



Code-driven NO, Normware SI!

4 July 2024, *Annual Conference of “Law, Science and Technology”*, Bologna
Panel on AI, Regulation, Ethics

Giovanni Sileno, g.sileno@uva.nl

Socially Intelligent Artificial Systems (SIAS),
Informatics Institute, University of Amsterdam

Prologue:
on the continuity between
institutions and computations

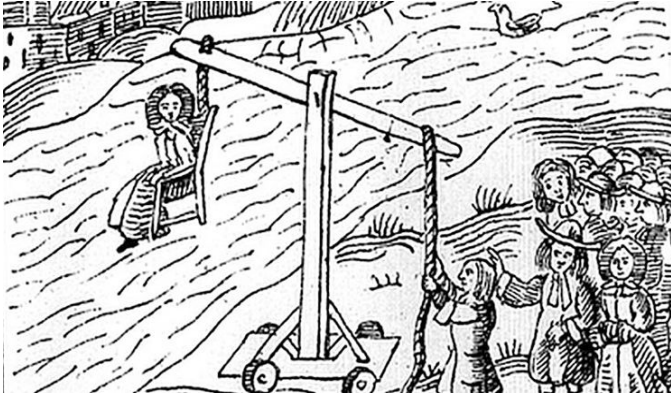
Ordalia or the “judgment of God”

In medieval times, juridical disputes that could not be or were willingly not regulated by human means were solved by calling for the “the judgment of God”.

Two main types:

- **unilateral**, e.g. proof of water, of fire, of bread
- **bilateral**, or judicial duel

Ordalia or the “judgment of God”



**water
ordeal**

*“If you are innocent,
God will save you”*



fire ordeal



**boiling
water
ordeal**

Ordalia or the “judgment of God”



an interesting
variation:

host or altar bread ordeal

*“If you are **guilty**,
God will chock you”*

Ordalia or the “judgment of God”



*“If you are innocent,
God will make you win”*



Ordalia or the “judgment of God”



*“If you are innocent,
God will make you win”*

Detailed process
(arbiters, choice of
weapons, rules of
attack). Once set, the
duel it is just a matter of
execution.



Ordalia or the “judgment of God”



Detailed process
(arbiters, choice of
weapons, rules of
attack). Once set, the

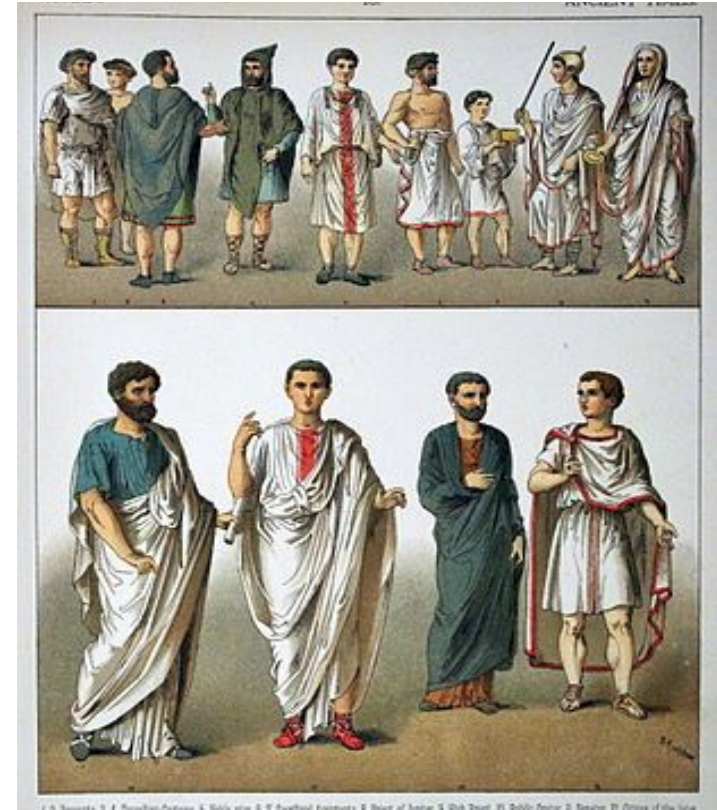
“If you are innocent,

fast to set, prompt response, clear (and definitive) result

Trials, eg. *per formulas*

In Roman times:

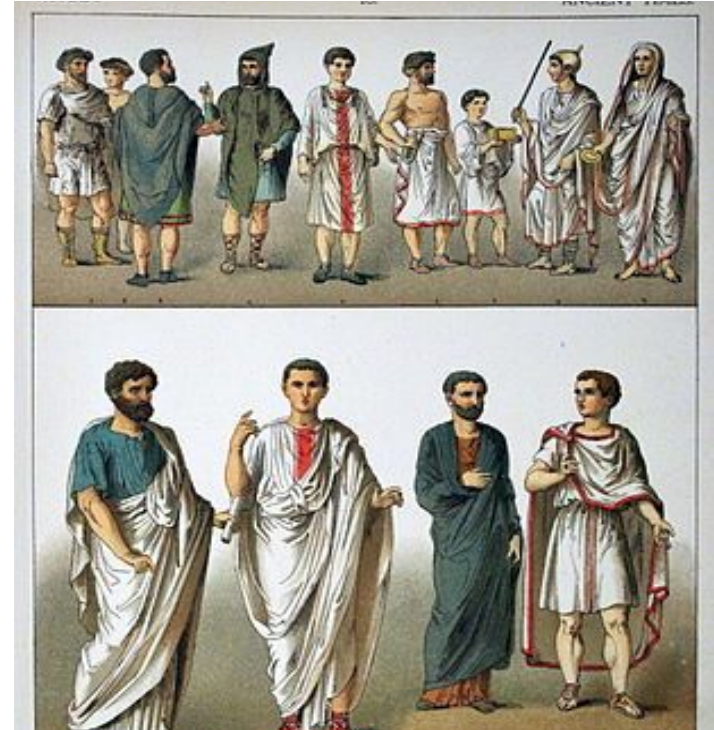
- parties were required to be in front of a *magistratus*
- the *magistratus* led the parties to write a *formula*, representing adequately the dispute and providing a “judgment template”
- a private judge gave judgment following the formula



Trials, eg. *per formulas*

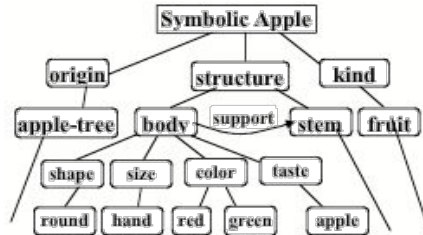
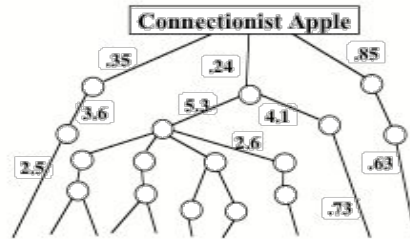
In Roman times:

- parties were required to be in front of a *magistratus*
- the *magistratus* led the parties to write a *formula*, representing adequately the dispute and providing a “judgment template”
- a private judge gave judgment following the formula



slow to set, but verbalized and written, might be not definitive

Two main approaches to AI



sub-symbolic AI

associationistic methods:

neural networks, deep learning, gen AI

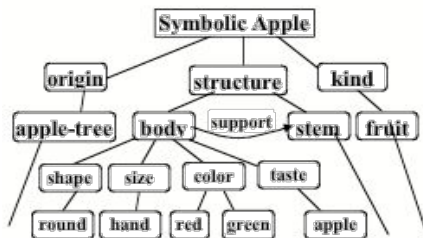
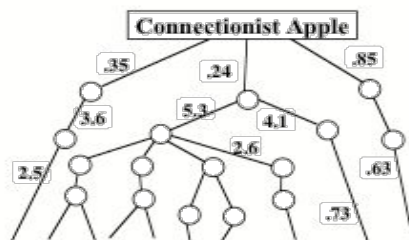
symbolic AI

logic-based methods,

expert systems, rule-based systems

formal argumentation

Two main approaches to AI



sub-symbolic AI

associationistic methods:

neural networks, deep learning, gen AI

grounded, *not intelligible*

symbolic AI

logic-based methods,

expert systems, rule-based systems

formal argumentation

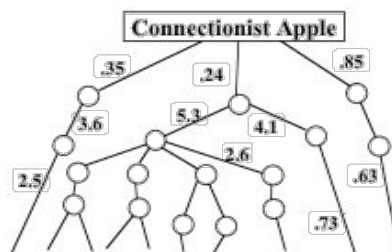
not grounded, *intelligible*



sub-symbolic AI



APPLE

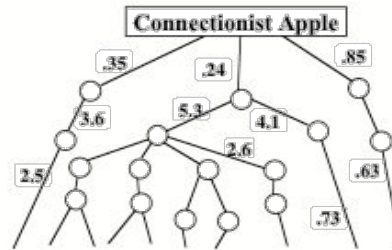




sub-symbolic AI



APPLE



If the model has been already trained

fast to set, prompt response, clear (and definitive) result

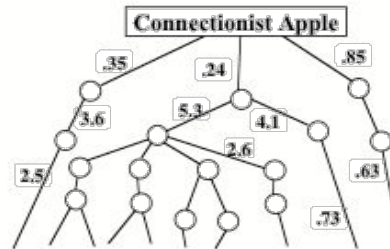


sub-symbolic AI



APPLE

**Tacit
Graded, Fast
Automatic**



If the model has been already trained

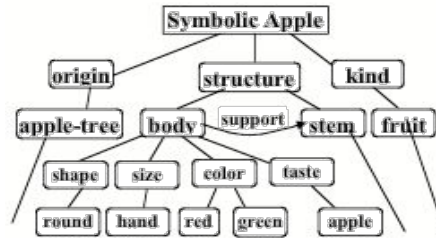
fast to set, prompt response, clear (and definitive) result

...
...
... RED SWEET
... FRUIT

symbolic AI



APPLE

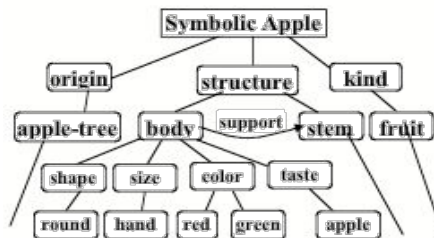


...
...
... RED SWEET
... FRUIT
...

symbolic AI



APPLE



Explicit, Crisp
Less-Fast
Automated

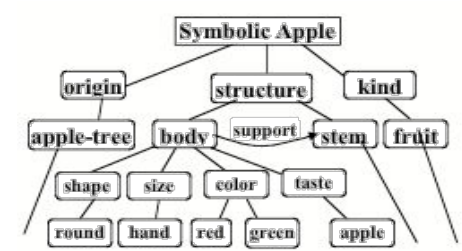
Explicit, Crisp
Less-Fast
Automated

...
...
... RED SWEET
... FRUIT
...

symbolic AI



APPLE

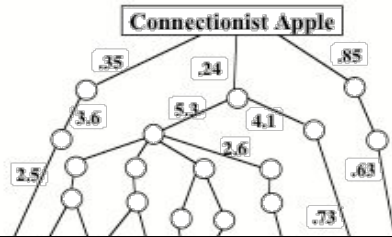


You have to write down all evidence/rules

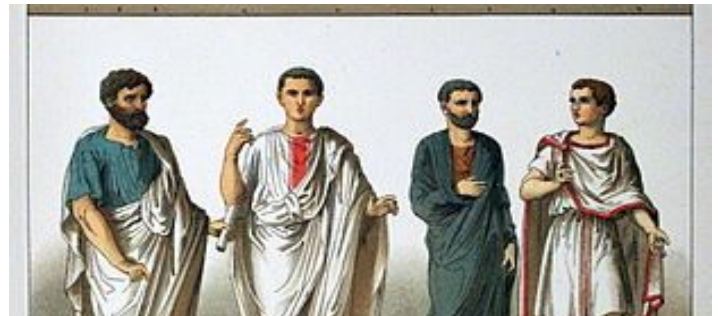
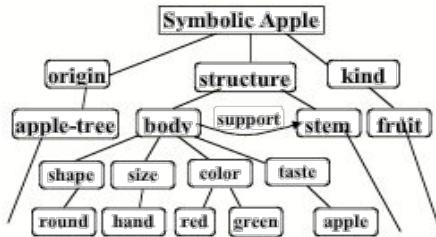
If you bring new evidence/rules

slow to set, but verbalized and written, might be not definitive

Computational vs Institutional systems?



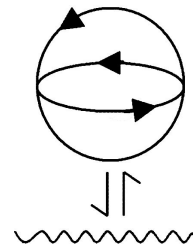
fast to set, prompt response, clear (and definitive) result



slow to set, but verbalized and written, might be not definitive

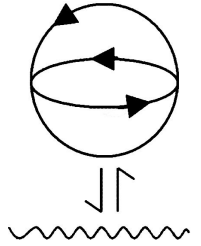
What they have in common?

*One way to understand this correspondence
is by remounting to the concept of systems...*



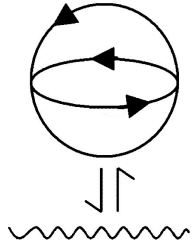
What they have in common?

In open settings, any system experiences at some point increases of **complexity**



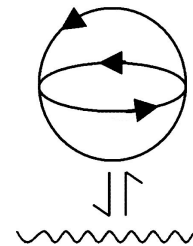
What they have in common?

In open settings, any system experiences at some point increases of **complexity**
roughly: proportional to number of system failures



What they have in common?

In open settings, any system experiences at some point increases of **complexity**
roughly: proportional to number of system failures

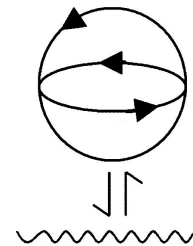


The system either:

- simplifies the environment, or, more in general,
- finds a simplified informational niche, or
- increases its internal complexity, or
- eventually collapses.

Simplifying systems

In open settings, any system experiences at some point increases of **complexity**
roughly: proportional to number of system failures

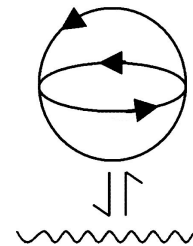


The system either:

- ~~simplifies the environment, or, more in general,~~
- finds a simplified informational niche, or
- increases its internal complexity, or
- ~~eventually collapses.~~

Simplifying systems

In open settings, any system experiences at some point increases of **complexity**
roughly: proportional to number of system failures

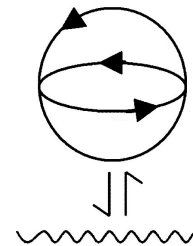


The system either:

- finds a simplified informational niche
- increases its internal complexity

Simplifying systems

In open settings, any system experiences at some point increases of **complexity**
roughly: proportional to number of system failures



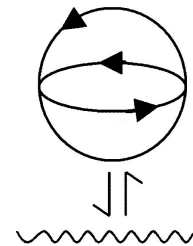
The system either:

- finds a simplified informational niche
- increases its internal complexity

to facilitate interactions (reducing failures, or enabling prompt response to them)

Simplifying systems

In open settings, any system experiences at some point increases of **complexity**
roughly: proportional to number of system failures

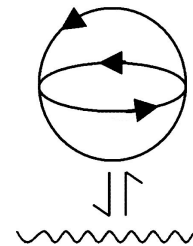


The system either:

- finds a simplified informational niche ← **Ordeals**
- ← **Institutional procedures**
- increases its internal complexity ← **Humans in the loop (judges, legislative bodies, ..)**

Simplifying systems

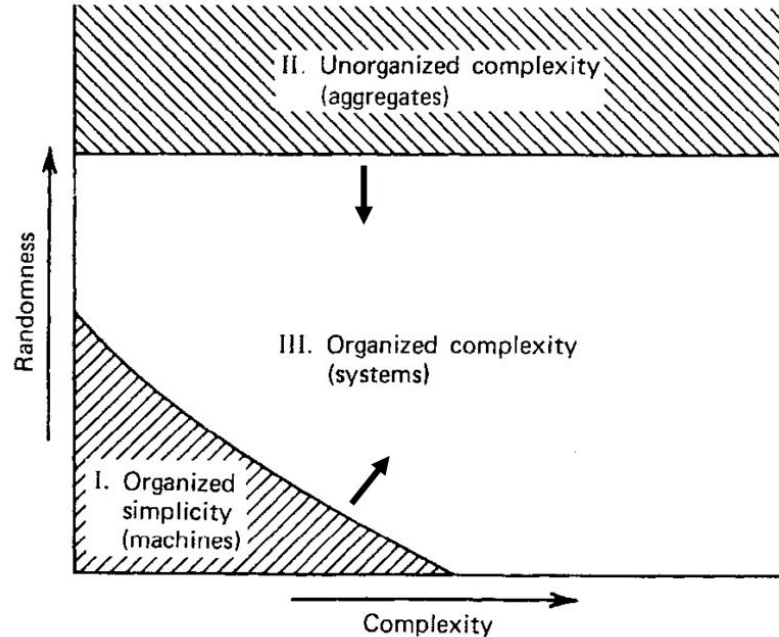
In open settings, any system experiences at some point increases of **complexity**
roughly: proportional to number of system failures



The system either:

- finds a simplified informational niche ← **Machine-learning AI**
- ← **Rule-based AI**
- increases its internal complexity ← **Humans in the loop (users, designers, ..)**

Computational and institutional systems are both information-processing systems promoting some form of “order” in the world!



Part I:
types of "law",
and a more nuanced distinction

text-driven law

data-driven law

code-driven law

Hildebrandt, M., Code Driven Law. Scaling the Past and Freezing the Future, In: Is Law Computable? Critical Perspectives on Law and Artificial Intelligence, eds. Markou, Deakin, Hart Publishers (2020)

text-driven law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

code-driven law

Hildebrandt, M., Code Driven Law. Scaling the Past and Freezing the Future, In: Is Law Computable? Critical Perspectives on Law and Artificial Intelligence, eds. Markou, Deakin, Hart Publishers (2020)

text-driven law

legal activity performed by humans by means of sources of norms such as statute and case law

data-driven law

automatic decision-making or predictions used in support derived from statistical/inductive methods

code-driven law

Hildebrandt, M., Code Driven Law. Scaling the Past and Freezing the Future, In: Is Law Computable? Critical Perspectives on Law and Artificial Intelligence, eds. Markou, Deakin, Hart Publishers (2020)

text-driven law

legal activity performed by humans by means of sources of norms such as statute and case law

data-driven law

automatic decision-making or predictions used in support derived from statistical/inductive methods

code-driven law

legal norms or policies that have been articulated in computer code

Hildebrandt, M., Code Driven Law. Scaling the Past and Freezing the Future, In: Is Law Computable? Critical Perspectives on Law and Artificial Intelligence, eds. Markou, Deakin, Hart Publishers (2020)

open-textured concepts
multi-interpretability
→ **it can be contested**

text-driven law

legal activity performed by humans by means of sources of norms such as statute and case law

data-driven law

automatic decision-making or predictions used in support derived from statistical/inductive methods

code-driven law

legal norms or policies that have been articulated in computer code

open-textured concepts
multi-interpretability
→ **it can be contested**

statistical closure

logical closure

text-driven law

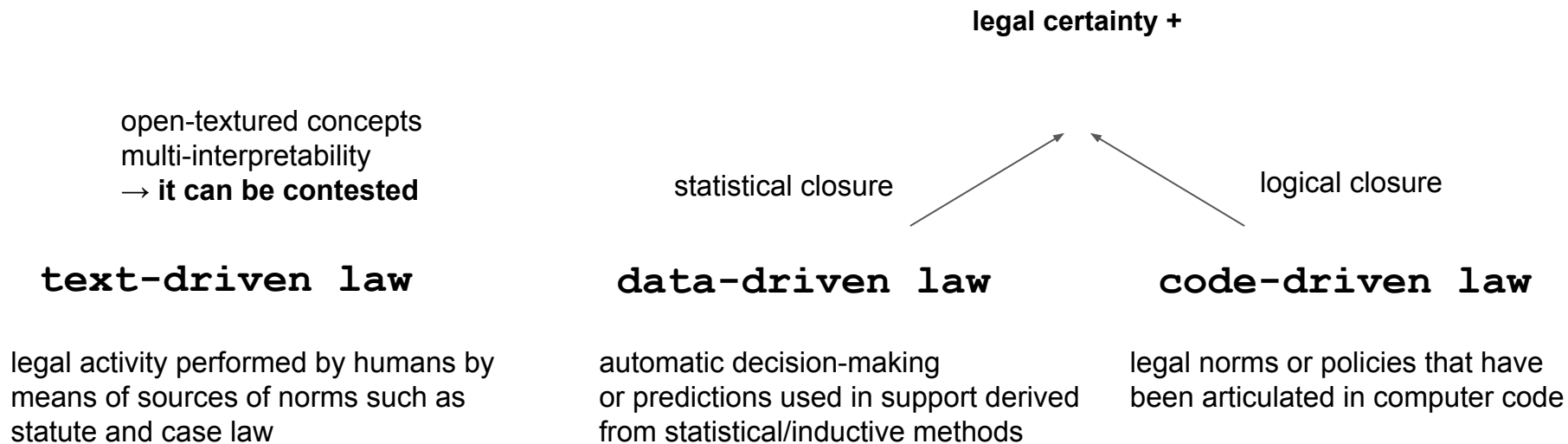
legal activity performed by humans by means of sources of norms such as statute and case law

data-driven law

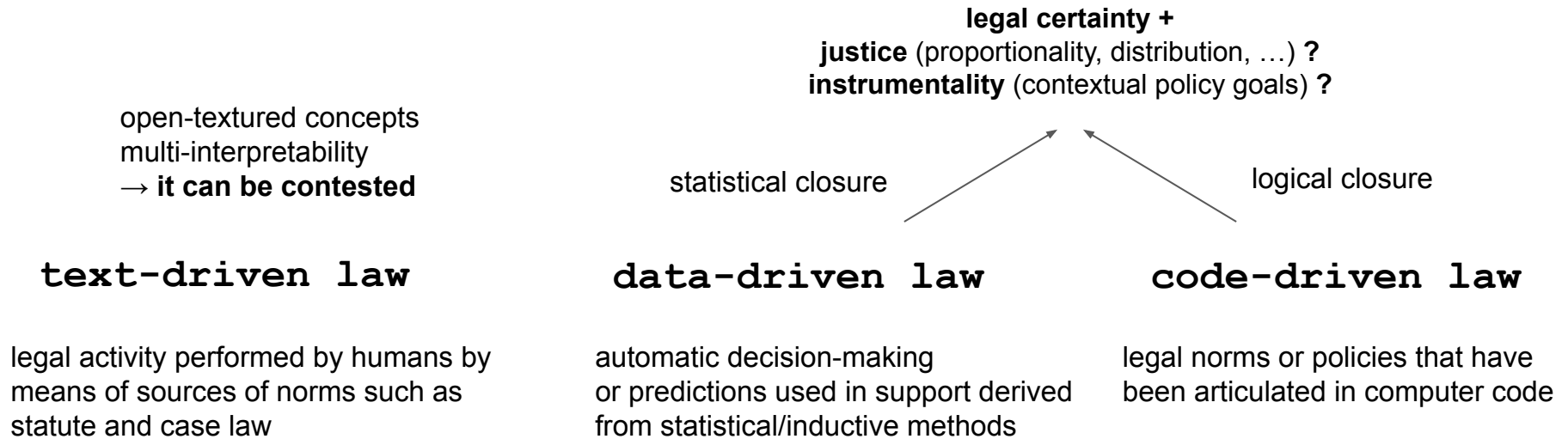
automatic decision-making or predictions used in support derived from statistical/inductive methods

code-driven law

legal norms or policies that have been articulated in computer code



Hildebrandt, M., Code Driven Law. Scaling the Past and Freezing the Future, In: Is Law Computable? Critical Perspectives on Law and Artificial Intelligence, eds. Markou, Deakin, Hart Publishers (2020)



COMPUTATIONAL LEGALISM

legal certainty +
justice (proportionality, distribution, ...) ?
instrumentality (contextual policy goals) ?

statistical closure

logical closure

text-driven law

legal activity performed by humans by means of sources of norms such as statute and case law

data-driven law

automatic decision-making or predictions used in support derived from statistical/inductive methods

code-driven law

legal norms or policies that have been articulated in computer code

legal certainty +
justice (proportionality, distribution, ...) ?
instrumentality (contextual policy goals) ?

statistical closure

logical closure

text-driven law

legal activity performed by humans by means of sources of norms such as statute and case law

data-driven law

automatic decision-making or predictions used in support derived from statistical/inductive methods

code-driven law

legal norms or policies that have been articulated in computer code

legal certainty +
justice (proportionality, distribution, ...) ?
instrumentality (contextual policy goals) ?

statistical closure

logical closure

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

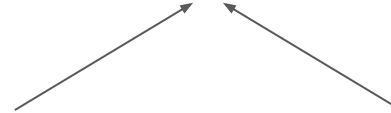
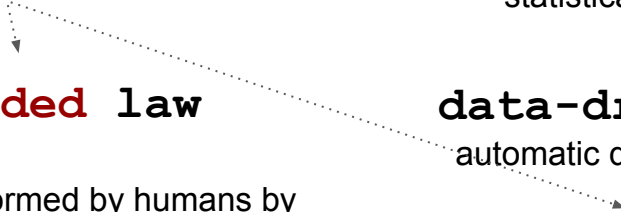
decision support systems

derived from statistical/inductive
methods

code-driven law

legal norms or policies that have
been articulated in computer code

in so far humans are involved



**TRAINING
DATASET**

instance + label
instance + label
instance + label
instance + label
.....

**legal certainty +
justice** (proportionality, distribution, ...) ?
instrumentality (contextual policy goals) ?

statistical closure

logical closure

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

code-driven law

legal norms or policies that have
been articulated in computer code

is the training data set
representative of
the domain in which
it will be applied?

is the rationality behind labeling
still the same?

TRAINING DATASET

instance + label
instance + label
instance + label
instance + label
.....

legal certainty +
justice (proportionality, distribution, ...) ?
instrumentality (contextual policy goals) ?

statistical closure

logical closure

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

code-driven law

legal norms or policies that have
been articulated in computer code

is the training data set
representative of
the domain in which
it will be applied?

is the rationality behind labeling
still the same?

TRAINING DATASET

instance + label
instance + label
instance + label
instance + label
.....

is the training method
producing adequate,
reproducible, robust results?

TRAINING METHOD

legal certainty +
justice (proportionality, distribution, ...) ?
instrumentality (contextual policy goals) ?

statistical closure

logical closure

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

code-driven law

legal norms or policies that have
been articulated in computer code

is the training data set
representative of
the domain in which
it will be applied?

is the rationality behind labeling
still the same?

TRAINING DATASET

instance + label
instance + label
instance + label
instance + label
.....

is the training method
producing adequate,
reproducible, robust results?

TRAINING METHOD

legal certainty +
justice (proportionality, distribution, ...) ?
instrumentality (contextual policy goals) ?

statistical closure
→ **it can be contested**

logical closure

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

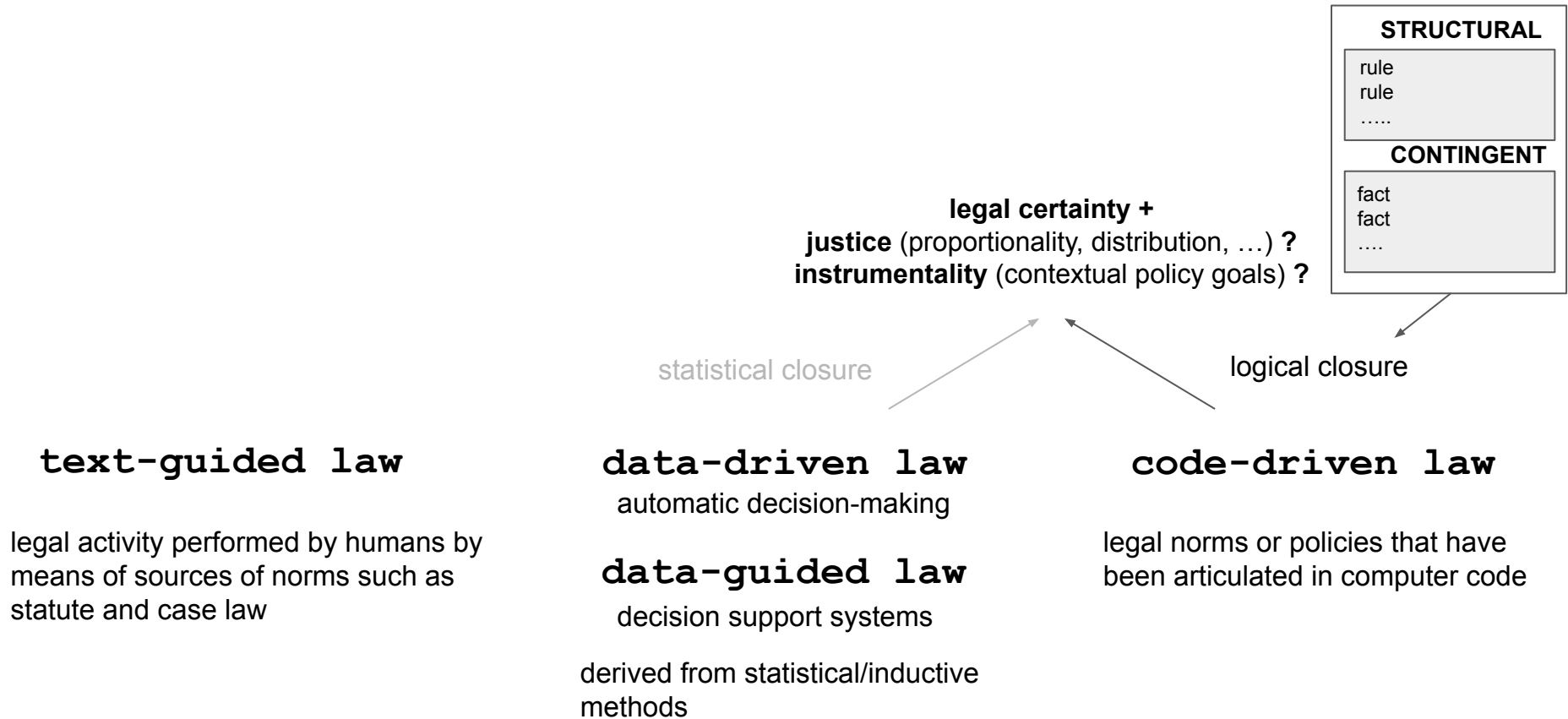
data-guided law

decision support systems

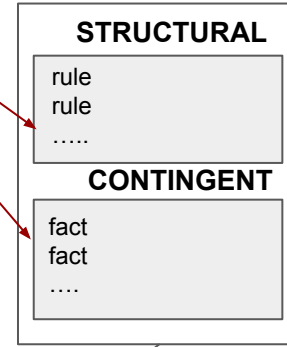
derived from statistical/inductive
methods

code-driven law

legal norms or policies that have
been articulated in computer code



were all the relevant rules considered?
were all the relevant facts given?
is the formalization correct?
is the reasoning tool robust?



legal certainty +
justice (proportionality, distribution, ...) ?
instrumentality (contextual policy goals) ?

statistical closure

logical closure

text-guided law

legal activity performed by humans by means of sources of norms such as statute and case law

data-driven law

automatic decision-making

data-guided law

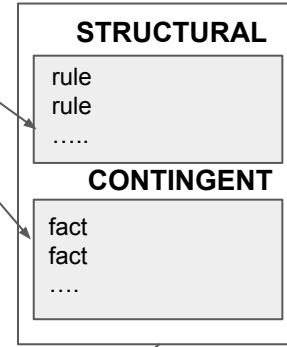
decision support systems

derived from statistical/inductive methods

code-driven law

legal norms or policies that have been articulated in computer code

were all the relevant rules considered?
were all the relevant facts given?
is the formalization correct?
is the reasoning tool robust?



legal certainty +
justice (proportionality, distribution, ...) ?
instrumentality (contextual policy goals) ?

statistical closure

logical closure
→ **it can be contested**

text-guided law

legal activity performed by humans by means of sources of norms such as statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive methods

code-driven law

legal norms or policies that have been articulated in computer code

the COMPUTATIONAL LEGALISM “ideal” works
only in so far we forget all what may go wrong



legal certainty ?
justice (proportionality, distribution, ...) ?
instrumentality (contextual policy goals) ?

statistical closure

logical closure

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

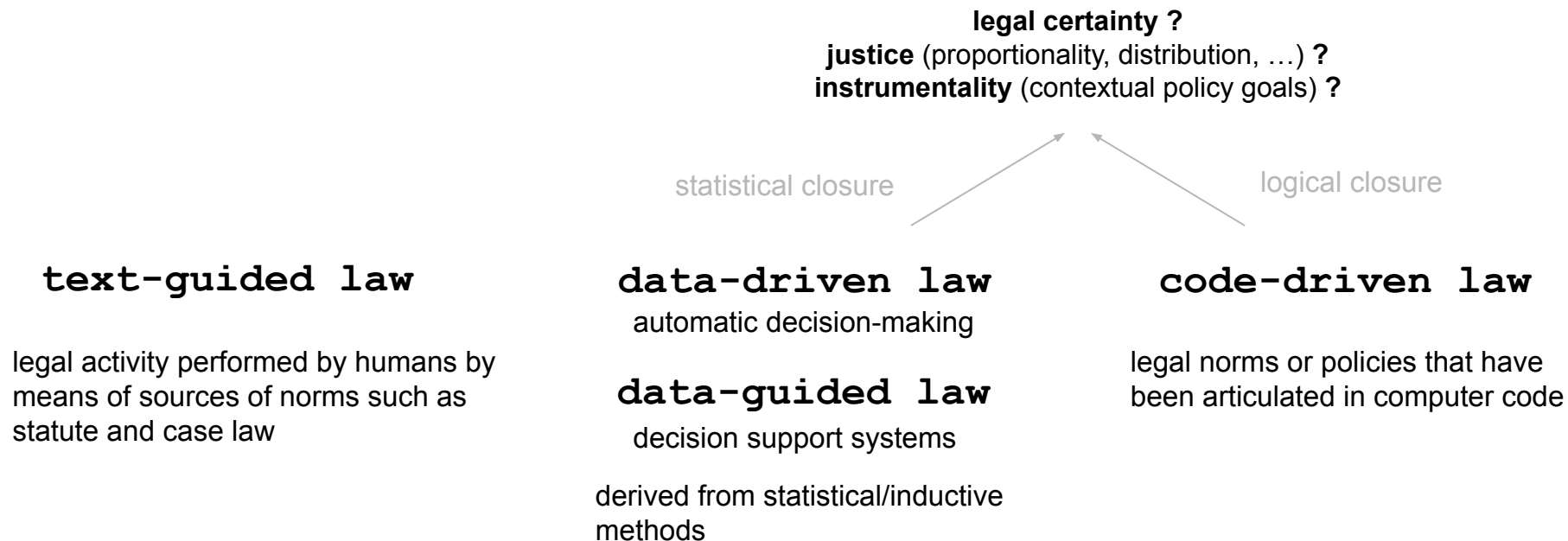
decision support systems

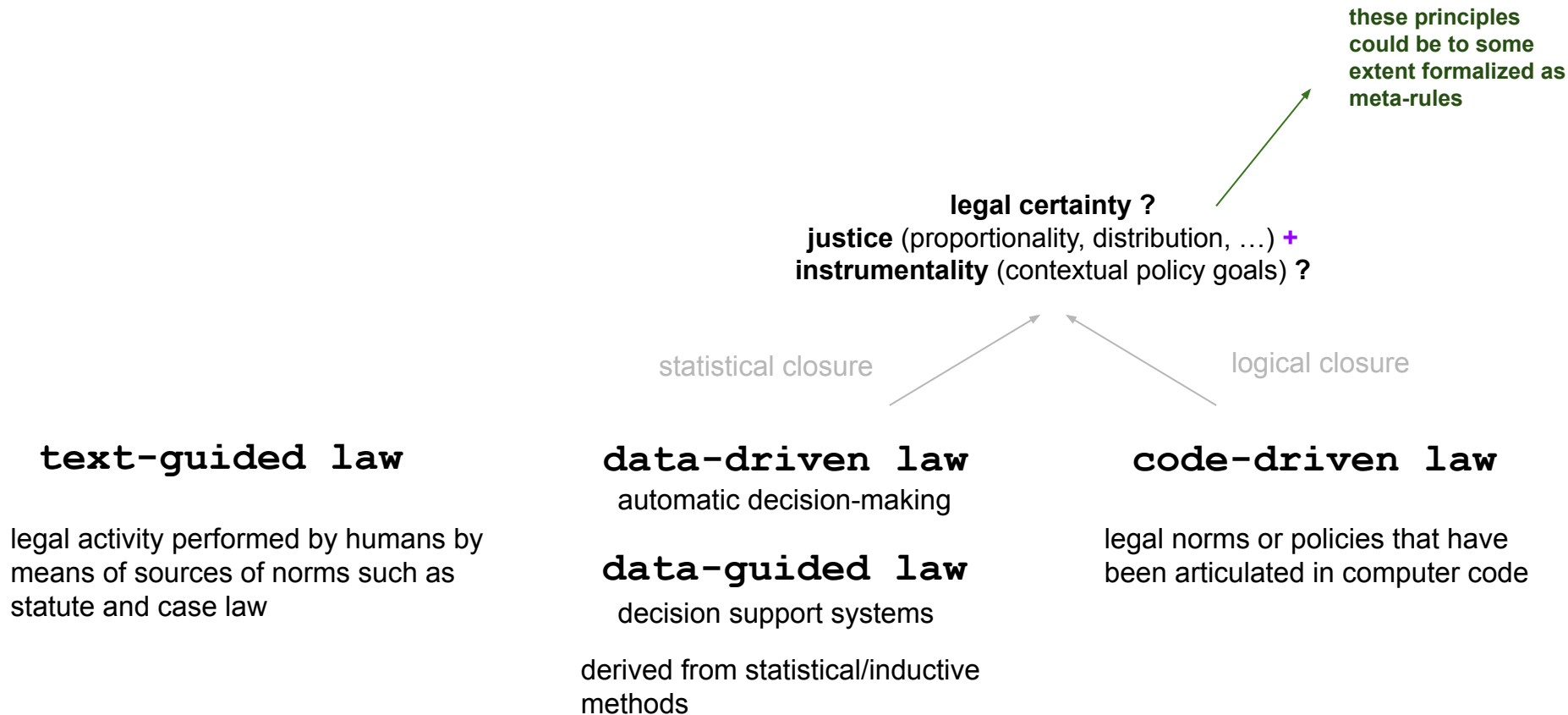
derived from statistical/inductive
methods

code-driven law

legal norms or policies that have
been articulated in computer code

is this all?





these goals
could be made explicit

these principles
could be to some
extent formalized as
meta-rules

legal certainty ?

justice (proportionality, distribution, ...) +
instrumentality (contextual policy goals) +

statistical closure

logical closure

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

code-driven law

legal norms or policies that have
been articulated in computer code

can we be sure about a certain formalization?

these principles could be to some extent formalized as meta-rules

these goals could be made explicit

legal certainty ?

justice (proportionality, distribution, ...) +
instrumentality (contextual policy goals) +

statistical closure

logical closure

text-guided law

legal activity performed by humans by means of sources of norms such as statute and case law

data-driven law

automatic decision-making

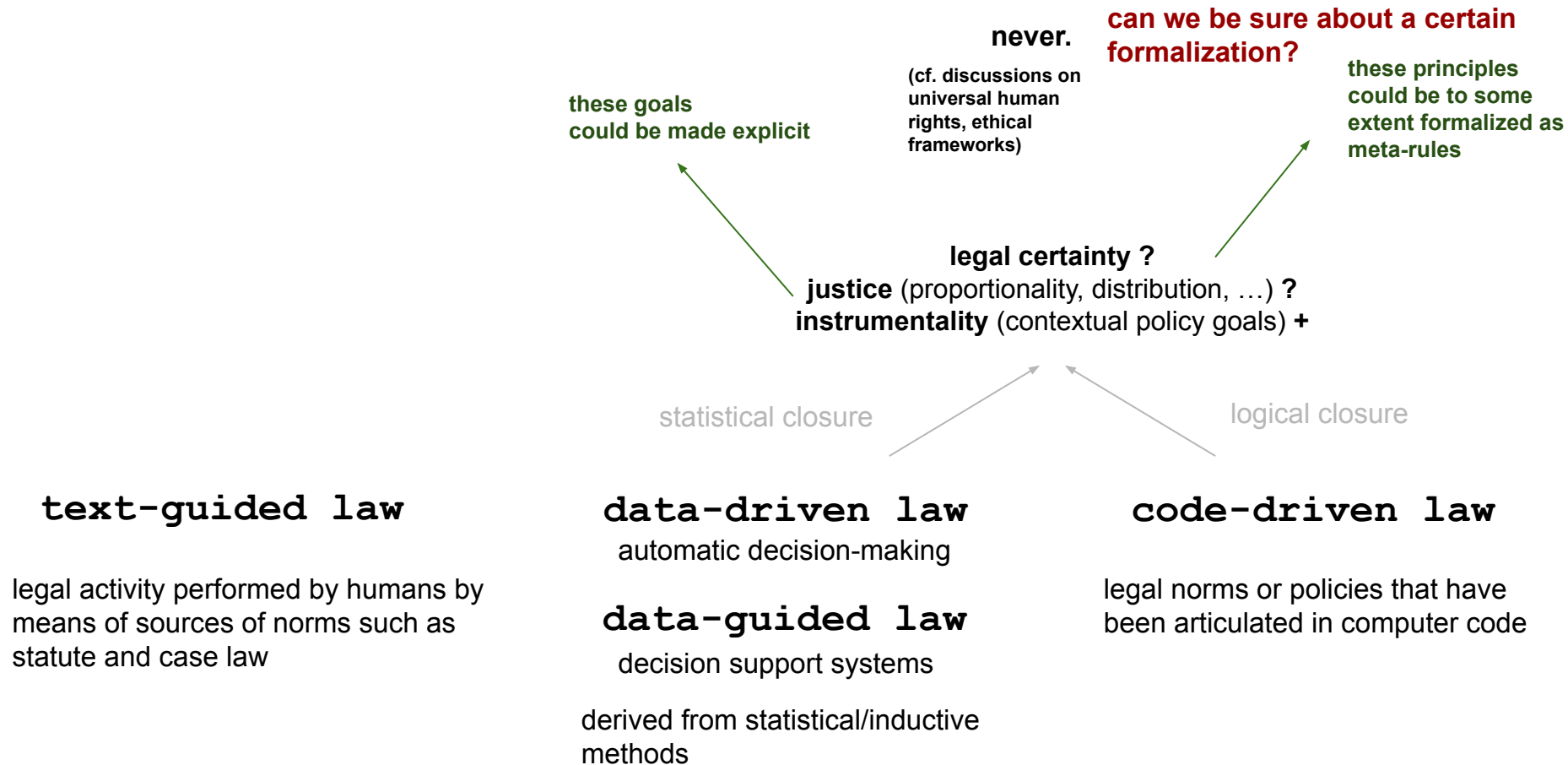
data-guided law

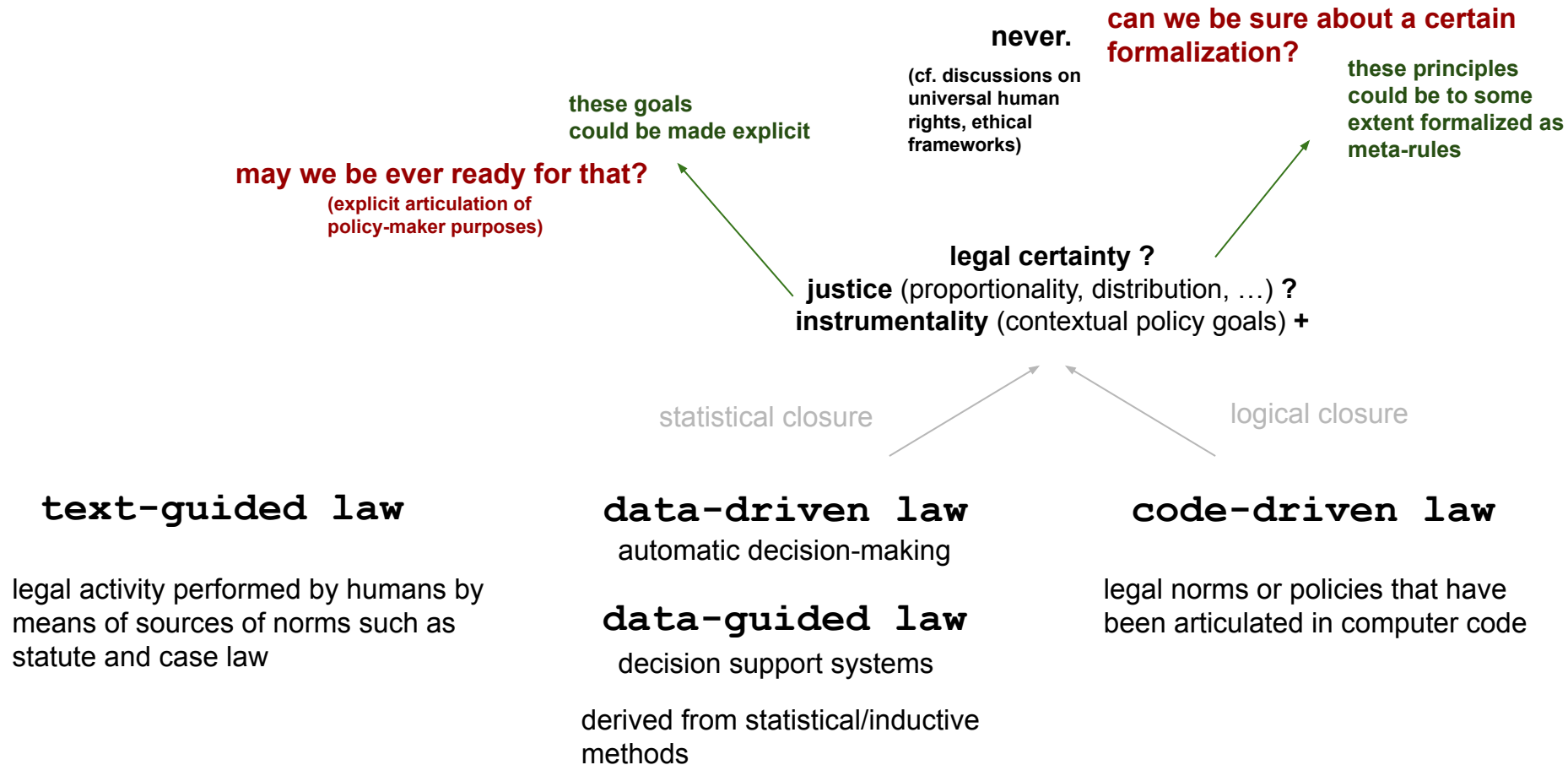
decision support systems

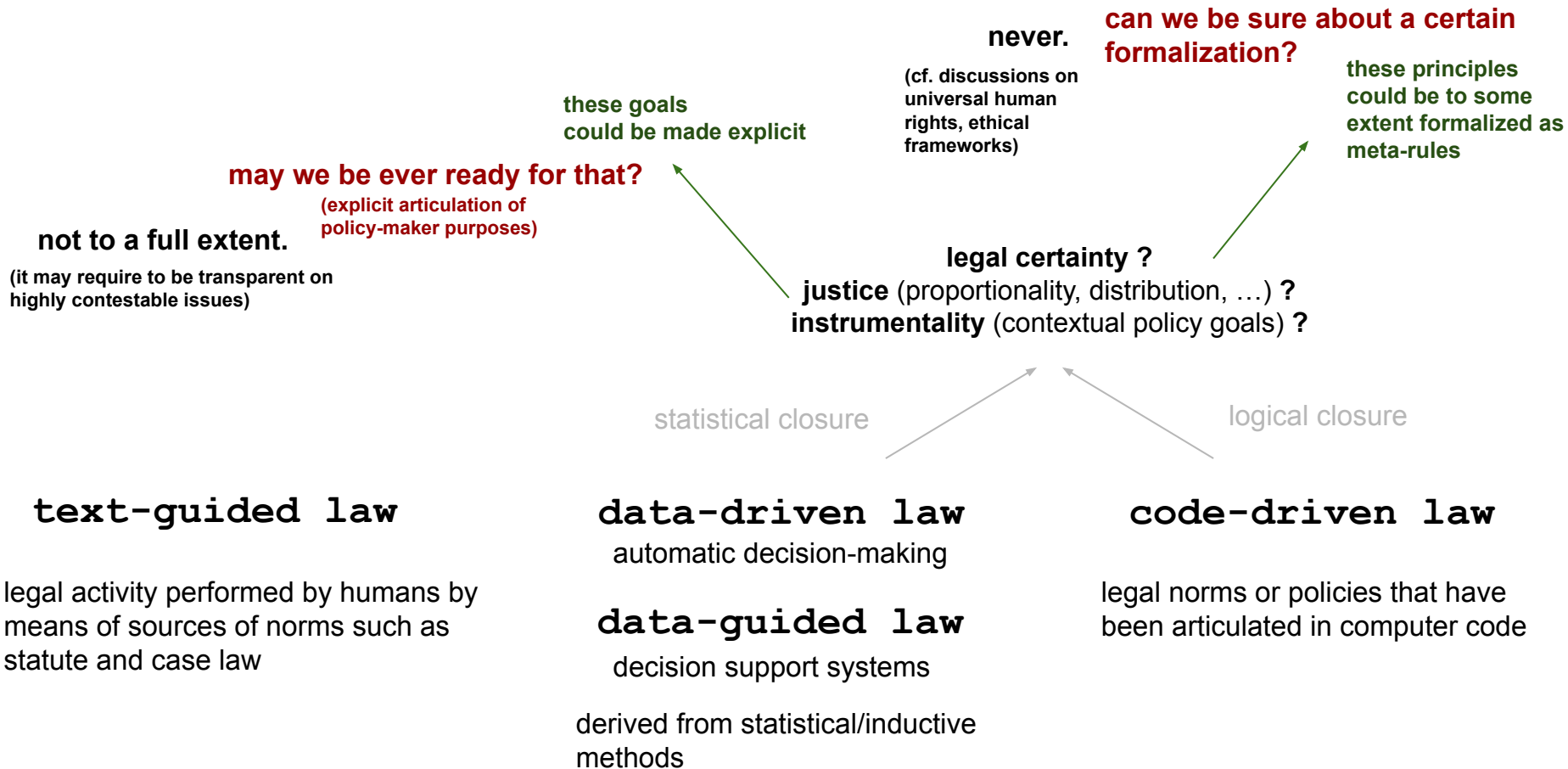
derived from statistical/inductive methods

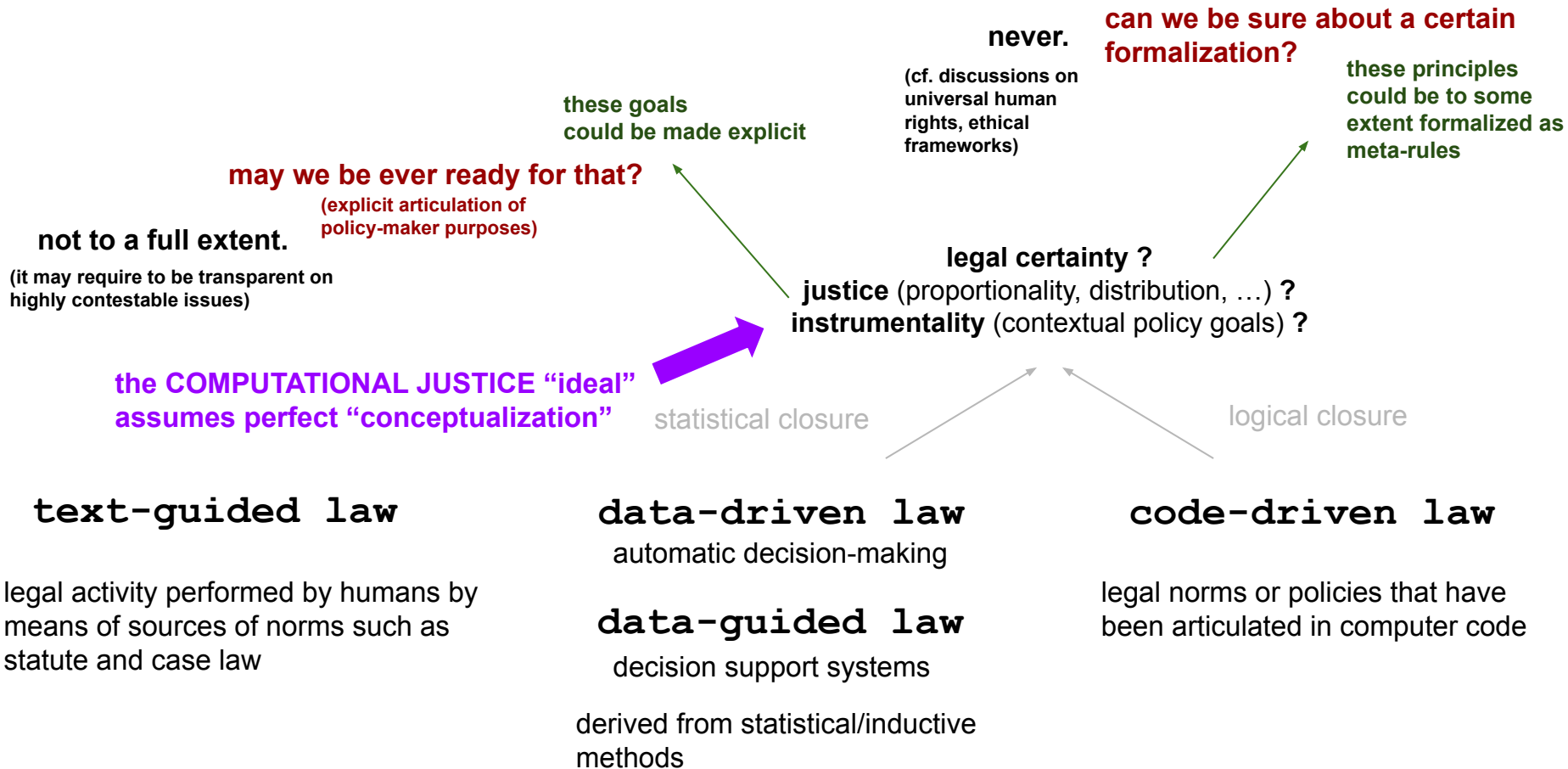
code-driven law

legal norms or policies that have been articulated in computer code









the COMPUTATIONAL LEGALISM “ideal”
works only in so far we forget all what
may go wrong

the COMPUTATIONAL JUSTICE “ideal”
assumes perfect “conceptualization”

legal certainty ?
justice (proportionality, distribution) ?
instrumentality (contextual policy goals) ?

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

code-driven law

legal norms or policies that have
been articulated in computer code

~~the COMPUTATIONAL LEGALISM “ideal”
works only in so far as we forget all what
may go wrong~~

let's accept things may go wrong at all level

the COMPUTATIONAL JUSTICE “ideal”
assumes perfect “conceptualization”

legal certainty ?

justice (proportionality, distribution) ?

instrumentality (contextual policy goals) ?

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

code-driven law

legal norms or policies that have
been articulated in computer code

~~the COMPUTATIONAL LEGALISM “ideal”
works only in so far as we forget all what may
go wrong~~

let's accept things may go wrong at all level



people or other actors should
be able to *appeal*

the COMPUTATIONAL JUSTICE “ideal”
assumes perfect “conceptualization”

legal certainty ?

justice (proportionality, distribution) ?

instrumentality (contextual policy goals) ?

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

code-driven law

legal norms or policies that have
been articulated in computer code

~~the COMPUTATIONAL LEGALISM “ideal”
works only in so far as we forget all what
may go wrong~~

let's accept things may go wrong at all level



people or other actors should
be able to *appeal*

legal certainty ?
justice (proportionality, distribution) ?
instrumentality (contextual policy goals) ?

~~the COMPUTATIONAL JUSTICE “ideal”
assumes perfect conceptualization”~~

let's accept we need to integrate
systematically feedbacks from
higher-order courts, jurisprudence or
other normative sources

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

code-driven law

legal norms or policies that have
been articulated in computer code

~~the COMPUTATIONAL LEGALISM “ideal”
works only in so far as we forget all what
may go wrong~~

let's accept things may go wrong at all level



people or other actors should
be able to **appeal**

legal certainty ?
justice (proportionality, distribution) ?
instrumentality (contextual policy goals) ?

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

~~the COMPUTATIONAL JUSTICE “ideal”
assumes perfect conceptualization”~~

let's accept we need to integrate
systematically feedbacks from
higher-order courts, jurisprudence or
other normative sources



courts or other actors should
be able to **quash/overrule**

code-driven law

legal norms or policies that have
been articulated in computer code

~~the COMPUTATIONAL LEGALISM “ideal”
works only in so far as we forget all what
may go wrong~~

let's accept things may go wrong at all level



people or other **actors** should
be able to **appeal**

legal certainty ?
justice (proportionality, distribution) ?
instrumentality (contextual policy goals) ?

including computational actors
possibility of **continuous, automated testing/verification**

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

~~the COMPUTATIONAL JUSTICE “ideal”
assumes perfect conceptualization”~~

let's accept we need to integrate
systematically feedbacks from
higher-order courts, jurisprudence or
other normative sources



courts or other **actors** should
be able to **quash/overrule**

code-driven law

legal norms or policies that have
been articulated in computer code

~~the COMPUTATIONAL LEGALISM “ideal”
works only in so far as we forget all what
may go wrong~~

let's accept things may go wrong at all level



people or other actors should
be able to ***appeal***

legal certainty ?
justice (proportionality, distribution) ?
instrumentality (contextual policy goals) ?

including computational actors
possibility of **continuous, automated testing/verification**

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

~~the COMPUTATIONAL JUSTICE “ideal”
assumes perfect conceptualization”~~

let's accept we need to integrate
systematically feedbacks from
higher-order courts, jurisprudence or
other normative sources



courts or other actors should
be able to ***quash/overrule***

code-driven law

monolithical systems

code-guided law

ecological system including
interfaces with humans

derived from legal norms or policies that
have been articulated in computer code
(or from other methods)

~~the COMPUTATIONAL LEGALISM “ideal”
works only in so far we forget all what
may go wrong~~

let's accept things may go wrong at all level



people or other actors should
be able to **appeal**

legal certainty ?
justice (proportionality, distribution) ?
instrumentality (contextual policy goals) ?

including computational actors
possibility of **continuous, automated testing/verification**

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

~~the COMPUTATIONAL JUSTICE “ideal”
assumes perfect conceptualization”~~

let's accept we need to integrate
systematically feedbacks from
higher-order courts, jurisprudence or
other normative sources



courts or other actors should
be able to **quash/overrule**

code-driven law

monolithical systems

code-guided law

ecological system including
interfaces with humans

derived from legal norms or policies that
have been articulated in computer code
(or from other methods). **(its authority
inferior to human authorities)**

~~the COMPUTATIONAL LEGALISM “ideal”
works only in so far as we forget all what
may go wrong~~

let's accept things may go wrong at all level

↓
people or other actors should
be able to **appeal**

legal certainty ?
justice (proportionality, distribution) ?
instrumentality (contextual policy goals) ?

including computational actors
possibility of continuous, automated testing/verification

text-guided law

legal activity performed by humans by
means of sources of norms such as
statute and case law

data-driven law

automatic decision-making

data-guided law

decision support systems

derived from statistical/inductive
methods

~~the COMPUTATIONAL JUSTICE “ideal”
assumes perfect conceptualization”~~

let's accept we need to integrate
systematically feedbacks from
higher-order courts, jurisprudence or
other normative sources

↓
courts or other actors should
be able to **quash/overrule**

code-driven law

monolithical systems

code-guided law

ecological system including
interfaces with humans

derived from legal norms or policies that
have been articulated in computer code
(or from other methods). **(its authority
inferior to human authorities)**

a serious technological gap exists today...

Part II:

what is normware?

A tentative ontology



HARDWARE

- physical device
- when running
⇒ **physical process**
- situated in a
physical environment



SOFTWARE

- symbolic device
- when running
⇒ **symbolic process**
- relies on
physical processes

A tentative ontology



HARDWARE

- **physical device**
- when running
⇒ **physical process**
- situated in a
physical environment



SOFTWARE

- **symbolic device**
- when running
⇒ **symbolic process**
- relies on
physical processes



NORMWARE

- ?
- when running
⇒ ?
- relies on ?

A tentative ontology



HARDWARE

- physical device
- when running
⇒ physical process
- situated in a
physical environment



SOFTWARE

- symbolic device
- when running
⇒ symbolic process
- relies on
physical processes



NORMWARE

- ?
- when running
⇒ ?
- relies on ?

**ARTIFACT
dimension**

**PROCESS
dimension**

Normware as artifacts...

Normware as artifacts:

1. directives concerning regulation

the cookie jar
must be full



you are prohibited
to eat cookies

you can not eat
cookies

Normware as artifacts:

1. directives concerning regulation

aiming to regulate situations in the world



the cookie jar
must be full



aiming to regulate behaviour



you are prohibited
to eat cookies

you can not eat
cookies

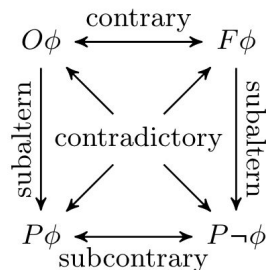
Normware as artifacts:

1. directives concerning regulation



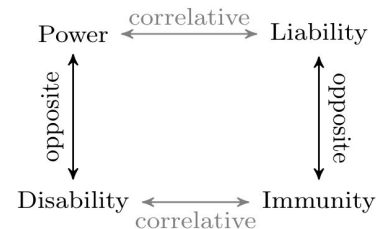
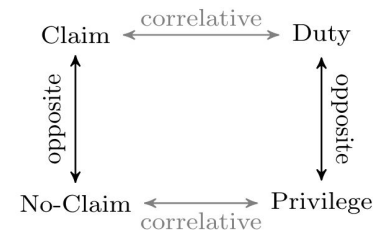
we can have explicit normative specifications...

Order Deny , Allow
Deny from all
Allow from `example.org`
access-control policies



*languages based on
deontic logic*

*languages based
on Hohfeld's
primitives*



Normware as artifacts:

1. directives concerning regulation



but programs in themselves
are already mandatory in nature!

`a := 2 + 2`

`?mother(maggie, bart)`

`animal :- dog.`

computer **has** to perform $2 + 2$...

computer **has** to prove that...

computer **has** to make `animal` true if `dog` is true

Normware as artifacts:

2. directives concerning terminology/meaning

what is a *cookie*?

what is a *jar*?

the cookie jar
must be full

what does it mean to be *full*?



who is *you*?

what is *eating*?

you are prohibited
to eat cookies

you can not eat
cookies

Normware as artifacts:

3. directives concerning expectations

eating cookies → cookies are destroyed → the jar is not full

the cookie jar
must be full

practical normative
reasoning
always require
some world
knowledge



you are prohibited
to eat cookies

you can not eat
cookies

Normware as artifacts:

4. devices intended to regulate

doors regulate entrances



behaviour can be regulated even if we do not have access to the inner decision-making mechanism!



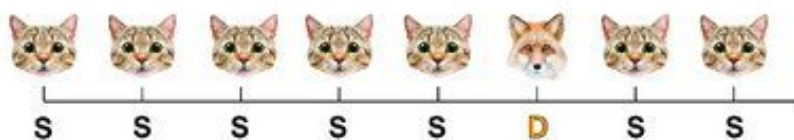
semaphores regulate traffic

Normware as artifacts:

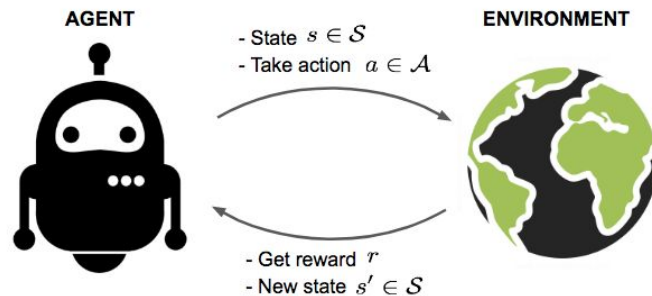
4. devices intended to regulate

black-boxes (eg. ML models) are also artifacts expressing some form of normativity/normality

“is this a cat?”



“how to (best) behave in certain conditions?”

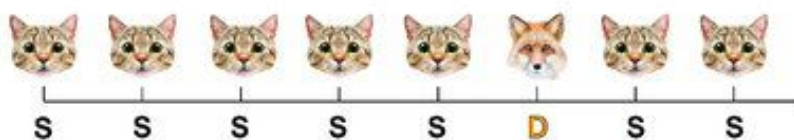


Normware as artifacts:

4. devices intended to regulate

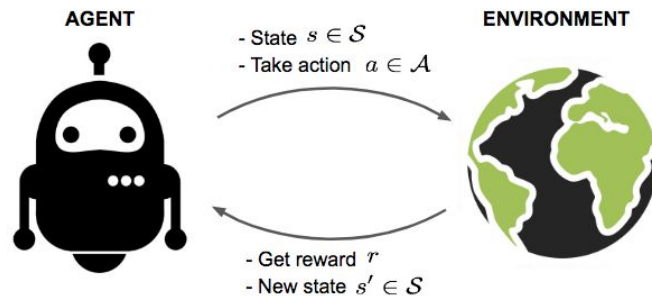
black-boxes (eg. ML models) are also artifacts expressing some form of normativity/normality

“is this a cat?”



from a **functional point of view**, they also count as normware!

“how to (best) behave in certain conditions?”



Normware as **processes**

Normware as processes:

1. regulation as control

Whether artificial or natural, designed or emergent,

what counts in control is

- the existence of some ***reference*** (the **target** of control),
- which the entity is set to either *approach* or *avoid* (the **direction** of control).

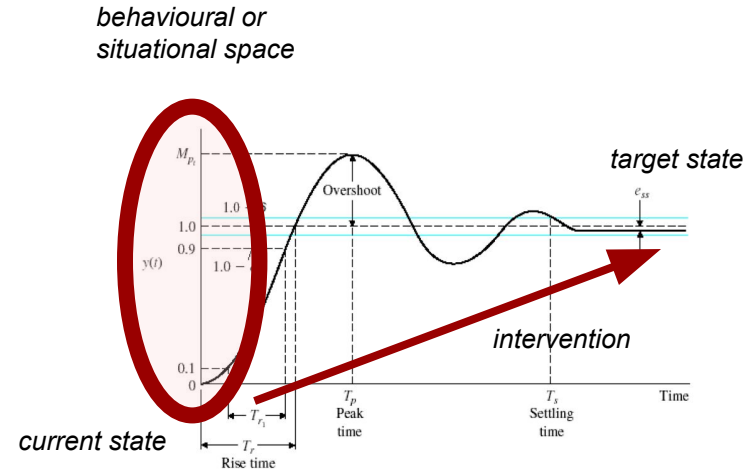
Normware as processes:

1. regulation as control

Whether artificial or natural, designed or emergent,

what counts in control is

- the existence of some **reference** (the **target** of control),
- which the entity is set to either *approach* or *avoid* (the **direction** of control).



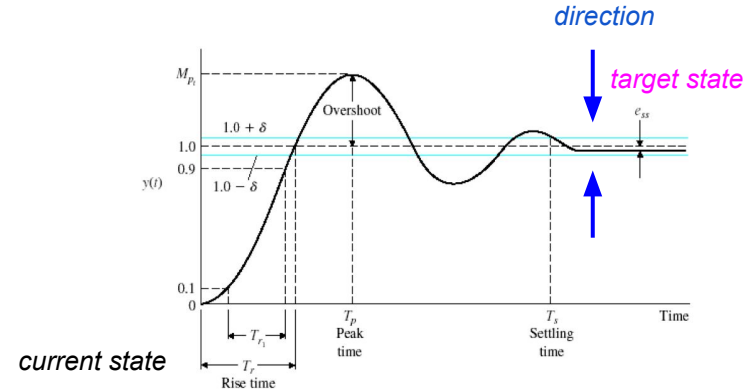
Normware as processes:

1. regulation as control

Whether artificial or natural, designed or emergent,

what counts in control is

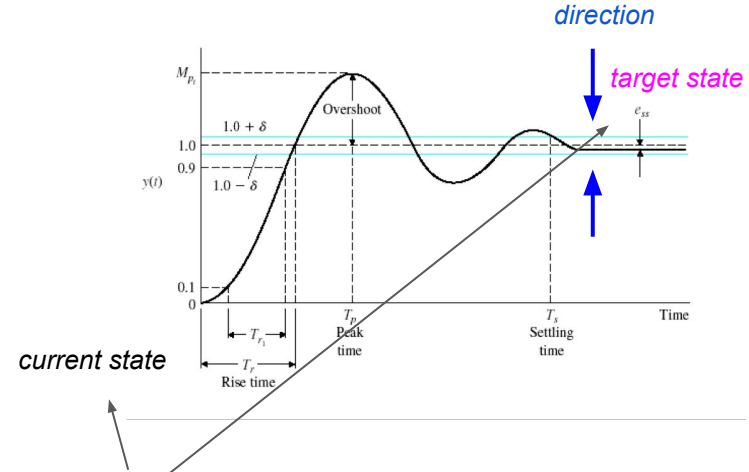
- the existence of some **reference** (the **target** of control),
- which the entity is set to either *approach* or *avoid* (the **direction** of control).



by defining directives by this **control signature** (target, direction), any regulative mechanisms can be abstracted from its implementation.

Normware as processes:

1. regulation as control

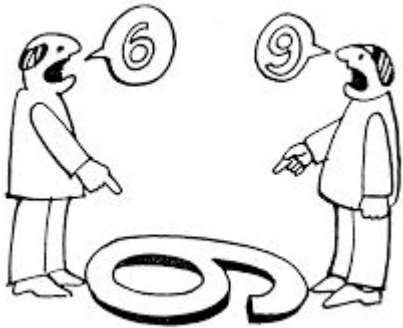


what defines the references though?

Normware as processes:

2. higher-order indetermination

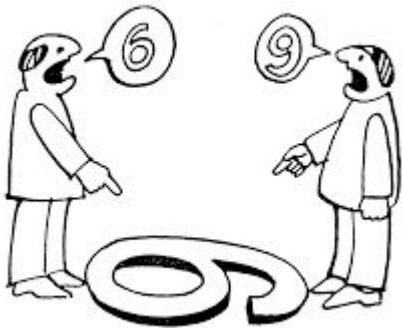
indeterminacy of references



Normware as processes:

2. higher-order indetermination

indeterminacy of references



indeterminacy of directives

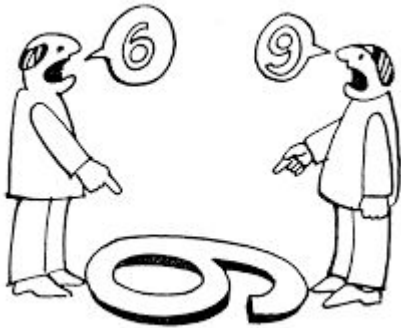
Normware as processes:

2. higher-order indetermination



indeterminacy of references

antinomies



indeterminacy of directives

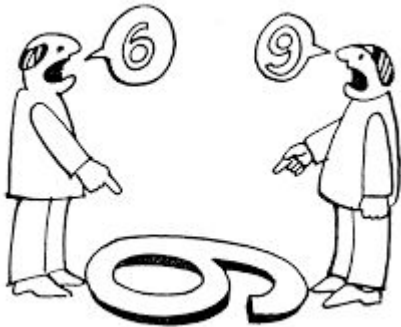
Normware as processes:

2. higher-order indetermination



indeterminacy of references

antinomies

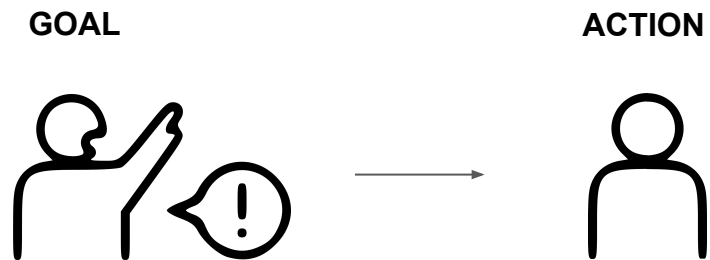


indeterminacy of directives

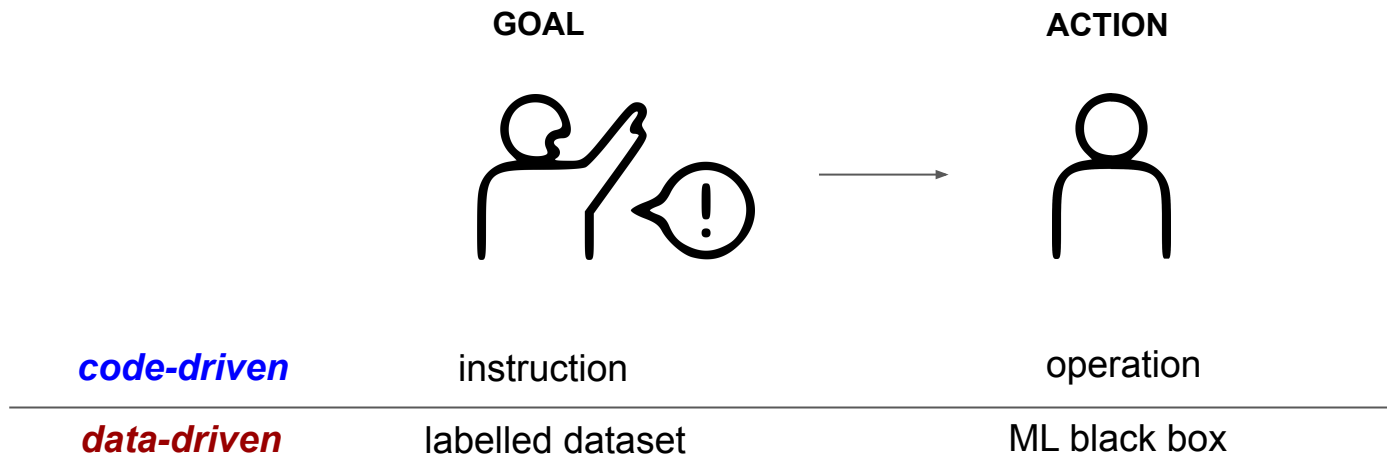


mechanisms of conflict resolution
are needed at systematic level!

Normware: first-order control

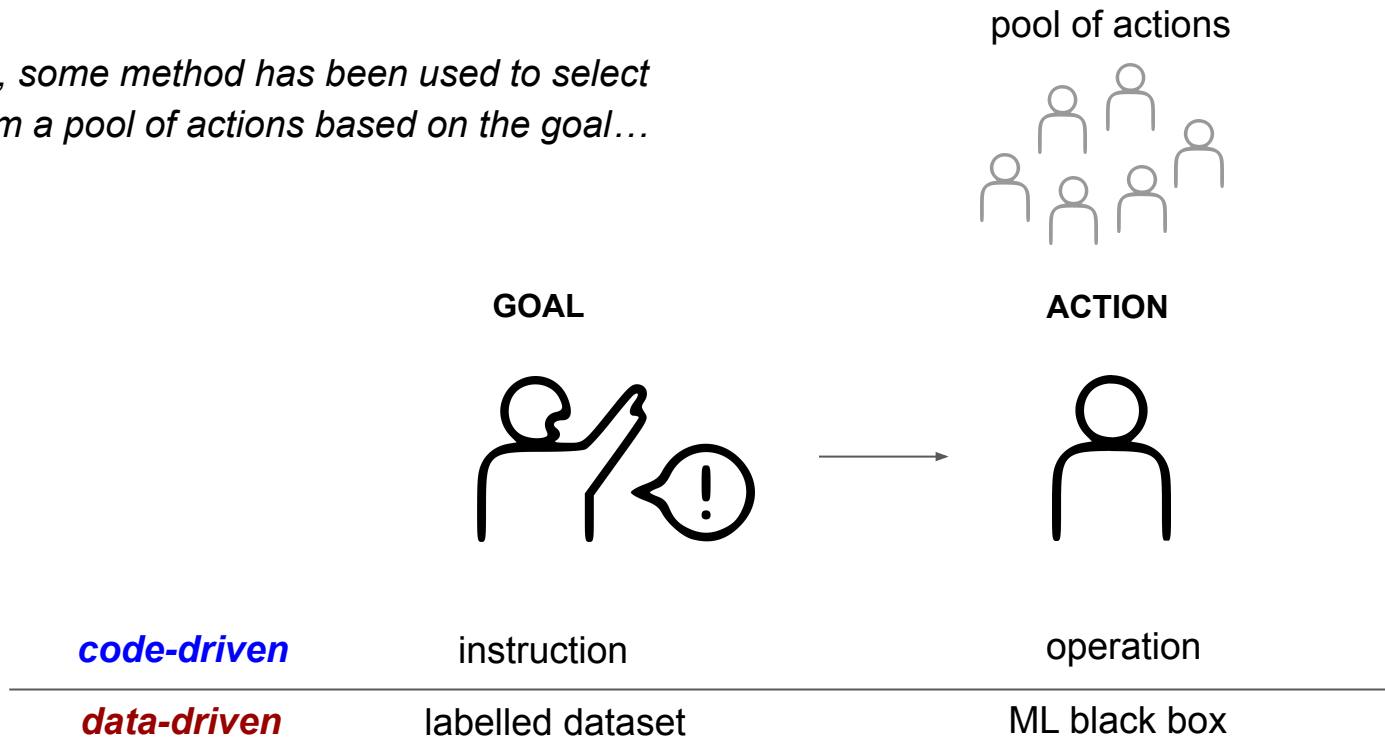


Normware: first-order control



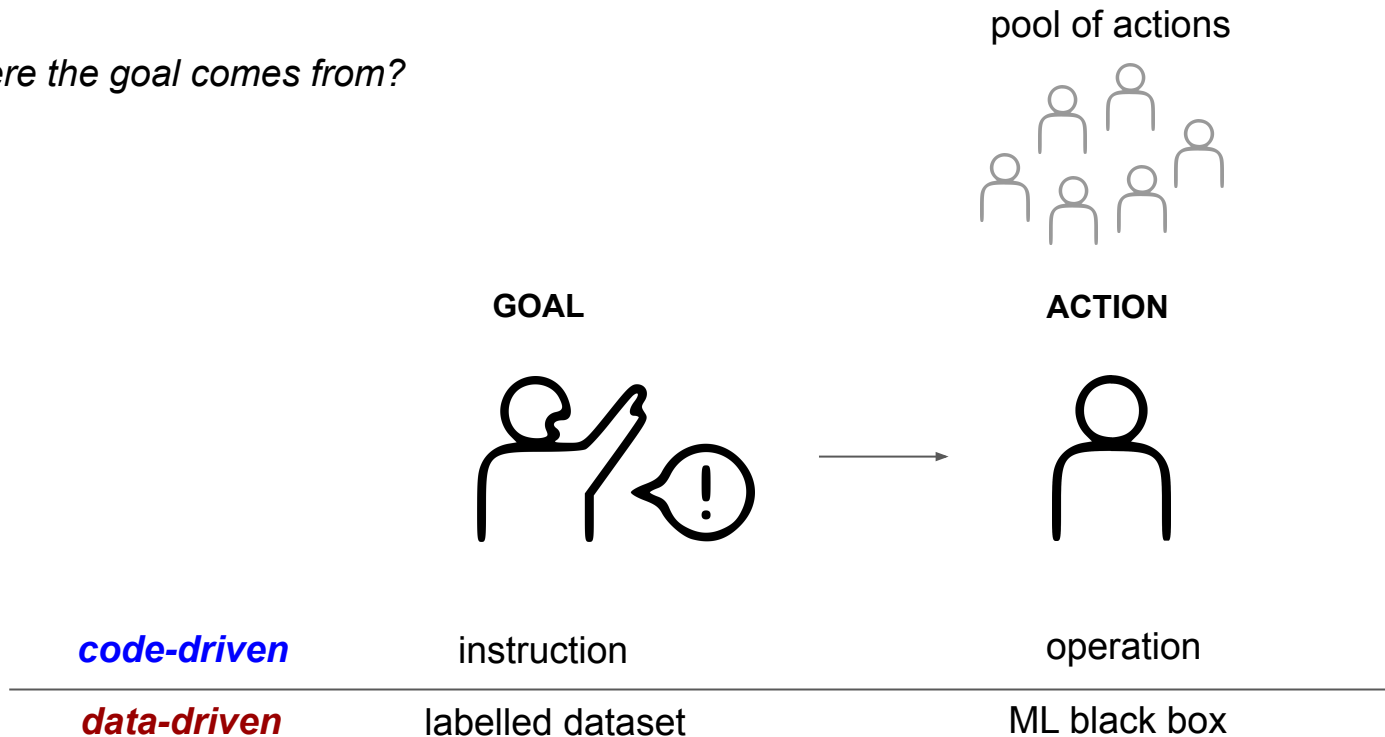
Normware: first-order control

In both cases, some method has been used to select the action from a pool of actions based on the goal...



Normware: first-order control

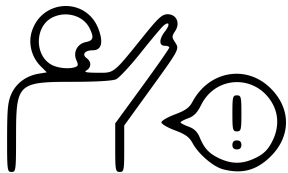
But then, where the goal comes from?



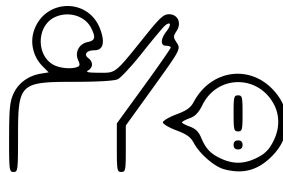
Normware: second-order control

But then, where the goal comes from?
We add depth!

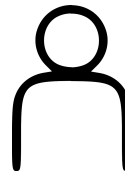
STRATEGIC GOAL



TACTICAL GOAL



ACTION



pool of actions



Normware: second-order control

But then, where the goal comes from?
We add depth!

pool of goals



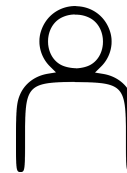
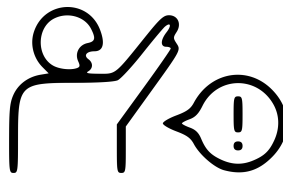
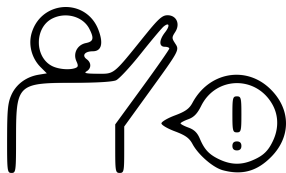
pool of actions



STRATEGIC GOAL

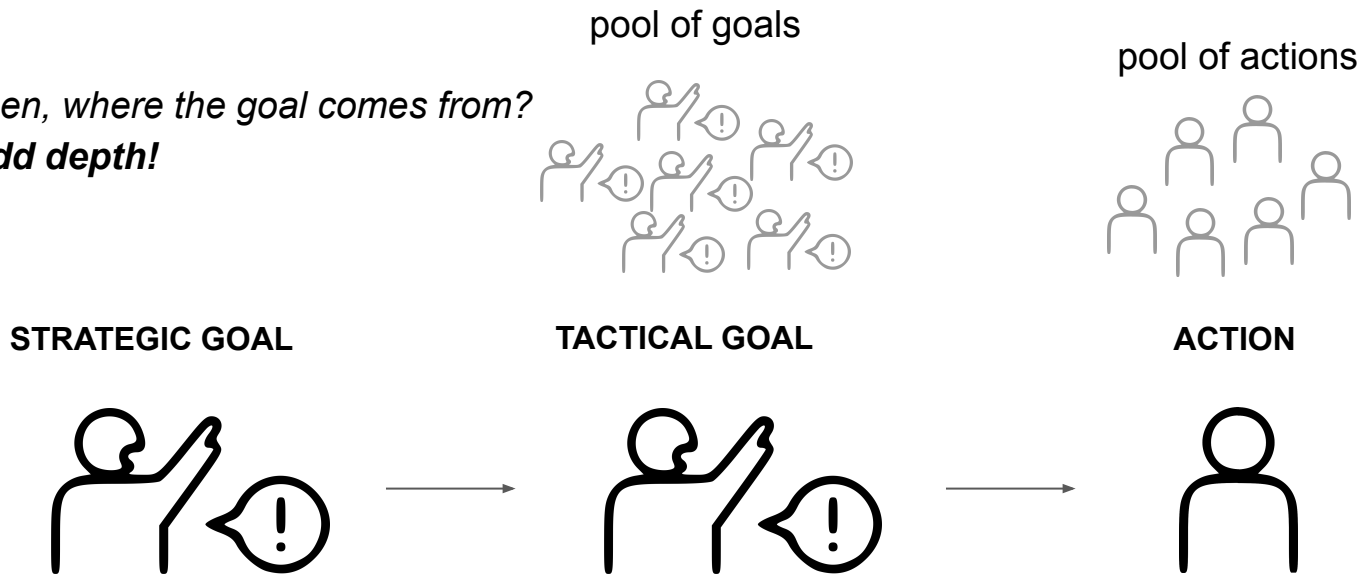
TACTICAL GOAL

ACTION



Normware: second-order control

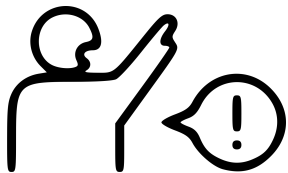
*But then, where the goal comes from?
We add depth!*



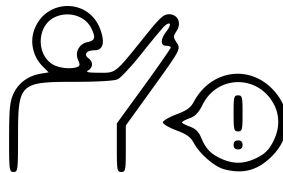
Normware: second-order control

cybernetic view on systems: policy, intelligence, operations

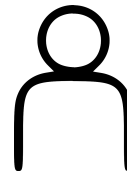
STRATEGIC GOAL



TACTICAL GOAL

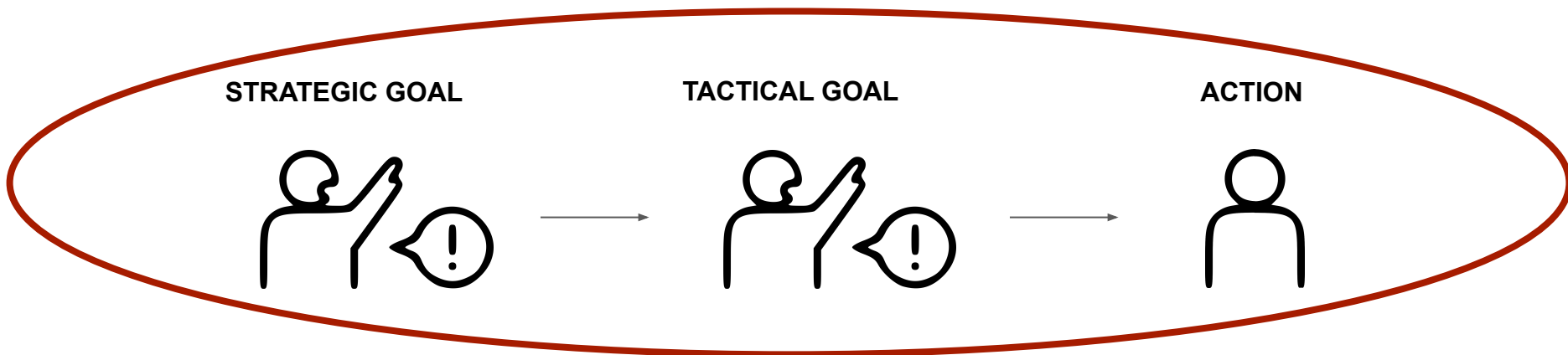


ACTION



Normware: second-order control

cybernetic view on systems: policy, intelligence, operations



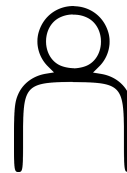
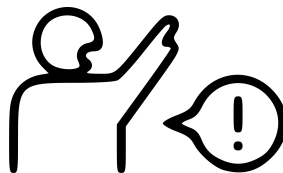
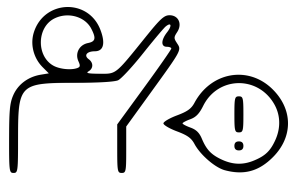
...yet it is about a single “organism”, not an “ecology”



**“TOTALITARIAN”
architecture**

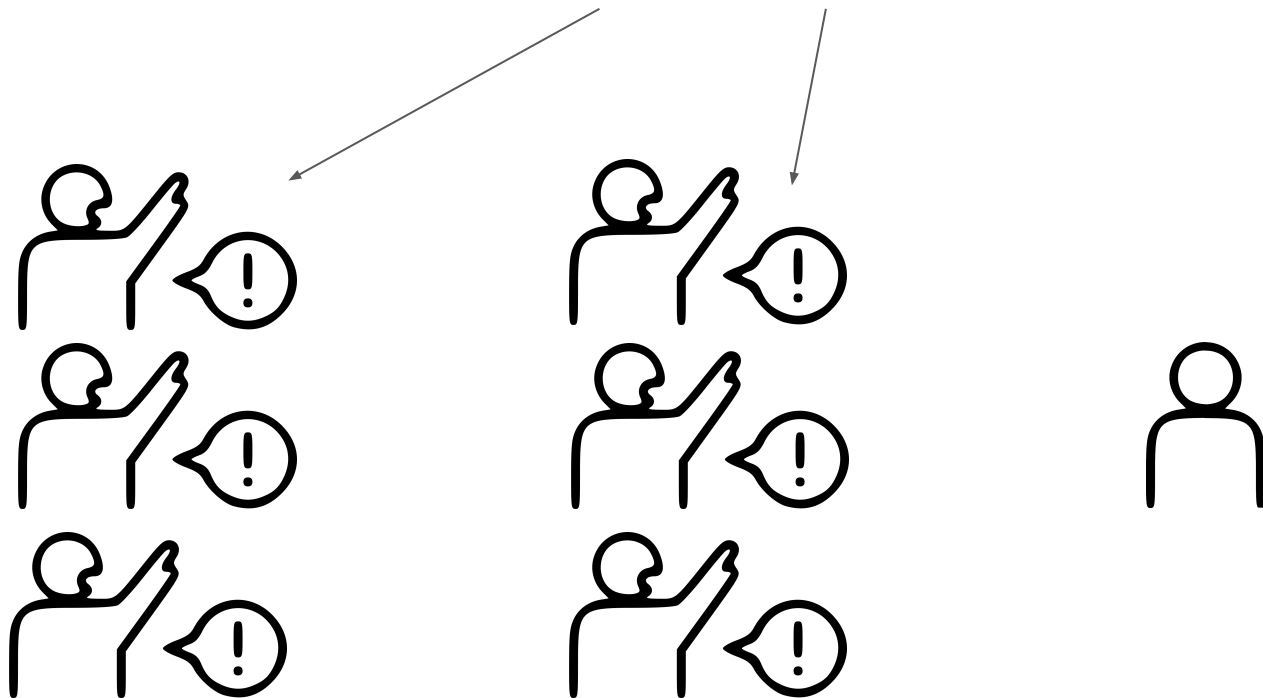
Normware: plural second-order control

*we need to acknowledge the presence of several **autonomous entities**,*



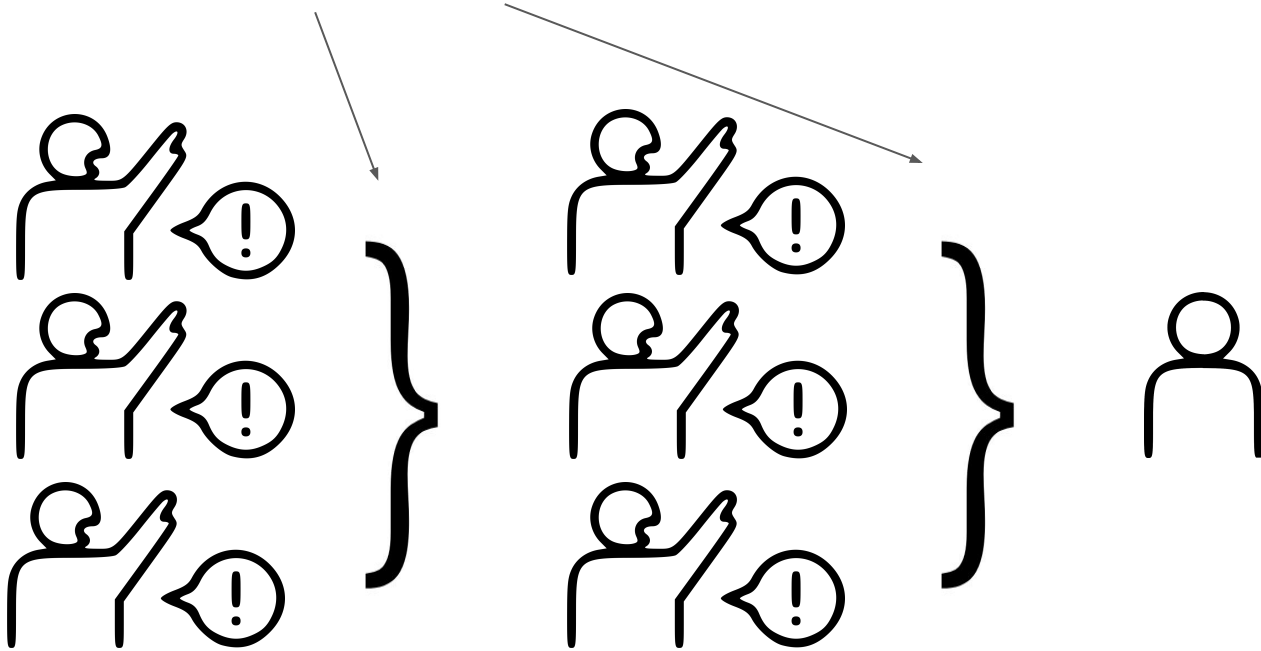
Normware: plural second-order control

*we need to acknowledge the presence of several **autonomous entities**,*



Normware: plural second-order control

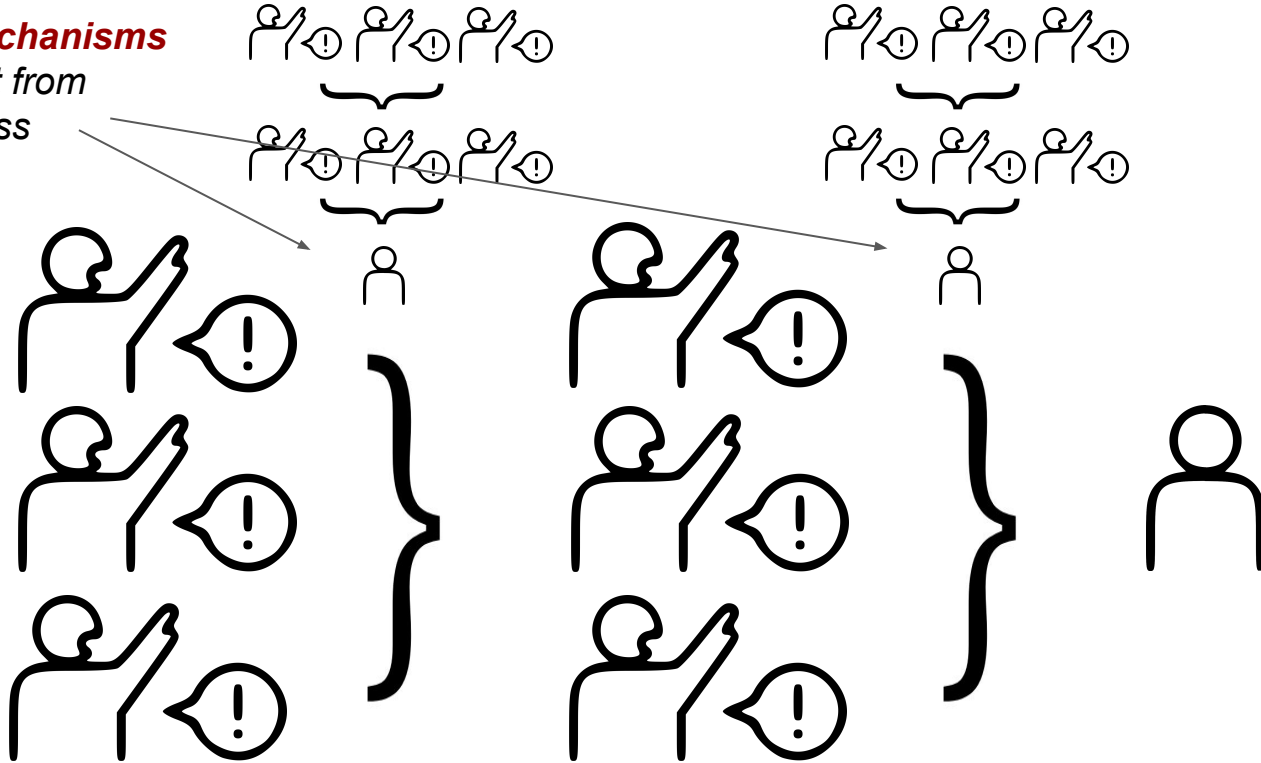
we need to acknowledge the presence of several *autonomous entities*,
and adequate conflict *resolution mechanisms*



Normware: plural second-order control

resolution mechanisms

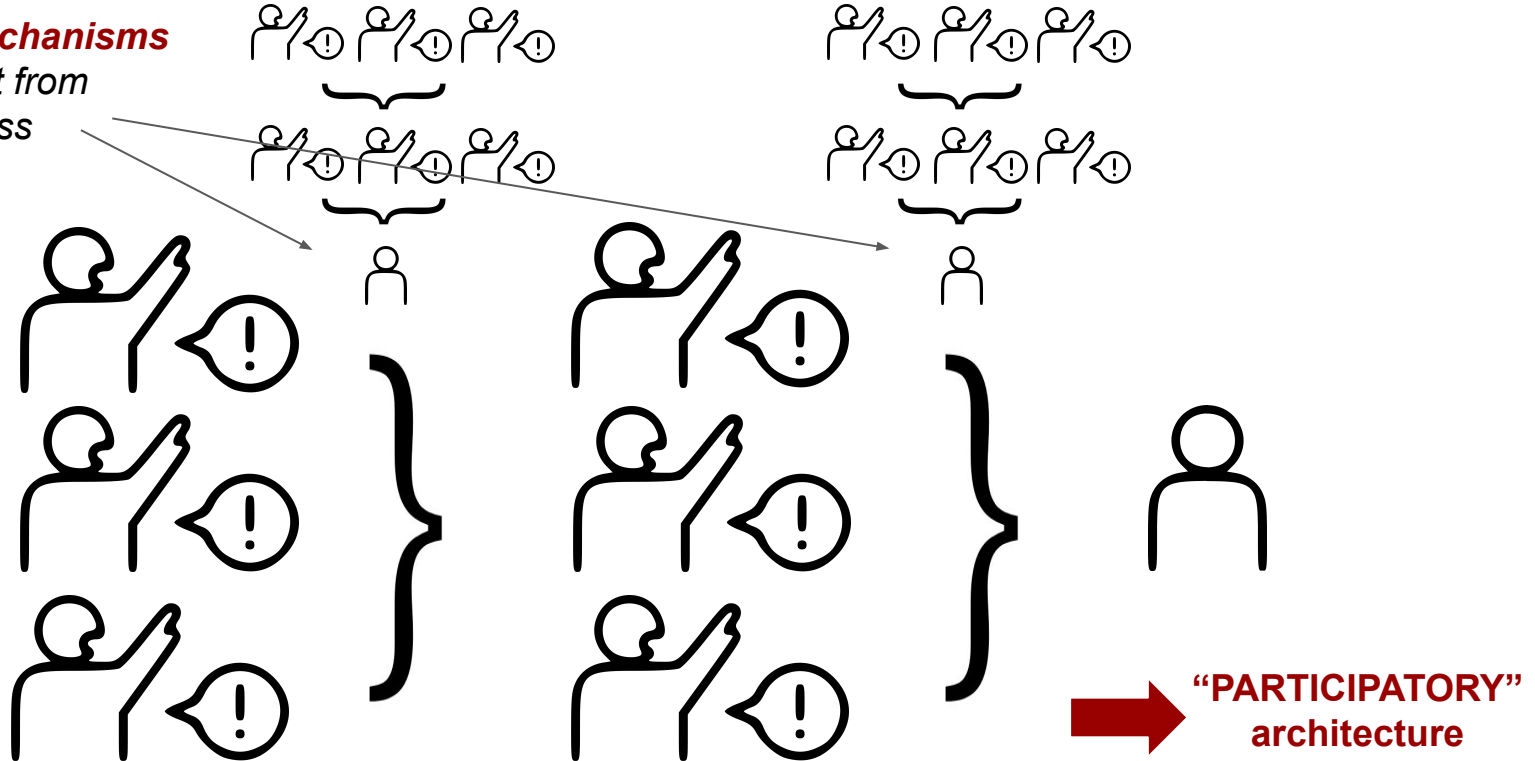
*may also result from
a similar process*



Normware: plural second-order control

resolution mechanisms

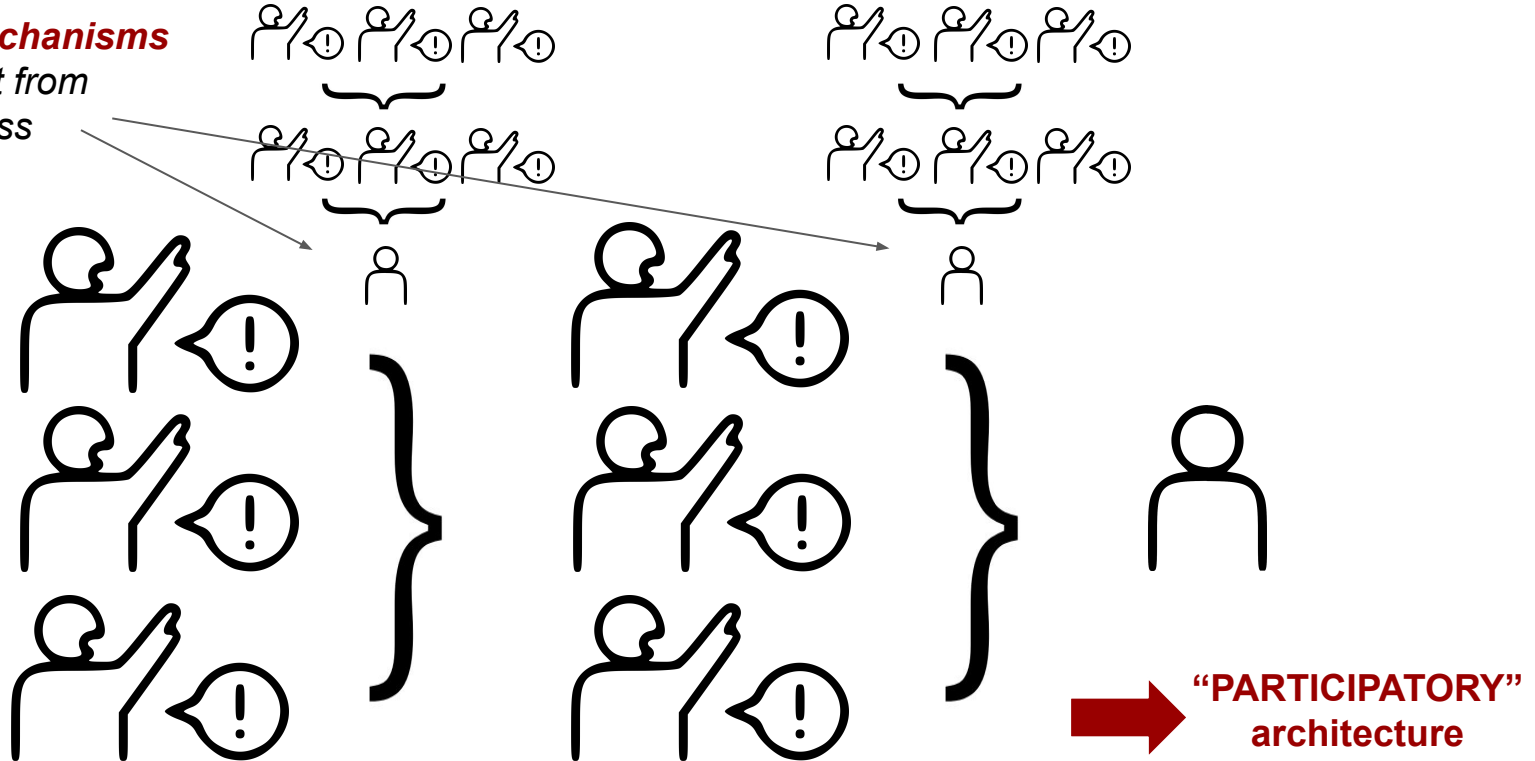
*may also result from
a similar process*



Normware: plural second-order control

resolution mechanisms

*may also result from
a similar process*



A less tentative ontology



HARDWARE

physical device

when running
⇒ **physical process**

situated in a
physical environment



SOFTWARE

symbolic device

when running
⇒ **symbolic process**

relies on
physical processes



NORMWARE

coordination device

when running
⇒ **coordination process**

relies on **symbolic** (possibly
hard-coded) **processes**

Relevant research directions for normware?

- **Artefact level:** going up in the abstraction ladder of specification languages...

Imperative programming

Declarative programming

Policy-based programming

desires/preferences as individual policies

norms as collective policies

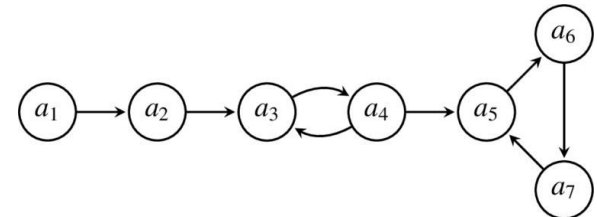
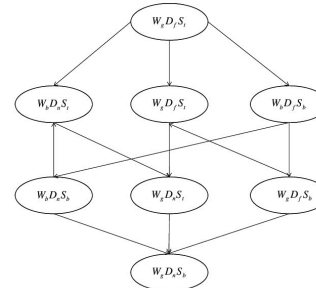
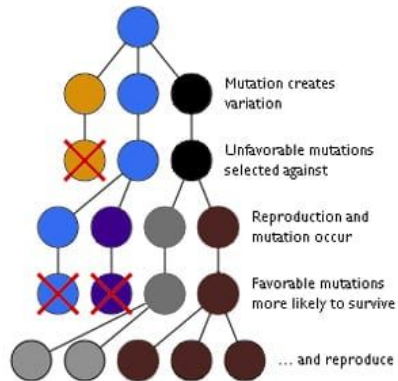
***Normative
specifications***

***Agent-based
Programming***

HOW
WHAT
WHY

Relevant research directions?

- **Process level:** better understanding/application of mechanisms of resolution, eg.
 - *preferential aggregation*
 - *voting systems (computational social theory)*
 - *formal argumentation frameworks*
 - *evolutionary algorithms?...*
- 



Key points

- There is continuity between institutional and computational activities.
- We don't need institutions to become more mechanical, institutional constructs need to be brought into the computational realm.
- As we have a plurality of normative sources & mechanisms, we need a plurality of computational normative sources & mechanisms.

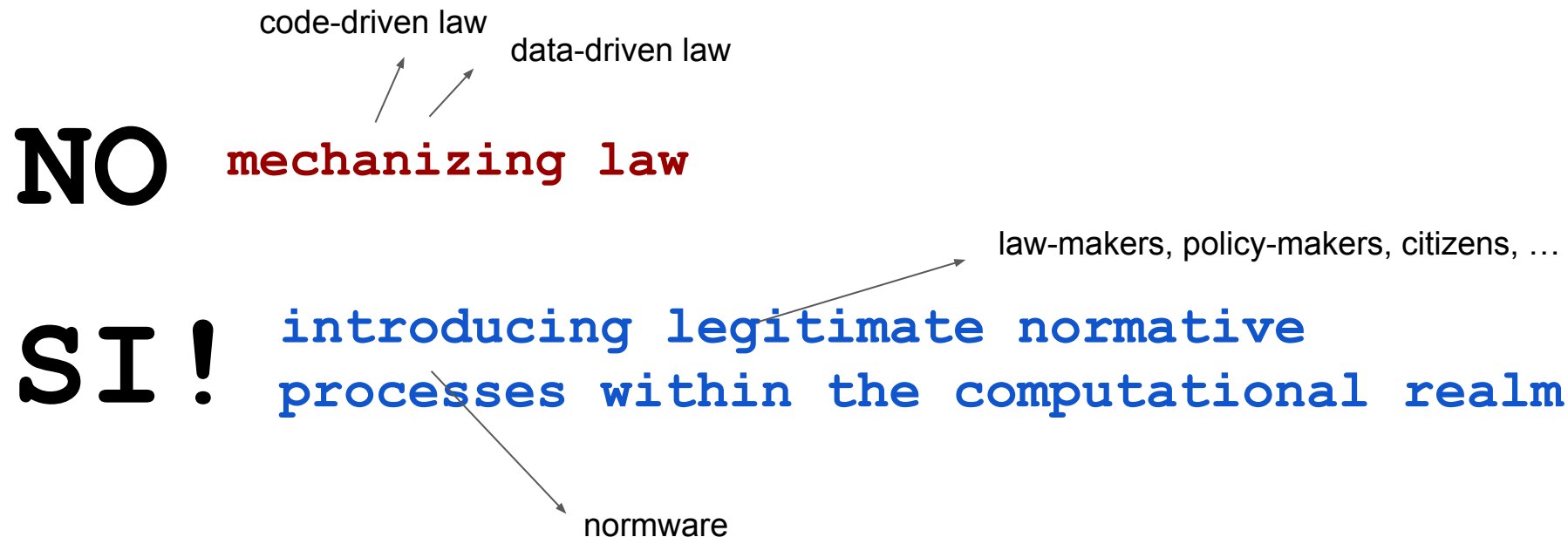
 **normware**

Contemporary socio-technical challenge:

NO mechanizing law

SI ! introducing legitimate normative
processes within the computational realm

Contemporary socio-technical challenge:





Code-driven NO, Normware SI!

4 July 2024, *Annual Conference of “Law, Science and Technology”*, Bologna
Panel on AI, Regulation, Ethics

Giovanni Sileno, g.sileno@uva.nl
Socially Intelligent Artificial Systems (SIAS),
Informatics Institute, University of Amsterdam