Cheaper (& correct) blockchain protocols and programs

Maria A Schett
mail@maria-a-schett.net

2021-10-01@ Languages, Systems, and Data Lab @ UCSC

Goals

1. **blockchains** are **fun**

2. **cheaper protocols** through (nearly) **'telepatic' computers** [PODC21]

3. 🤑 through **cheaper programs** [CAV20]

? ⇒ !

Blockchain

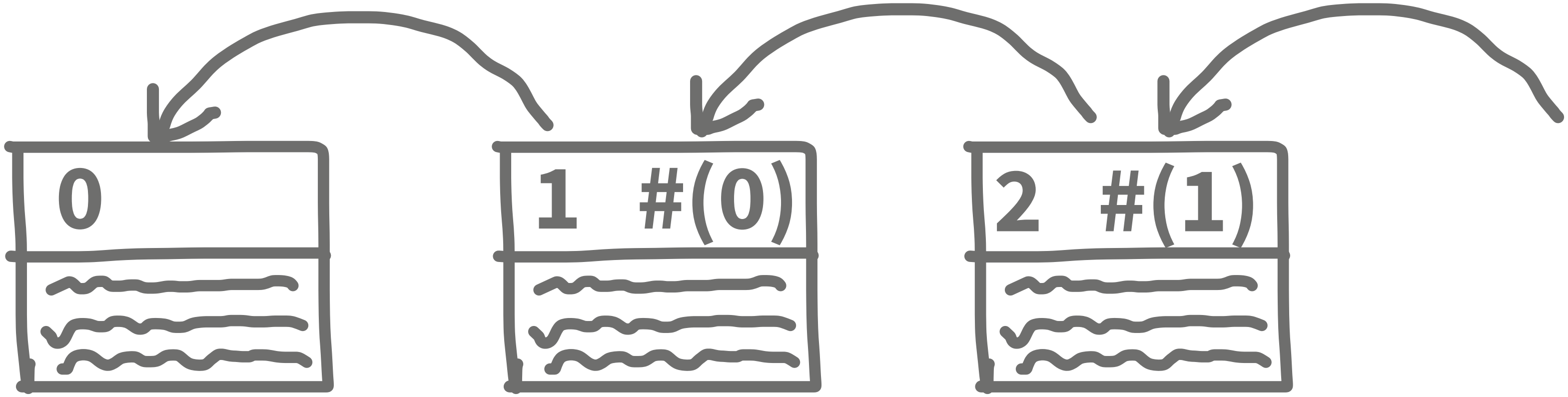
"no standard technical definition but is a **loose umbrella term"**

referring to

"systems that bear varying levels of **resemblance to Bitcoin and its ledger"** [1]

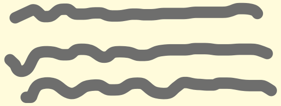
[1] A Narayanan & J Clark. Bitcoin’s Academic Pedigree. Queue, 15(4):20, 2017

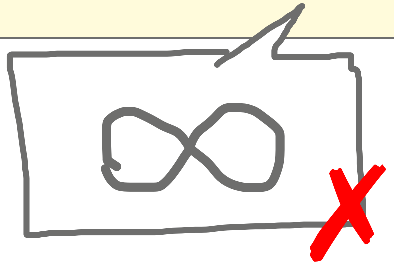
Data structure



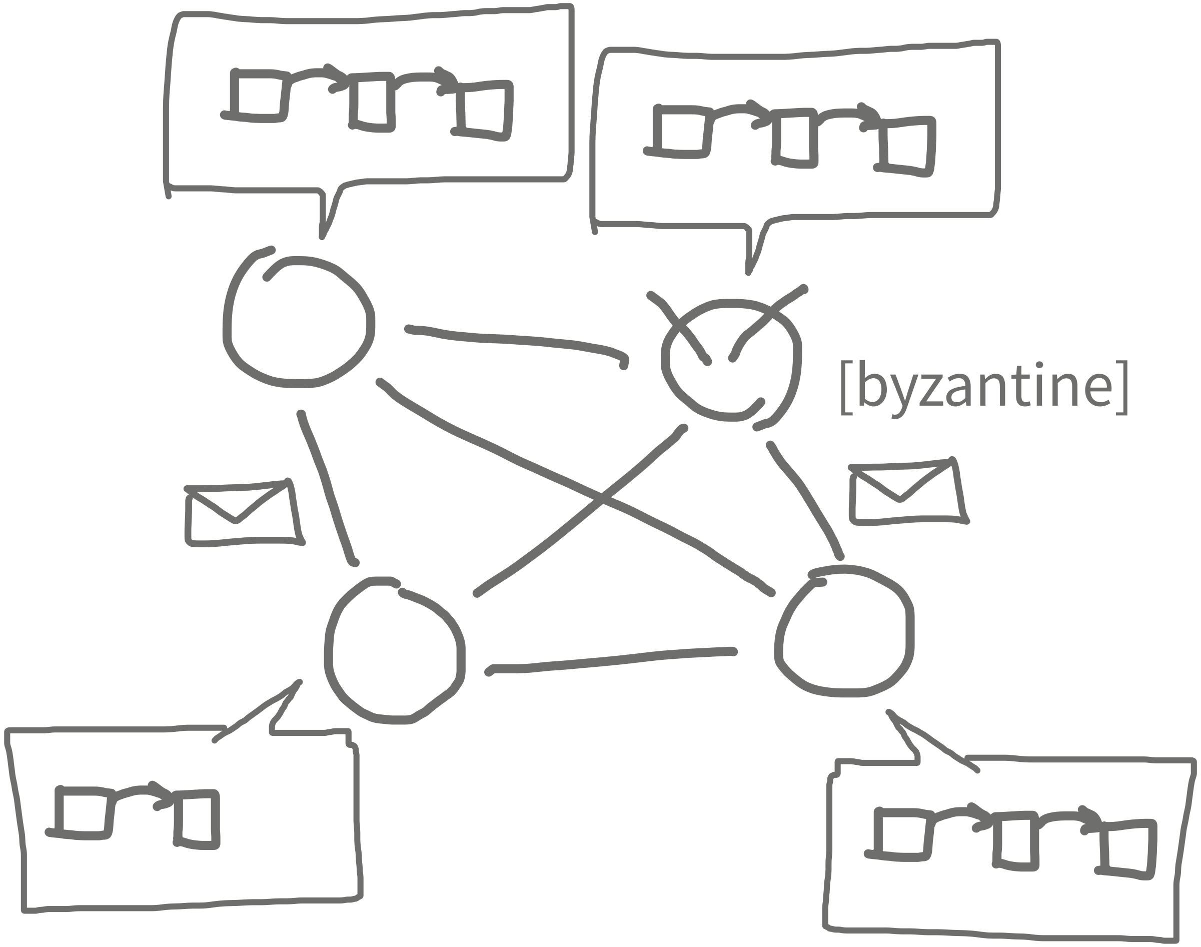
[genesis block]

[Ethereum]

 == transactions || "smart contracts"



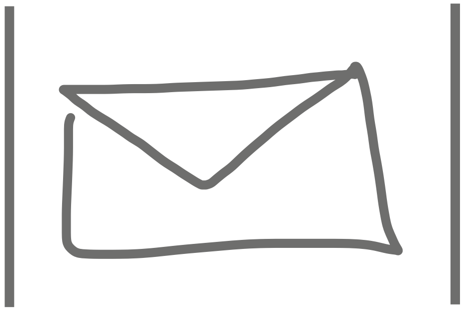
Govern shared state



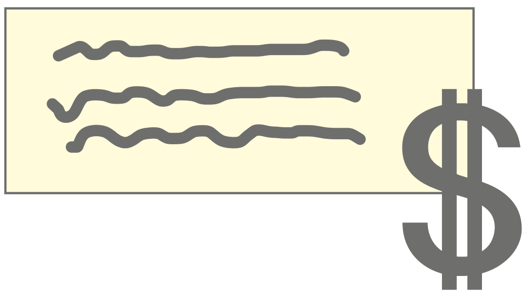
EXPENSIVE!

Cheaper blockchain …

protocols



programs

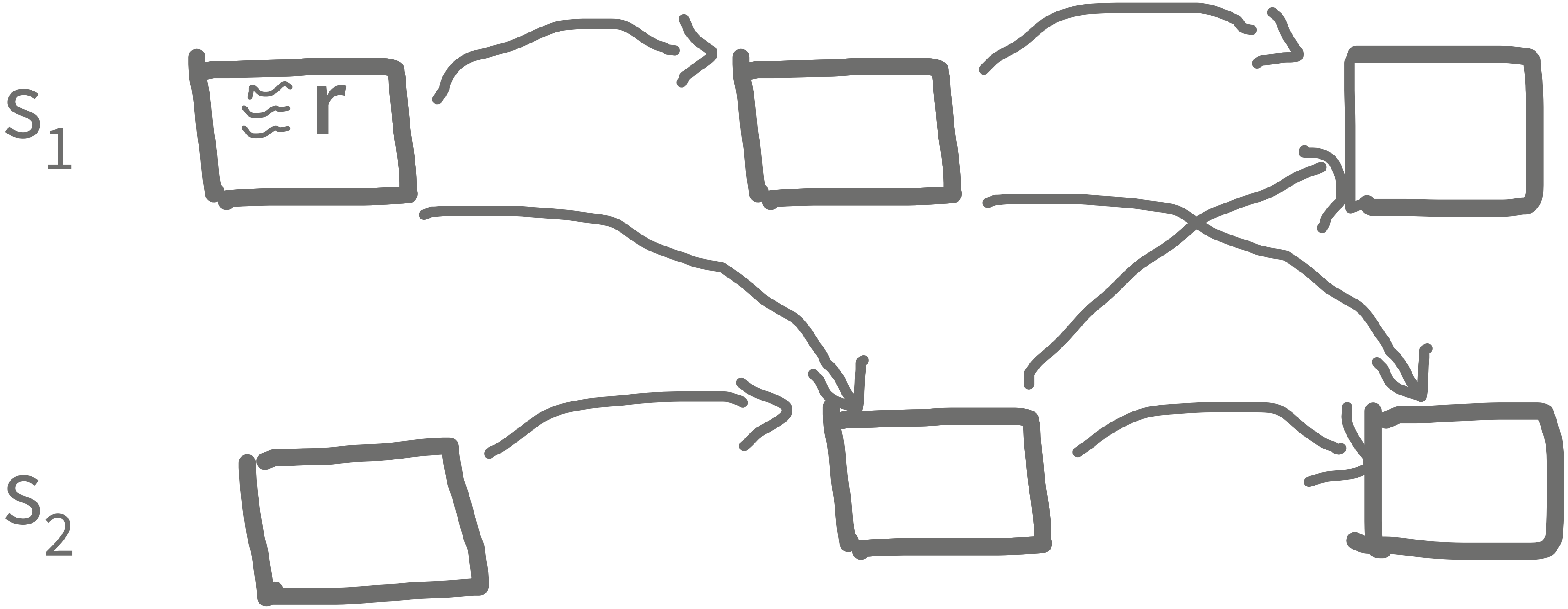


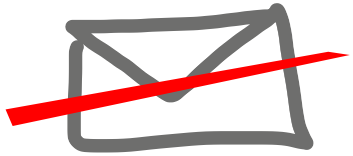
guaranteeing correctness

Cheaper blockchain protocols

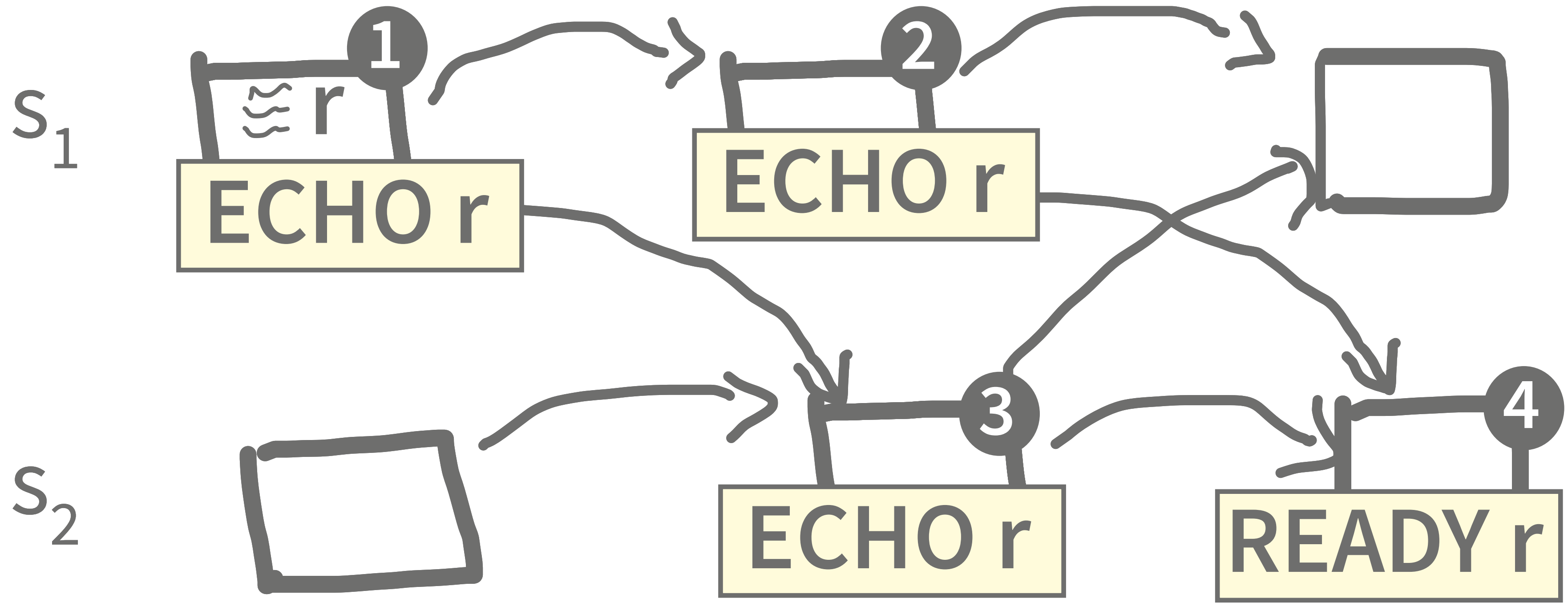
-chain to -DAG

"log DAG"

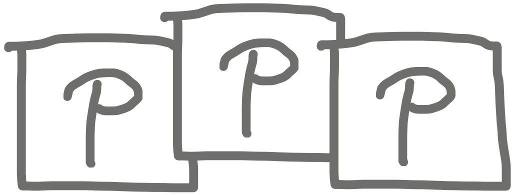


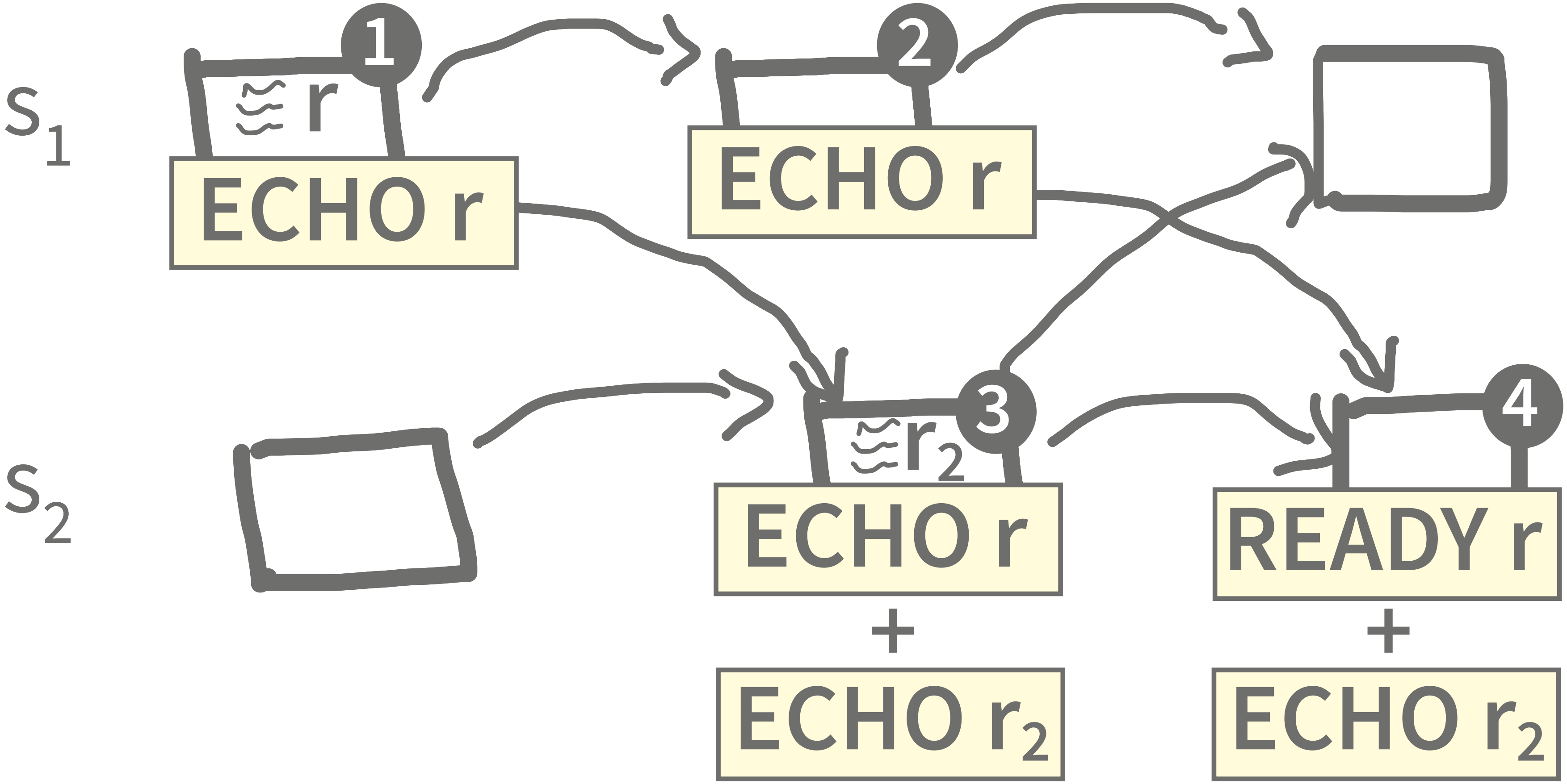


Interpret $\boxed{\mathcal{P}}$:= reliable broadcast of ≋ r



Interpret





S_1

ECHO r

ECHO r

S_2

ECHO r

+

ECHO r_2

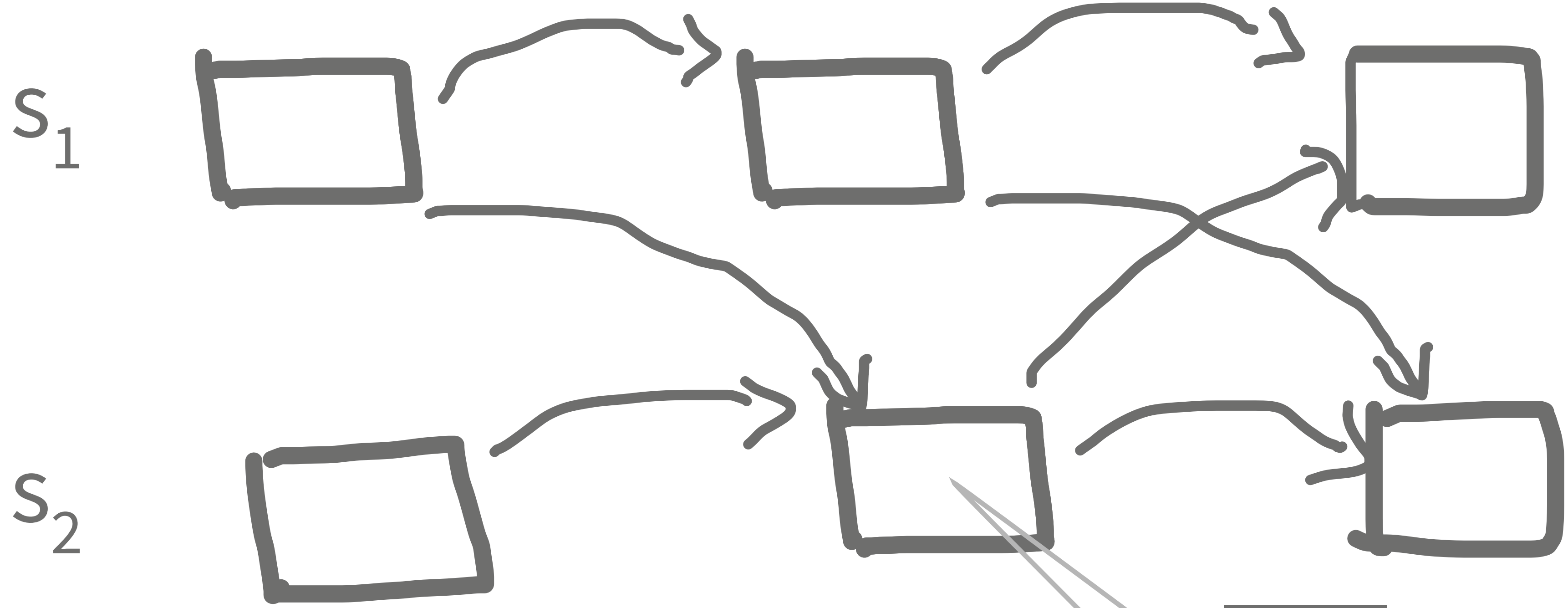
READY r

+

ECHO r_2

interpret protocol

Build a block DAG



creator: ○ **valid**

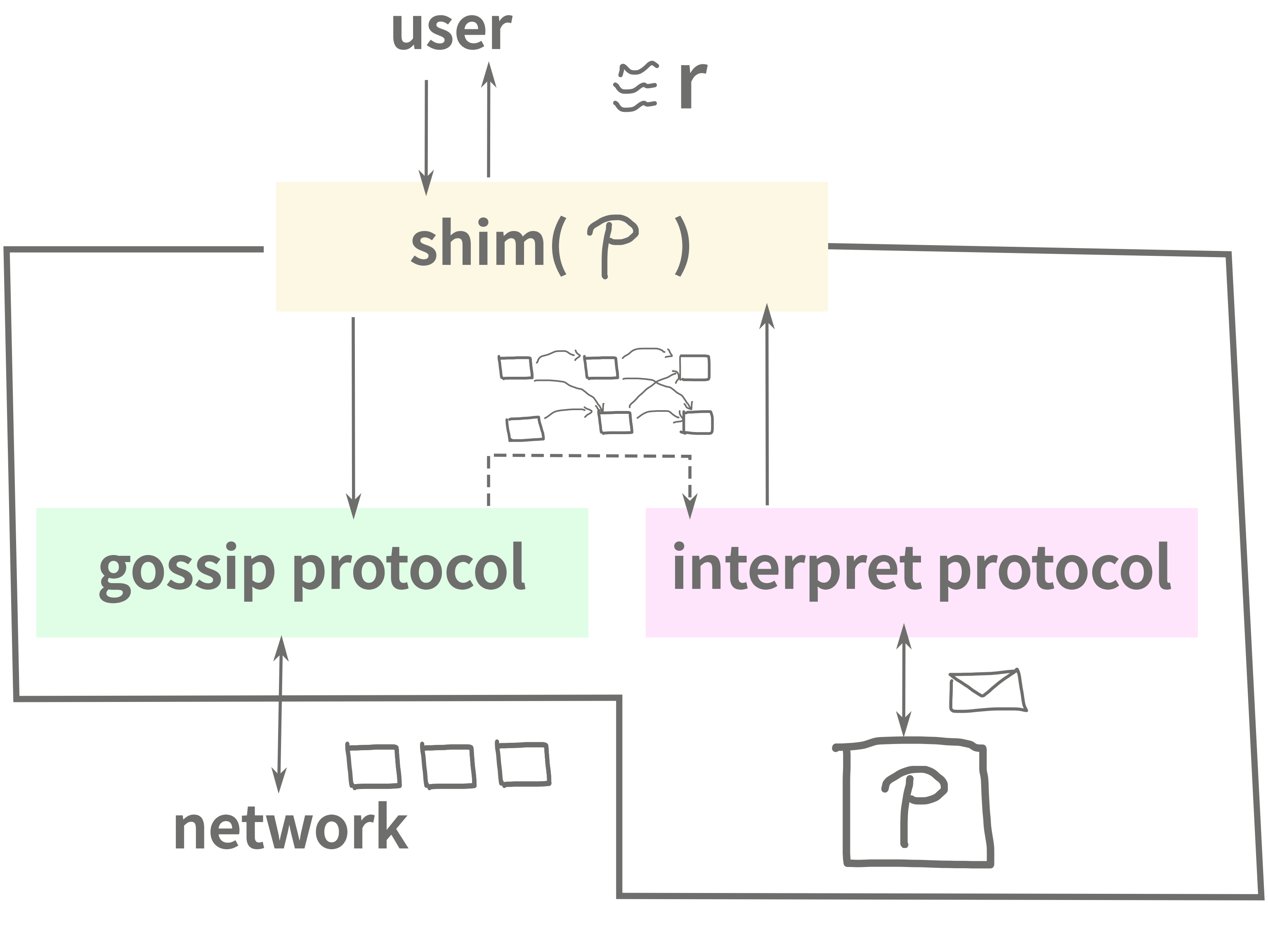
parent: □

preds: □ □

user requests: ≋

gossip protocol

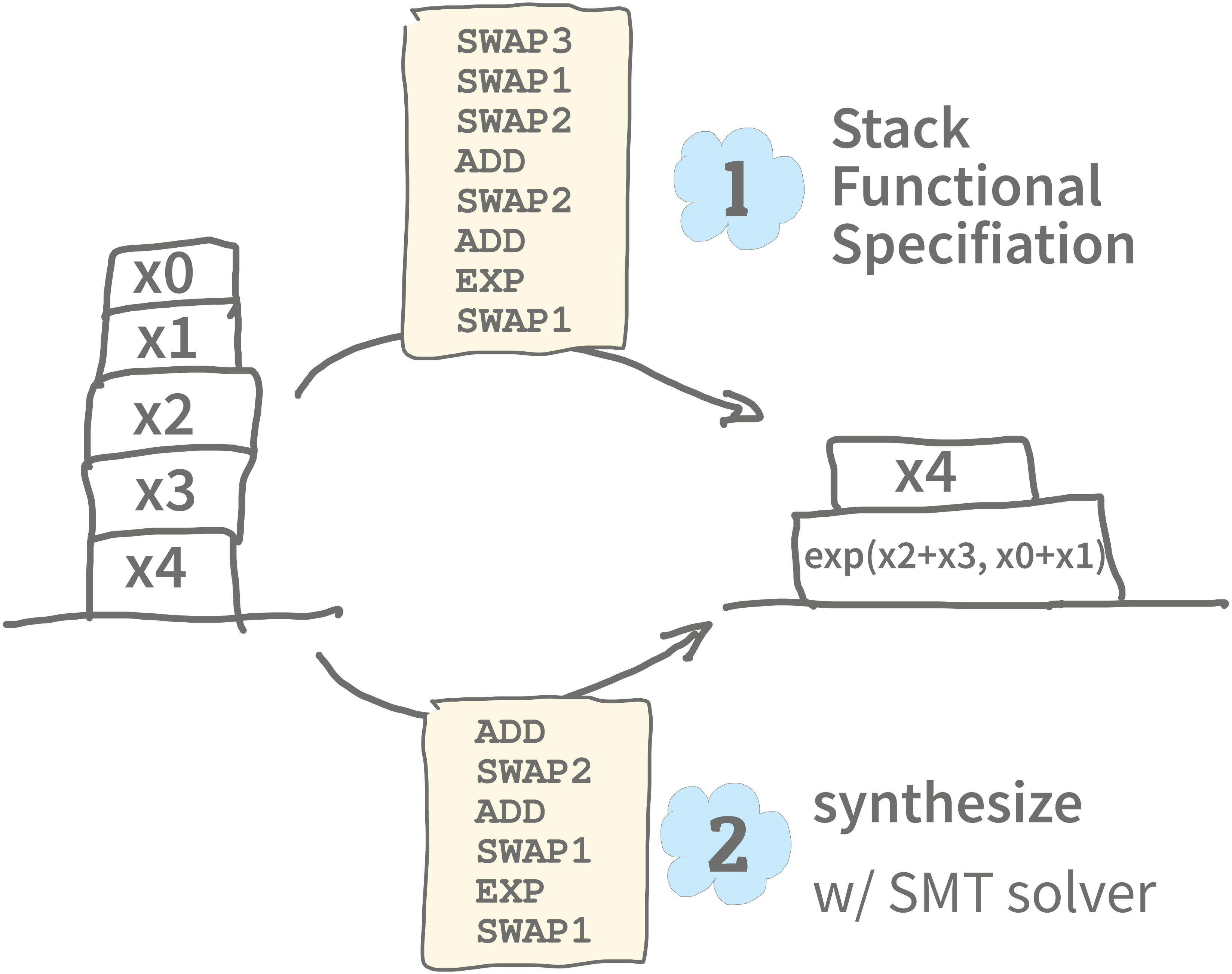
Block DAG framework



For every correct server
if protocol $\mathcal{P}$ has safety or liveness property $\mathbb{P}$
then shim($\mathcal{P}$) preserves $\mathbb{P}$.

idea: block DAG is a **reliable point-to-point link**

Cheaper blockchain programs



```
SWAP3
SWAP1
SWAP2
ADD
SWAP2
ADD
EXP
SWAP1
```

1 Stack Functional Specifiation

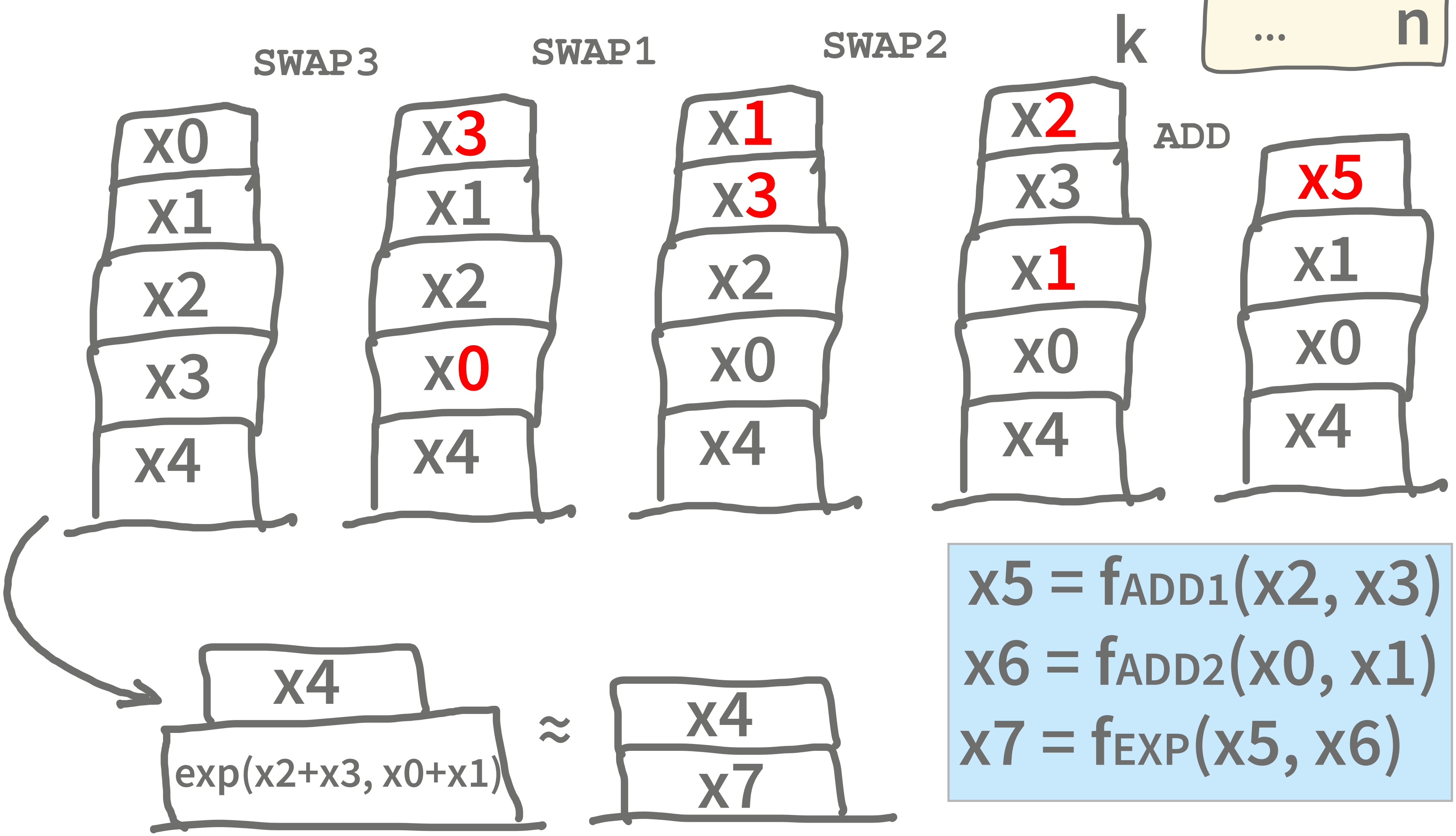
```
ADD
SWAP2
ADD
SWAP1
EXP
SWAP1
```

2 synthesize

w/ SMT solver

SFS (Stack Functional Specification)

```
SWAP3
SWAP1
SWAP2
ADD
...        n
```



$$x5 = f_{\mathrm{ADD1}}(x2, x3)$$
$$x6 = f_{\mathrm{ADD2}}(x0, x1)$$
$$x7 = f_{\mathrm{EXP}}(x5, x6)$$

SMT solvers

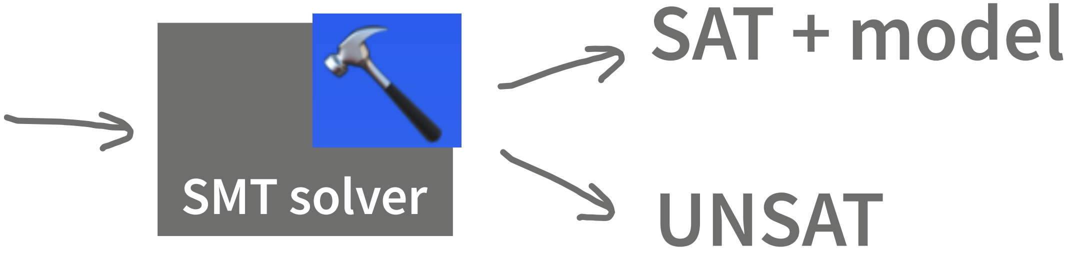
Satisfiability Modulo Theories

first-order logic

BV, LIA, unintepreted functions ...

[decidable]

```
∃ t, s, u.
  t = 3 ⇒
  s = t + 1 ∧ u
```



e.g. Z3

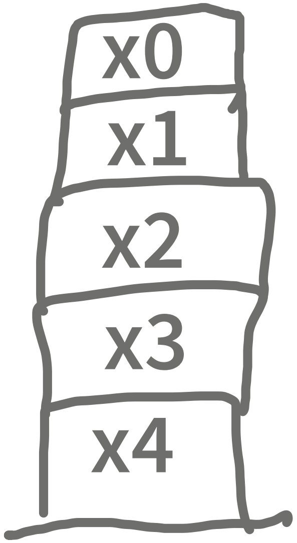
Synthesize

$\exists\, t_1 \ldots t_n.\ \_\_{t_1}\ \_\_{t_2}\ \cdots\ \_\_{t_n}$

SWAP1 ↦ 1

PUSH ↦ 2

f_{ADD1} ↦ 42 ...



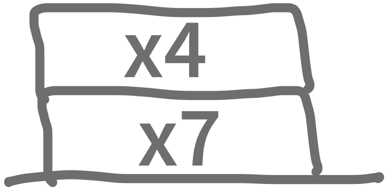
$s_{0,0} = x0$

$s_{1,0} = x1$

$s_{2,0} = x2$

$s_{3,0} = x3$

$s_{4,0} = x4$



$s_{0,n} = x4$

$s_{1,n} = x7$

Synthesize

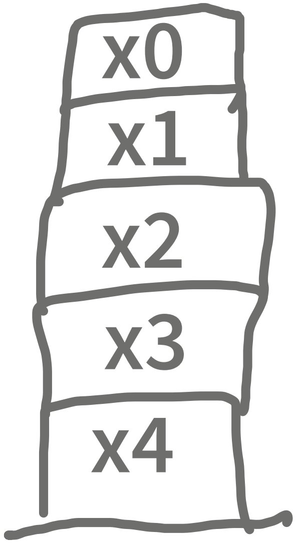
$\exists\ t_1 \ldots t_n.\ \_t_1\ \_t_2\ \cdots\ \_t_n$

SWAP1 ↦ 1

PUSH ↦ 2

f_{ADD1} ↦ 42 …

$$
\begin{aligned}
t_j = 1 \Rightarrow\ & s_{0,j+1} = s_{1,j+1}\ \wedge \\
& s_{1,j+1} = s_{0,j+1}\ \wedge \\
& \mathrm{move}(s_{2,j+1} \ldots s_{k,j+1})
\end{aligned}
$$



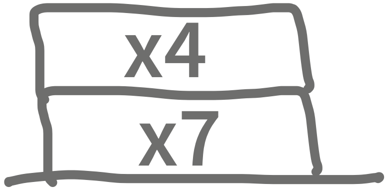
$s_{0,0} = x0$

$s_{1,0} = x1$

$s_{2,0} = x2$

$s_{3,0} = x3$

$s_{4,0} = x4$



$s_{0,n} = x4$

$s_{1,n} = x7$

Synthesize

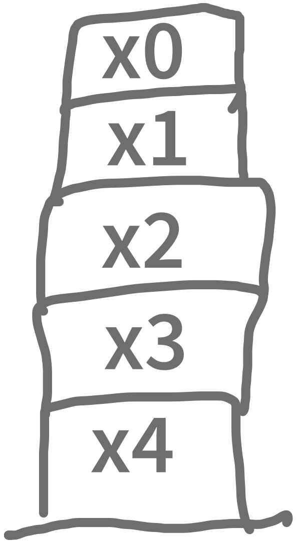
$\exists\, t_1 \ldots t_n.\ \_{t_1}\ \_{t_2}\ \cdots\ \_{t_n}$

SWAP1 ↦ 1

PUSH ↦ 2

f_{ADD1} ↦ 42 ...

$t_j = 2 \Rightarrow \ldots$



$s_{0,0} = x0$

$s_{1,0} = x1$

$s_{2,0} = x2$

$s_{3,0} = x3$

$s_{4,0} = x4$

Synthesize

$\exists\, t_1 \ldots t_n.\ \_t_1\ \_t_2\ \cdots\ \_t_n$

$\_a_1\ \_a_2\ \cdots\ \_a_n$

$\mathrm{SWAP1} \mapsto 1$

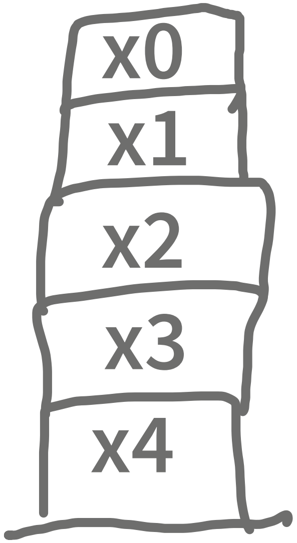
$\mathrm{PUSH} \mapsto 2$

$f_{\mathrm{ADD1}} \mapsto 42 \ldots$

$t_j = 2 \Rightarrow s_{0,j+1} = a_j \wedge$

$a_j < 2^{256} \wedge$

$\mathrm{move}(s_{0,j+1}, \ldots s_{k,j+1})$



$s_{0,0} = x0$

$s_{1,0} = x1$

$s_{2,0} = x2$

$s_{3,0} = x3$

$s_{4,0} = x4$

$x_i \mapsto 2^{256} + i$

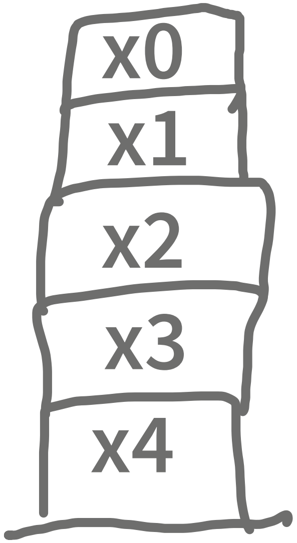
$\exists\ t_1 \ldots t_n.\ \_{t_1}\ \_{t_2}\ \cdots\ \_{t_n}$

SWAP1 ↦ 1

PUSH ↦ 2

$f_{\text{ADD1}} \mapsto 42 \ldots$

$$t_j = 42 \Rightarrow s_{0,j+1} = x2 \wedge$$
$$s_{1,j+1} = x3 \wedge$$
$$\text{move}(s_{2,j+1} \ldots s_{k,j+1})$$



$s_{0,0} = x0$

$s_{1,0} = x1$

$s_{2,0} = x2$

$s_{3,0} = x3$

$s_{4,0} = x4$

SFS

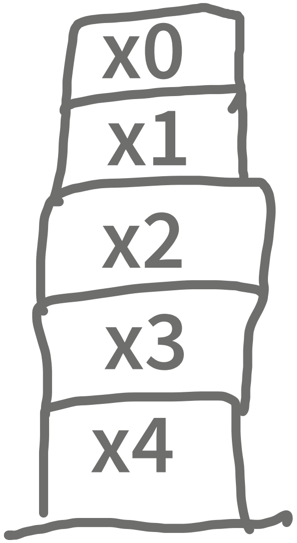
$x5 = f_{\text{ADD1}}(x2, x3)$

$\exists\ t_1 \ldots t_n.\ \_\_{t_1}\ \_\_{t_2}\ \cdots\ \_\_{t_n}$

$\mathrm{SWAP1} \mapsto 1$

$\mathrm{PUSH} \mapsto 2$

$f_{\mathrm{ADD1}} \mapsto 42 \ldots$



$s_{0,0} = x0$

$s_{1,0} = x1$

$s_{2,0} = x2$

$s_{3,0} = x3$

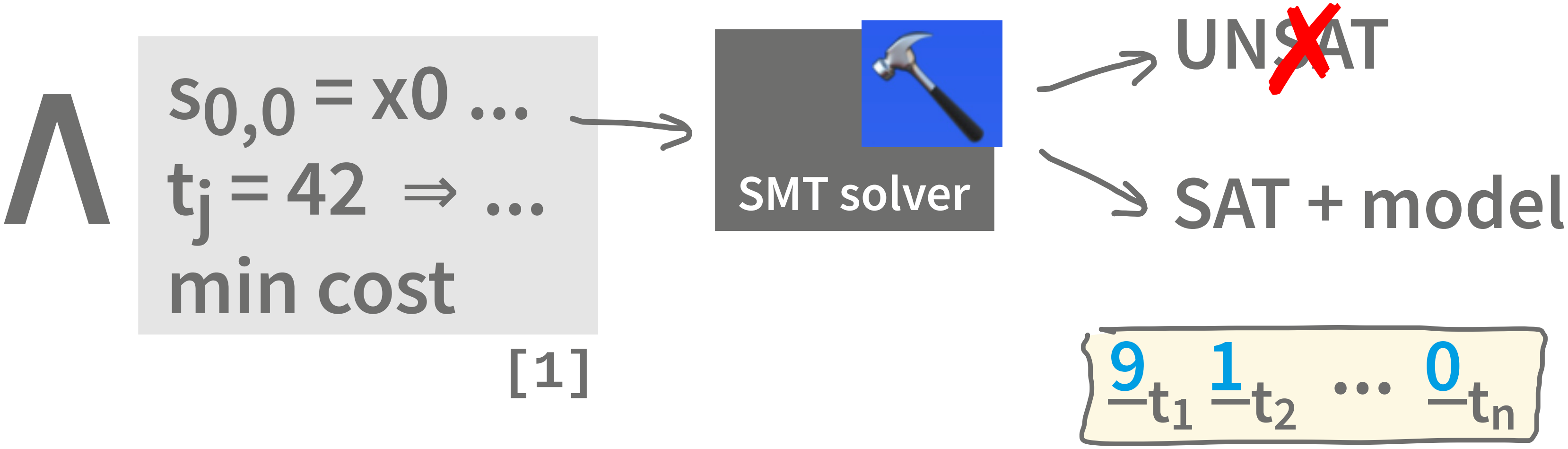
$s_{4,0} = x4$

$$t_j = 42 \Rightarrow s_{0,j+1} = x2 \wedge \\ s_{1,j+1} = x3 \wedge \\ \mathrm{move}(s_{2,j+1} \ldots s_{k,j+1})$$

$?

$$t_j = 42 \Rightarrow \underbrace{\mathrm{cost}}_{\mathrm{min}} + 3$$

Synthesize



$\bigwedge$

$s_{0,0} = x0 \ldots$

$t_j = 42 \Rightarrow \ldots$

min cost

[1]

SMT solver

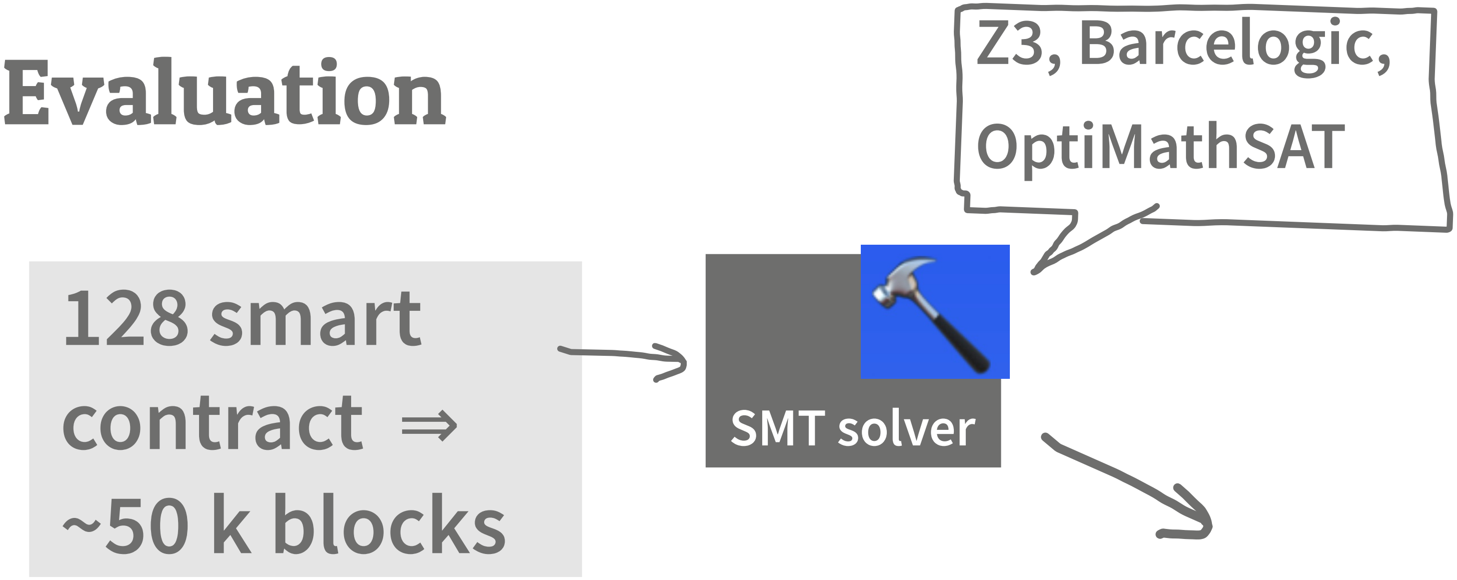
~~UNSAT~~

SAT + model

$\underline{9}_{t_1}\ \underline{1}_{t_2}\ \ldots\ \underline{0}_{t_n}$

[1] github.com/mariaschett/syrup-backend

Evaluation



Z3, Barcelogic, OptiMathSAT

128 smart contract ⇒ ~50 k blocks

SMT solver

~65% already optimal
~30% optimized
~2% time-out

transfer of AirdropToken; 500k called; saved 832 gas ⇒ 2815 $

Wrapping Up

Goals

3 🤑 through **cheaper programs** [CAV20]

2 **cheaper protocols** through (nearly) **'telepatic' computers** [PODC21]

1 **blockchains** are **fun**

[mail@] **maria-a-schett.net**

Thank you!