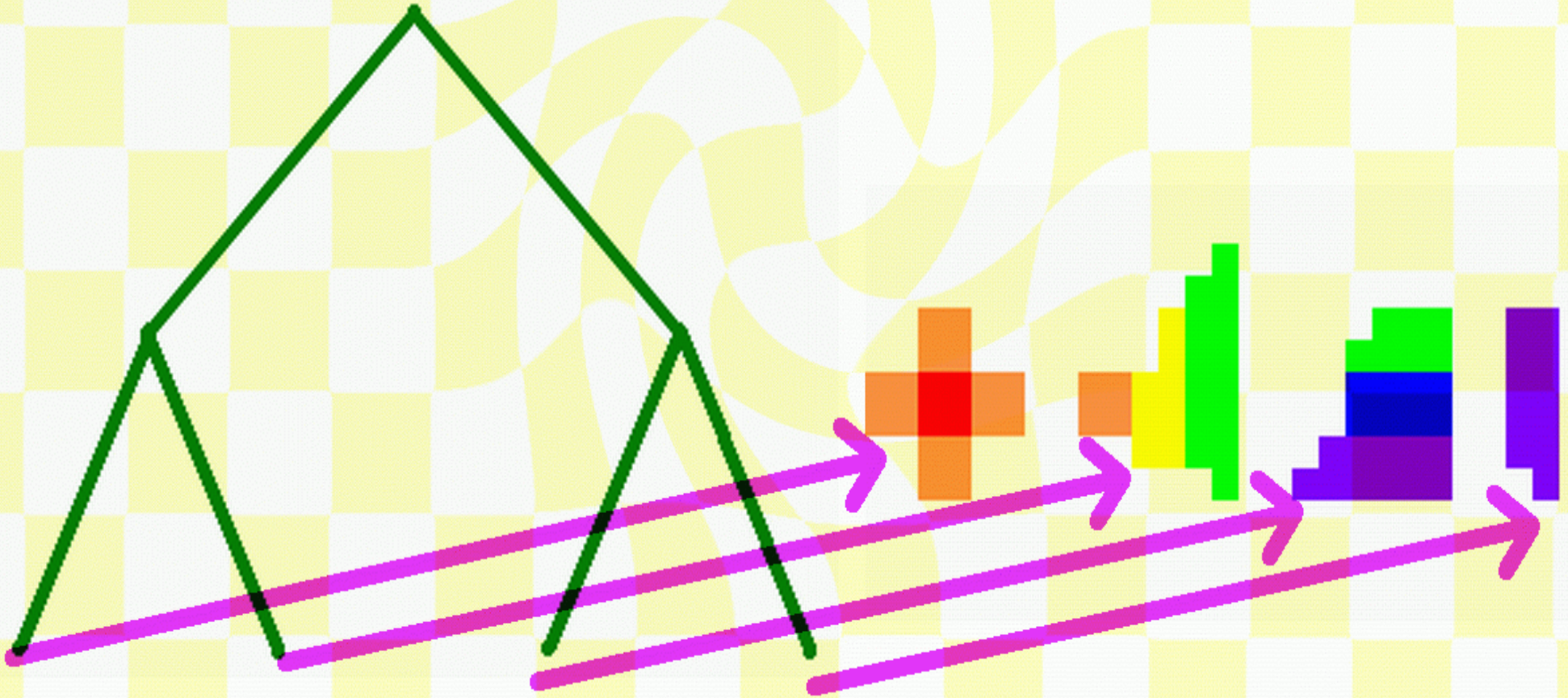
Games Based Safety Checking with MACE

Adam Bakewell & Dan Ghica
University of Birmingham, UK

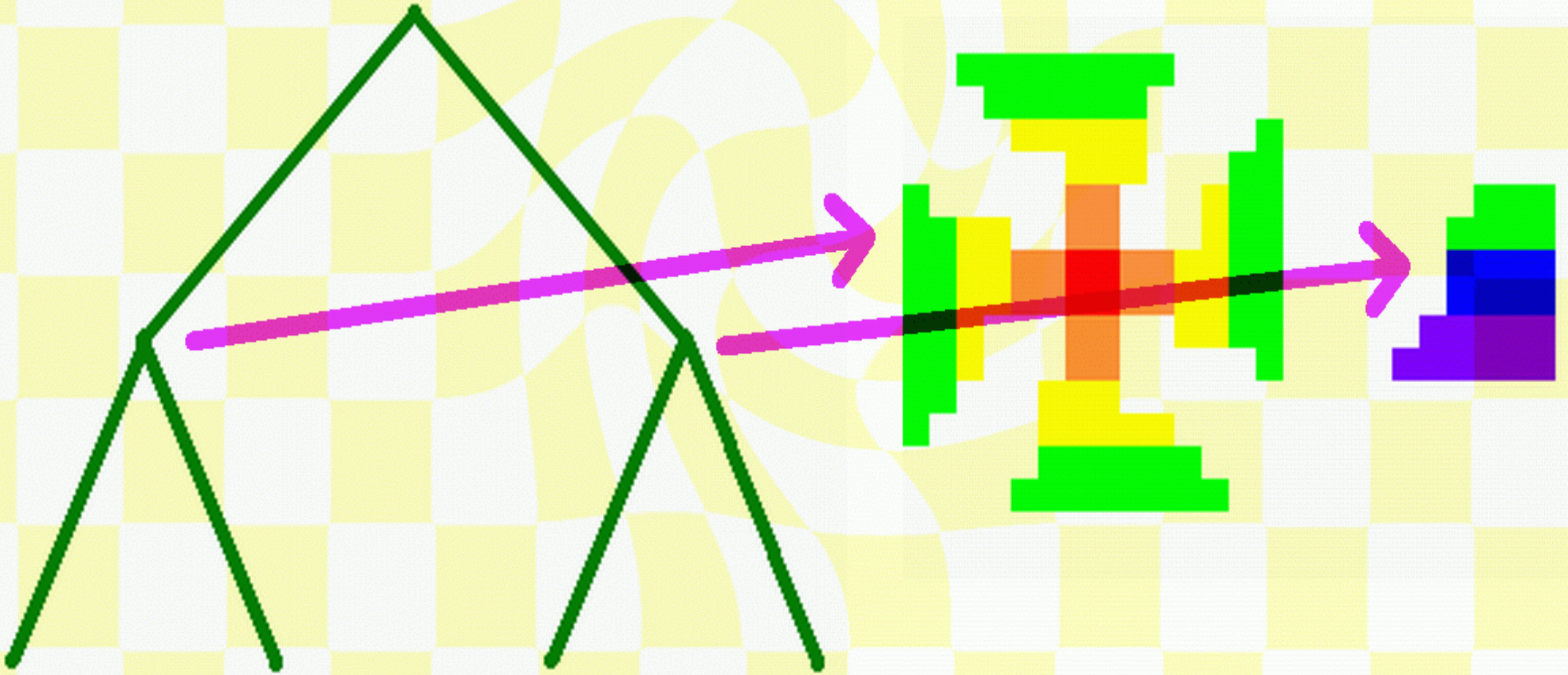
Games Based ★★★

★ Compositional Model:



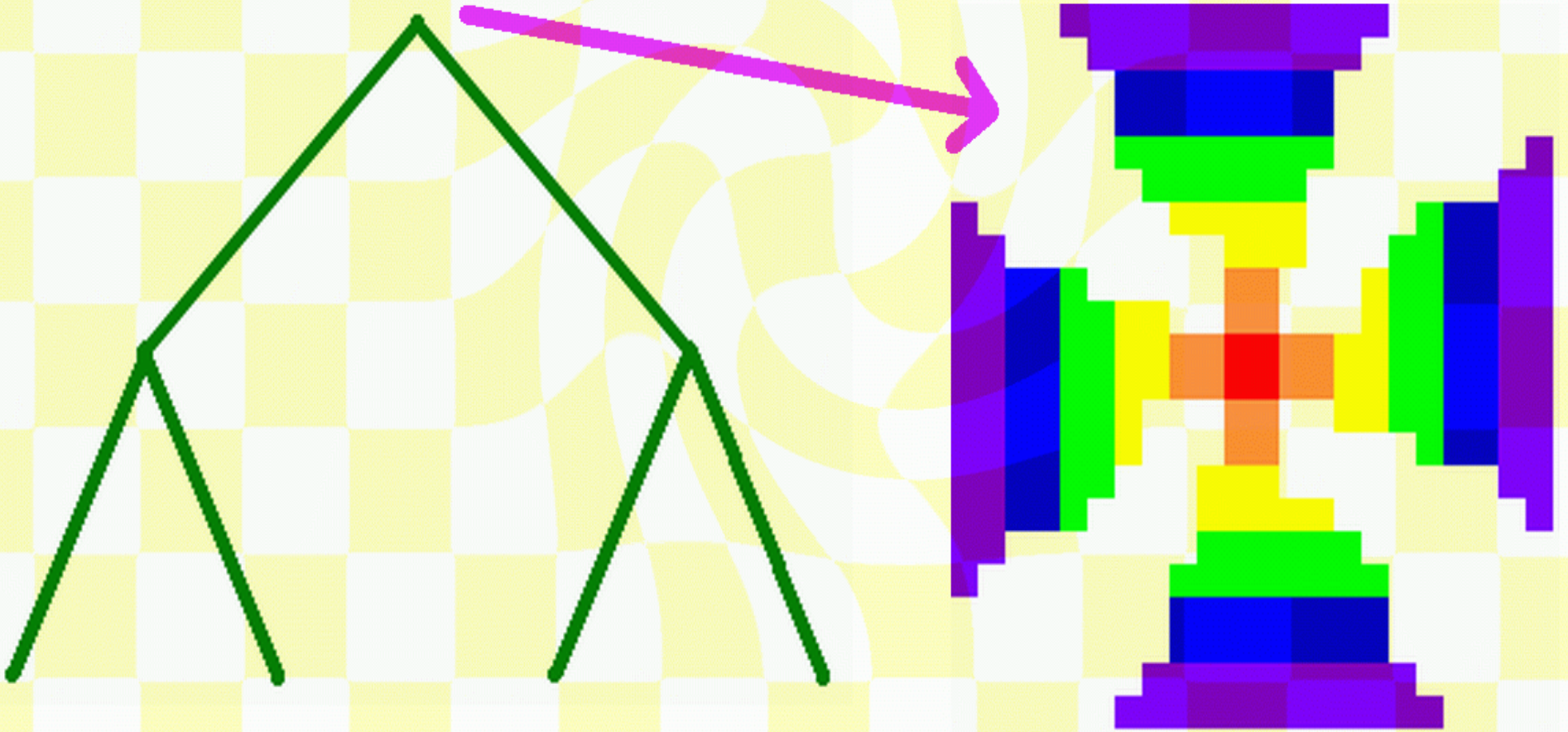
Games Based ★★★

★ Compositional Model:



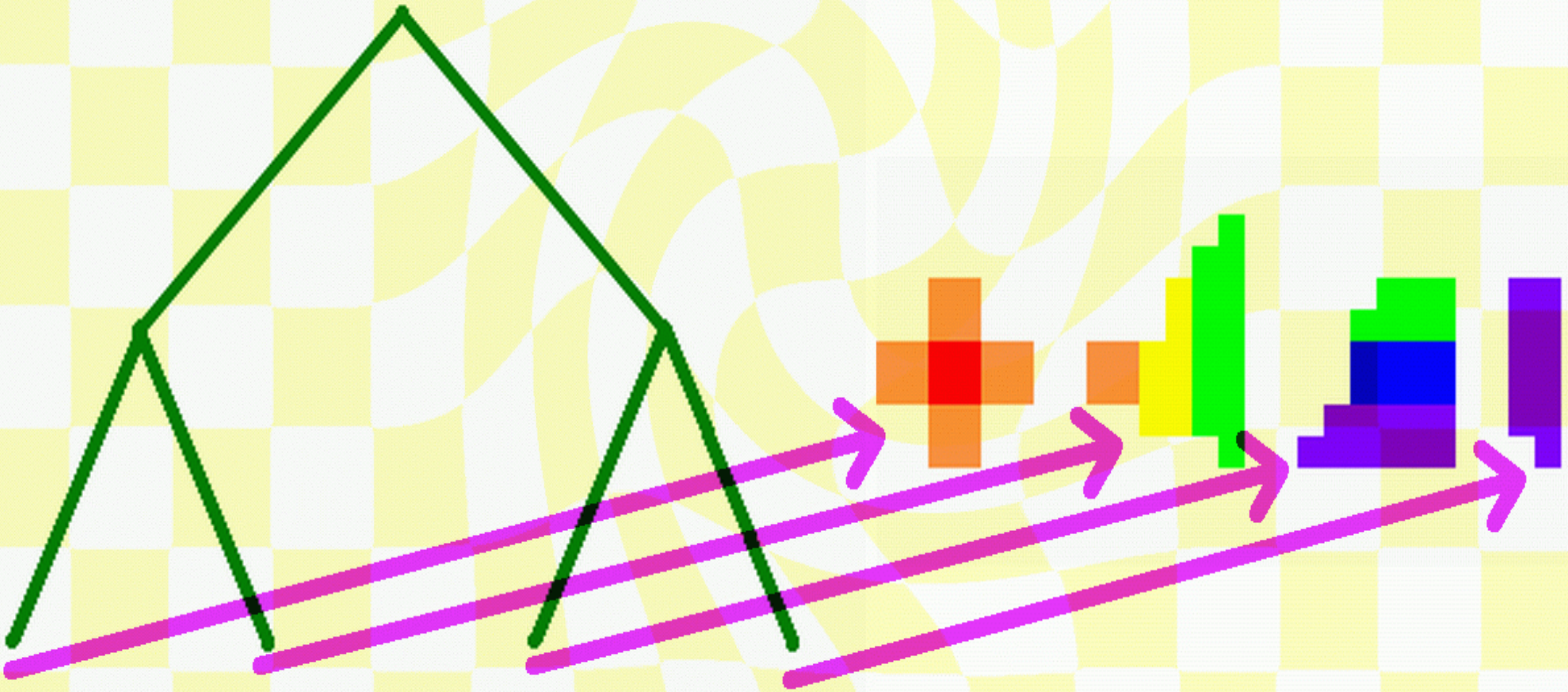
Games Based ★★★

★ Compositional Model:



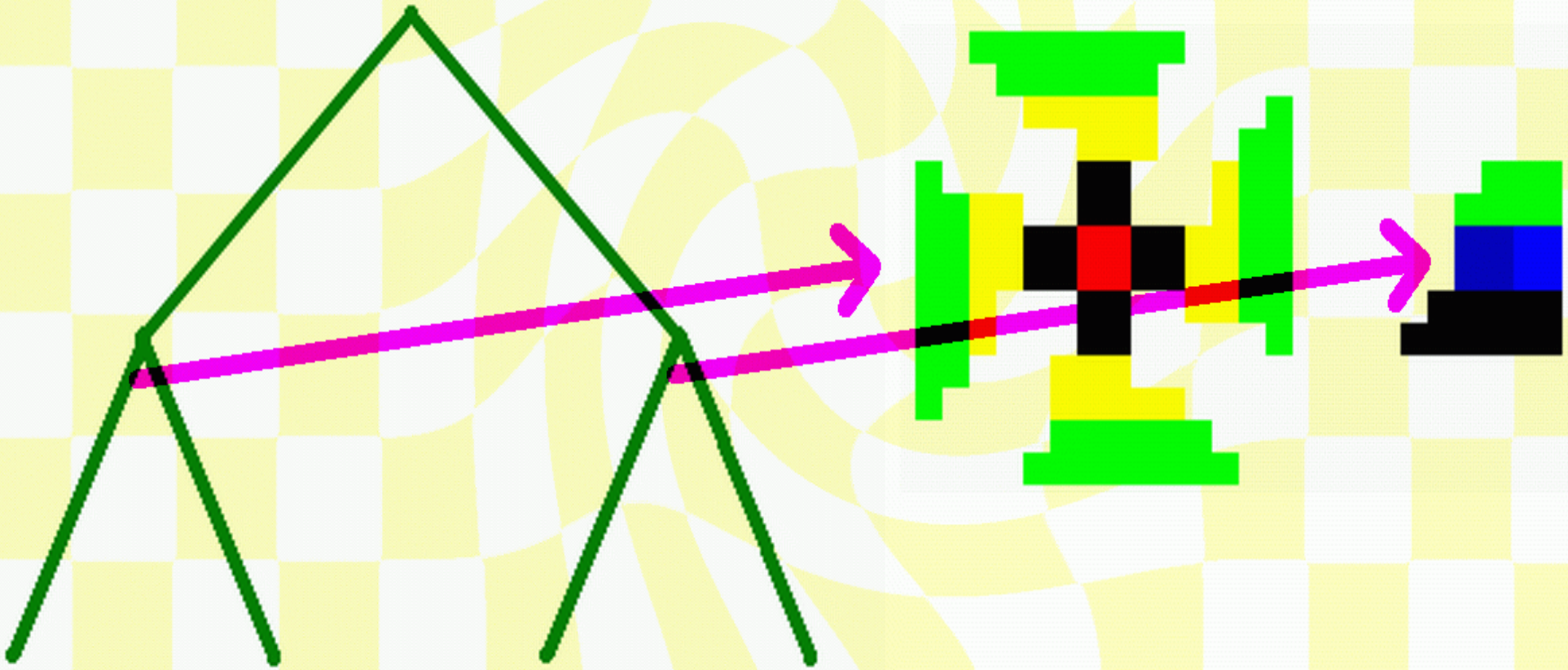
Games Based ★★★

★ Black-Box Model:



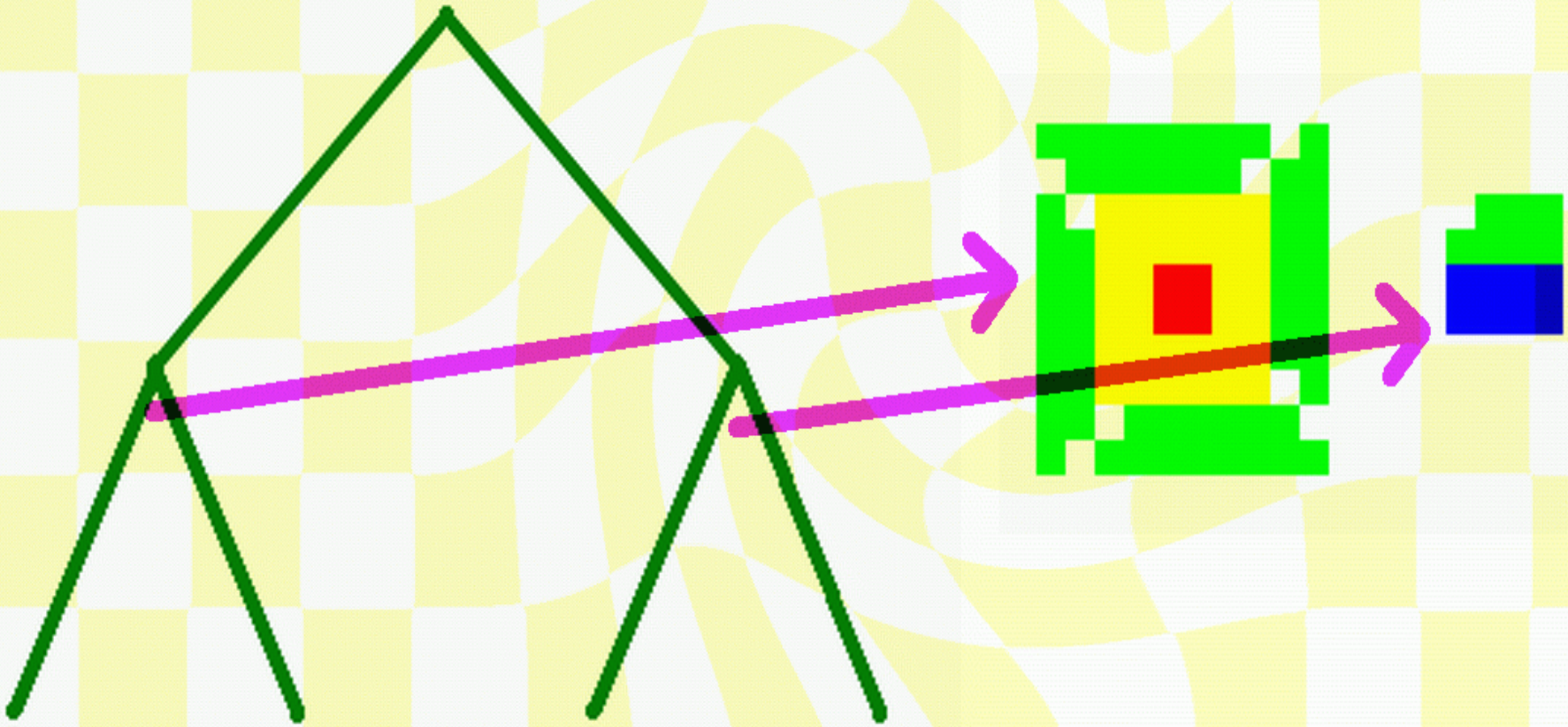
Games Based ★★★

★ Black-Box Model:



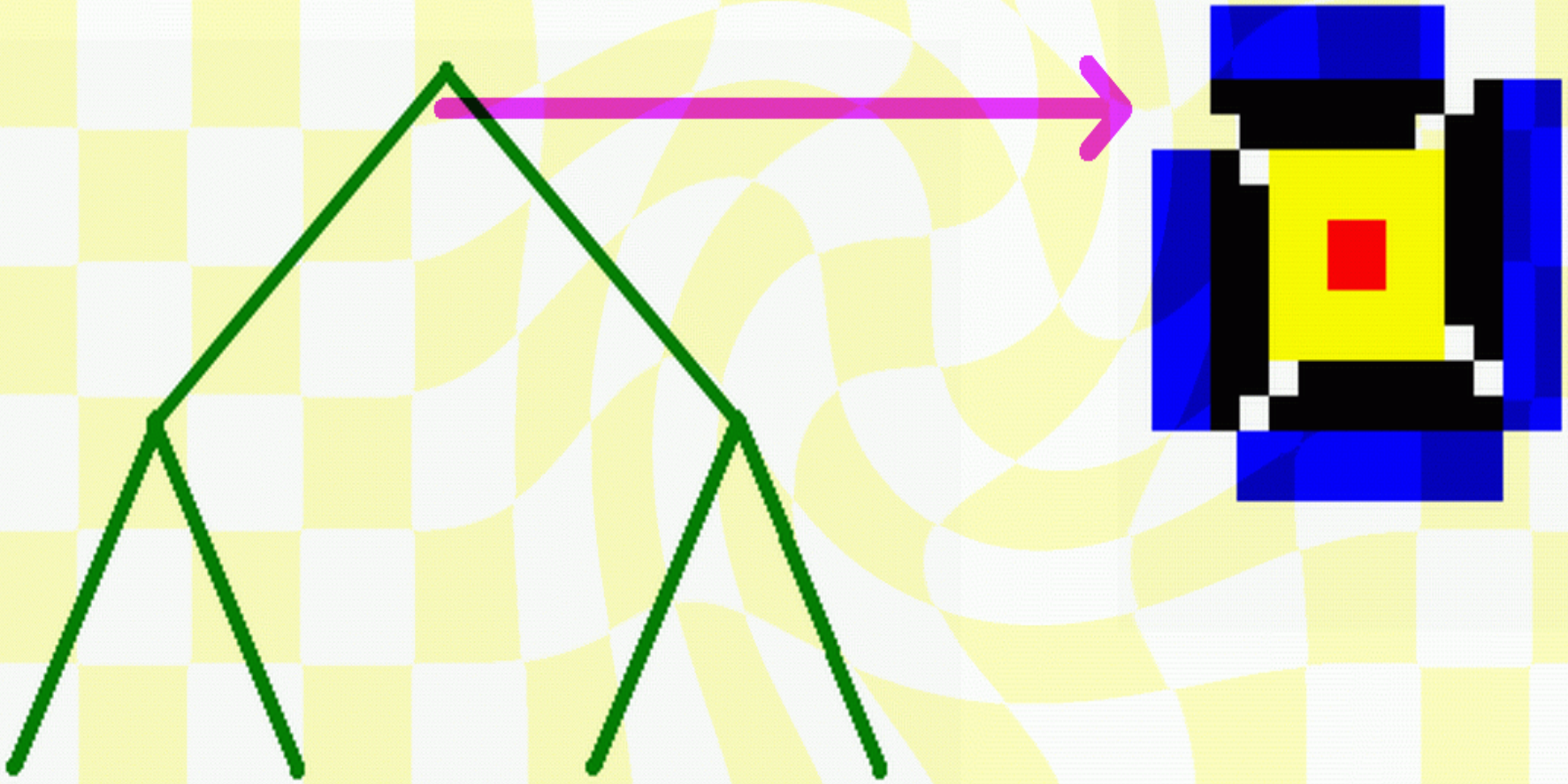
Games Based ...

★ Black-Box Model:



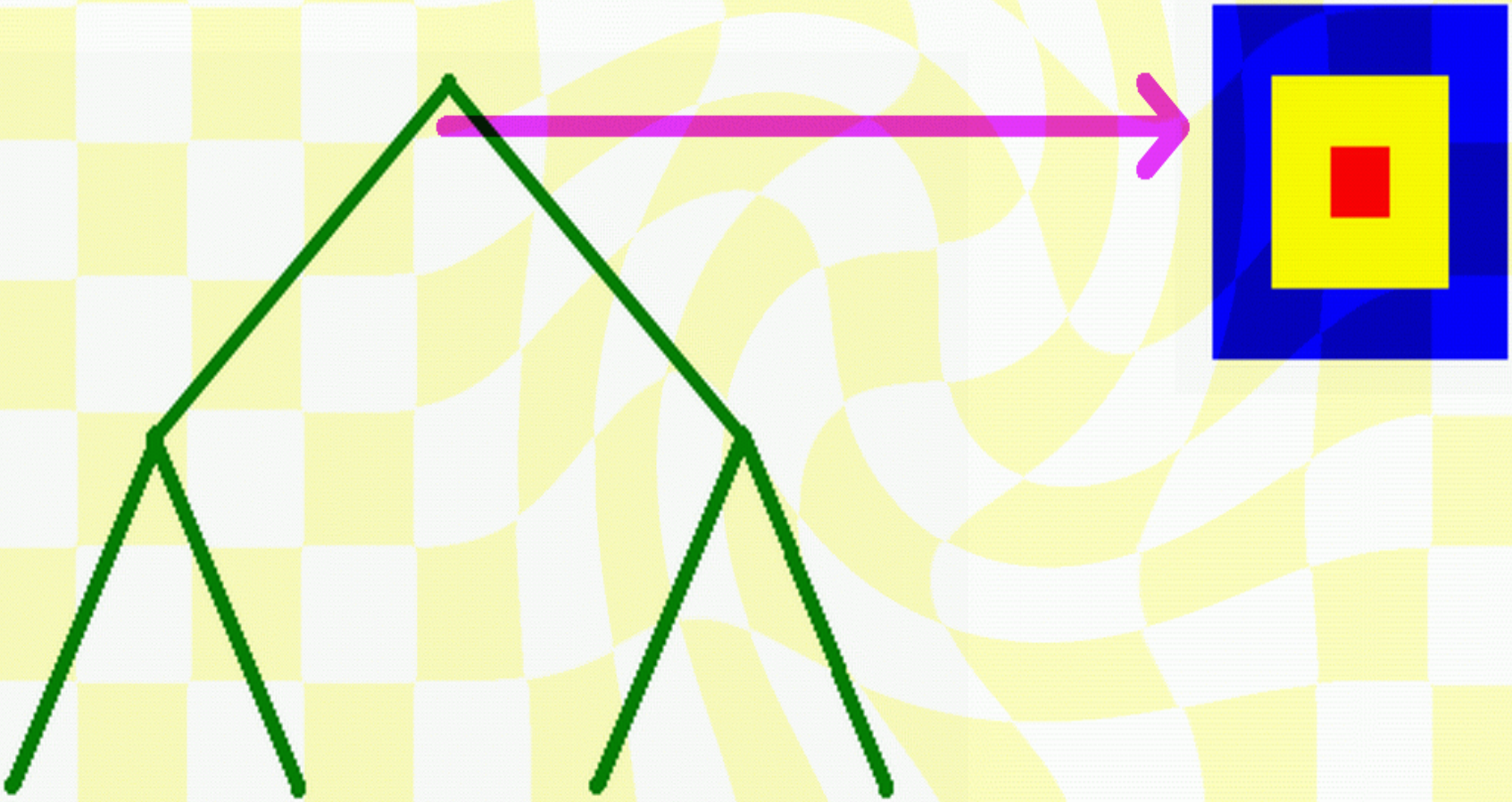
Games Based ...

★ Black-Box Model:



Games Based ★★★

★ Black-Box Model:



Games Based ...

★ Fully abstract:

- Error in program

<=>

Reachable error action in model

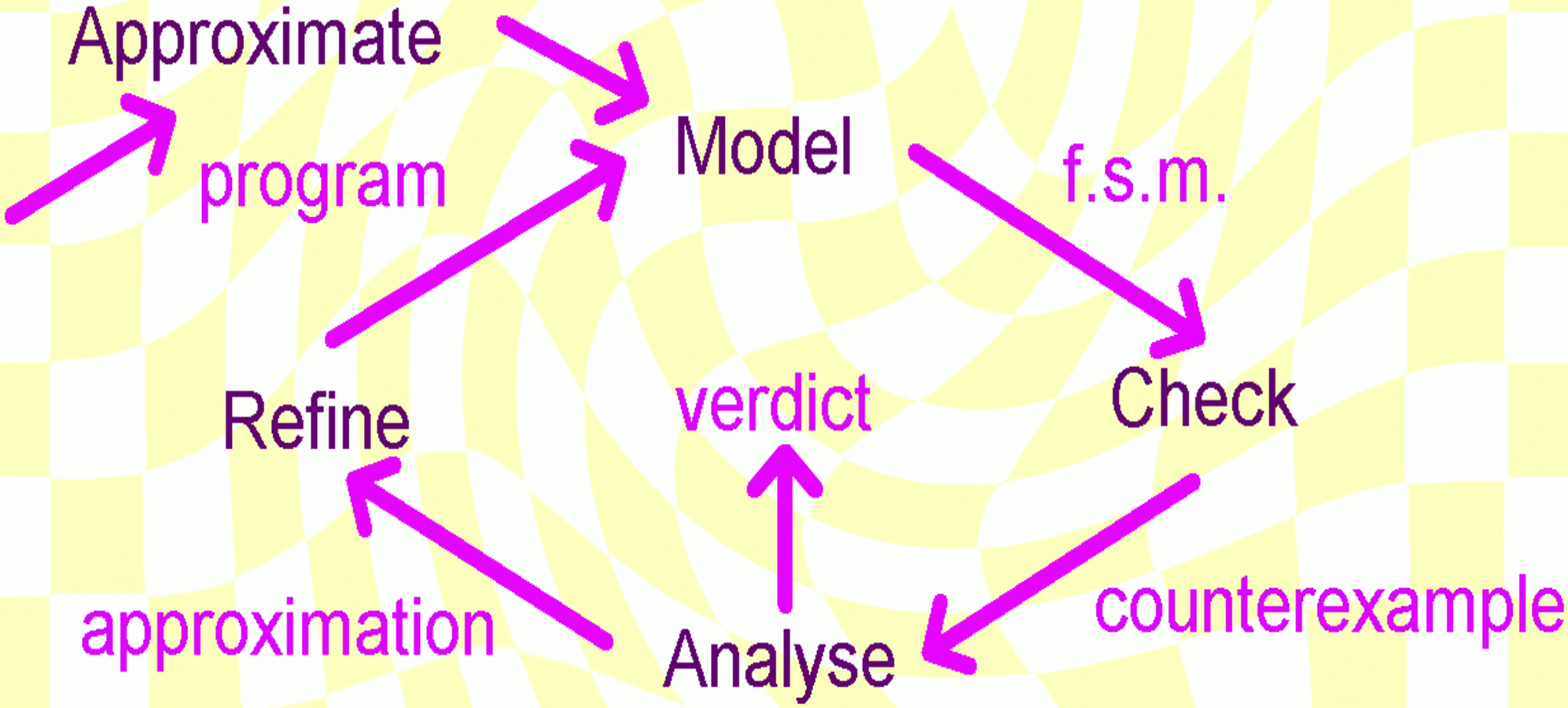
Safety Checking

★ Program unsafe iff
reachable error action in model

★ Implemented [2003-4] - works ... in principle!

Safety Checking

★ **GAMECHECKER** [2005-6] adds CEGAR:



★★★ with MAGE

★ Lazy Modelling:

- compute model parts by need

★ Lazy Checking:

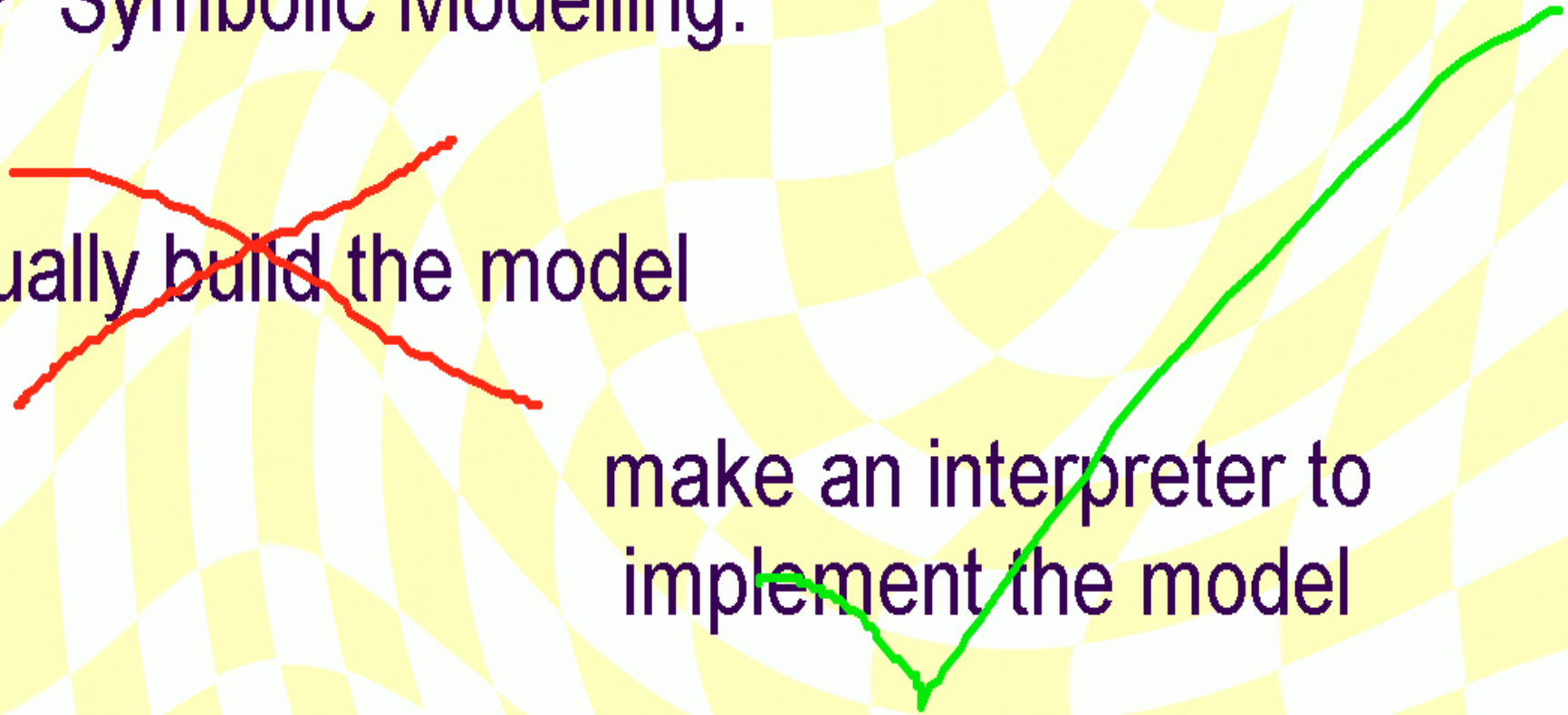
- stop checking at first error

★★★ with MAGE

★ Symbolic Modelling:

~~actually build the model~~

make an interpreter to implement the model



... with MACE

- Cheap Counterexample Test:
  - no theorem prover needed
  - accept approximated counterexamples

★★★ with MAGE

★ Cheap Refinement:

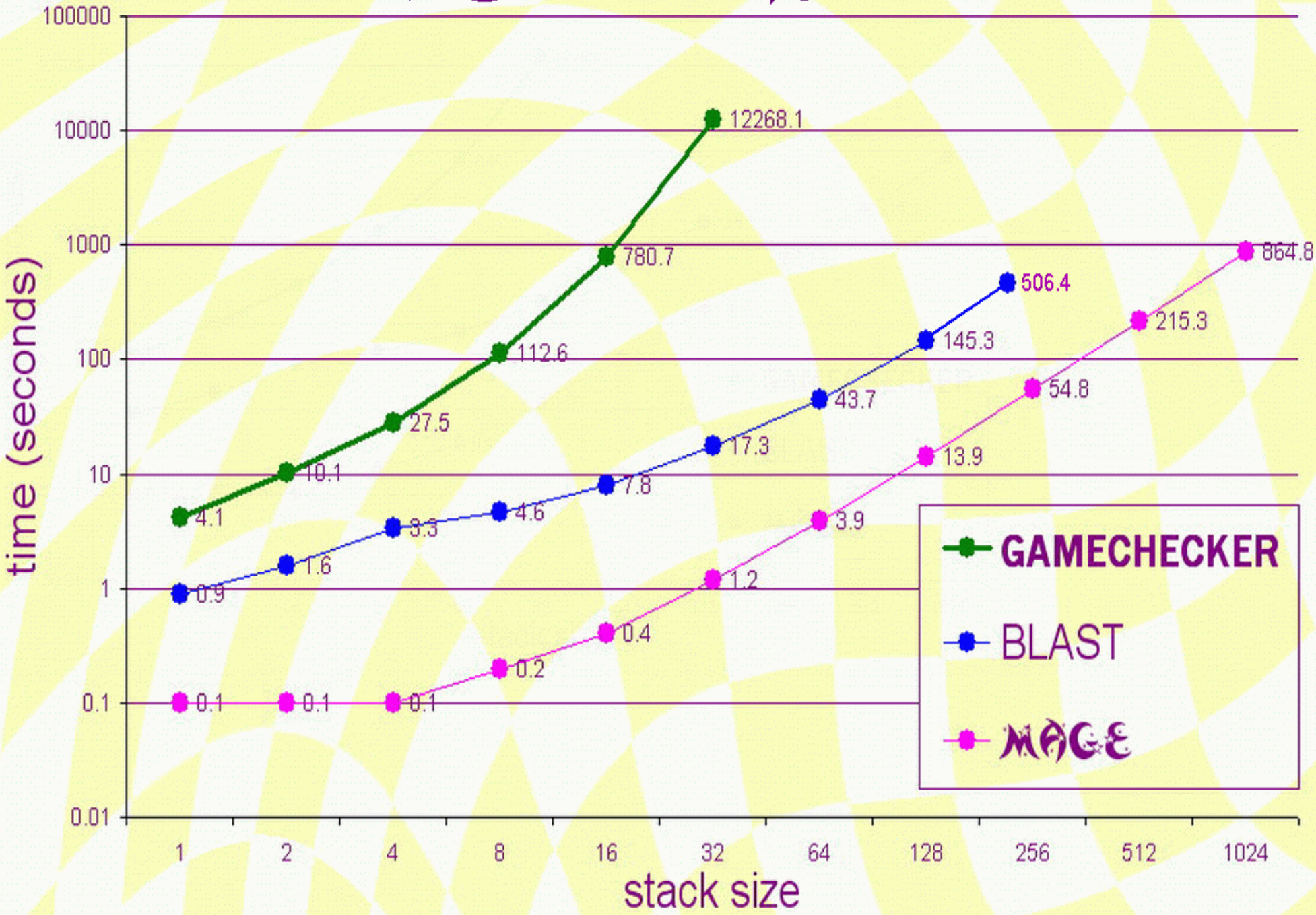
- adjust types of some leaf terms

EXAMPLE

```
oflo : com,
uflo : com,
input : nat32,
output : nat32,
analyse : com -> com -> com
```

```
new buffer : nat32[64] in
new top : nat32 := 0 in
let push=if top=64 then oflo else buffer[top++]:=input
let pop=if top=0 then uflo else output:=buffer[--top]
analyse(push,pop)
```

RESULTS


time (seconds)
100000
10000
1000
100
10
1
0.1
0.01
1
2
4
8
16
32
64
128
256
512
1024
stack size
12268.1
780.7
112.6
27.5
10.1
4.1
506.4
145.3
43.7
17.3
7.8
4.6
3.3
1.6
0.9
864.8
215.3
54.8
13.9
3.9
1.2
0.4
0.2
0.1
0.1
0.1
GAMECHECKER
BLAST
MAGE