

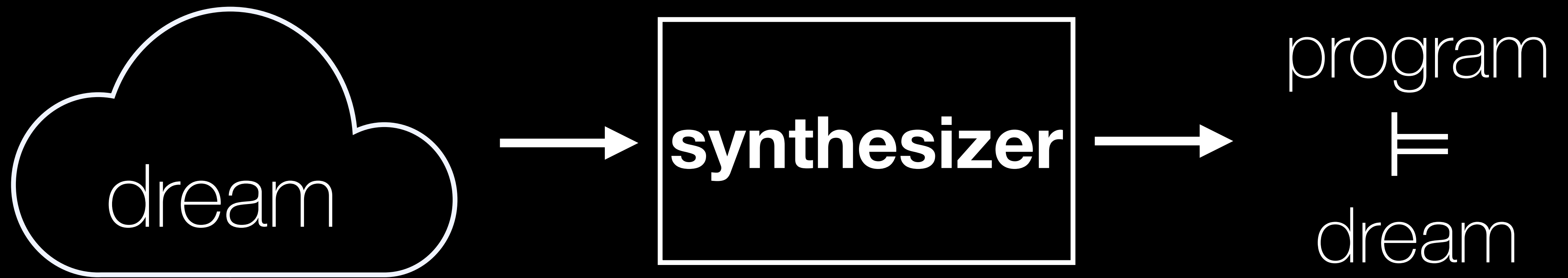
# a taste of synthesis

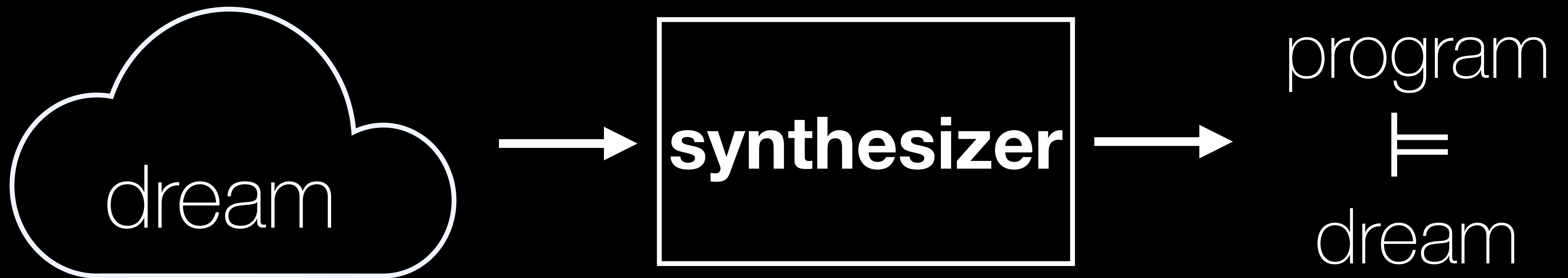
**aws albarghouthi** university of wisconsin–madison











## Synthesis: Dreams $\Rightarrow$ Programs

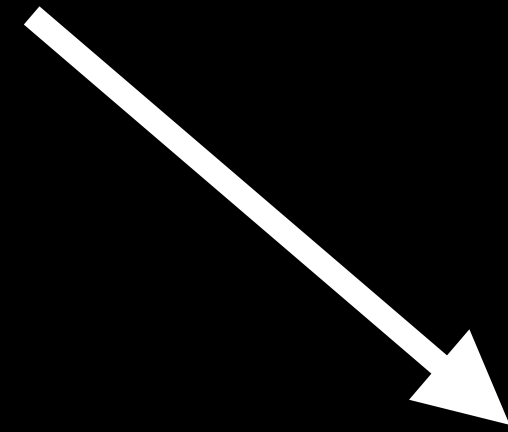
ZOHAR MANNA AND RICHARD WALDINGER

**synthesizer**



program

specification

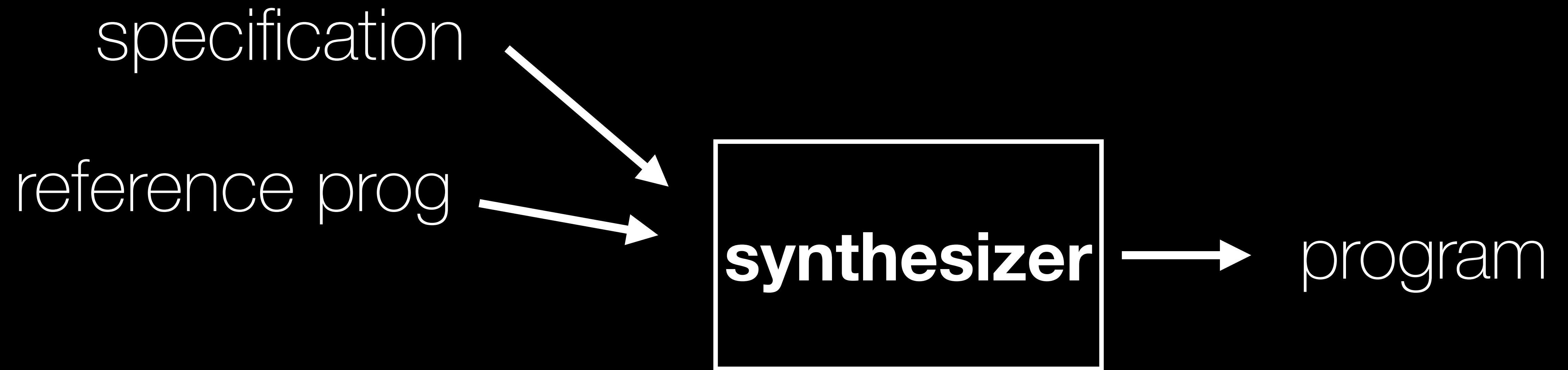


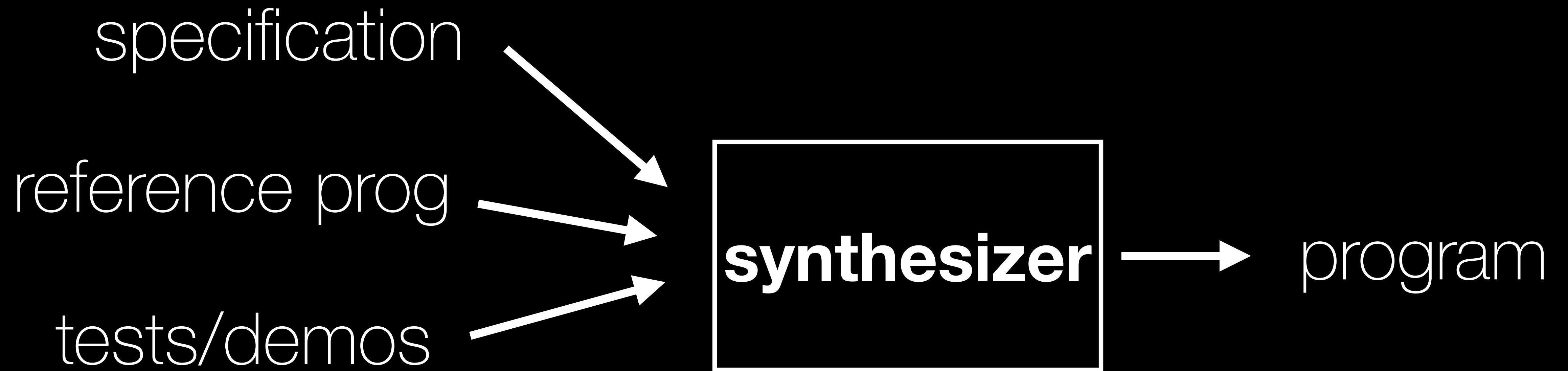
**synthesizer**

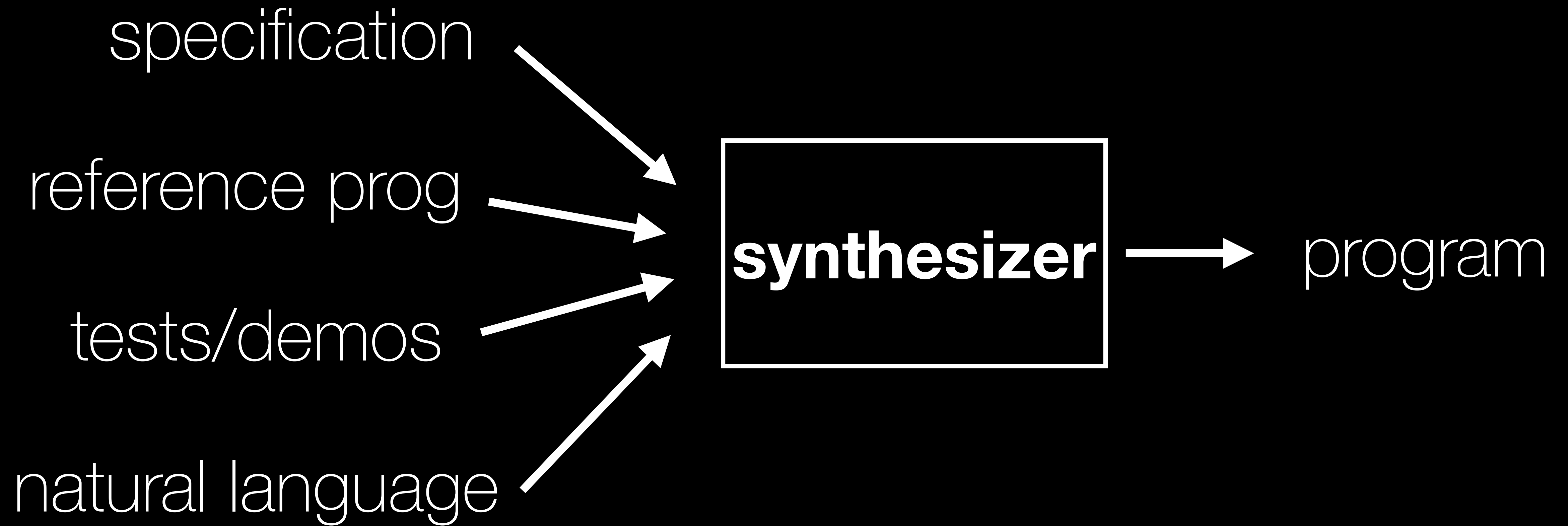


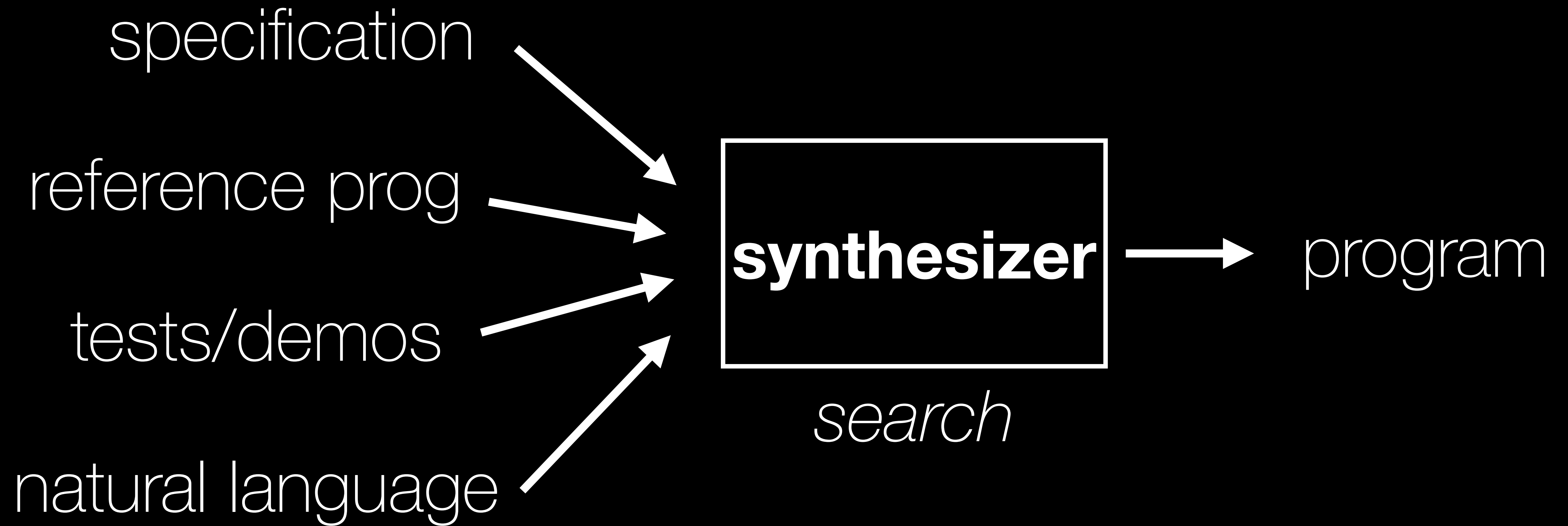
program

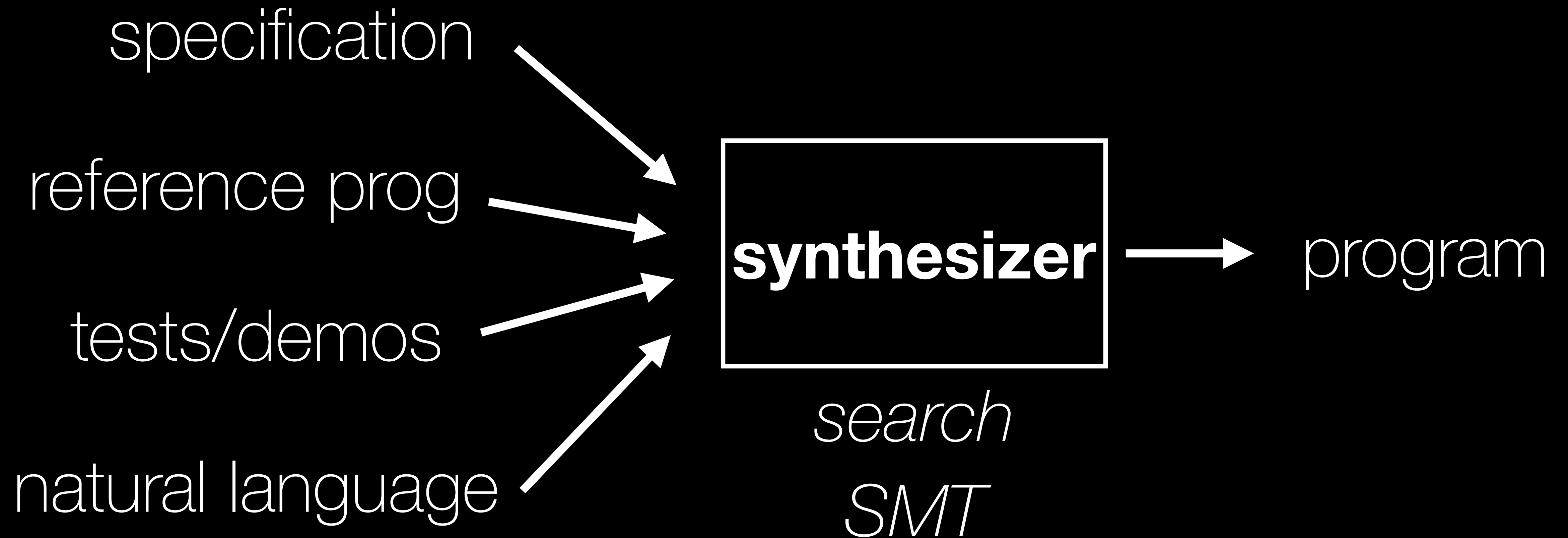


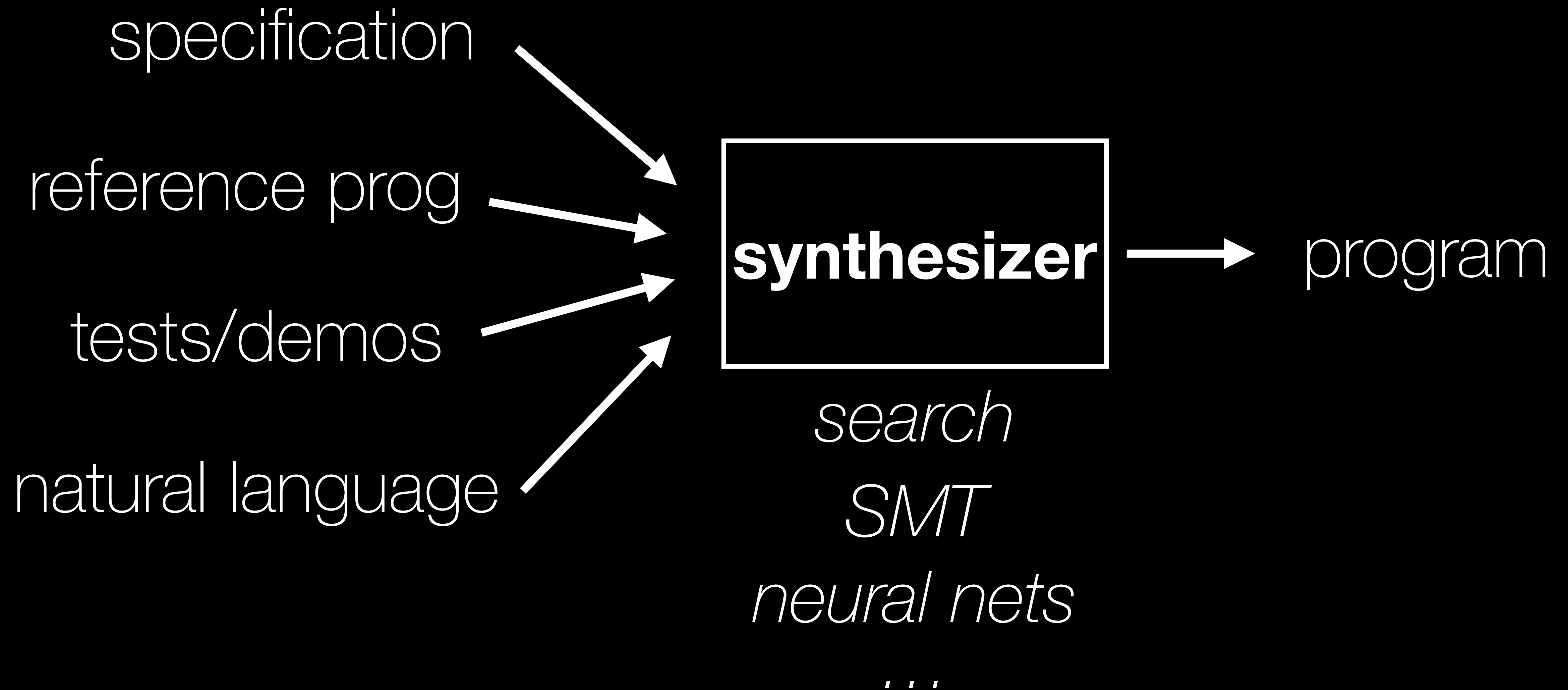












# reference implementation

```
int p_naive(int x):  
    y = x * 2  
    return y
```

# reference implementation

```
int p_naive(int x):  
    y = x * 2  
    return y
```

```
int p_smart(int x):  
    y = x << ??  
    return y
```



# reference implementation

```
int p_naive(int x):  
    y = x * 2  
    return y
```

≡

```
int p_smart(int x):  
    y = x << ??  
    return y
```

# reference implementation

```
int p_naive(int x):  
    y = x * 2  
    return y
```

≡

```
int p_smart(int x):  
    y = x << ??  
    return y
```

synthesizer replaces ?? with 1

# reference implementation

```
int p_naive(int x):  
    y = x * 2  
    return y
```

≡

```
int p_smart(int x):  
    y = x << ??  
    return y
```

synthesizer replaces ?? with 1

## Combinatorial Sketching for Finite Programs

Armando Solar-Lezama, Liviu Tancau, Rastislav Bodik, Vijay Saraswat\*, Sanjit Seshia

UC Berkeley

{asolar,tancau,bodik,sseshia}@eecs.berkeley.edu

\*IBM T.J. Watson Research Center

vsaraswa@us.ibm.com

# reference implementation

slow  $\equiv$  fast

# reference implementation

slow  $\equiv$  fast  
sequential  $\equiv$  parallel

# reference implementation

slow	≡	fast
sequential	≡	parallel
insecure	≡	secure

# reference implementation

slow	≡	fast
sequential	≡	parallel
insecure	≡	secure
precise	~	imprecise

# examples & demonstration

## **Automating String Processing in Spreadsheets Using Input-Output Examples**

Sumit Gulwani

Microsoft Research, Redmond, WA, USA



# examples & demonstrations





# examples & demonstrations





# examples & demonstrations





# examples & demonstrations

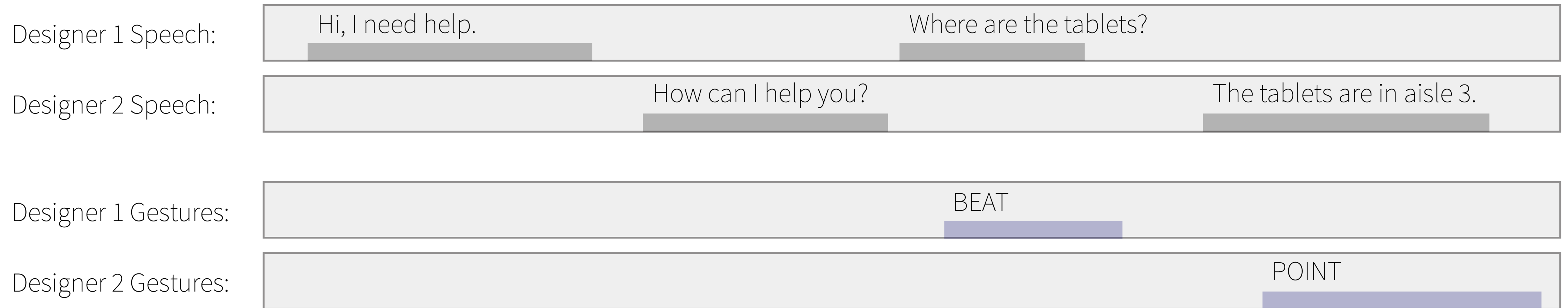




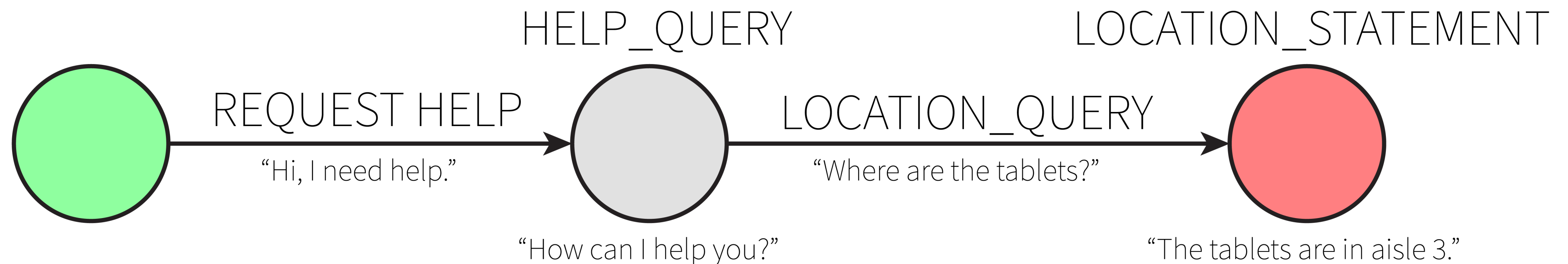
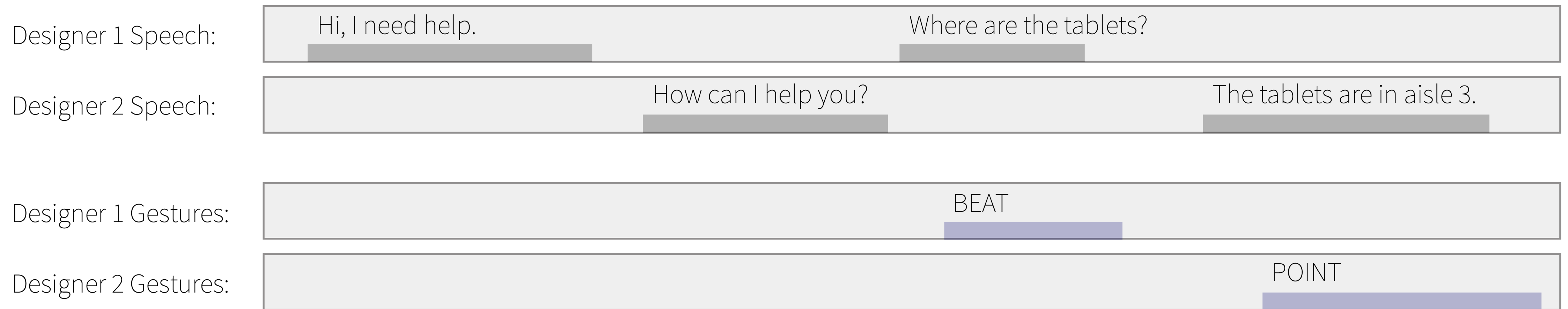
# examples & demonstrations



# examples & demonstrations



# examples & demonstrations



# natural language

find the number of papers in CAV 2019



# natural language

find the number of papers in CAV 2019

```
SELECT count(Pub.pid)
FROM Pub JOIN Conf ON Pub.cid = Conf.cid
WHERE Conf.name = "CAV" AND Pub.year = 2019
```

# natural language

find the number of papers in CAV 2019

```
SELECT count(Pub.pid)
FROM Pub JOIN Conf ON Pub.cid = Conf.cid
WHERE Conf.name = "CAV" AND Pub.year = 2019
```

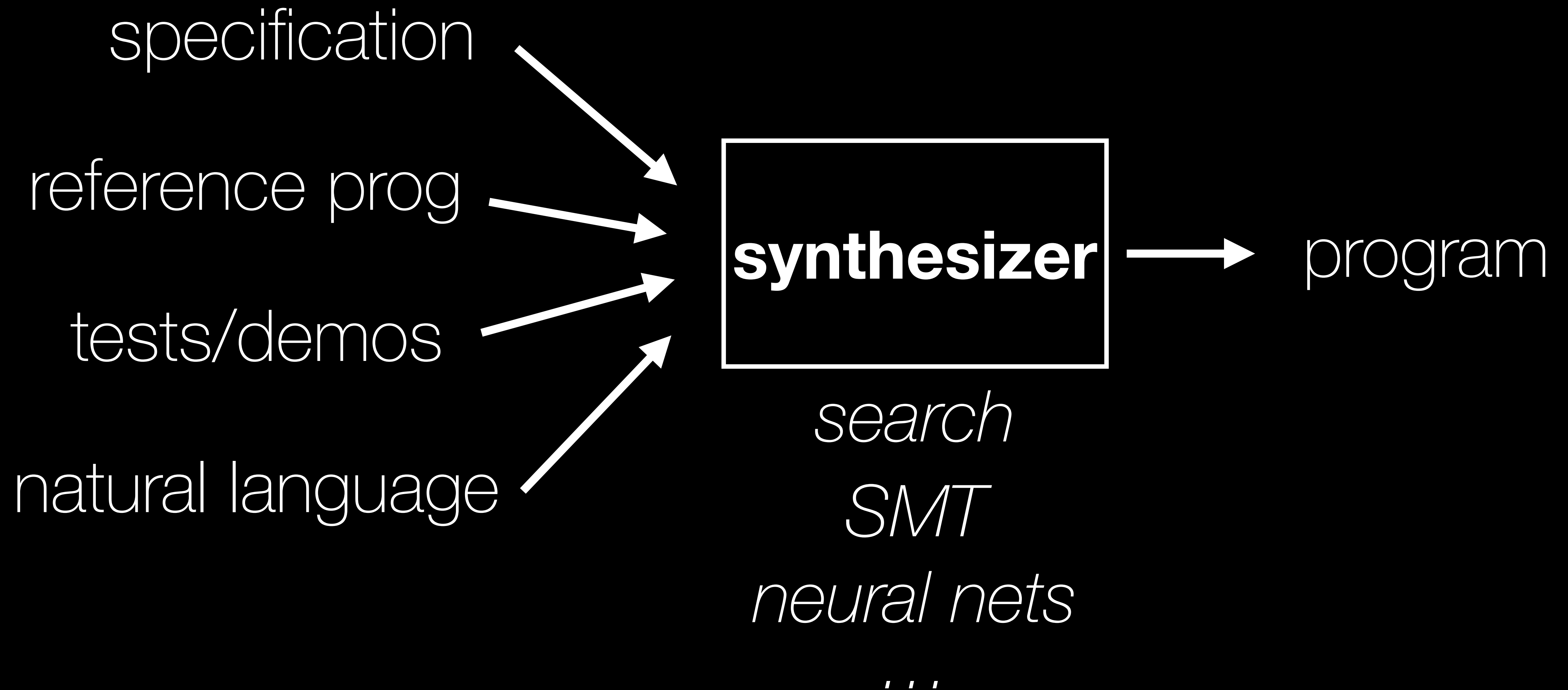
## **SQLizer: Query Synthesis from Natural Language**

NAVID YAGHMAZADEH, The University of Texas at Austin

YUEPENG WANG, The University of Texas at Austin

ISIL DILLIG, The University of Texas at Austin

THOMAS DILLIG, The University of Texas at Austin



```
int p_naive(int x):  
    y = x * 2  
    return y
```

≡

```
int p_smart(int x):  
    y = x << h  
    return y
```

```
int p_naive(int x):  
    y = x * 2  
    return y
```

≡

```
int p_smart(int x):  
    y = x << h  
    return y
```

### definition 1

find a value for **h** such that *for all* values of **x**,  
**p\_naive(x) = p\_smart(x)**.

```
int p_naive(int x):  
    y = x * 2  
    return y
```

≡

```
int p_smart(int x, h):  
    y = x << h  
    return y
```

```
int p_naive(int x):  
    y = x * 2  
    return y
```

≡

```
int p_smart(int x, h):  
    y = x << h  
    return y
```



```
int p_naive(int x):  
    y = x * 2  
    return y
```

≡

```
int p_smart(int x, h):  
    y = x << h  
    return y
```



## definition 2

find a value for **h** such that *for all* values of **x**,  
 $p\_naive(x) = p\_smart(x, h)$ .



```
int p_naive(int x):  
    y = x * 2  
    return y
```

≡

```
int p_smart(int x, h):  
    y = x << h  
    return y
```

$R_n(x, y)$ 

```
int p_naive(int x):  
    y = x * 2  
    return y
```

 $\equiv$  $R_s(x, h, y)$ 

```
int p_smart(int x, h):  
    y = x << h  
    return y
```

$R_n(x, y)$ 

```
int p_naive(int x):  
    y = x * 2  
    return y
```

 $\equiv$  $R_s(x, h, y)$ 

```
int p_smart(int x, h):  
    y = x << h  
    return y
```

### definition 3

find a value for  $h$  such that *for all* values of  $x$  and  $y$ ,  
 $R_n(x, y) \iff R_s(x, h, y)$  is true

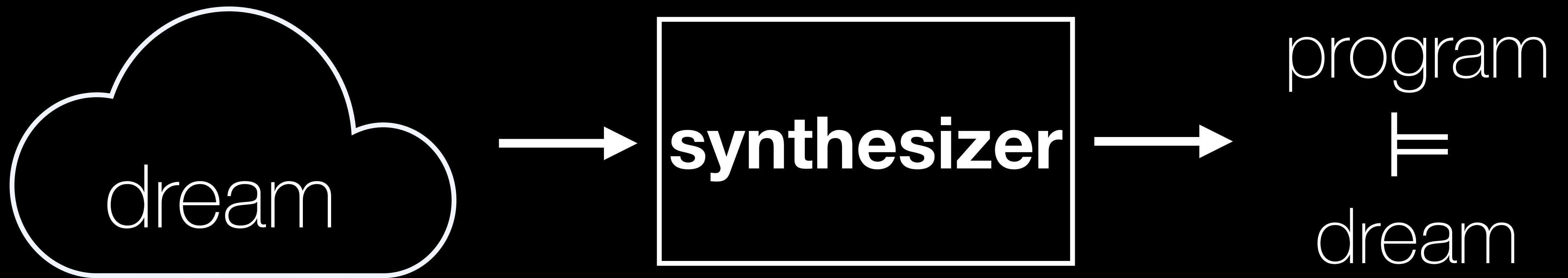
$R_n(x, y)$ 

```
int p_naive(int x):  
    y = x * 2  
    return y
```

 $\equiv$  $R_s(x, h, y)$ 

```
int p_smart(int x, h):  
    y = x << h  
    return y
```

$$\forall x, y. R_n(x, y) \iff R_s(x, h, y)$$



[github.com/barghouthi/vmw19](https://github.com/barghouthi/vmw19)