

# A SPECTRUM OF SOUNDNESS AND PERFORMANCE

Ben Greenman



Northeastern University

Expressions

$e$

Dynamic Type

$\text{Dyn}$

Types

$\tau$

Precision relation

$\sqsubseteq$

Typing judgment

$\vdash e : \tau$

Evaluation Syntax

$e$

Semantics



Expressions

$e$

Dynamic Type

$\text{Dyn}$

Types

$\tau$

Precision relation

$\sqsubseteq$

Typing judgment

$\vdash e : \tau$

Evaluation Syntax

$e$

Semantics



# Pair of Languages

$e_S = x$

- |  $e_S e_S$
- |  $\lambda(x:\tau)e_S$
- | ....
- |  $\text{dyn } \tau e_D$

$\tau = \text{Int}$

- |  $\text{Nat}$
- |  $(\tau \times \tau)$
- |  $(\tau \rightarrow \tau)$

$e_D = x$

- |  $e_D e_D$
- |  $\lambda(x)e_D$
- | ....
- |  $\text{stat } \tau e_S$

# Typing judgment(s)

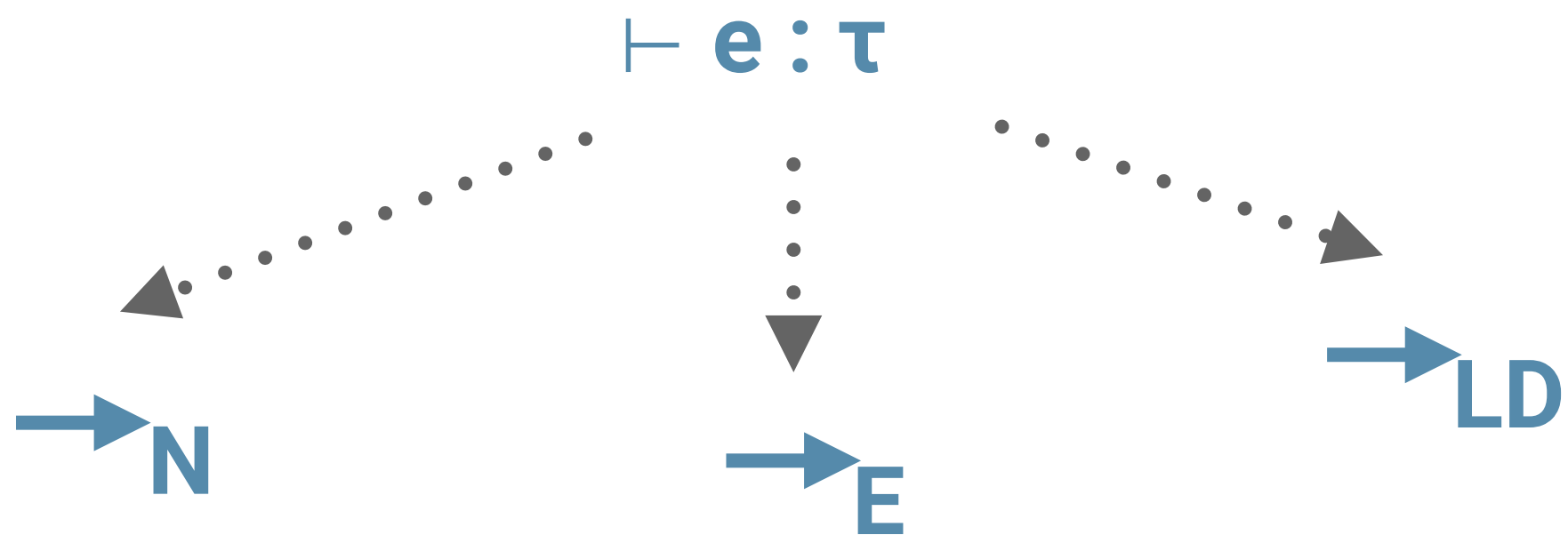
 $\vdash e_S : \tau$  $\vdash e_D : \tau$  $\vdash e_D$ 

---

 $\vdash \text{dyn } \tau \ e_D : \tau$  $\vdash e_S : \tau$ 

---

 $\vdash \text{stat } \tau \ e_S$



"types enforce levels of abstraction"

If  $\text{dyn } \tau \ v_D \rightarrow_N v_S$

then  $\vdash v_S : \tau$

# Boundary Terms

$\text{dyn Int } v_D \rightarrow_N v_D$  if  $v_D$  is an integer

$\text{dyn Nat } v_D \rightarrow_N v_D$  if  $v_D$  is a natural

$\text{dyn } (\tau \times \tau') v_D \rightarrow_N (\text{dyn } \tau v, \text{dyn } \tau' v')$   
if  $v_D = (v, v')$

$\text{dyn } (\tau \rightarrow \tau') v_D \rightarrow_N (\text{mon } (\tau \rightarrow \tau') v)$   
if  $v_D$  is a function



# Core Language

$e_S = \dots$

$| \text{mon } (\tau \rightarrow \tau') v_D$

$e_D = \dots$

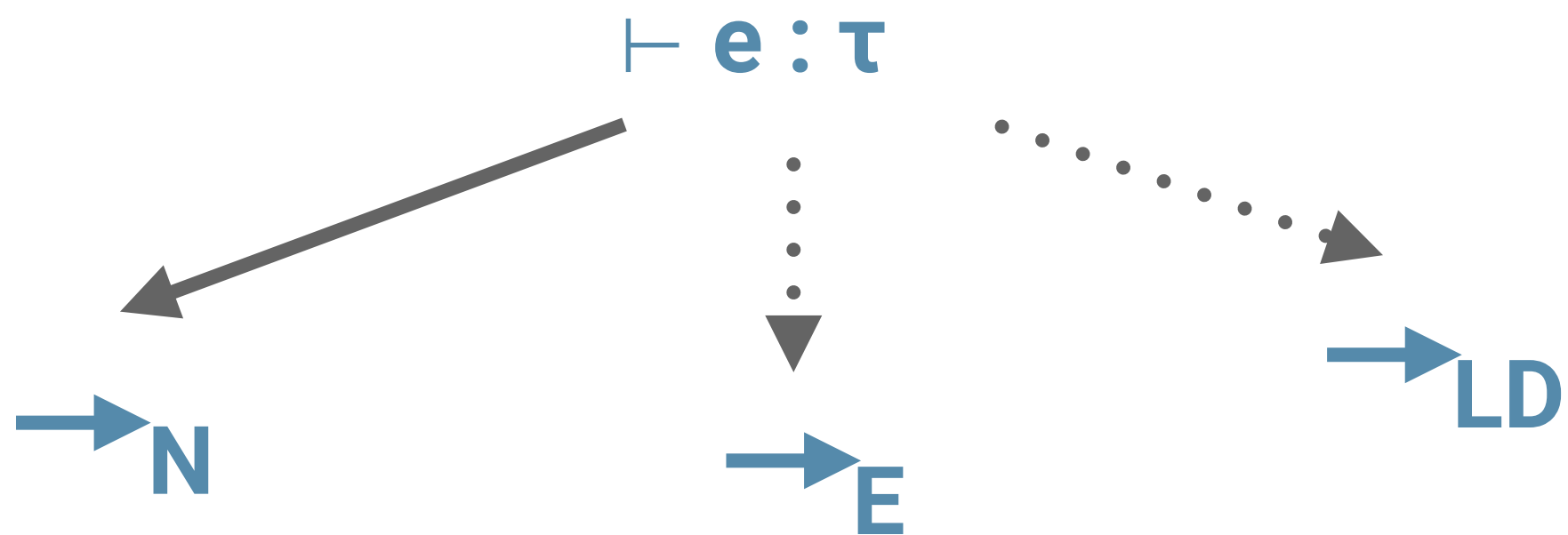
$| \text{mon } (\tau \rightarrow \tau') v_S$

$(\text{mon } (\tau \rightarrow \tau') v_D) v_S \rightarrow_N (\text{stat } \tau' (v_D (\text{dyn } \tau v_S)))$

# Soundness (Natural Embedding)

If  $\vdash e : \tau$  then either:

- $e \rightarrow_N^* v_s$  and  $\vdash v_s : \tau$
- $e \rightarrow_N^* \text{BoundaryErr}$
- $e \rightarrow_N^* E[e_D]$  and  $e_D \rightarrow_N \text{DynErr}$
- $e$  diverges



"types should not affect semantics"

If  $\text{dyn } \tau \ v_D \rightarrow_E v$   
then  $\vdash v$

# Boundary Terms

**dyn Int  $v_D$   $\rightarrow_E$   $v$**

**dyn Nat  $v_D$   $\rightarrow_E$   $v$**

**dyn ( $\tau \times \tau'$ )  $v_D$   $\rightarrow_E$   $v$**

**dyn ( $\tau \rightarrow \tau'$ )  $v_D$   $\rightarrow_E$   $v$**

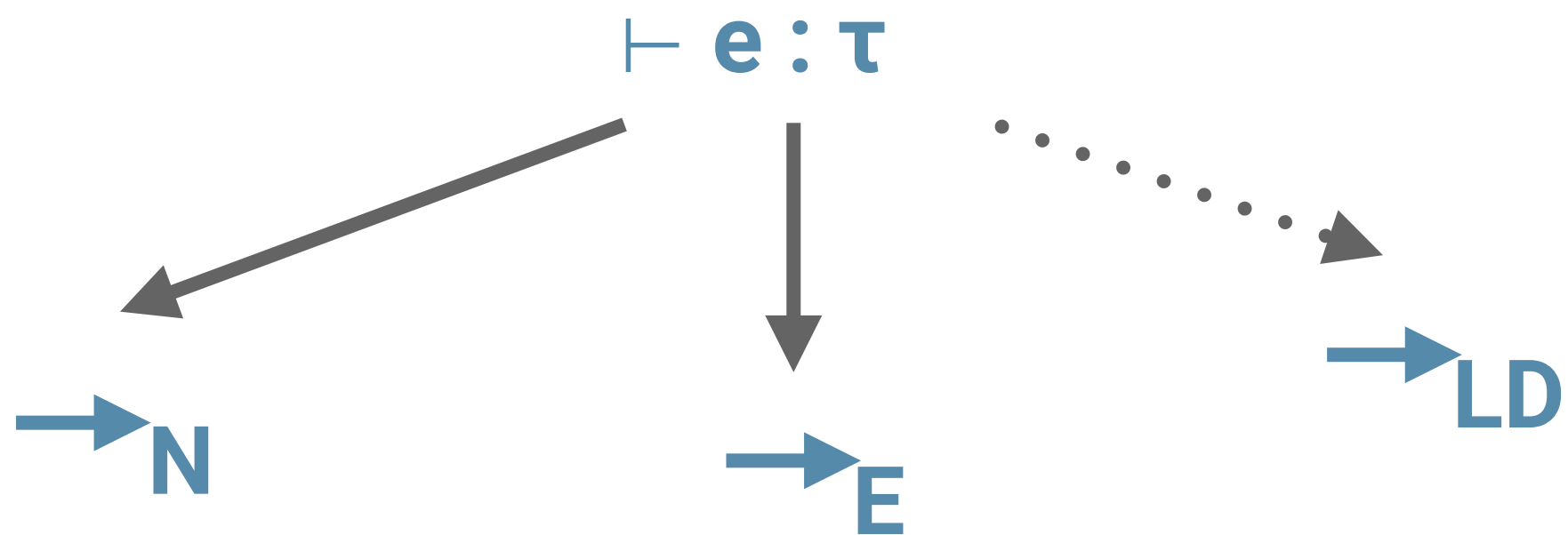
# Core Language

Same as the surface language!

# Soundness (Erasure Embedding)

If  $\vdash e : \tau$  then either:

- $e \rightarrow_E^* v$  and  $\vdash v$
- $e \rightarrow_E^* \text{BoundaryErr}$
- $e \rightarrow_E^* \text{DynErr}$
- $e$  diverges





"types prevent undefined behavior"

"types prevent undefined behavior"

**(1 2)**

**$\vdash (-3 * -4) : \text{Nat}$**

**(fst #<fun>)**

**(#<fun> + #<fun>)**

"types prevent undefined behavior"

"types prevent undefined behavior"

If  $\text{dyn } \tau \ v_D \rightarrow_{LD} v$

then  $\vdash v : \llbracket \tau \rrbracket$

where  $\llbracket \text{Int} \rrbracket = \text{Int}$                        $\llbracket \text{Nat} \rrbracket = \text{Nat}$

$\llbracket (\tau \times \tau) \rrbracket = \text{Pair}$        $\llbracket (\tau \rightarrow \tau) \rrbracket = \text{Fun}$

# Boundary Terms

$\text{dyn Int } v_D \rightarrow_{LD} v_D$  if  $v_D$  is an integer

$\text{dyn Nat } v_D \rightarrow_{LD} v_D$  if  $v_D$  is a natural

$\text{dyn } (\tau \times \tau') v_D \rightarrow_{LD} v_D$  if  $v_D$  is a pair

$\text{dyn } (\tau \rightarrow \tau') v_D \rightarrow_{LD} v_D$  if  $v_D$  is a function

✓  $(2 + (\text{dyn Int } \#<\text{fun}>))$

✗  $(2 + (\text{fst } (\text{dyn (Int x Int)} (\#<\text{fun}>, \#<\text{fun}>))))$

✗  $(2 + ((\text{dyn (Int} \rightarrow \text{Int)} \#<\text{fun}>) 3))$

**(2 + (dyn Int #<fun>))**

**(2 + (fst (dyn (Int x Int) (#<fun>, #<fun>))))**

**~~> (2 + (check Int (fst ....)))**

**(2 + ((dyn (Int->Int) #<fun>) 3))**

**~~> (2 + (check Int ((dyn ....) ....)))**

**$(\lambda(x : \text{Nat}) (x * x))$**

**$\sim\sim>$**



$(\lambda(x : \text{Nat}) (x * x))$

$\sim\sim> (\lambda(x : \text{Nat}) (\text{check Nat } x) ; (x * x))$

# Core Language

$e_S = \dots$

| **check**  $[\tau]$   $e_S$

$e_D = \dots$

**check**  $[\tau]$   $v \rightarrow_{LD} v$

if  $v$  matches constructor  $[\tau]$

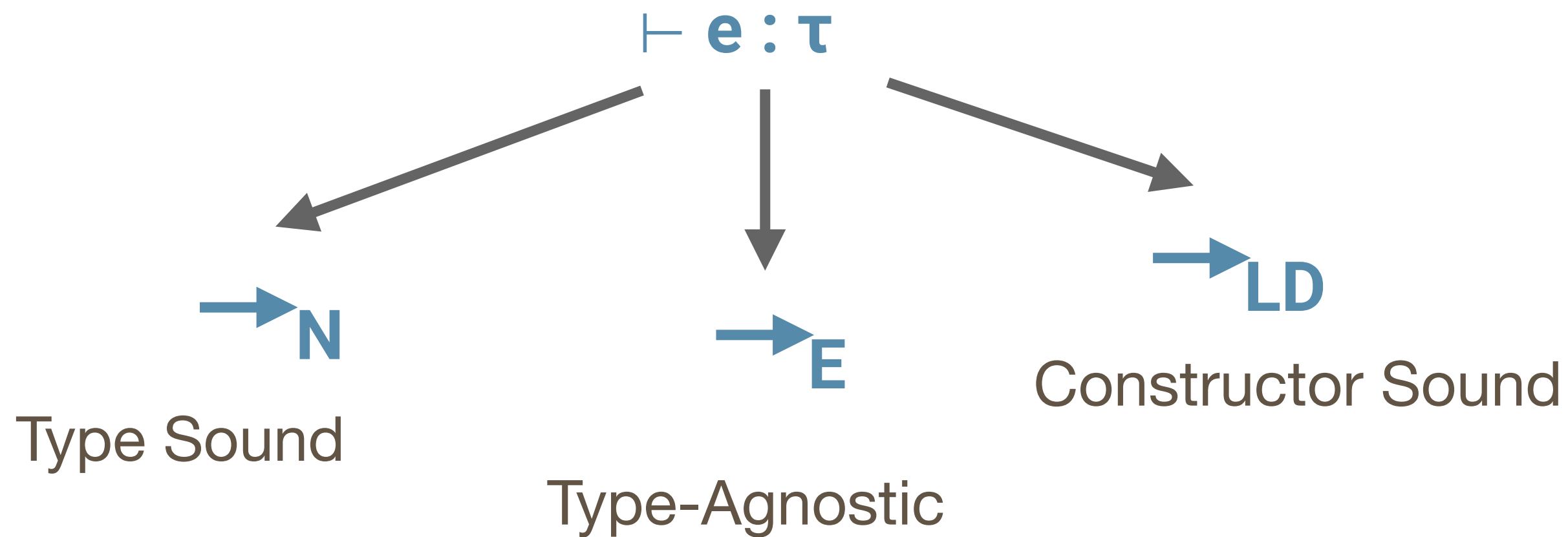
$\vdash e : \tau \rightsquigarrow e'$

$\vdash e \rightsquigarrow e'$

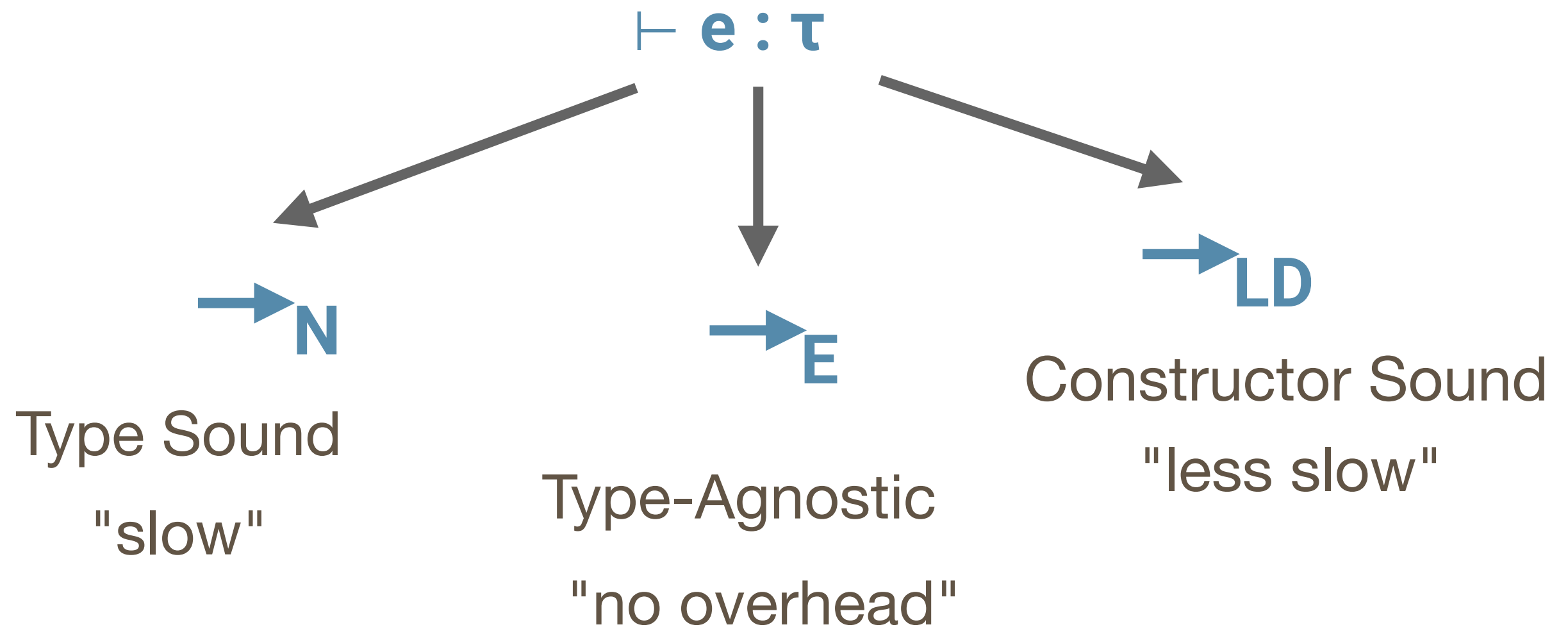
# Soundness (Locally-Defensive)

If  $\vdash e : \tau$  then  $\vdash e : \tau \rightsquigarrow e'$  and either:

- $e' \rightarrow_{LD}^* v_s$  and  $\vdash v_s : \lfloor \tau \rfloor$
- $e' \rightarrow_{LD}^* \text{BoundaryErr}$
- $e' \rightarrow_{LD}^* E[e_D]$  and  $e_D \rightarrow_{LD} \text{DynErr}$
- $e'$  diverges

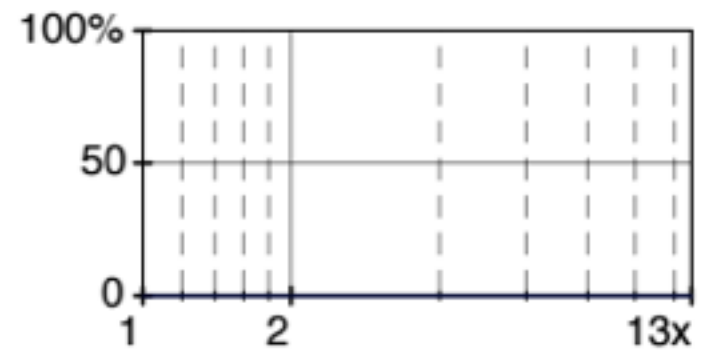


# Performance?

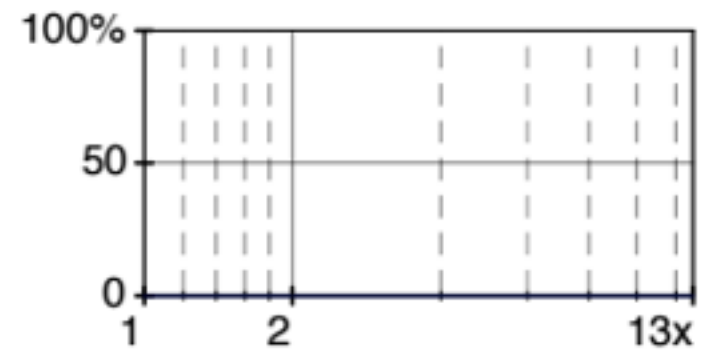


# Performance

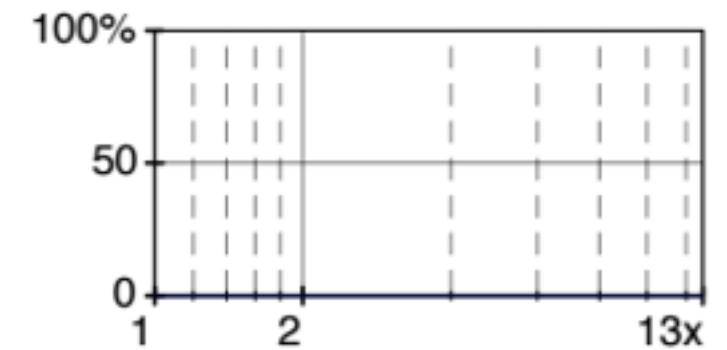
**sieve** 4 configurations



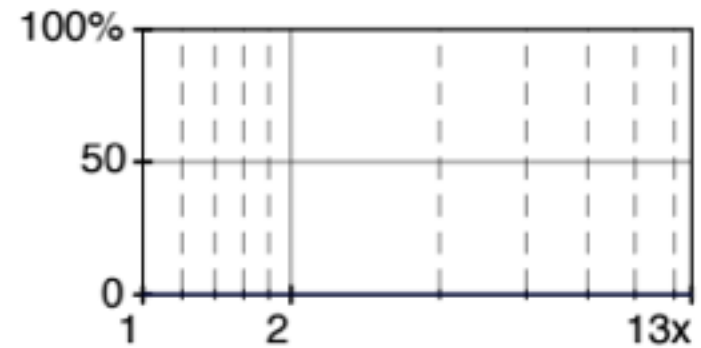
**fsm** 16 configurations



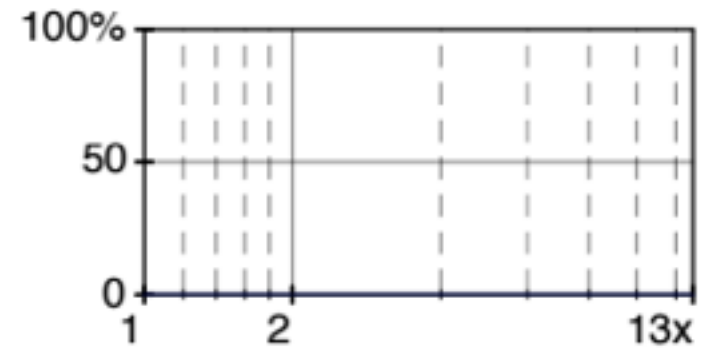
**morsecode** 16 configurations



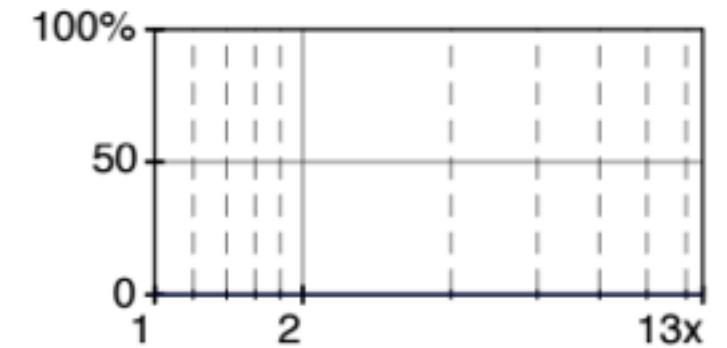
**zombie** 16 configurations



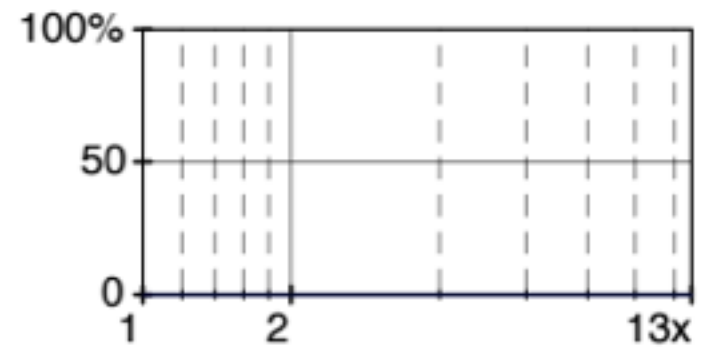
**suffixtree** 64 configurations



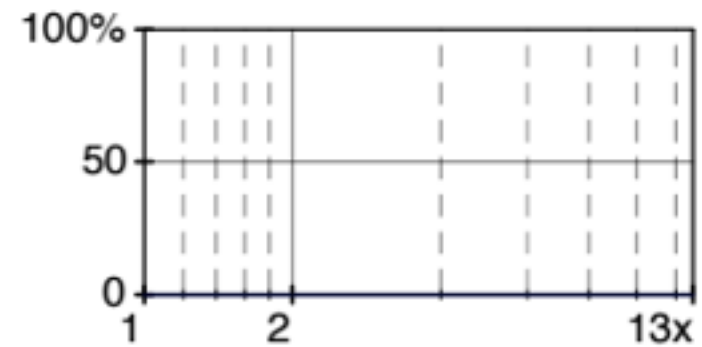
**kcfa** 128 configurations



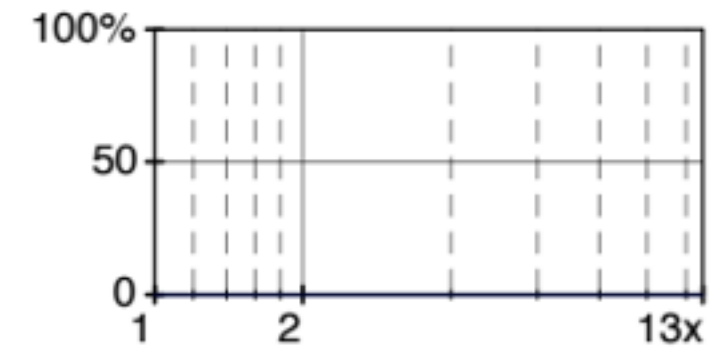
**snake** 256 configurations



**tetris** 512 configurations

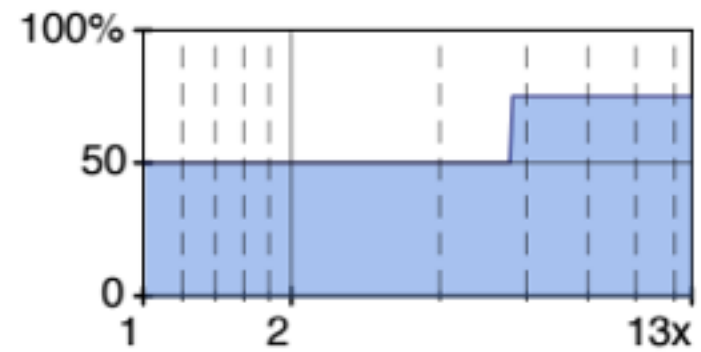


**synth** 1,024 configurations

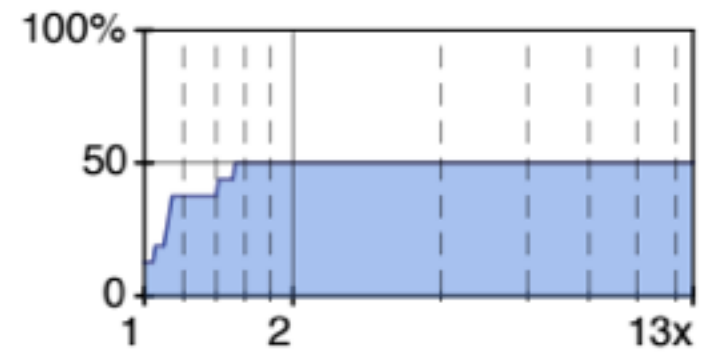


# Performance

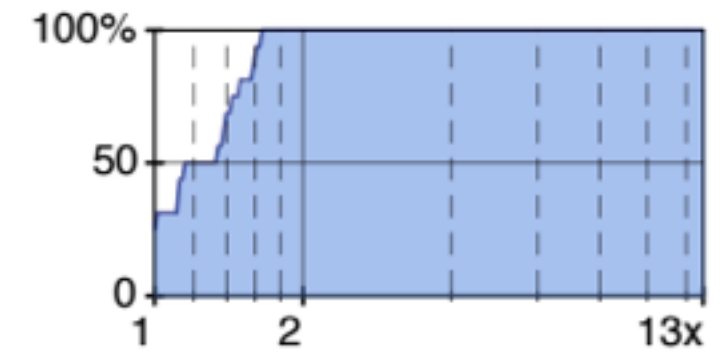
**sieve** 4 configurations



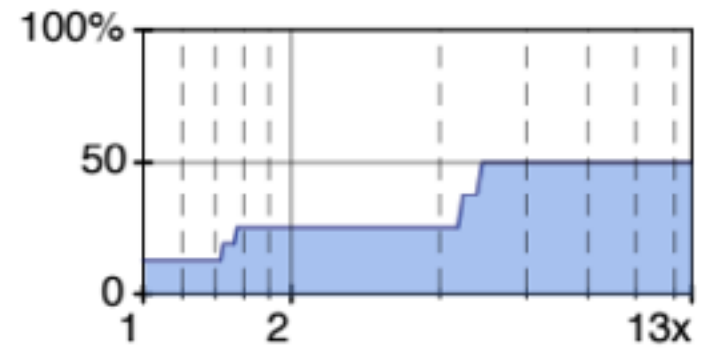
**fsm** 16 configurations



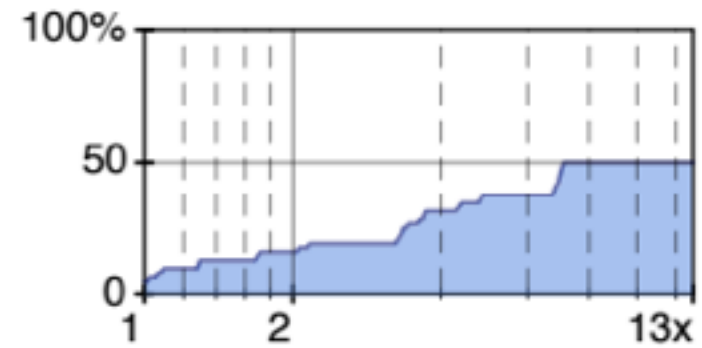
**morsecode** 16 configurations



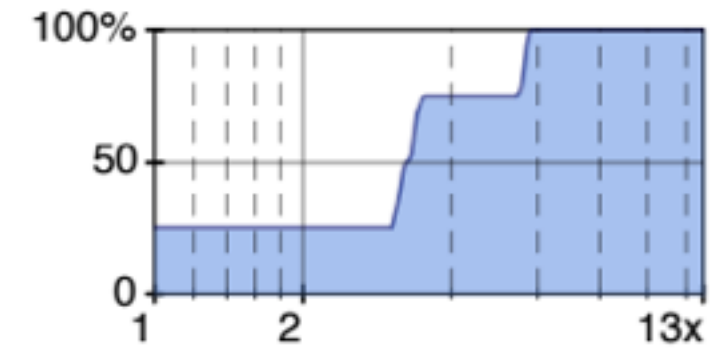
**zombie** 16 configurations



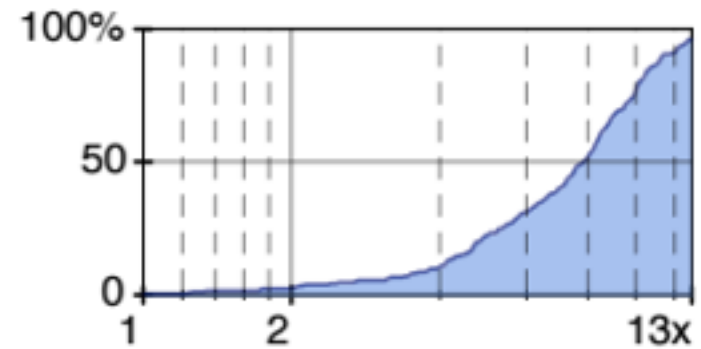
**suffixtree** 64 configurations



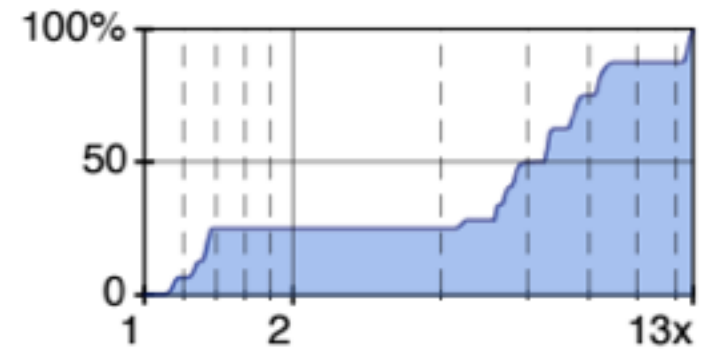
**kcfa** 128 configurations



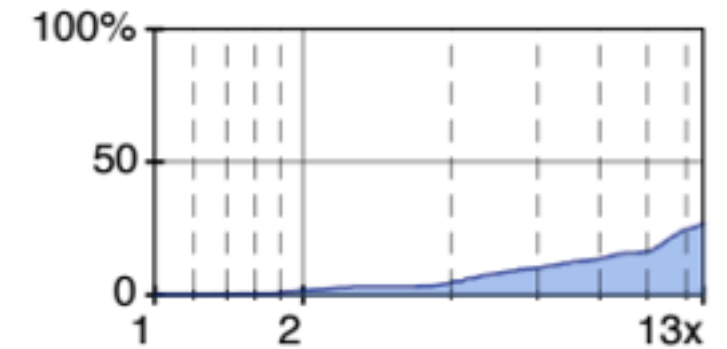
**snake** 256 configurations



**tetris** 512 configurations

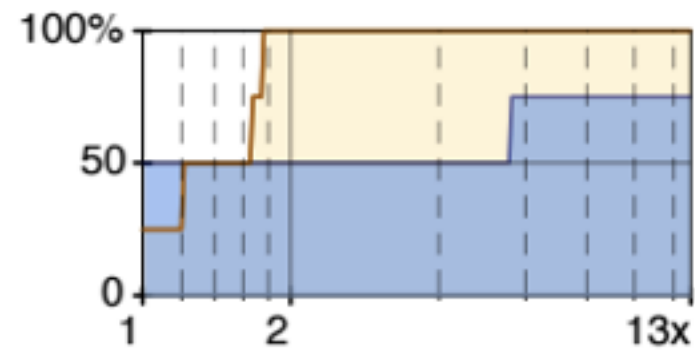


**synth** 1,024 configurations

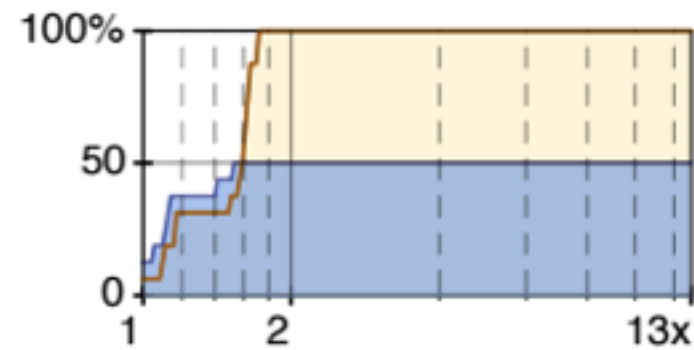


# Performance

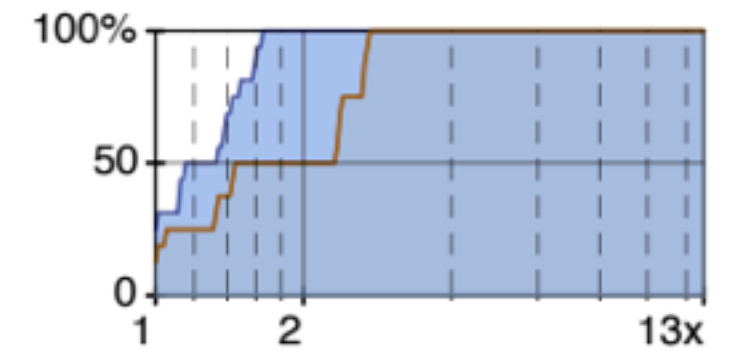
**sieve** 4 configurations



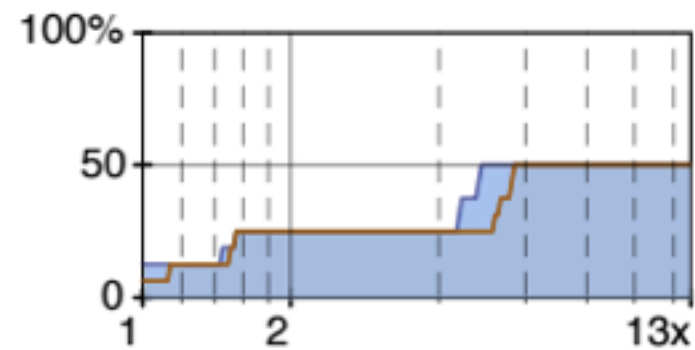
**fsm** 16 configurations



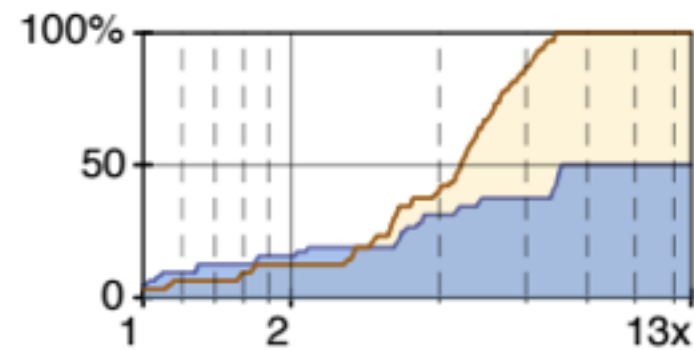
**morsecode** 16 configurations



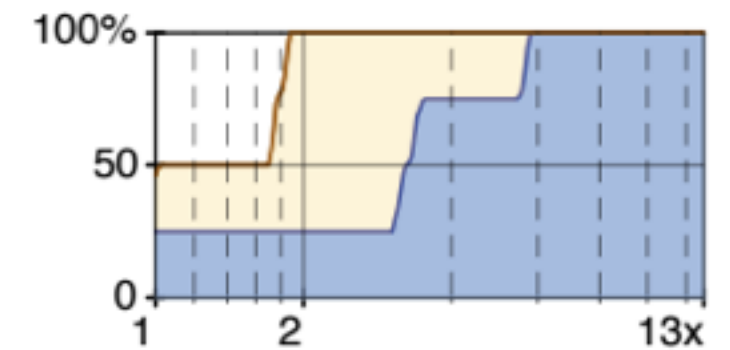
**zombie** 16 configurations



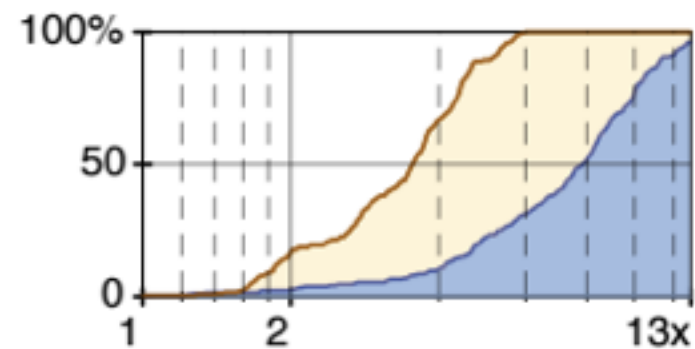
**suffixtree** 64 configurations



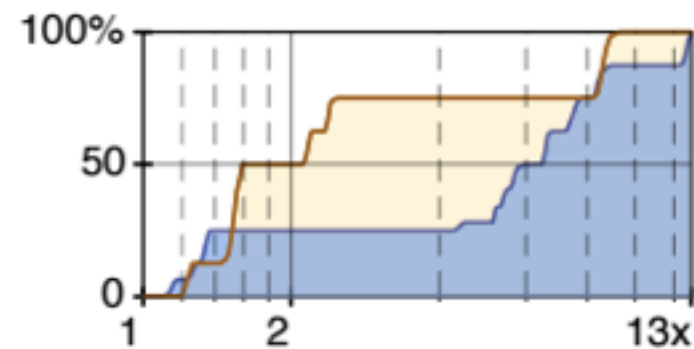
**kcfa** 128 configurations



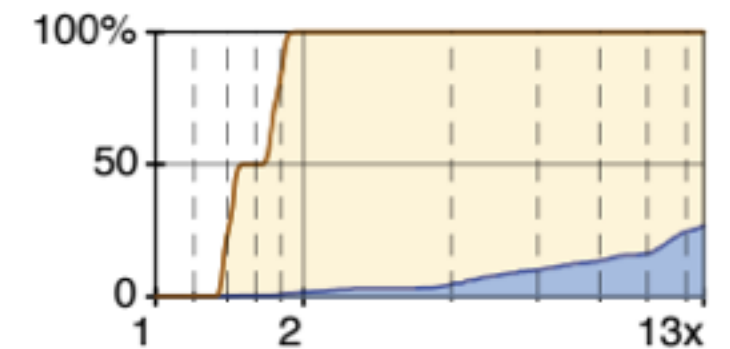
**snake** 256 configurations



**tetris** 512 configurations

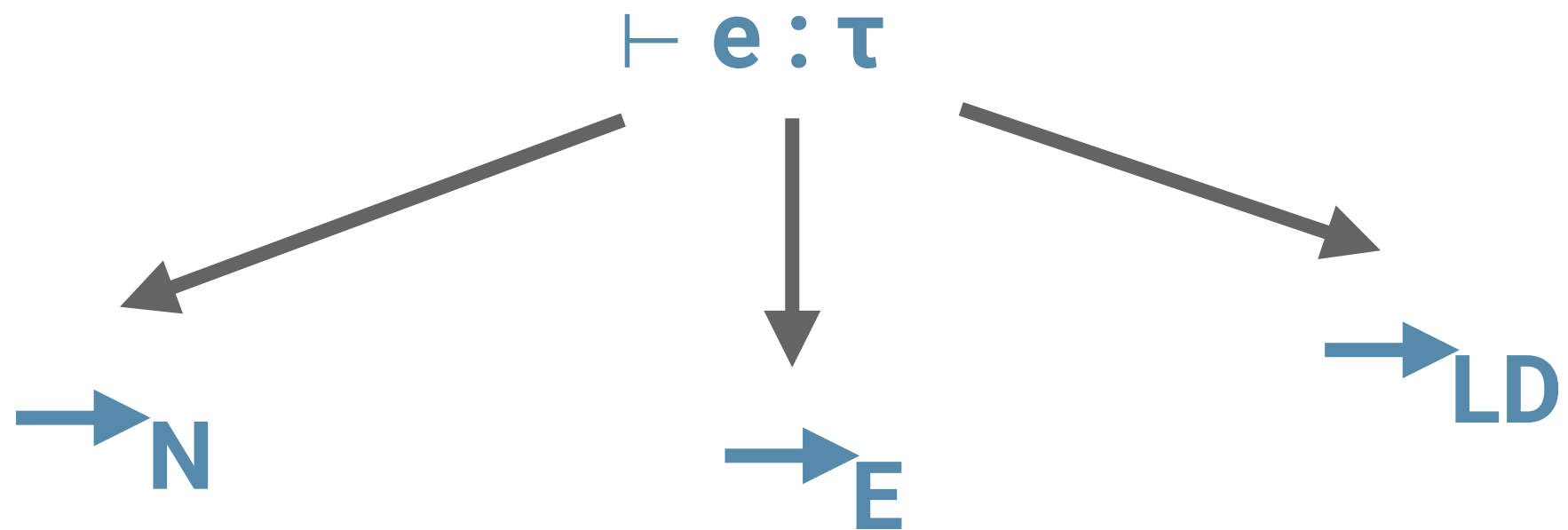


**synth** 1,024 configurations





# Metatheory



# Metatheory

- Classic type soundness for boundary-free terms  $\vdash e : \tau$   $\rightarrow_N \rightarrow_E \rightarrow_{LD}$
- For boundaries of base type,  $\rightarrow_N =_{\sim} \rightarrow_{LD}$
- For boundaries of base type,  $\rightarrow_E \neq \rightarrow_{LD}$
- For errors,  $\rightarrow_N \leq \rightarrow_{LD} \leq \rightarrow_E$

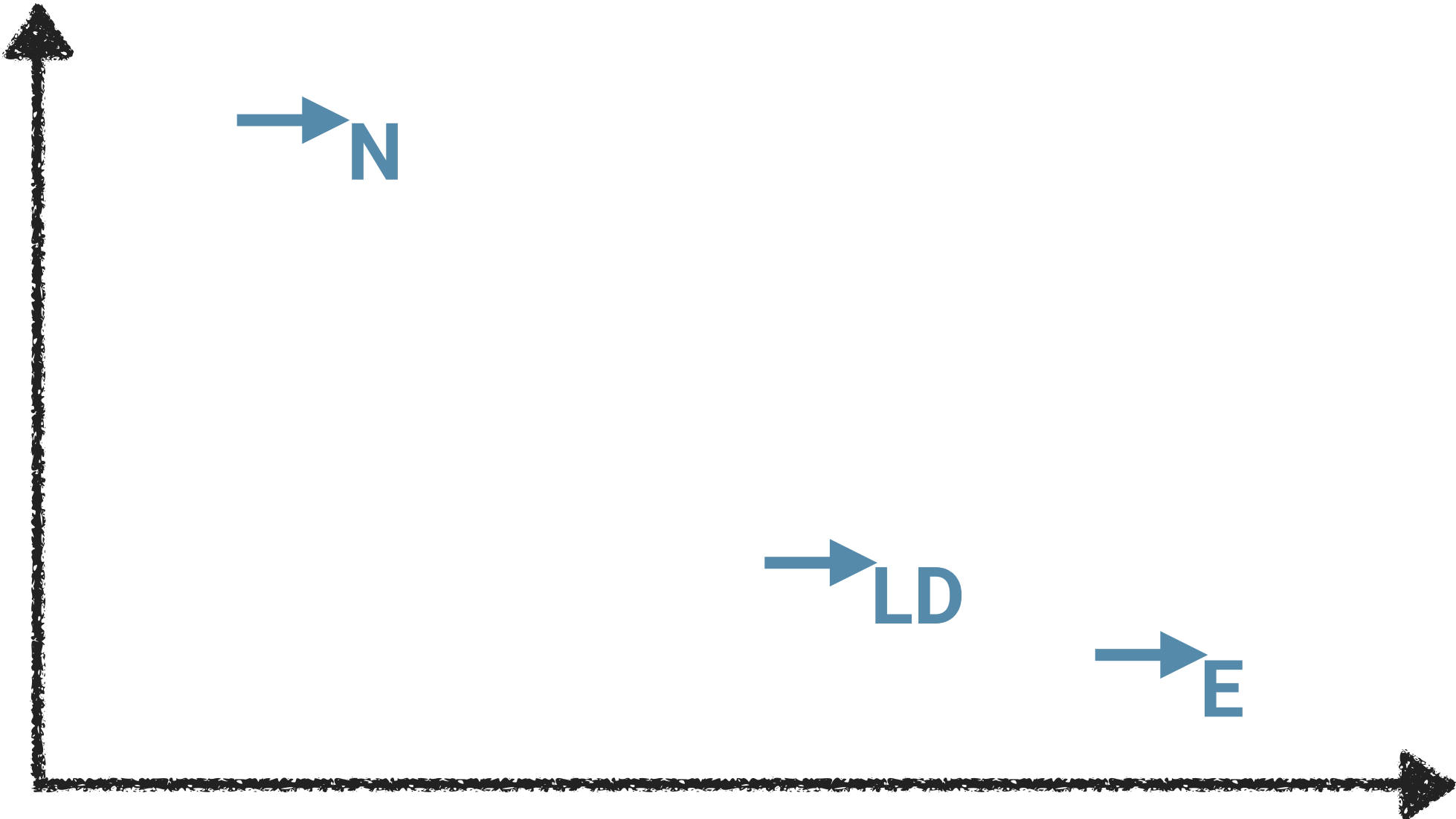
Soundness

→ N

→ LD

→ E

Performance



Soundness

→ N

→ LD

→ E

People?

Performance

