Evolving Reasons for Tests or

Can we gain something from Directly Searching SE Decision Spaces?

Robert Feldt

CREST Open Workshop, UCL, London, 2011-05-12

HOSE Lab (Human-fOcused SE)

Chalmers University of Technology
Sweden

What is Engineering?

What is Engineering?

“Finding good-enough (technical) solutions to real-world problems”

What is Engineering?

“Finding good-enough (technical) solutions to real-world problems”

“Making trade-offs & Balancing competing constraints”

What is Engineering?

What is Engineering?

Engineering is
a **Co-operative Game** to
Explore a Space of possibilities and
Making **Justified Decisions**

What is Engineering?

Engineering is
a **Co-operative Game** to
Explore a Space of possibilities and
Making **Justified Decisions**

and then **Spelling them out**

What is Engineering?

Engineering is
a **Co-operative Game** to
Explore a Space of possibilities and
Making **Justified Decisions**

and then **Spelling them out**

(and **Setting up for the Next Round** of the game...)

[Cockburn2006]

What is Engineering?

Preparing for Decision

Engineering is
a **Co-operative Game** to
Explore a Space of possibilities and
Making **Justified Decisions**

and then **Spelling them out**

(and **Setting up for the Next Round** of the game...)

[Cockburn2006]

What is Engineering?

Engineering is
a **Co-operative Game** to
Explore a Space of possibilities and
Making **Justified Decisions**
and then **Spelling them out**

(and **Setting up for the Next Round** of the game...)

Preparing for Decision

Making Decisions

[Cockburn2006]

What is Engineering?

Engineering is
a **Co-operative Game** to
Explore a Space of possibilities and
Making **Justified Decisions**
and then **Spelling them out**
(and **Setting up for the Next Round** of the game...)

Preparing for Decision

Making Decisions

Explaining/Solidifying Decisions

[Cockburn2006]

What is Engineering?

Engineering is
a **Co-operative Game** to
Explore a Space of possibilities and
Making **Justified Decisions**
and then **Spelling them out**
(and **Setting up for the Next Round** of the game...)

Preparing for Decision

Making Decisions

Explaining/Solidifying Decisions

Justifying Decisions

[Cockburn2006]

What is Engineering?

Engineering is
a **Co-operative Game** to
Explore a Space of possibilities and
Making **Justified Decisions**
and then **Spelling them out**
(and **Setting up for the Next Round** of the game...)

[Cockburn2006]

Preparing for Decision

Making Decisions

Explaining/Solidifying Decisions

Justifying Decisions

Changing Decisions

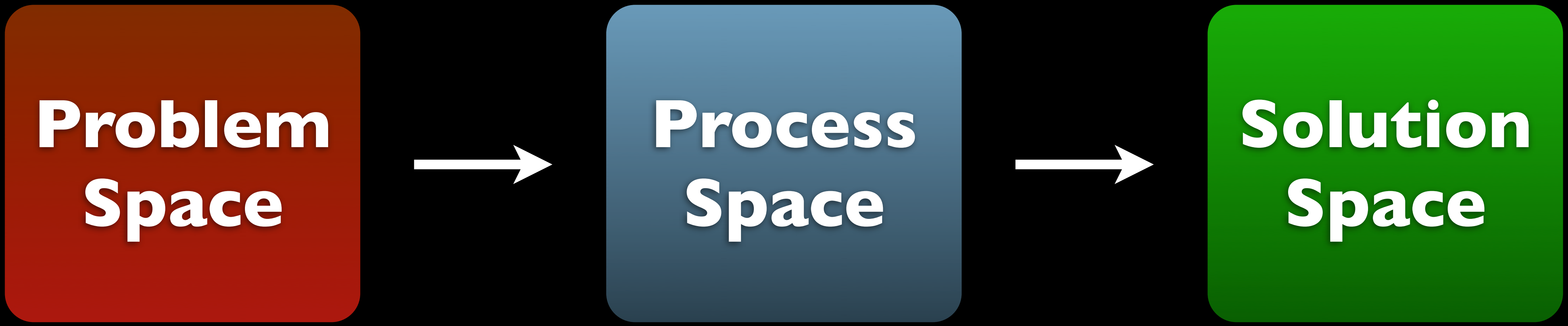
SBSE motivation

- Fits search/optimization like a glove so we argue that SBSE:
  - Easily Applicable ("We already have many spaces!" & "Requires relatively little expertise!")
  - Generic ("Same idea/search in many spaces!")
  - Robust ("Works even if info *incomplete, fuzzy, ...*")
  - Realistic ("Caters for multiple objectives")
  - Insight-rich ("We can learn about spaces we search manually")
  - Scalable ("CPU's gets cheaper and faster!")
  - Less Biased ("Fewer assumptions, that might be wrong!")
    - Than Humans & Than other Engineering Disciplines

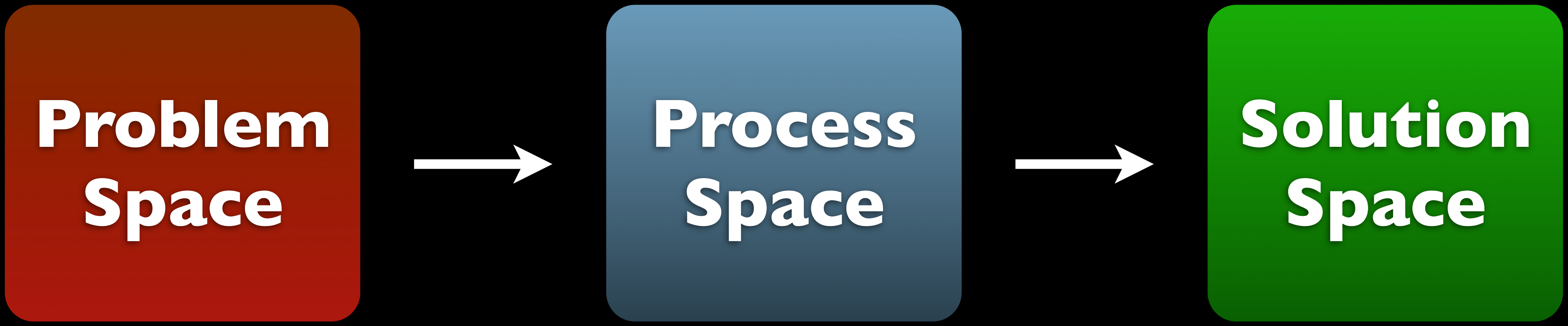
Solution Space

Problem Space

Solution Space

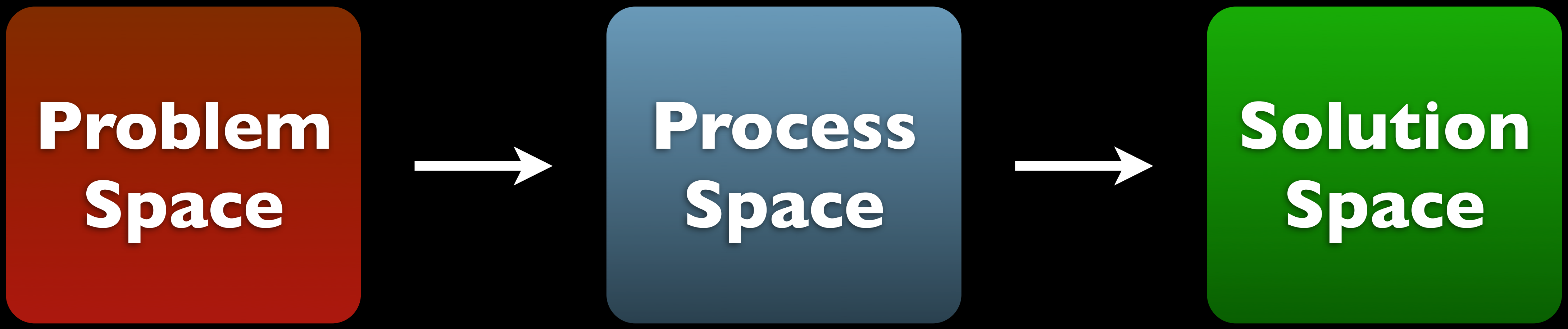


Problem Space → Process Space → Solution Space



Problem Space → Process Space → Solution Space

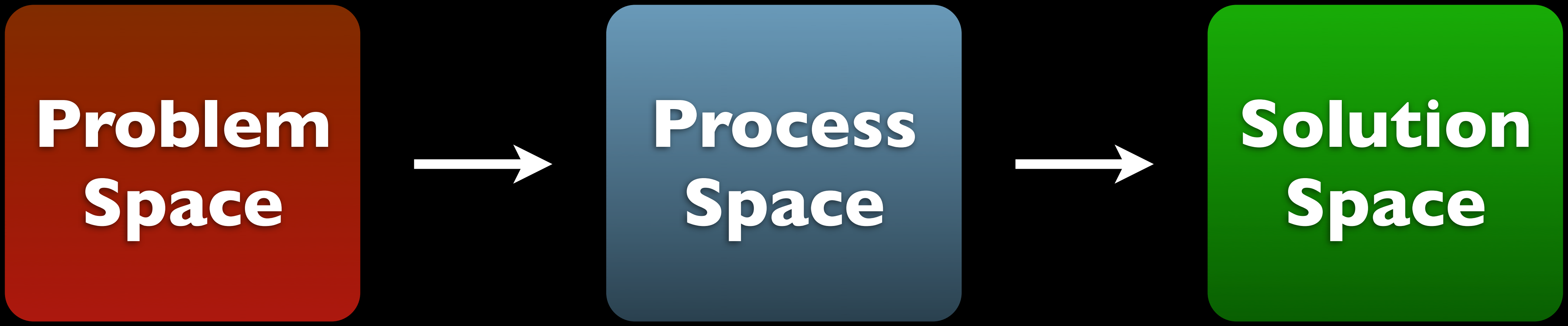
It Executes!



Problem Space → Process Space → Solution Space

It Executes!

It AutoCreates what Executes!



Problem Space

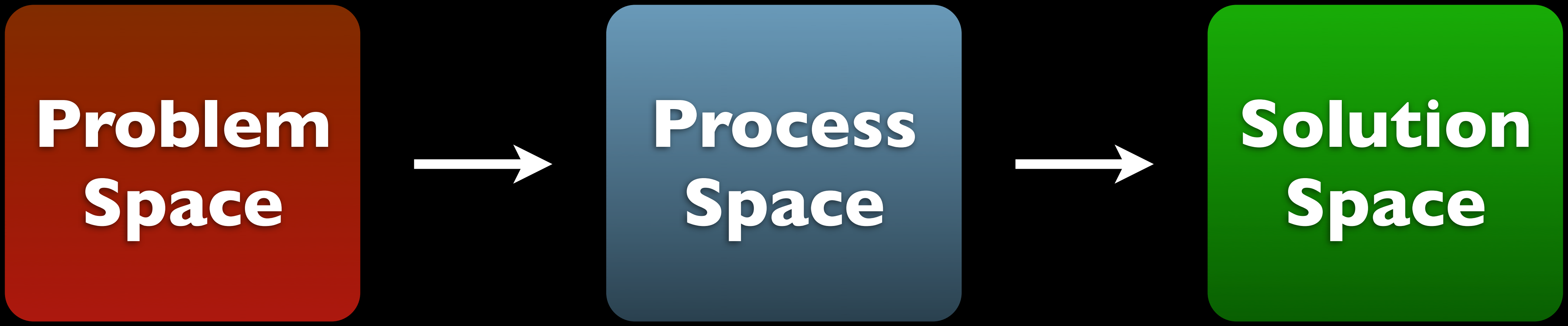
Process Space

Solution Space

It Executes!

It AutoCreates what Executes!

It Searches while Executing!



Problem Space

Process Space

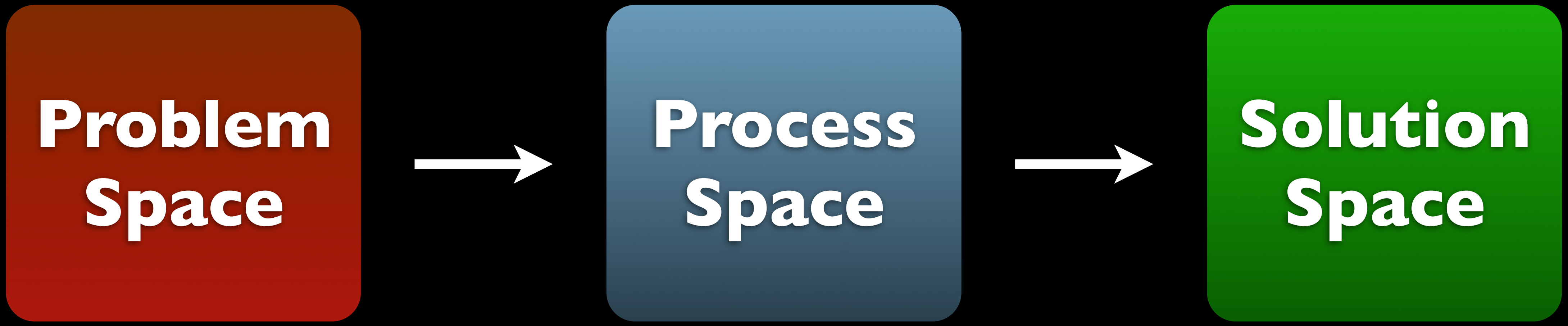
Solution Space

It Helps ManuCreate what Executes!

It Executes!

It AutoCreates what Executes!

It Searches while Executing!



Problem Space → Process Space → Solution Space

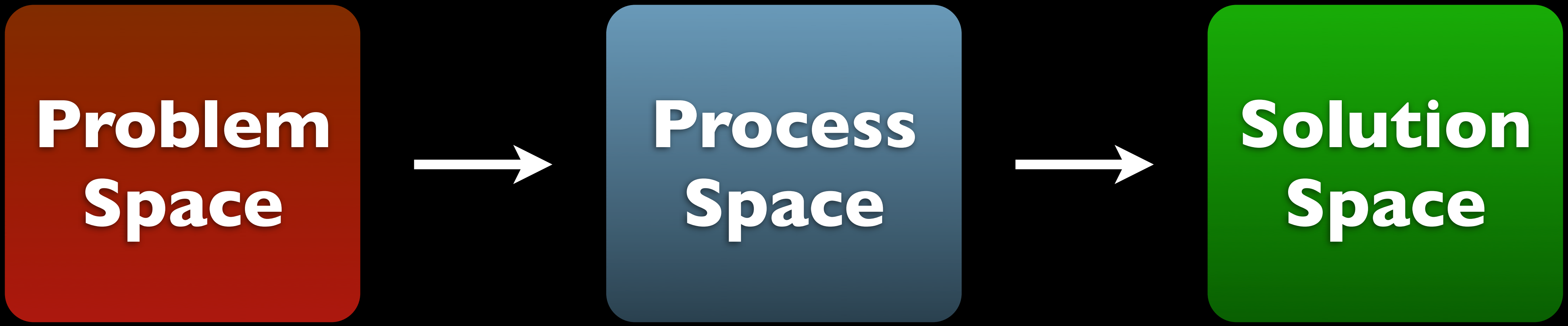
It Helps ManuCreate what Executes!

It Helps with How to ManuCreate!

It Executes!

It AutoCreates what Executes!

It Searches while Executing!



Problem Space → Process Space → Solution Space

It Promotes Understanding!

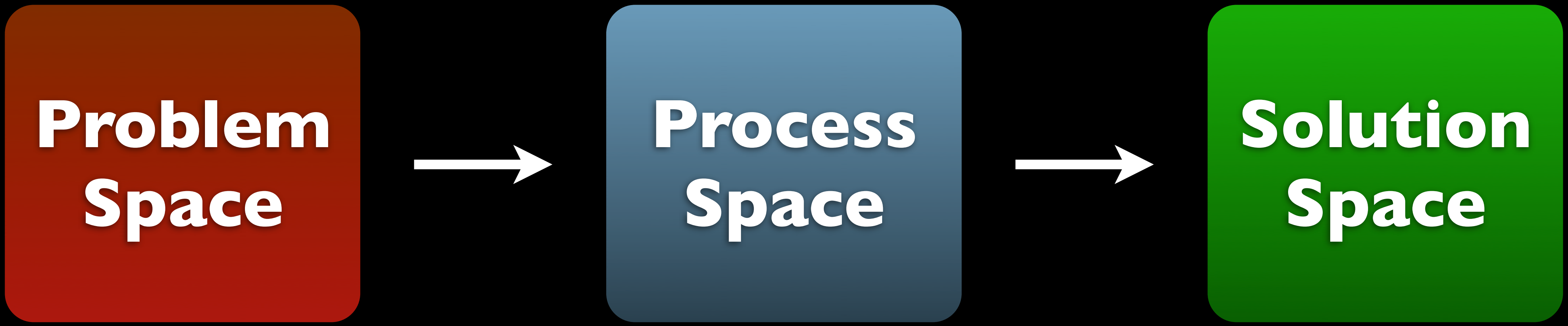
It Helps ManuCreate what Executes!

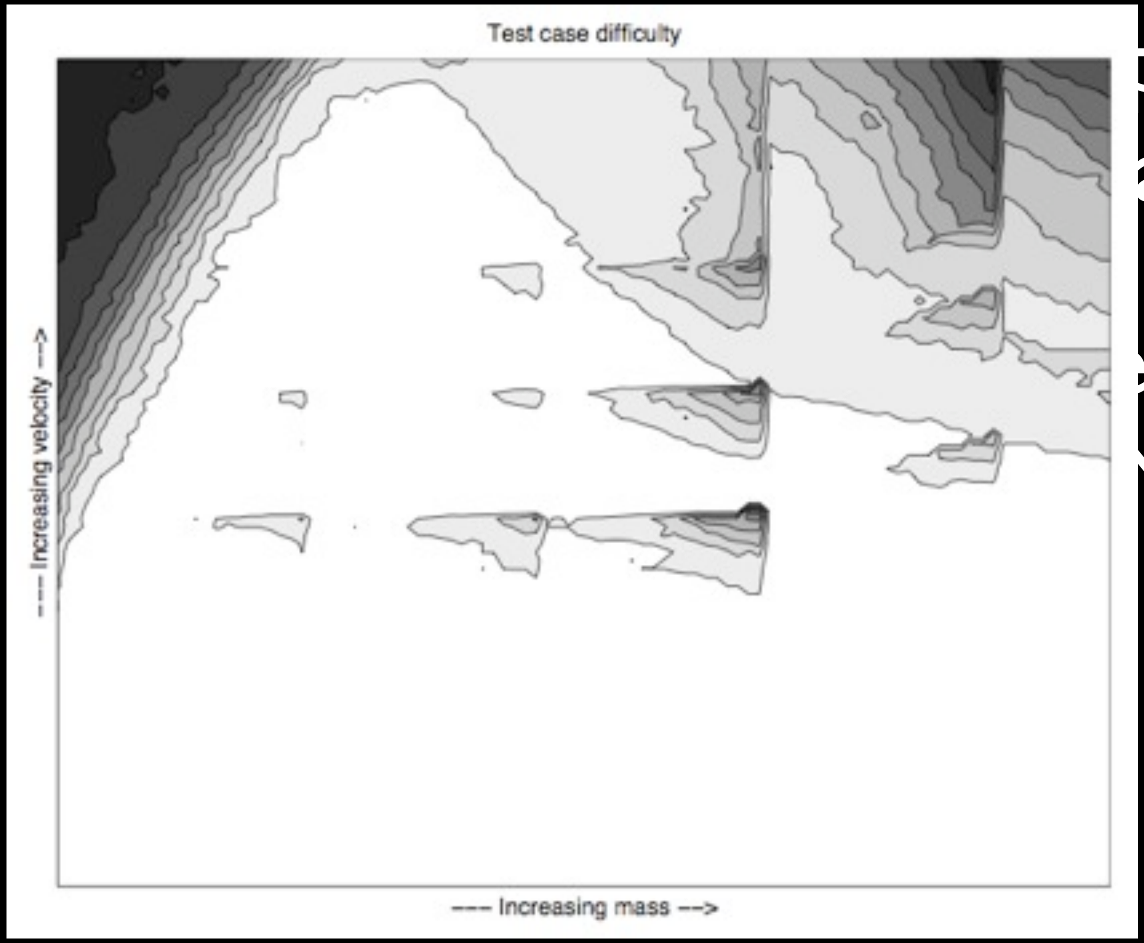
It Helps with How to ManuCreate!

It Executes!

It AutoCreates what Executes!

It Searches while Executing!





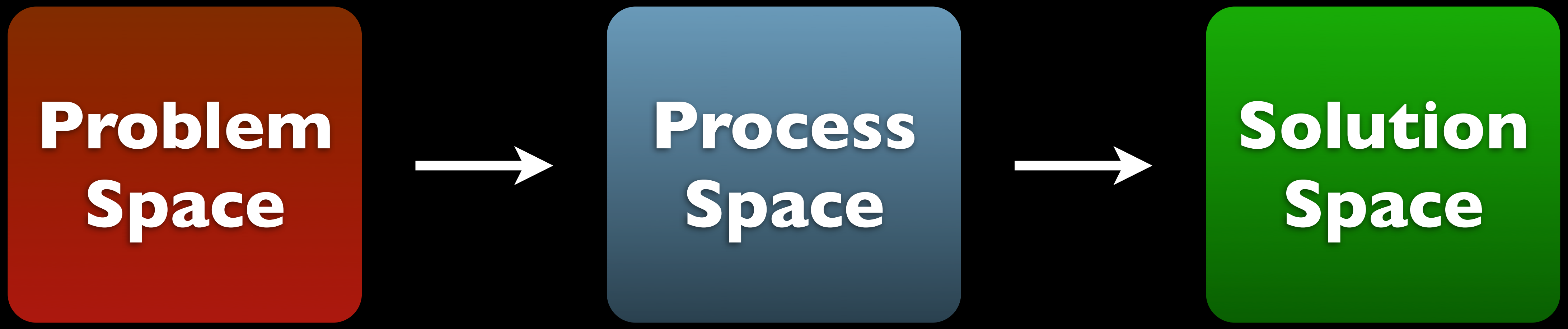
ps ManuCreate
at Executes!

elps with How
ManuCreate!

It Executes!

It AutoCreates
what Executes!

It Searches while
Executing!



Problem Space → Process Space → Solution Space

It Promotes Understanding!

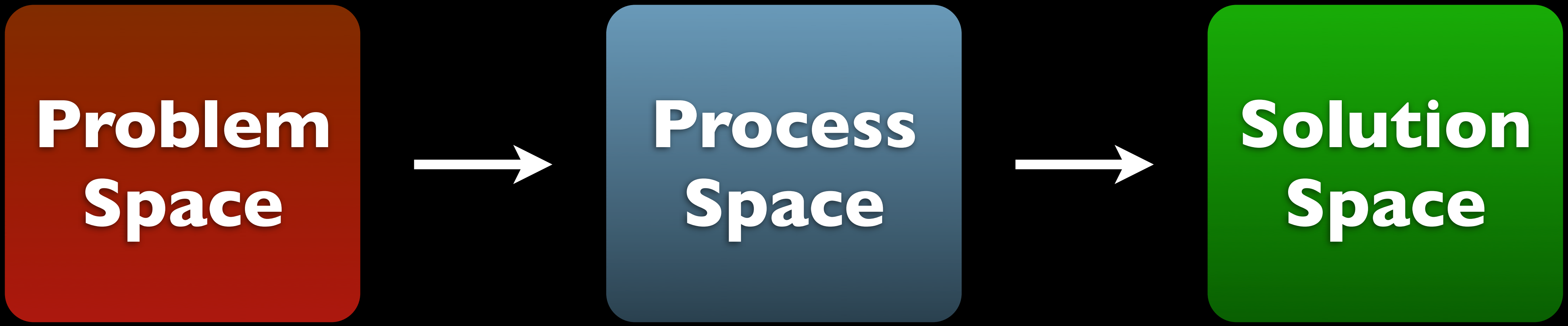
It Helps ManuCreate what Executes!

It Helps with How to ManuCreate!

It Executes!

It AutoCreates what Executes!

It Searches while Executing!



It Promotes Understanding!

It Helps ManuCreate what Executes!

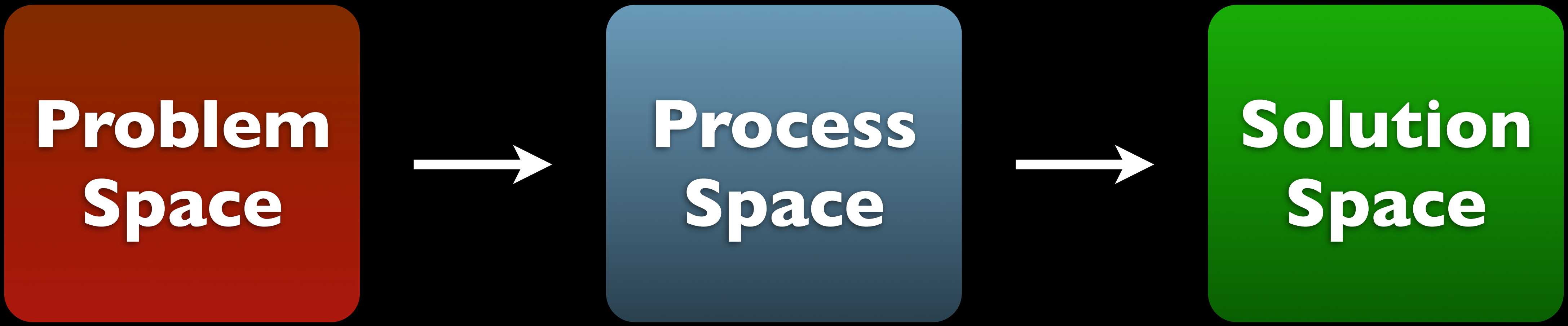
It Helps with How to ManuCreate!

It Executes!

It AutoCreates what Executes!

It Searches while Executing!

Avoid or Avoid Spelling out Decision



It Promotes Understanding!

It Helps ManuCreate what Executes!

Avoid or Avoid Spelling out Decision

It Helps with How to ManuCreate!

Select/Prio Decisions

It Executes!

It AutoCreates what Executes!

It Searches while Executing!

Problem Space → Process Space → Solution Space

It Promotes Understanding!

“Preparing” for Decision

It Helps ManuCreate what Executes!

It Helps with How to ManuCreate!

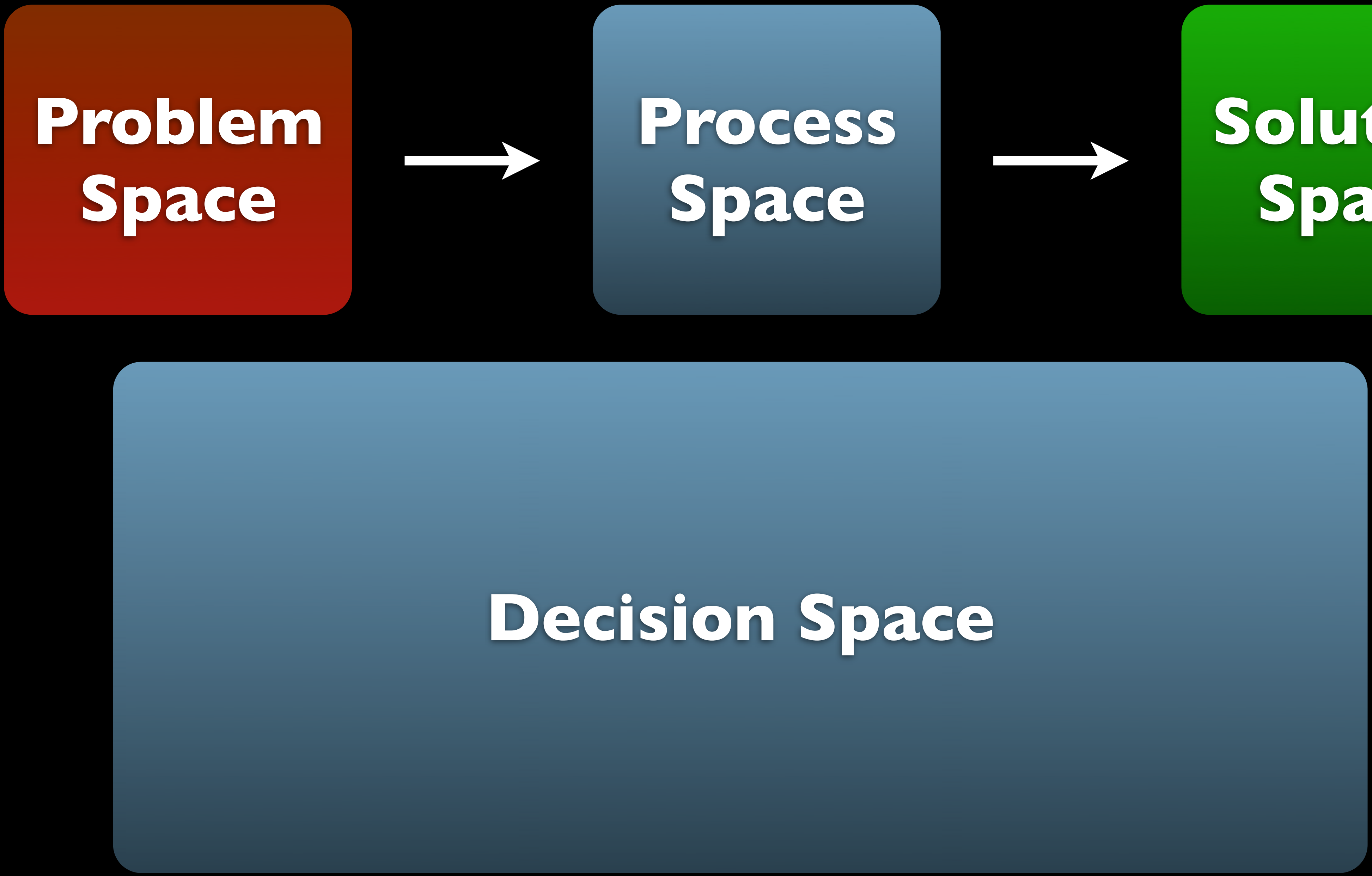
Select/Prio Decisions

It Executes!

It AutoCreates what Executes!

It Searches while Executing!

Avoid or Avoid Spelling out Decision



Problem Space → Process Space → Solution Space

Decision Space

Engineering Decisions

- Reasoning and justification involved is often:
  - Ill-defined
  - Ill-structured
  - Incomplete
  - Use Inconsistent and contradicting Information
  - Support not only logic, facts and probability but hunches, gut feelings, strange ideas

Engineering Decisions

- Reasoning and justification involved is often:
  - Ill-defined
  - Ill-structured
  - Incomplete
  - Use Inconsistent and contradicting Information
  - Support not only logic, facts and probability but hunches, gut feelings, strange ideas

Suited to Search and Optimization!

Engineering Decisions

- Reasoning and justification involved is often:
  - Ill-defined
  - Ill-structured
  - Incomplete
  - Use Inconsistent and contradicting Information
  - Support not only logic, facts and probability but hunches, gut feelings, strange ideas

Suited to Search and Optimization!

=> Possibilistic Reasoning

What would it mean in reality?

What would it mean in reality?

Real-world feedback from SW Engineers:

"Great that we can generate tests automatically but there are too many and we find too many bugs we do not want to bother with"

What would it mean in reality?

Real-world feedback from SW Engineers:
“Great that we can generate tests automatically but there are too many and we find too many bugs we do not want to bother with”

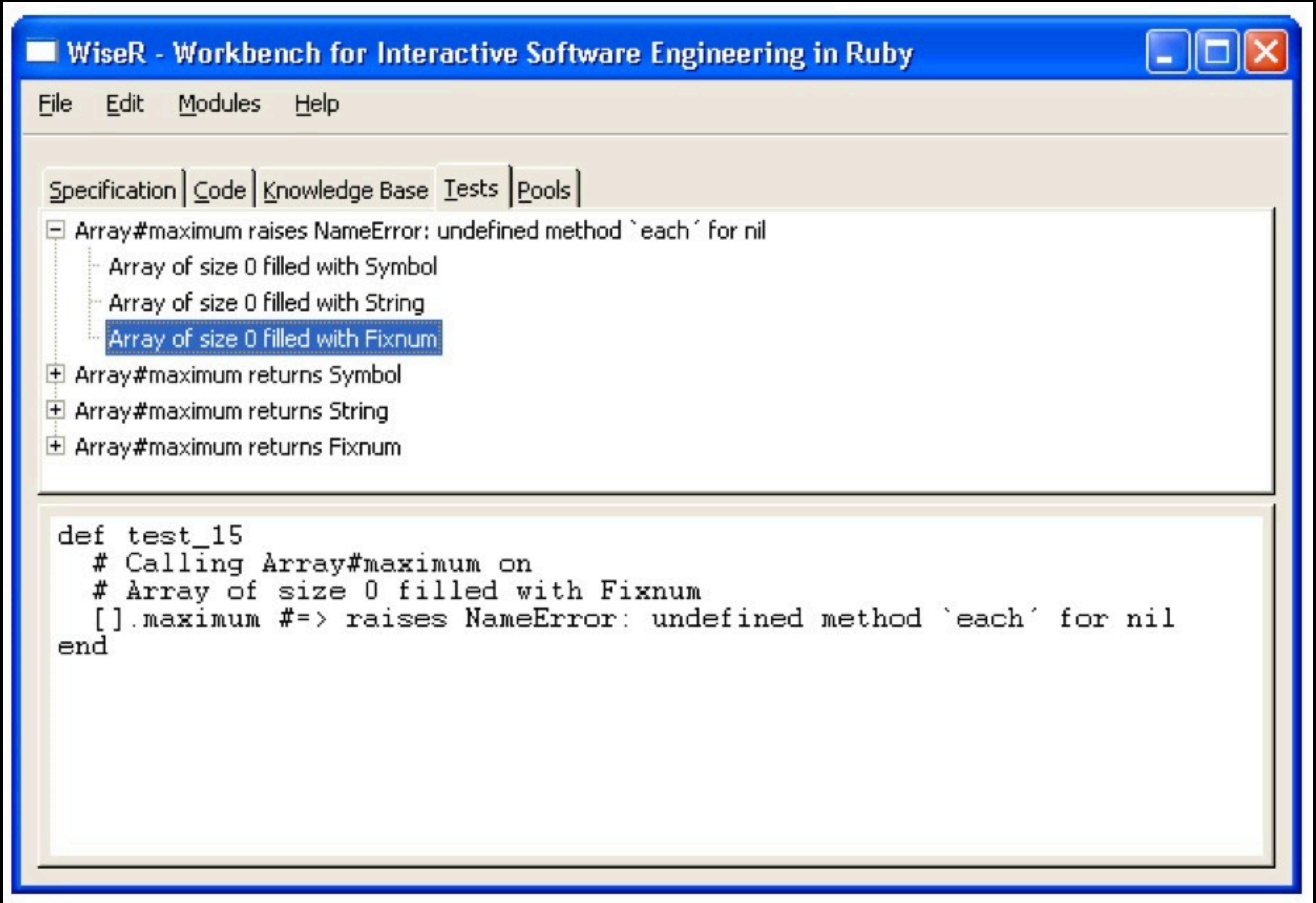
Can we directly justify the tests we generate?

What would it mean in reality?

Real-world feedback from SW Engineers:
“Great that we can generate tests automatically but there are too many and we find too many bugs we do not want to bother with”

Can we directly justify the tests we generate?

Which decisions/justification space can we search?


WiseR - Workbench for Interactive Software Engineering in Ruby
File
Edit
Modules
Help
Specification
Code
Knowledge Base
Tests
Pools

- Array#maximum raises NameError: undefined method `each´ for nil
  - Array of size 0 filled with Symbol
  - Array of size 0 filled with String
  - Array of size 0 filled with Fixnum
- Array#maximum returns Symbol
- Array#maximum returns String
- Array#maximum returns Fixnum

```
def test_15
  # Calling Array#maximum on
  # Array of size 0 filled with Fixnum
  [].maximum #=> raises NameError: undefined method `each´ for nil
end
```

Reasons for triangle(3,3,3)?

Reasons for triangle(3,3,3)?

Specification:

“`triangle` is a method and takes 3 integer arguments that are the length of its sides. Output is ‘equilateral’ if all sides are the same.

...”

Reasons for triangle(3,3,3)?

Specification:

“`triangle` is a method and takes 3 integer arguments that are the length of its sides. Output is ‘equilateral’ if all sides are the same. ...”

I want a high quality system

Reasons for triangle(3,3,3)?

Specification:

“`triangle` is a method and takes 3 integer arguments that are the length of its sides. Output is ‘equilateral’ if all sides are the same. ...”

I want a high quality system

For a high quality system, I want to test all methods

Reasons for triangle(3,3,3)?

Specification:

“`triangle` is a method and takes 3 integer arguments that are the length of its sides. Output is ‘equilateral’ if all sides are the same. …”

I want a high quality system

For a high quality system, I want to test all methods

triangle is a method

Reasons for triangle(3,3,3)?

Specification:

“`triangle` is a method and takes 3 integer arguments that are the length of its sides. Output is ‘equilateral’ if all sides are the same.

...”

I want a high quality system

For a high quality system, I want to test all methods

“VERB all Ts” => “VERB to a T”

triangle is a method

Reasons for triangle(3,3,3)?

Specification:

“`triangle` is a method and takes 3 integer arguments that are the length of its sides. Output is ‘equilateral’ if all sides are the same. ...”

I want a high quality system

“VERB all Ts” => “VERB to a T”

For a high quality system, I want to test all methods

to test is a VERB

triangle is a method

Reasons for triangle(3,3,3)?

Specification:

“`triangle` is a method and takes 3 integer arguments that are the length of its sides. Output is ‘equilateral’ if all sides are the same.

…”

I want a high quality system

For a high quality system, I want to test all methods

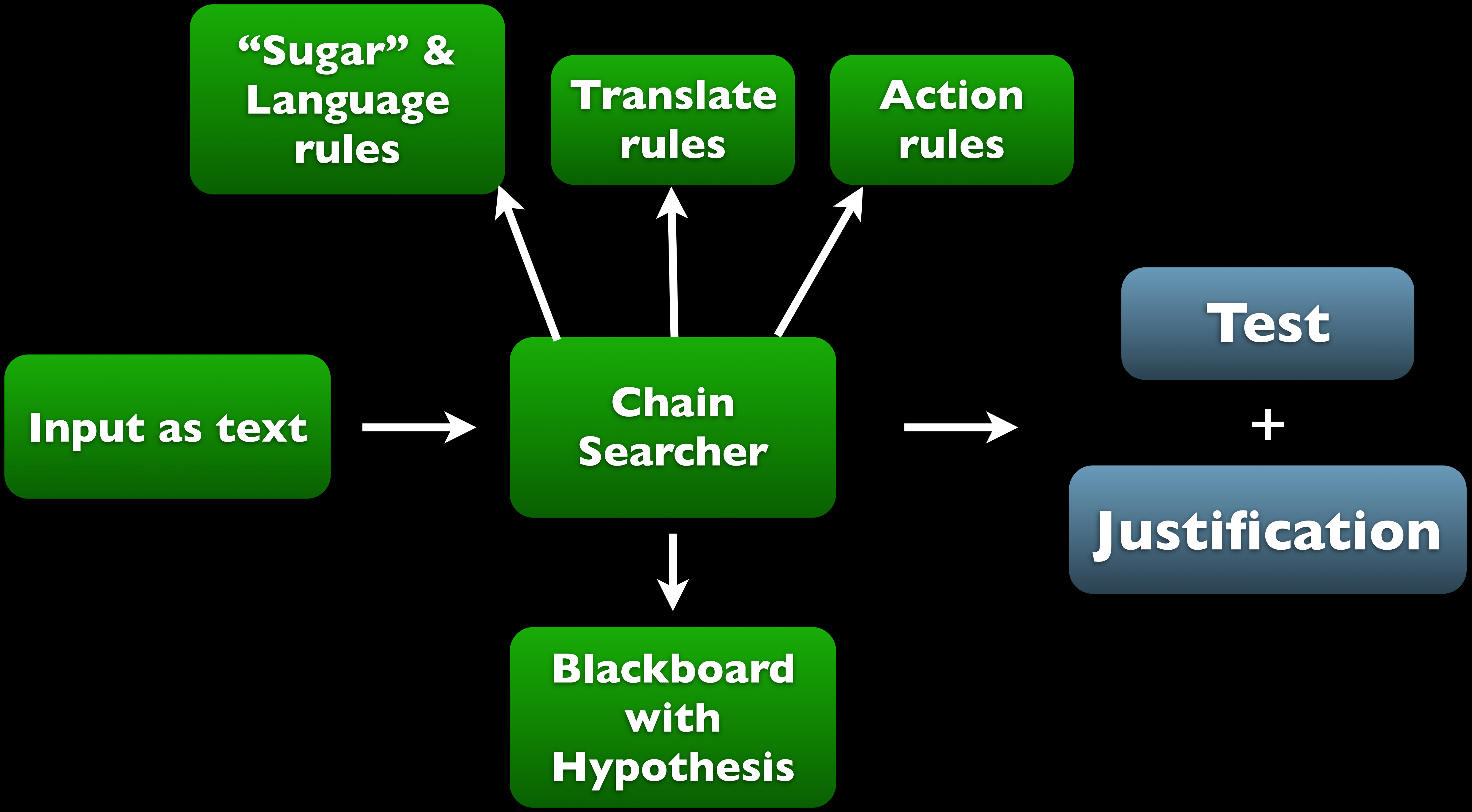
“VERB all Ts” => “VERB to a T”

to test is a VERB

triangle is a method

I want to test triangle

Possibilistic Reasoning



So, now what?

- Consider not only:
  - Which artefact am I searching for? and
  - Which activity do I support? but
  - Which engineering decision am I supporting? and
  - Can I more directly support that decision?
- Since benefits are:
  - Help engineers explore not only artefacts and info
  - Less assumptions means less missed (as long as we can make some progress) opportunities and "errors"

Extra

Normalized Compression Distance

- Conditional Kolmogorov Complexity K(X|Y)
- Cilibrasi: Use a compression algorithm, C!

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

- Non-negative number 0<=NCD<=1+e, where e depends on how good C approximates K