# Search Combinators

Tom Schrijvers

UNIVERSITEIT GENT

with Guido Tack, Pieter Wuille, Horst Samulowitz, Peter Stuckey

## Search heuristics are crucial.

# Support for Search?

General Purpose Programming Language

“everthing is possible, nothing is easy”

Solver-Provided Options

“everything is easy, nothing is possible”

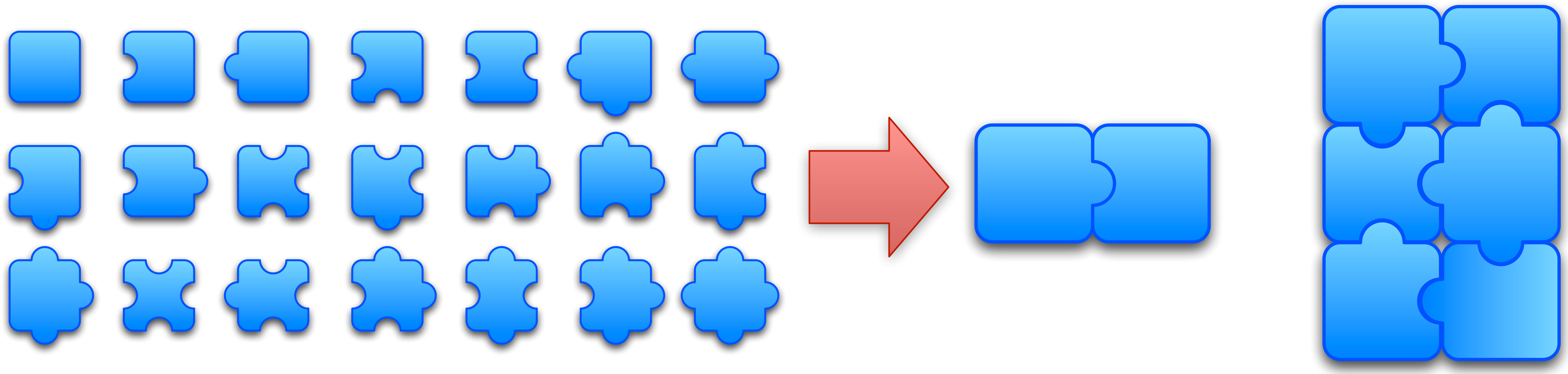
## Can we do better?

✓ Lots of expressivity and flexibility

✓ Lots of productivity through high-level specifications

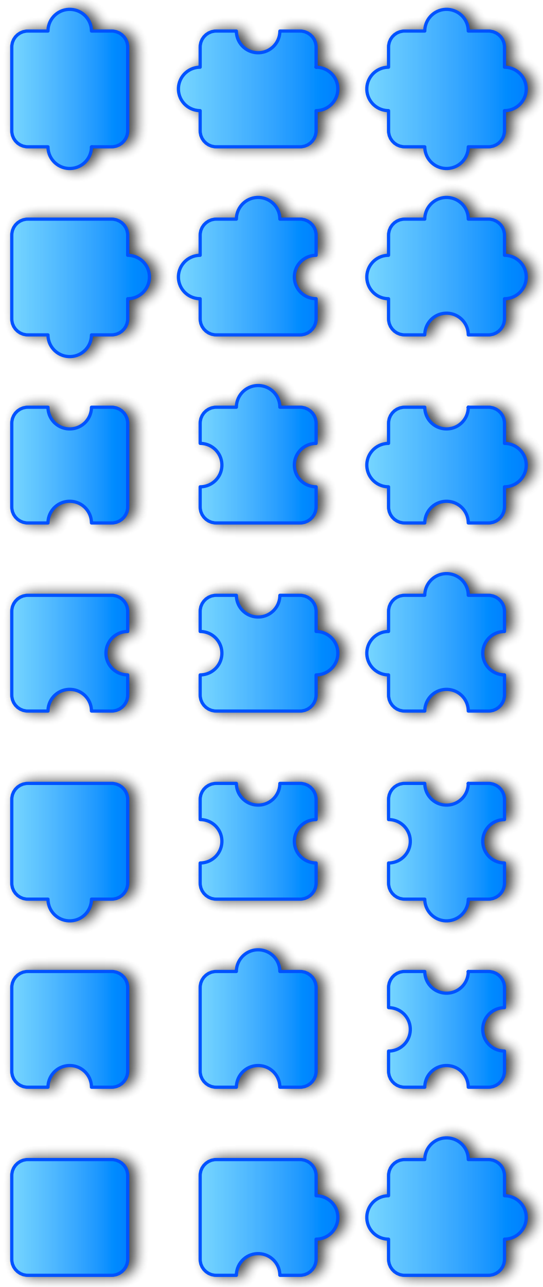
# Yes: Search Combinators

High-level building blocks



"Everything is possible and easy"

# Combinators



prune
let($v$,$e$,$s$)
assign($v$,$e$)
post($c$,$s$)
if($c$,$s_1$,$s_2$)
and([$s_1$, $s_2$,..., $s_n$])
or([$s_1$, $s_2$,..., $s_n$])
portfolio([$s_1$, $s_2$,..., $s_n$])
restart($c$,$s$)

# Reusable Abstractions

```
limit(c,s) ≡ if(c,s,prune)

for(v,l,u,s) ≡ ...

lds(s) ≡
   for(n,0,∞,
      limit(discrepancy ≤ n,s)
   )
```

# More Examples

`bab`$(obj,s)$
`restart_bab`$(obj,s)$
`dicho`$(obj,s,lb,ub)$
`id`$(s)$
`hot_start`$(c,s_1,s_2)$
...
`radiotherapy`

see paper

## Syntax

## vs.

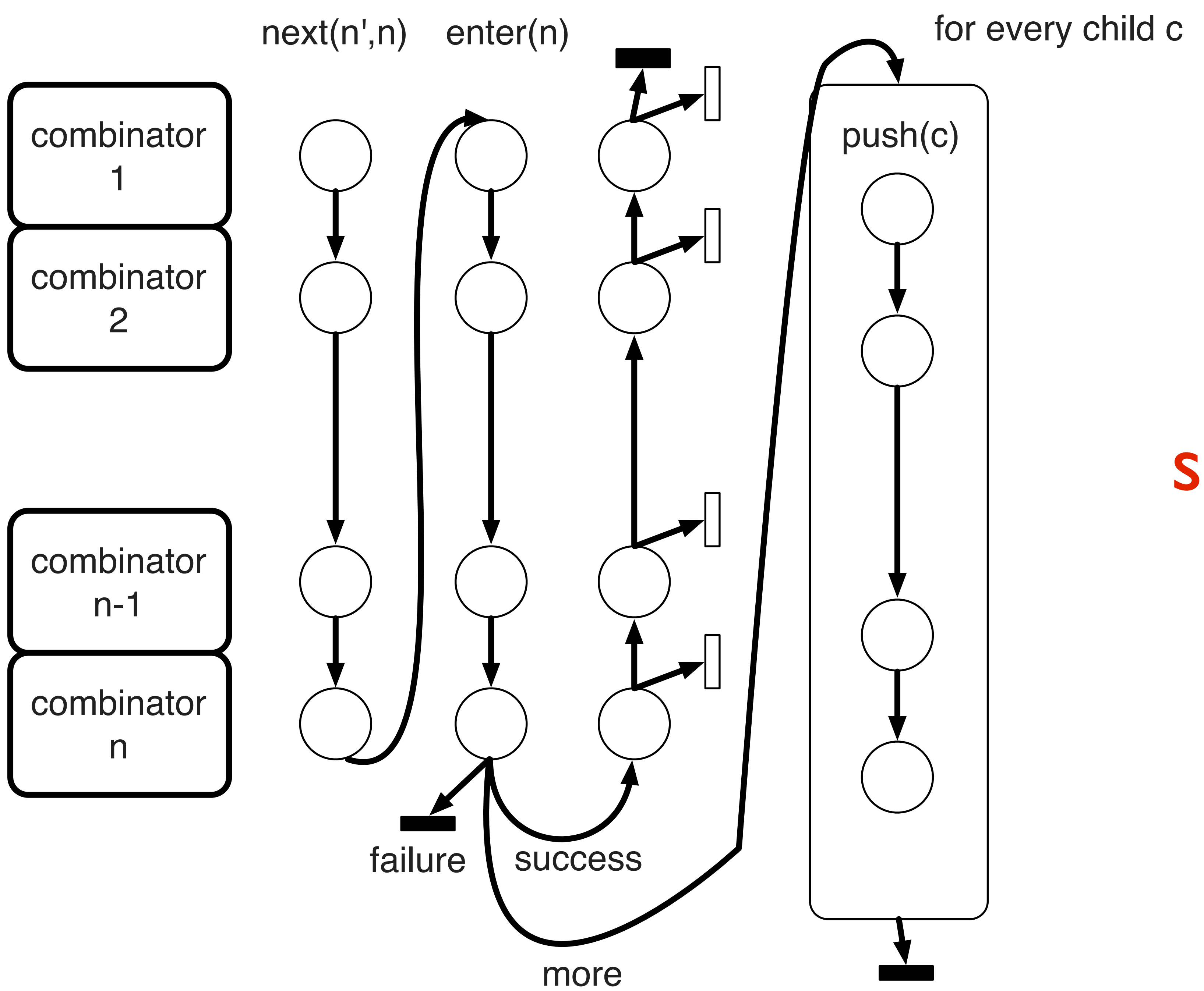
## Semantics

Syntax

vs.

Modular
Semantics

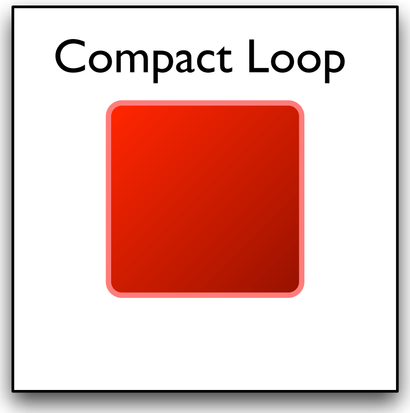
# Modular Mixin Design



Details:
see paper

# Implementations

| DSL | Haskell | C++ |
|---|---|---|



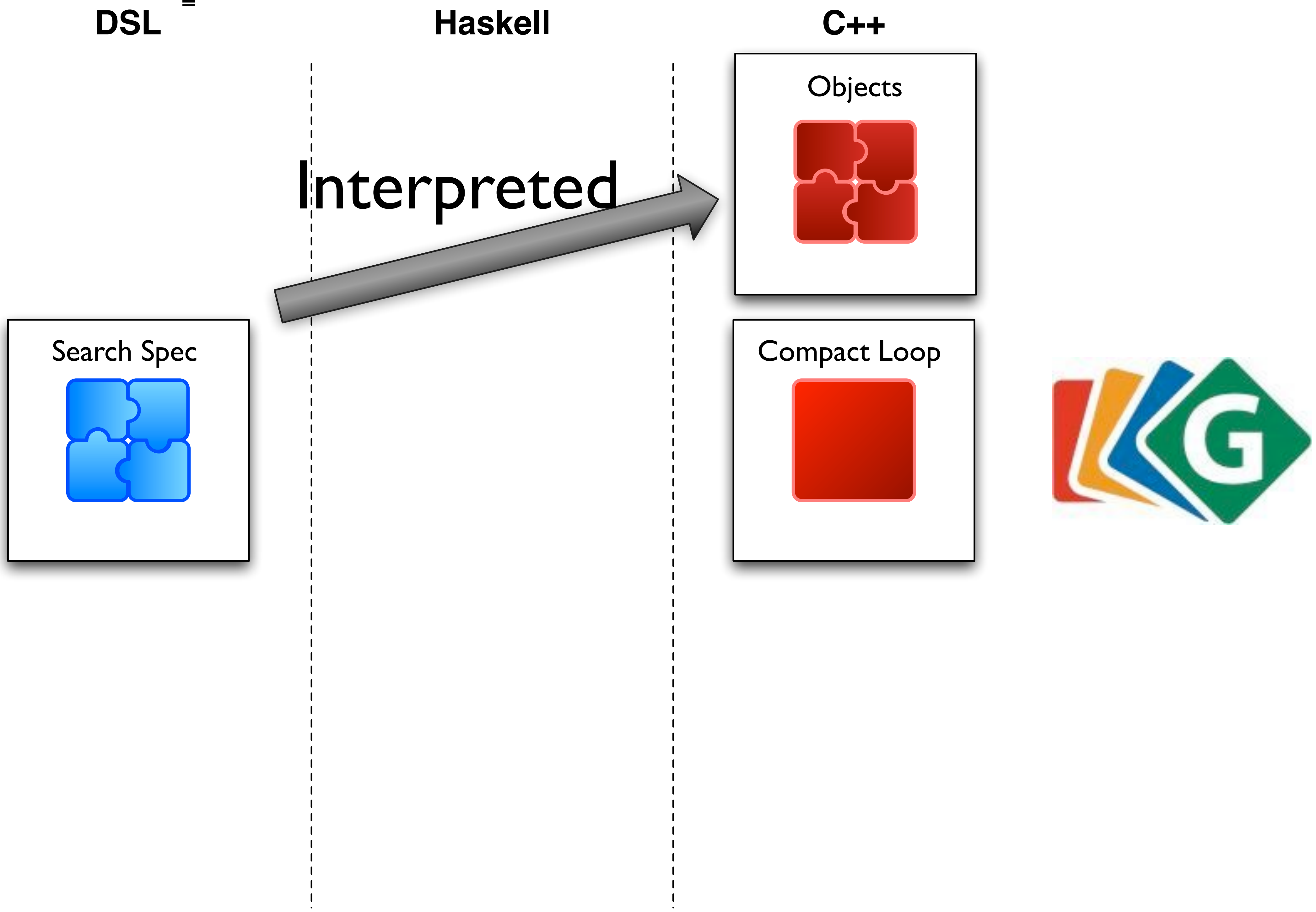
# Implementations

**DSL** | **Haskell** | **C++**

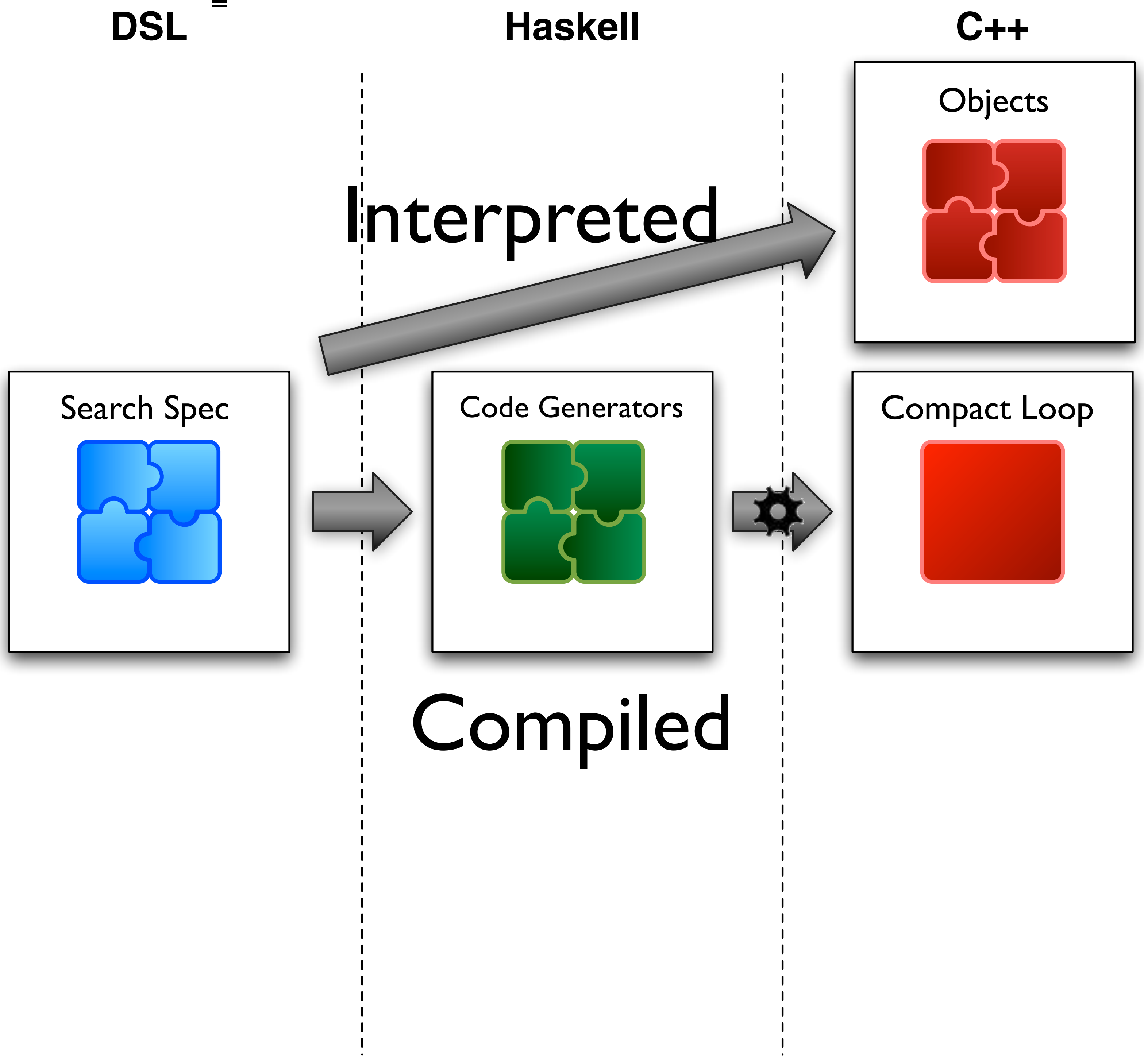
Search Spec

Compact Loop

# Implementations

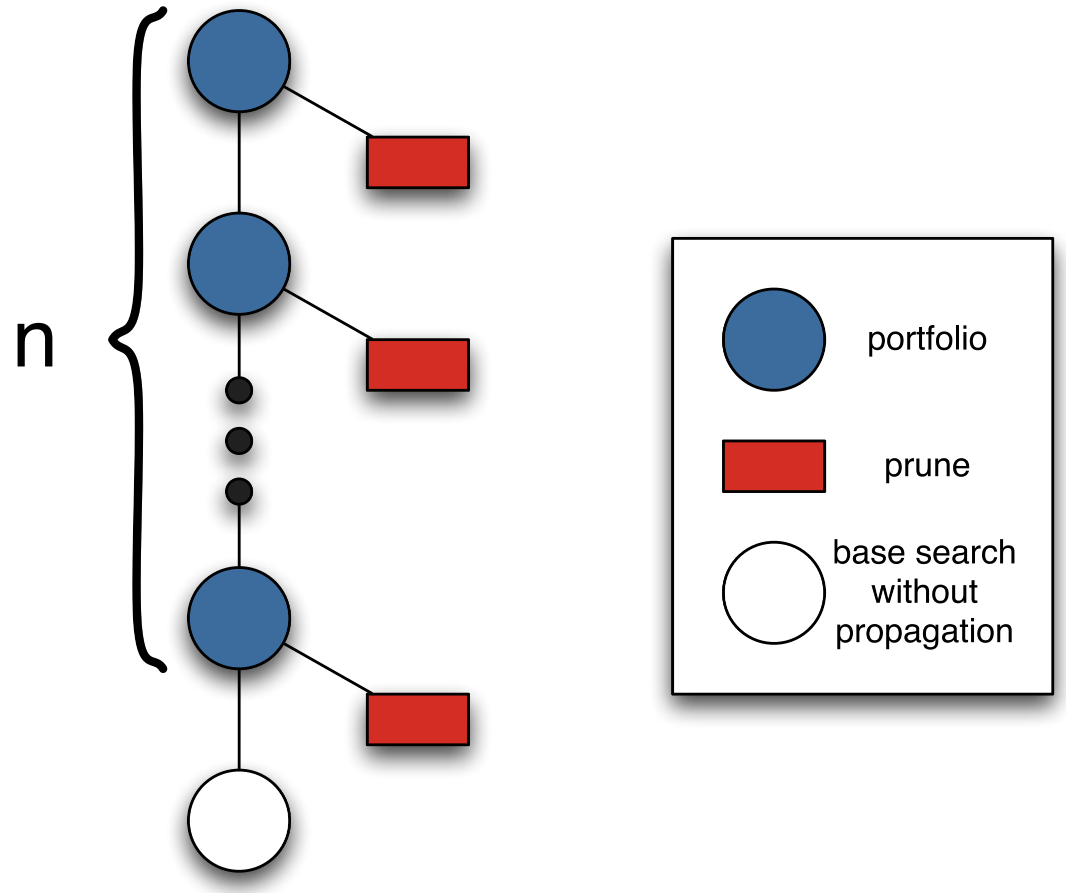


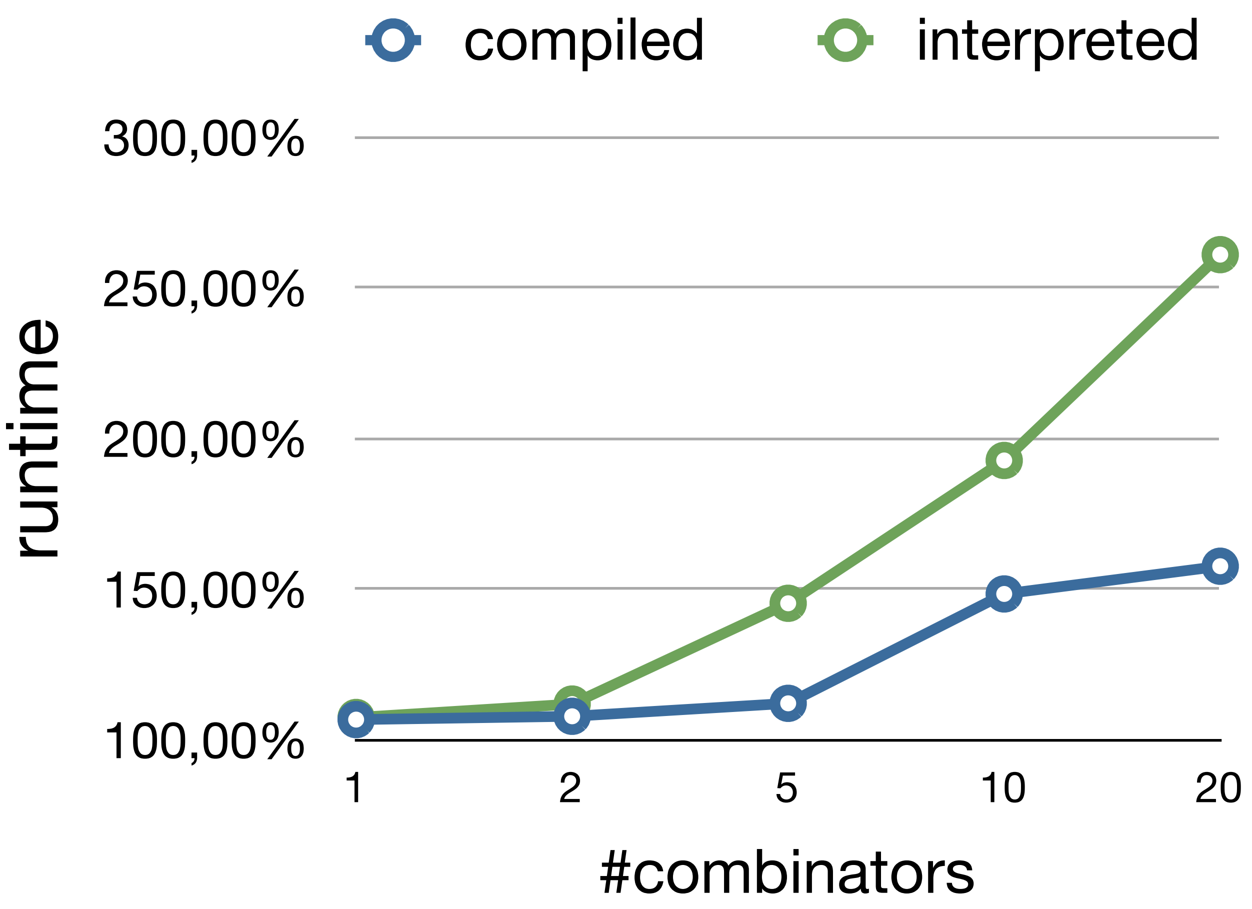
# Implementations



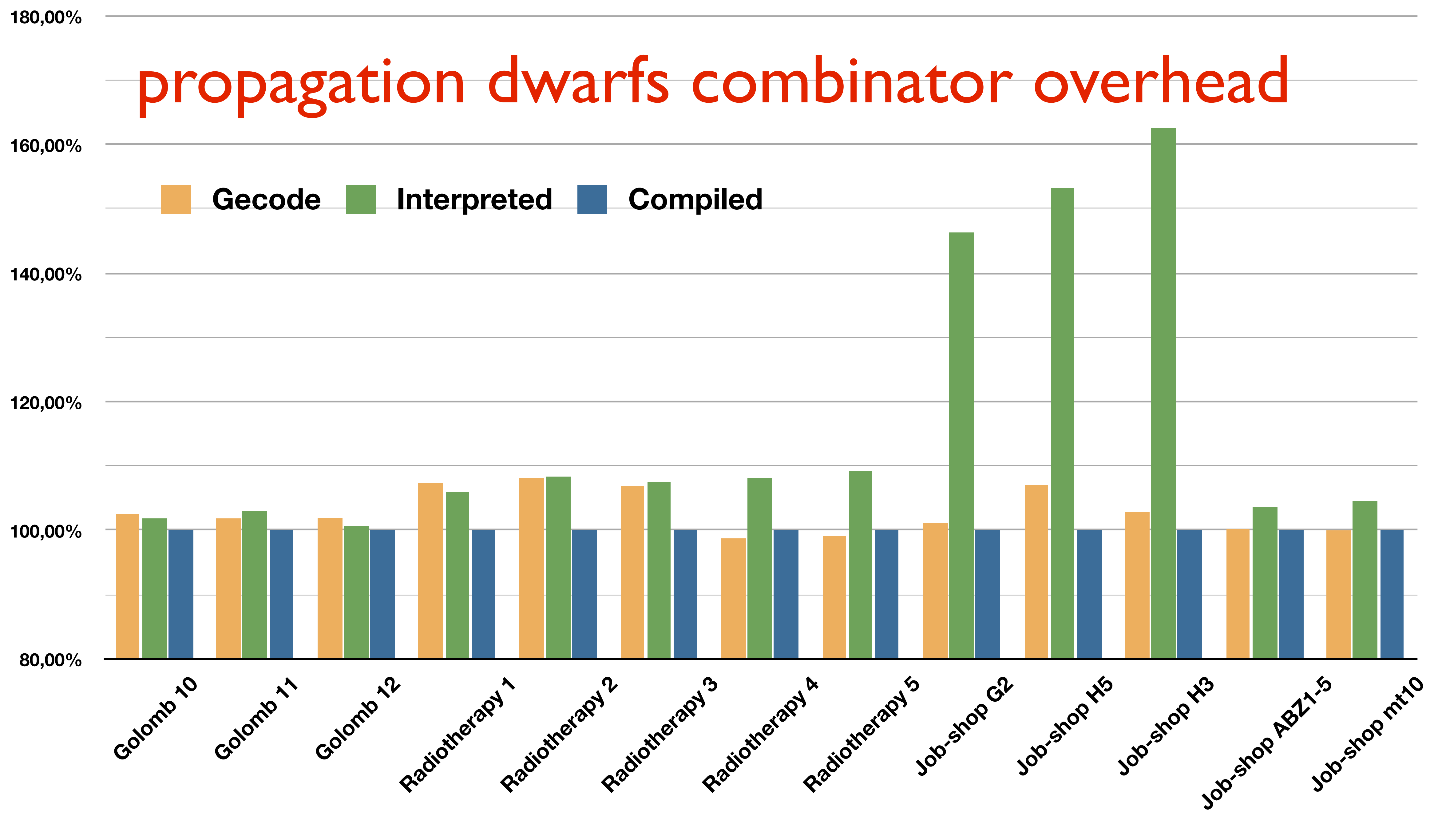
# Combinator Overhead?

## Worst-case Scenario





# In Practice



180,00%
160,00%
140,00%
120,00%
100,00%
80,00%
propagation dwarfs combinator overhead
Gecode
Interpreted
Compiled
Golomb 10
Golomb 11
Golomb 12
Radiotherapy 1
Radiotherapy 2
Radiotherapy 3
Radiotherapy 4
Radiotherapy 5
Job-shop G2
Job-shop H5
Job-shop H3
Job-shop ABZ1-5
Job-shop mt10

## Summary

- high-level modeling of search
- low-level modular implementation
- competitive performance compared to hand-coded algorithm

# Future Work

- Combinators for parallel search
- Other solving technology (e.g., LP)

➡ Combinators for hybrid search

# Thank You!

Full Paper Available

http://users.ugent.be/~tschrijv/