Esil: A hard pill to swallow

Esil
Esil
radare2
Evaluable Strings Intermediate Language
Esil
RADARE2
one architecture to emulate them all
$b8,C,=
$b4,H,=
$z,Z,=
17,a,==
@mikelloc
x86 e

Whoami

Twitter: @condr3t

Why Esil?

- Need for emulation
  - Debugging
  - Hooking and tracing
  - Halting problem

- We can

Last year

The Limits of Esil:

- memory mapping
- interaction with the world

Memory mapping

Interaction with the world

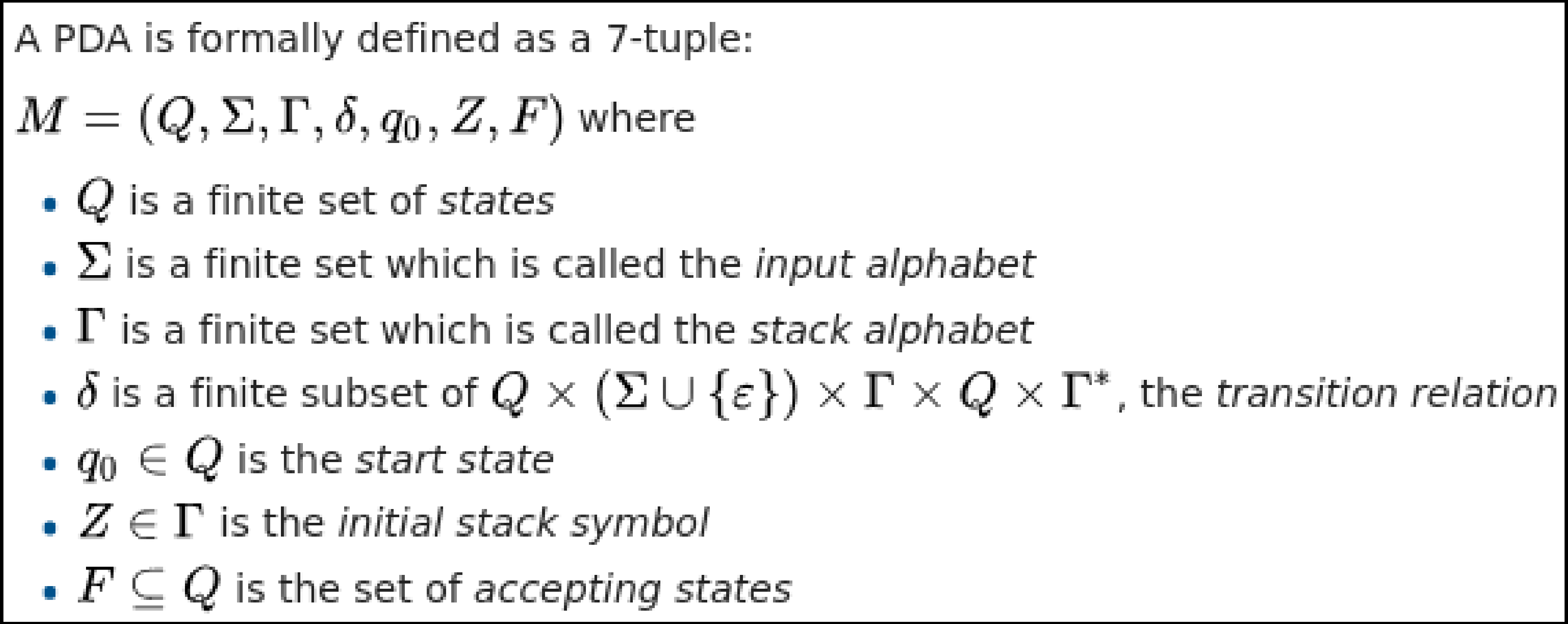
???

Esil Parser: Thinking with a stack

- How do stack machines work?
- Why does esil make use of this?

Stack Machines/PDA's

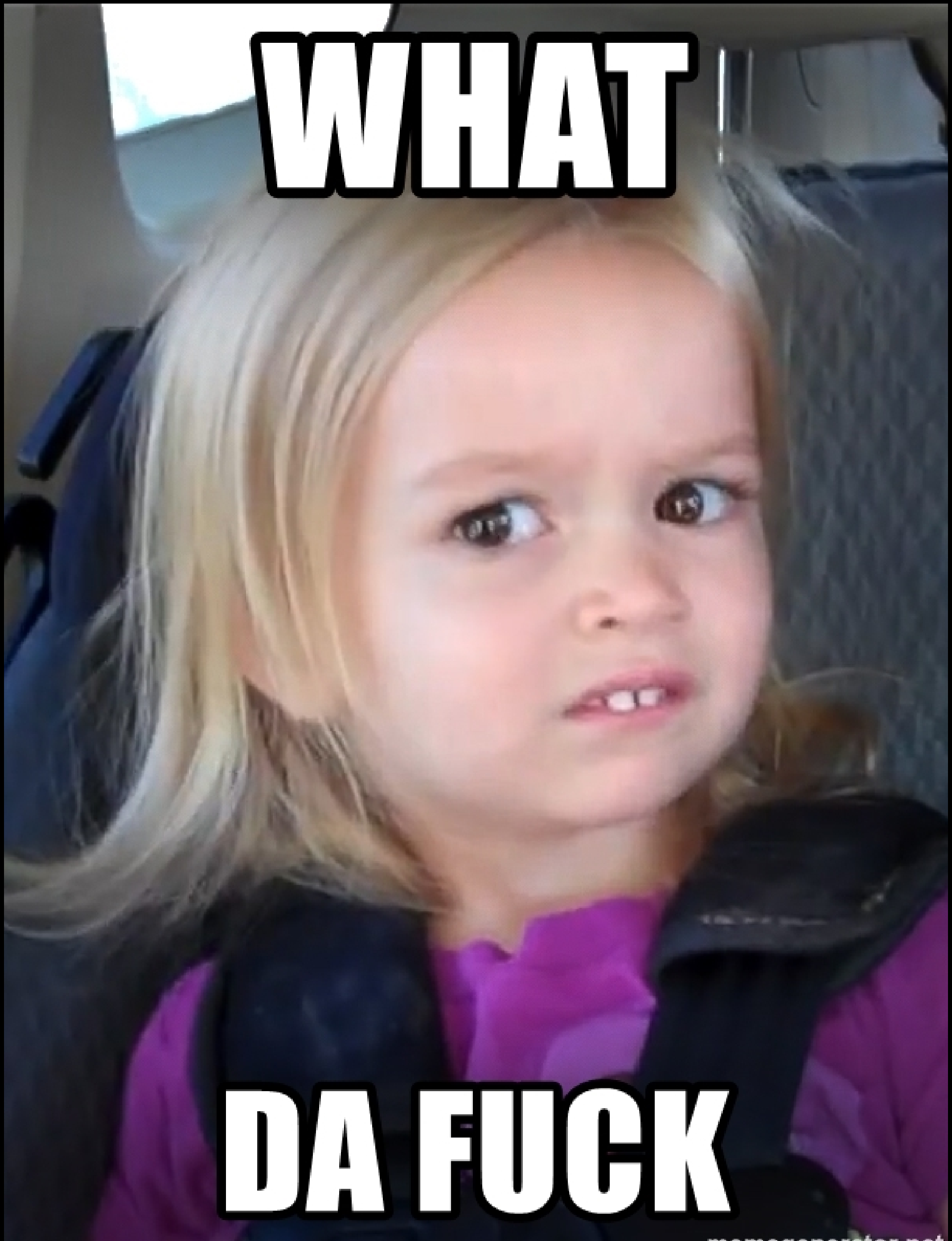
Wikipedia:



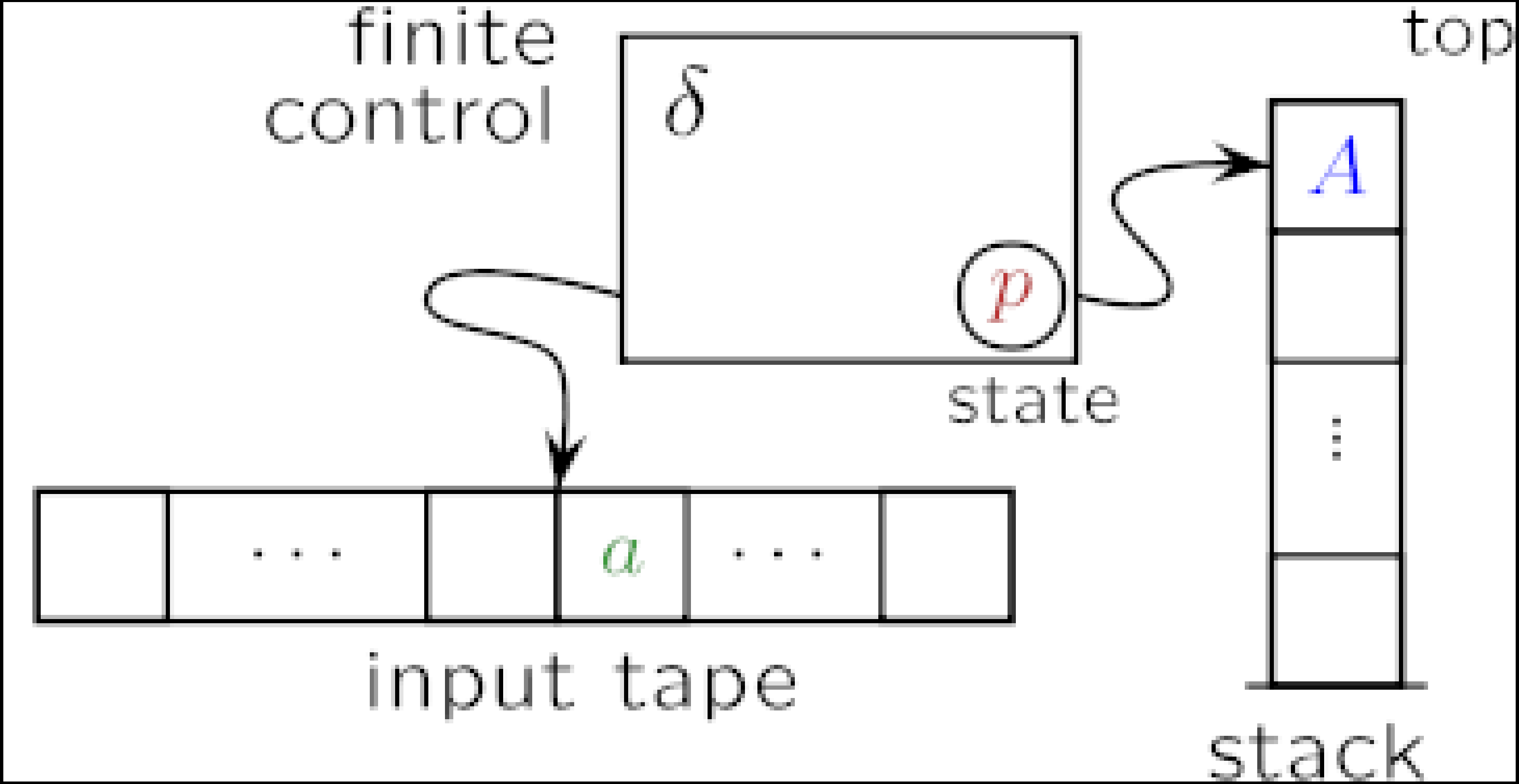
A PDA is formally defined as a 7-tuple:

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ where

- Q is a finite set of *states*
- Σ is a finite set which is called the *input alphabet*
- Γ is a finite set which is called the *stack alphabet*
- δ is a finite subset of $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times Q \times \Gamma^*$, the *transition relation*
- $q_0 \in Q$ is the *start state*
- $Z \in \Gamma$ is the *initial stack symbol*
- $F \subseteq Q$ is the set of *accepting states*

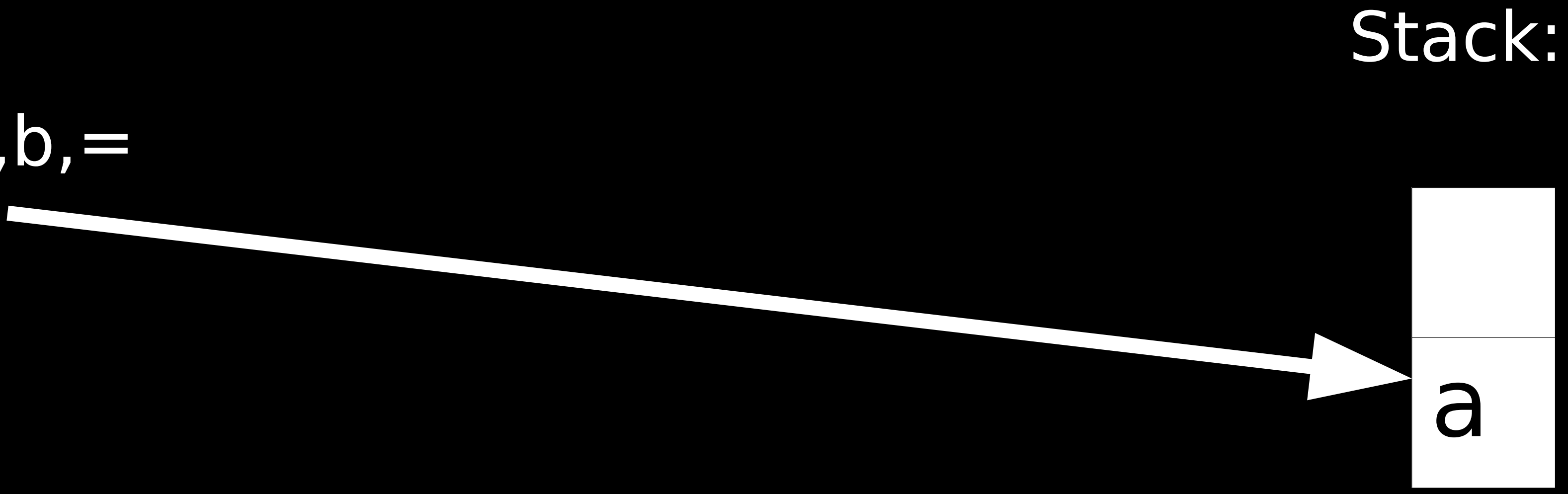

WHAT
DA FUCK
memegenerator.net

Wikipedia:


finite control
δ
top
A
p
state
$\cdots$
a
$\cdots$
$\vdots$
input tape
stack

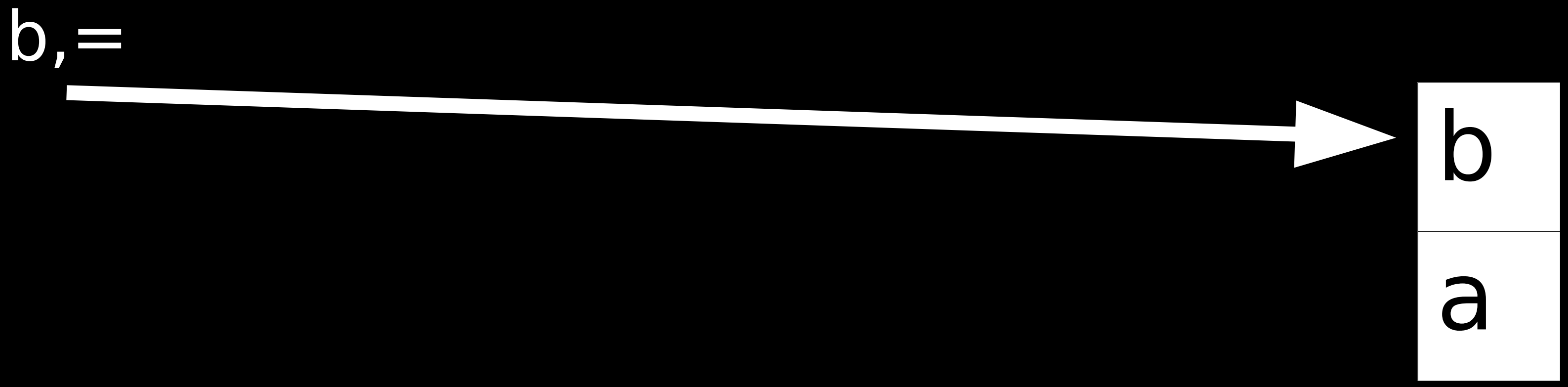
Example

a,b,=

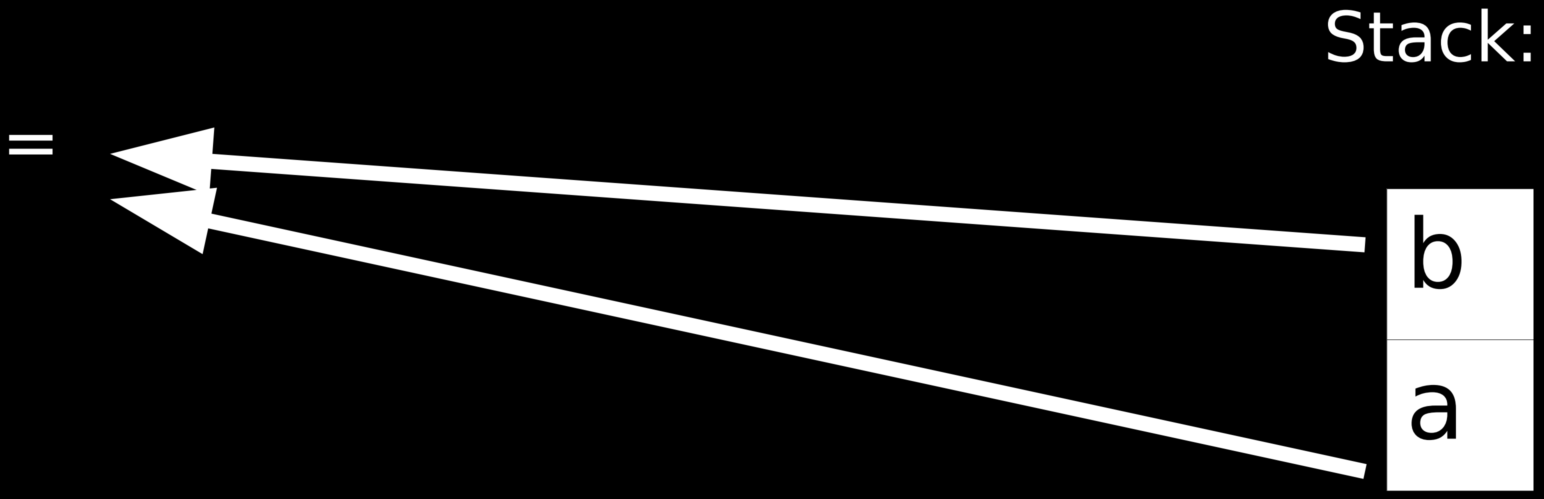


Example

Stack:

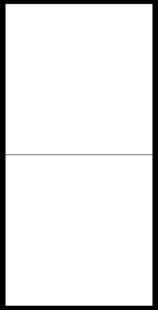


Example



Example

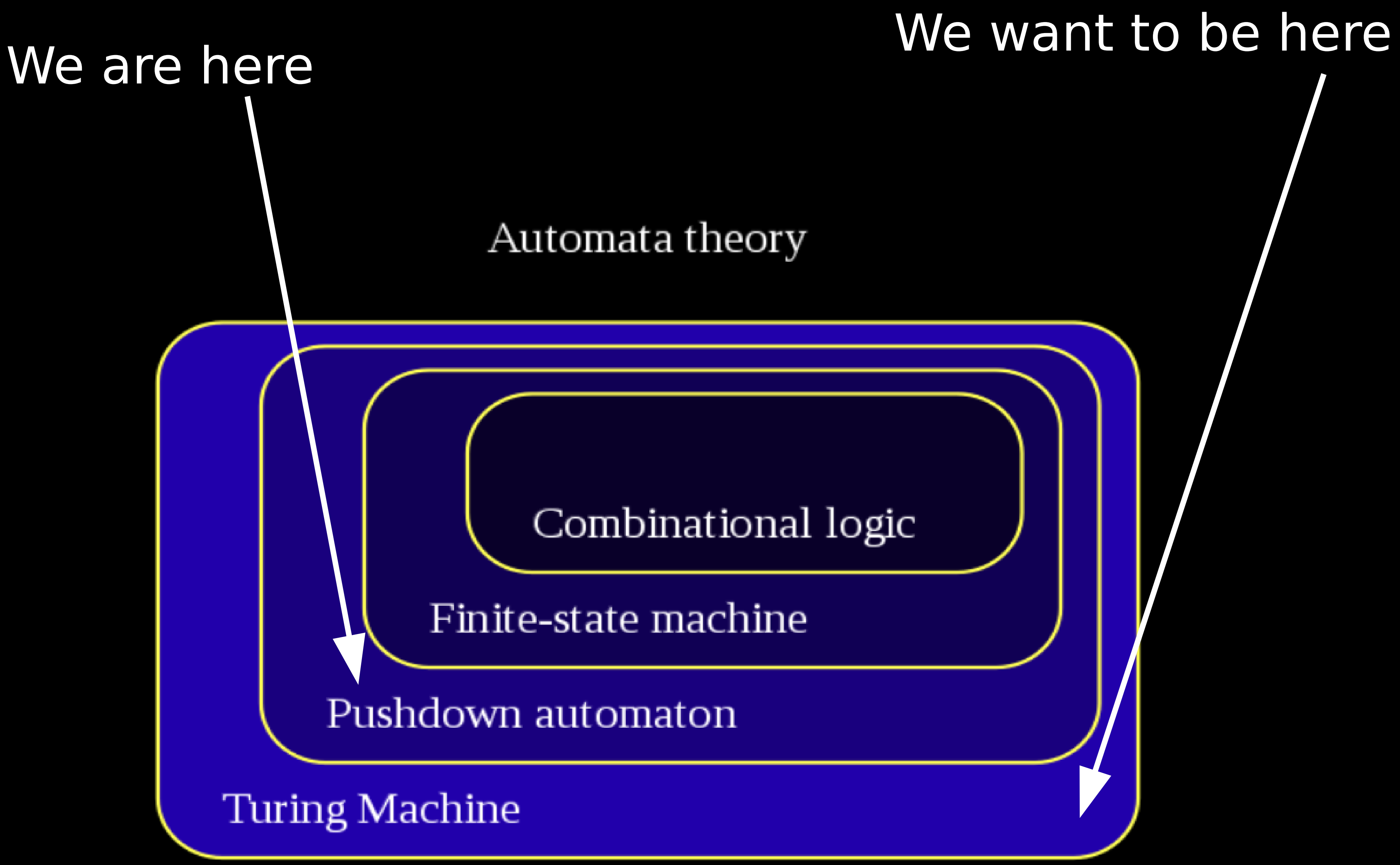
Stack:



Is this enough?

NO!

Is this enough?



Cheating part 1

- add random access operations to the stack
- add goto operations

DEMO: compute gcd in esil

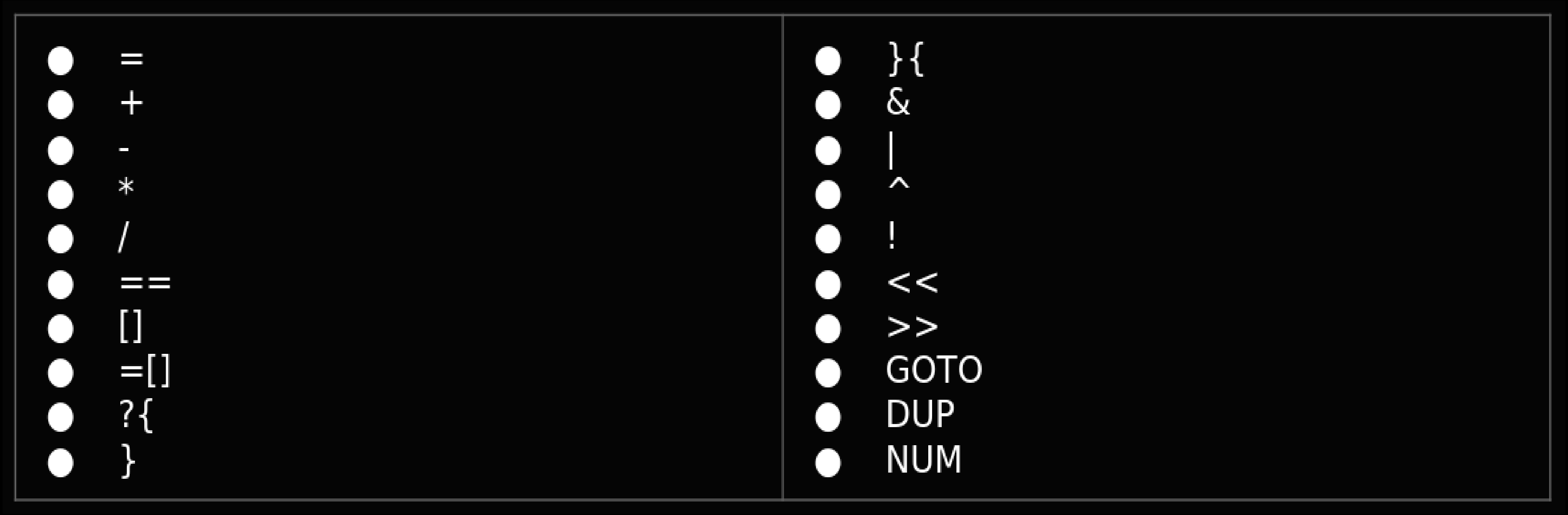
Is this enough?

NO!

Cheating part 2

- add register access
- add a “tape“ with random access

Esil Operations



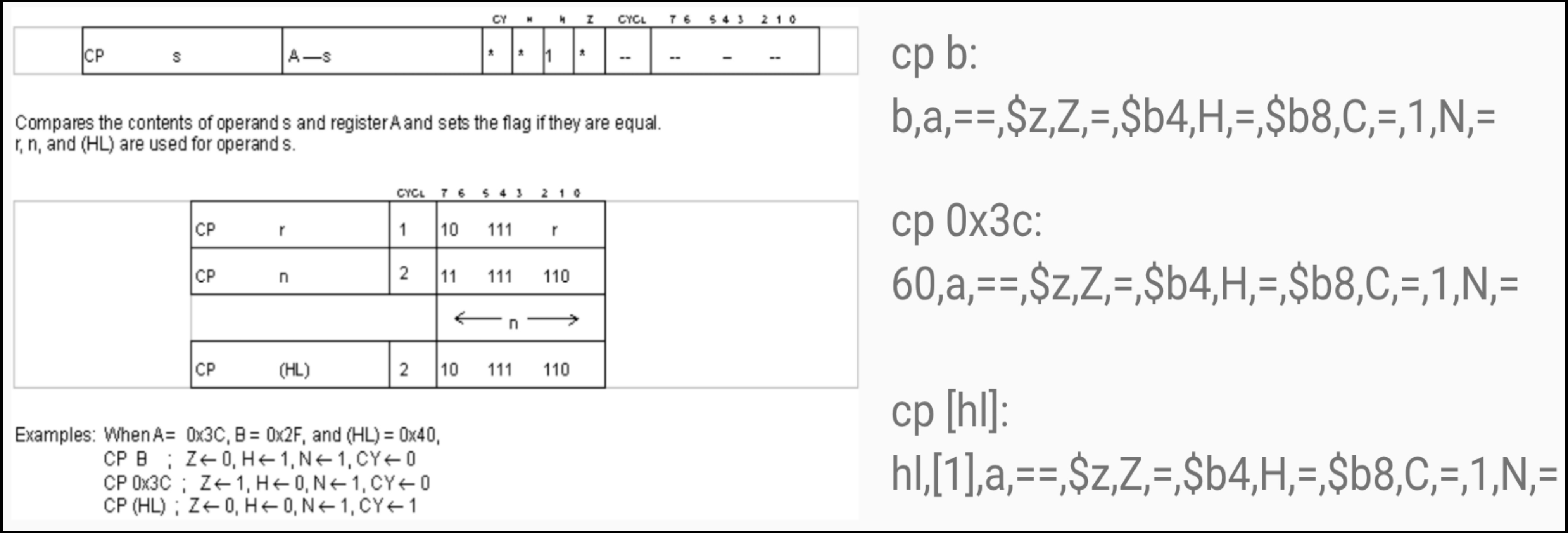
- =
- +
- -
- *
- /
- ==
- []
- =[]
- ?{
- }

- }{
- &
- |
- ^
- !
- <<
- >>
- GOTO
- DUP
- NUM

Sugar: Esil internal vars

- $ is prefix for access
  - $z for zeroflag
  - $cx for carry from bit x
  - $bx for borrow from bit x
- calculate flags
- update on every operation, that sets something
  - compares old and new value of the dst

Example: CP from gameboy



		CY	N	H	Z	CYCL	7 6	5 4 3	2 1 0
CP s	A —s	*	*	1	*	--	--	–	--

Compares the contents of operand s and register A and sets the flag if they are equal.
r, n, and (HL) are used for operand s.

	CYCL	7 6	5 4 3	2 1 0
CP r	1	10	111	r
CP n	2	11	111	110
		← n →		
CP (HL)	2	10	111	110

Examples: When A = 0x3C, B = 0x2F, and (HL) = 0x40,
CP B ; Z ← 0, H ← 1, N ← 1, CY ← 0
CP 0x3C ; Z ← 1, H ← 0, N ← 1, CY ← 0
CP (HL) ; Z ← 0, H ← 0, N ← 1, CY ← 1

```
cp b:
b,a,==,$z,Z,=,$b4,H,=,$b8,C,=,1,N,=

cp 0x3c:
60,a,==,$z,Z,=,$b4,H,=,$b8,C,=,1,N,=

cp [hl]:
hl,[1],a,==,$z,Z,=,$b4,H,=,$b8,C,=,1,N,=
```

But what about complex stuff?

Add custom operations in plugins

COSINUS IN ESIL-SPECS?
NO!!!!!
memegenerator.net

Example: custom operations

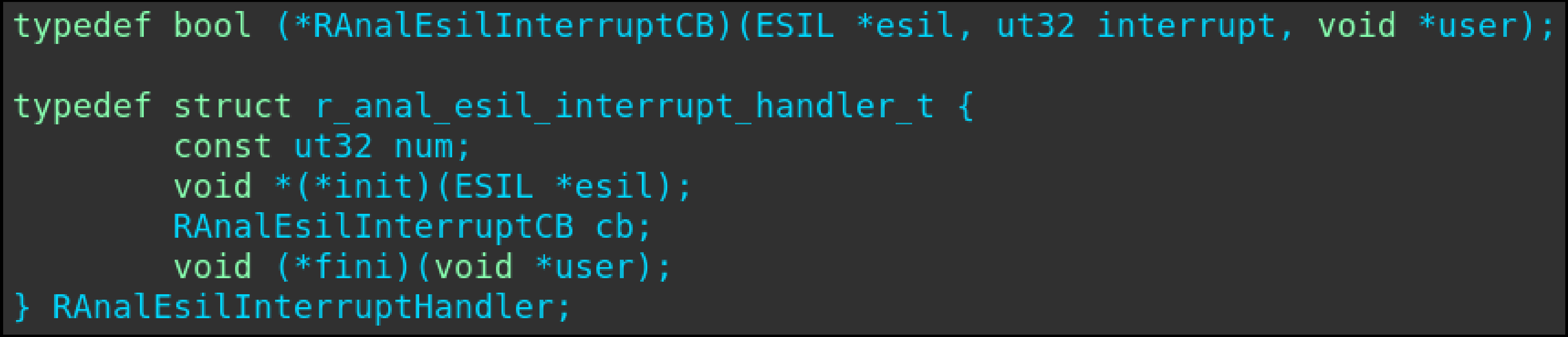
Interacting with the world

- Interrupts
- Memory mapped peripherals

New Toys: Interrupts ($)

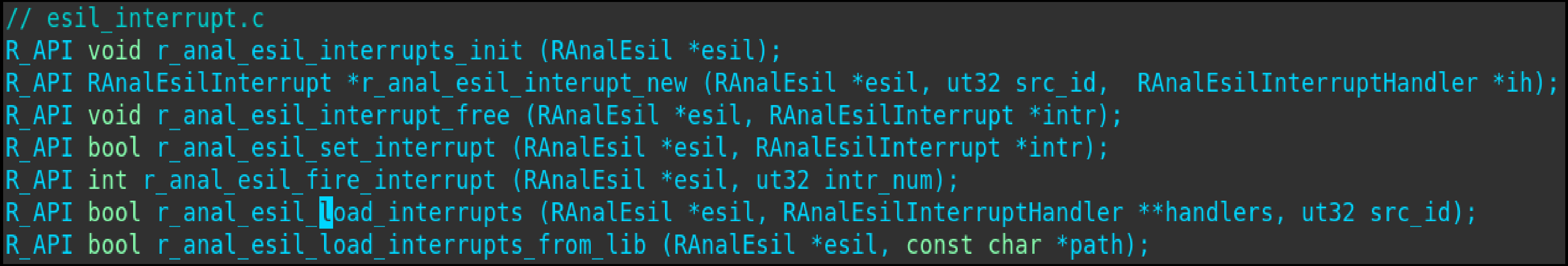
load them from shared libraries

– like plugins



```
typedef bool (*RAnalEsilInterruptCB)(ESIL *esil, ut32 interrupt, void *user);

typedef struct r_anal_esil_interrupt_handler_t {
        const ut32 num;
        void *(*init)(ESIL *esil);
        RAnalEsilInterruptCB cb;
        void (*fini)(void *user);
} RAnalEsilInterruptHandler;
```



```
// esil_interrupt.c
R_API void r_anal_esil_interrupts_init (RAnalEsil *esil);
R_API RAnalEsilInterrupt *r_anal_esil_interupt_new (RAnalEsil *esil, ut32 src_id,  RAnalEsilInterruptHandler *ih);
R_API void r_anal_esil_interrupt_free (RAnalEsil *esil, RAnalEsilInterrupt *intr);
R_API bool r_anal_esil_set_interrupt (RAnalEsil *esil, RAnalEsilInterrupt *intr);
R_API int r_anal_esil_fire_interrupt (RAnalEsil *esil, ut32 intr_num);
R_API bool r_anal_esil_load_interrupts (RAnalEsil *esil, RAnalEsilInterruptHandler **handlers, ut32 src_id);
R_API bool r_anal_esil_load_interrupts_from_lib (RAnalEsil *esil, const char *path);
```

Demo: Interrupts

TODO

- add more interrupt handlers
- create custom operations
- fix ?{
- memory mapped io (we have APIs for that)
- use internal vars

NOT TODO

- == push result on esil-stack
- add more operations to esil
  - we already have more than we need

EOF

Questions?