+x)?$$

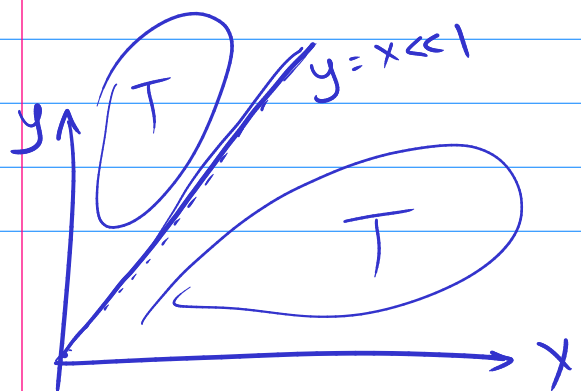
```

int x = input();
int y = x << 1;
assert (y = x + x);

```

$$\forall x$$

$$\forall y. y = x \ll 1 \text{ and } y = x + x?$$



$$\forall x$$

$$\forall y. y = x \ll 1 \Rightarrow y = x + x?$$

$\exists c. \forall x \forall y. y = x \ll c \Rightarrow y = x + x ?$

`int x = input`

`int y = x + x`

`int z = y << 1`

`assert (z = y + y)`

$\forall x.$

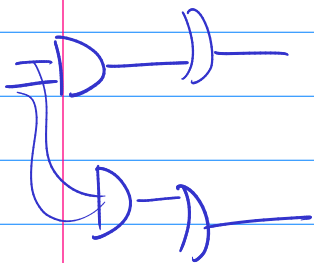
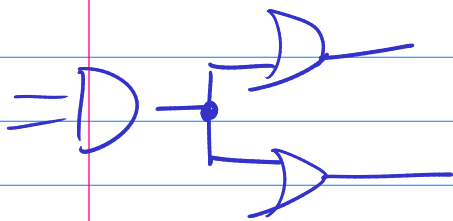
$((x+x) \ll 1) = \overset{\text{Inlining all computations}}{(x+x) + (x+x)}$

$\forall x. \forall y. \forall z.$

$y = x + x \Rightarrow$

$z = y \ll 1 \Rightarrow$

$z = y + y ?$

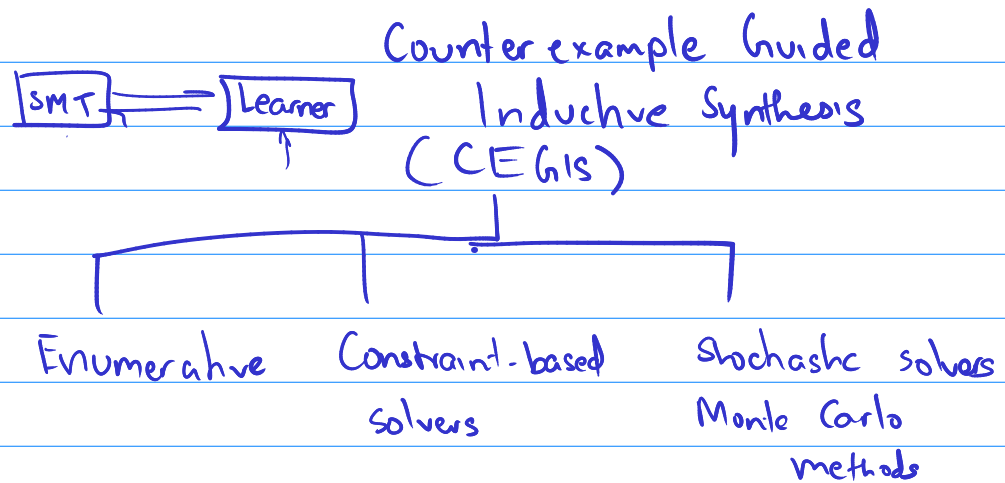


User Intent = Semantic spec + Syntactic spec

↑
What should it
do?

↑
How should it
look?

Algorithms for synthesis : Version spaces



- Deductive program synthesizers

Axioms + rewrite rules

Program Expression Graphs / Equality saturation