Versions and Applicability of Concept Definitions in Legal Ontologies

Szymon Klarman, **Rinke Hoekstra** and Marc Bron

Leibniz Center for Law
Faculty of Law, University of Amsterdam
http://www.leibnizcenter.org, hoekstra@uva.nl

OWLED 2008 DC, Gaithersburg

Outline

Context

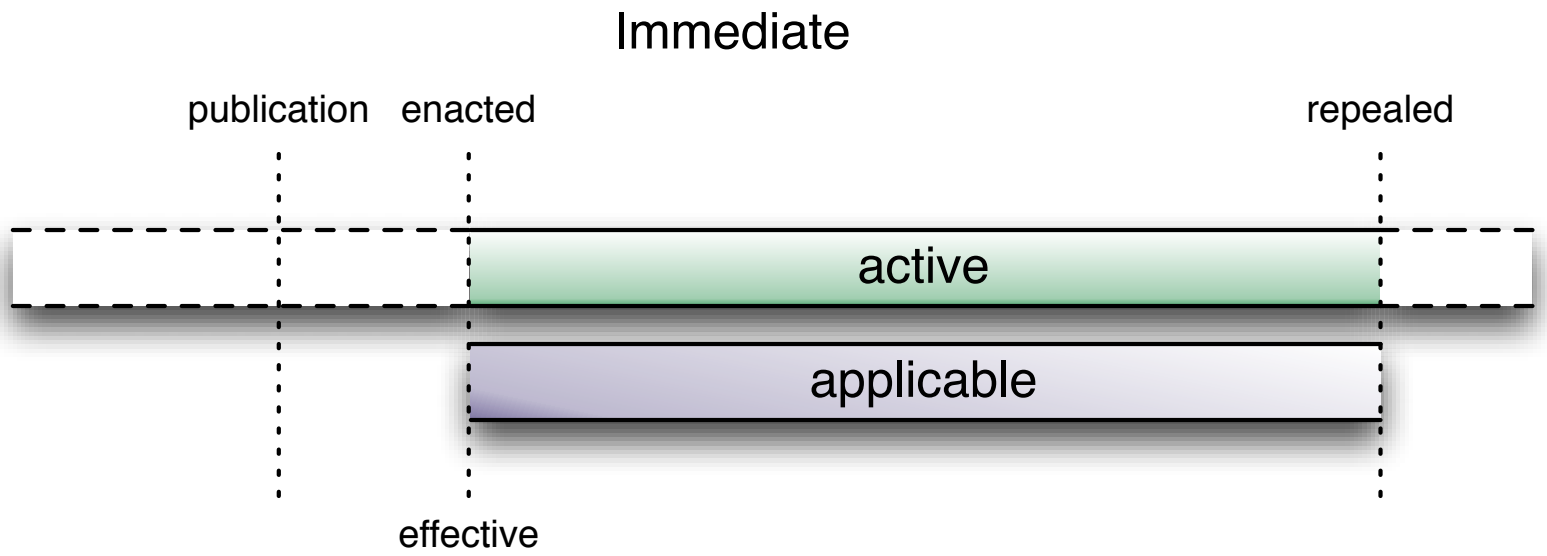
- Legal domain
  - Legislation, contracts, jurisprudence etc.
- Representation of
  - Norms
  - Definitions ($\rightarrow$ legal ontologies)
- . . . for the purpose of
  - Assessment, planning, simulation, harmonisation
- How to deal with versions?
  - Different *classification* of domain objects
  - *Reasoning* results in different outcome
  - Impact may be significant

Legal Ontologies

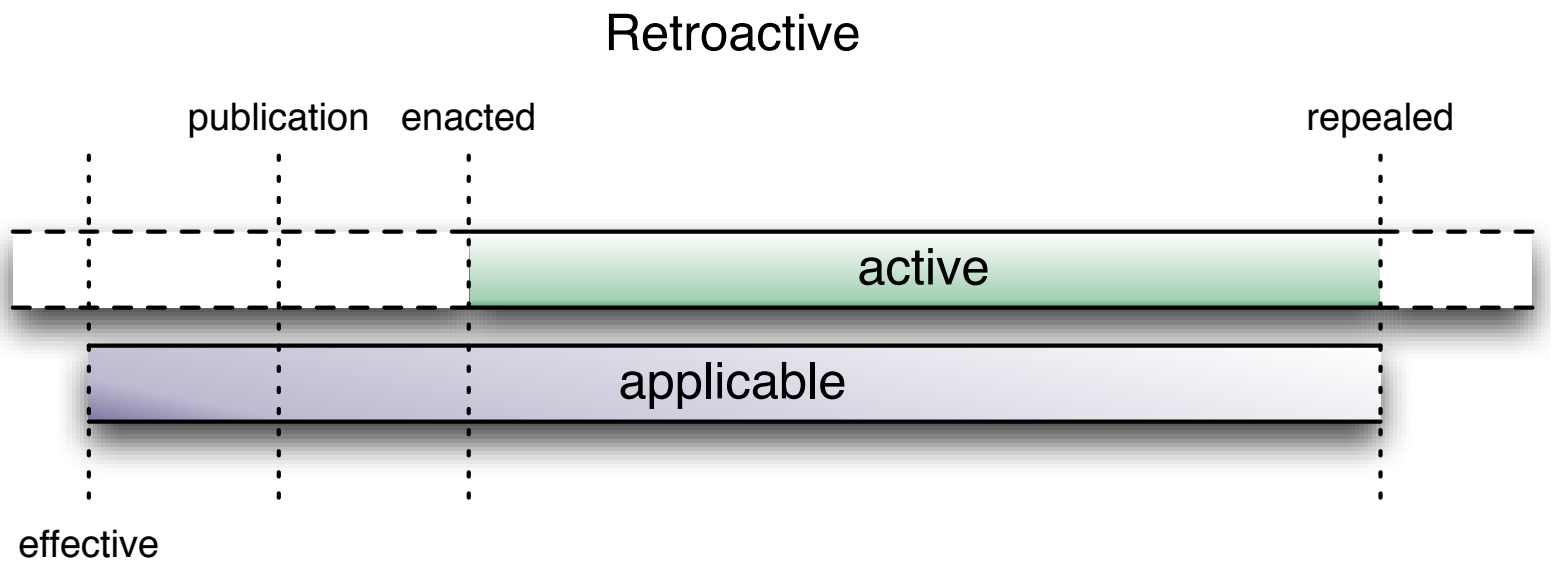
Variants of concept definitions in legislation

- Ordering on objects in the world
- Conflicts
  - Specificity (*lex specialis*)
  - Different issuers (*lex superior*)
  - **Through time** (*lex posterior*)
- Definitions hold independently, at the same time
- Complex determination of validity of definitions
  - Applicabiltity & Activity

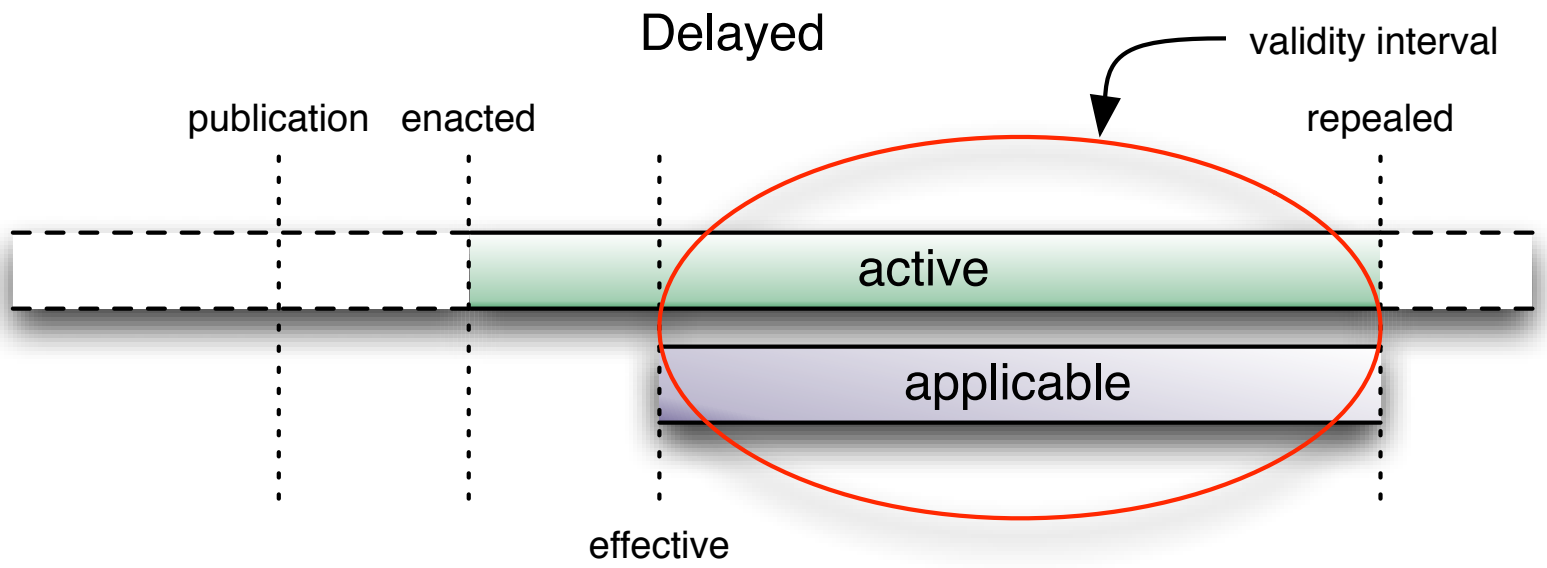
Validity Intervals



Validity Intervals



Validity Intervals



Ontology Evolution & Versioning

Approaches

- Evolution & Versioning
- Comparing two versions
- Time stamps (properties and classes)
  - Meta-ontology
  - Temporal Extension of KR language
  - Builtins
  - Rules

Drawbacks:

- Standard reasoning unaware of temporal information
  - Should infer conclusions that hold only at 'current' time
  - Relations may only hold between concepts that are both valid
- Snowball effect
- Rules may conflict with DL semantics, and lead to undecidability
- Multiple intervals?

Focus & Requirements

Versioning in Law

- Impact on classification of objects in a domain

Requirements

- General purpose representation formalism (OWL DL)
- Incremental versioning
  - New version of a concept should have *minimal impact*
- Co-existence of multiple (incompatible) versions
- Ability to switch between versions
- Reasoning on both versioned and version-independent concepts
- Validity depends on multiple intervals

Approach

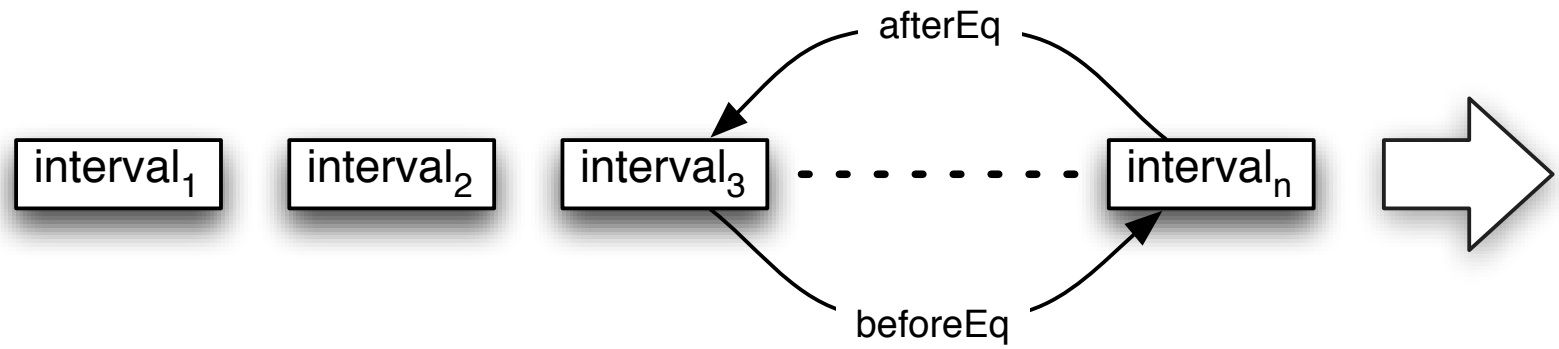
- Expressiveness of $\mathcal{SHOIN}$ ($\Rightarrow$ supported by OWL-DL).
- Representation:
  - A **dynamic concept** is a concept whose meaning changes over time
  - Each new **concept variant** is introduced as a *defined* class, subsumed by the dynamic concept class.
  - Concept variants are *valid* within some combination of intervals.
  - A DL reasoner classifies individuals as class members, based on the choice of a *current interval*.

Incorporating Time

Three layers:

- Timeline and current interval:
  - A **finite**, **discrete axis**,
  - undecomposable intervals + total ordering relation,
  - selection mechanism for current interval,
- Stamping individuals,
- Scoping concepts.

Timeline



$$\text{TimeInterval} \equiv \{\text{interval}_1, \ldots, \text{interval}_n\}$$

for any interval_i and interval_j, if $i \leq j$ then the ABox contains:

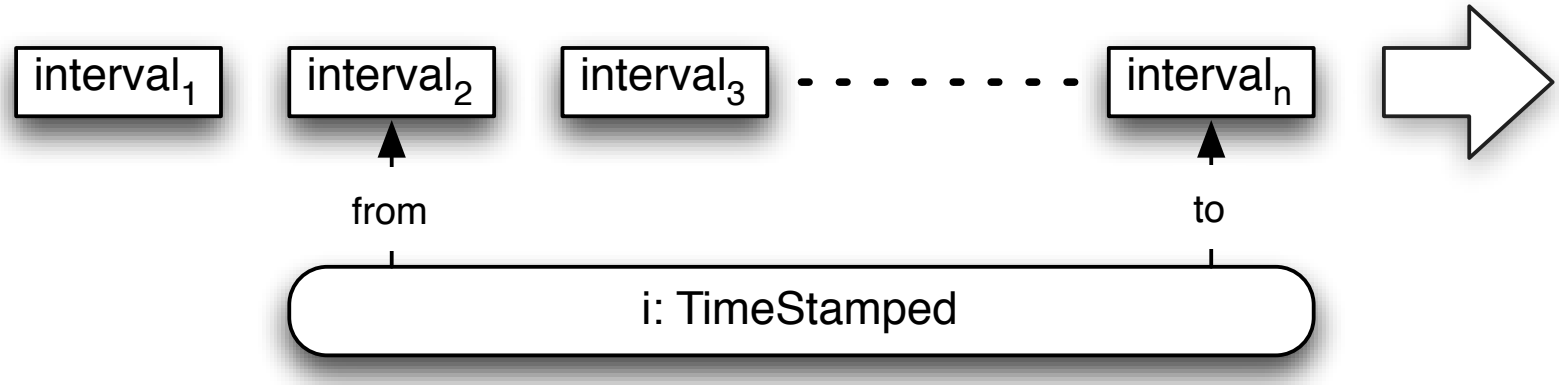
- $\text{beforeEq}(\text{interval}_i, \text{interval}_j)$
- $\text{afterEq}(\text{interval}_j, \text{interval}_i)$

We specify the **valid time** by defining:

$$\text{CurrentInterval} \equiv \{\text{interval}_i\}$$

where $i \in \{1, \ldots, n\}$

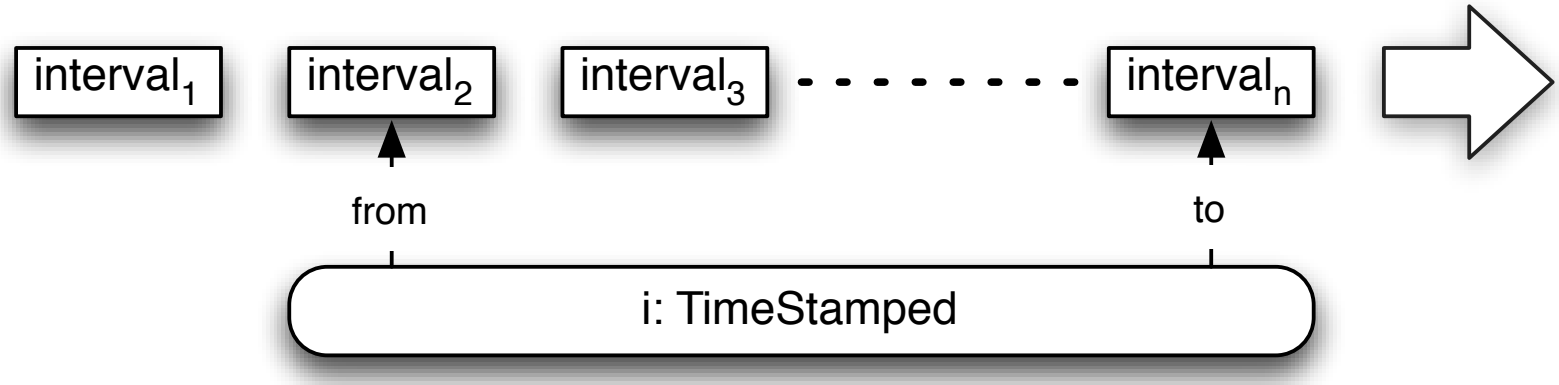
Timestamp



A **timestamp** marks two temporal limits (from and to):

$$\begin{aligned} \text{Timestamped} \equiv \quad & (\exists \text{from}.\text{TimeInterval} \sqcap \geq 1\ \text{from} \sqcap \leq 1\ \text{from}) \\ & \sqcap (\exists \text{to}.\text{TimeInterval} \sqcap \geq 1\ \text{to} \sqcap \leq 1\ \text{to}) \end{aligned}$$

Timestamp



An individual **exists** in the current interval only if it came into existence **before or during** the interval and ceased to exist **during or after** it.

$\exists \text{from}.(\exists \text{beforeEq}.\text{CurrentInterval}) \sqcap \exists \text{to}.(\exists \text{afterEq}.\text{CurrentInterval})$

GeneralTRestriction

More generally, we can specify a general temporal restriction:

$$\begin{aligned}\text{GeneralTRestriction} \equiv & \exists \text{from}.(\exists \text{beforeEq}.(\prod_{1 \leq i \leq m} \text{TConstraint}_i)) \\ & \sqcap \exists \text{to}.(\exists \text{afterEq}.(\prod_{1 \leq i \leq m} \text{TConstraint}_i))\end{aligned}$$

- Every TConstraint_i is a subset of time intervals.
- Every GeneralTRestriction contains at least:
  - $\text{TConstraint}_1 \equiv \text{CurrentInterval}$
  - $\text{TConstraint}_2 \equiv \exists \text{afterEq}.\{\text{interval}_i\} \sqcap \exists \text{beforeEq}.\{\text{interval}_j\}$

  where $1 \leq i \leq j \leq n$
- Two possibilities:
  - $\text{CurrentInterval}^{\mathcal{I}} \subseteq \text{TConstraint}_2^{\mathcal{I}}$
  - $\text{TConstraint}_1^{\mathcal{I}} \cap \text{TConstraint}_2^{\mathcal{I}} = \emptyset$

Dynamic Concept

A DynamicConcept is the union of its variants:

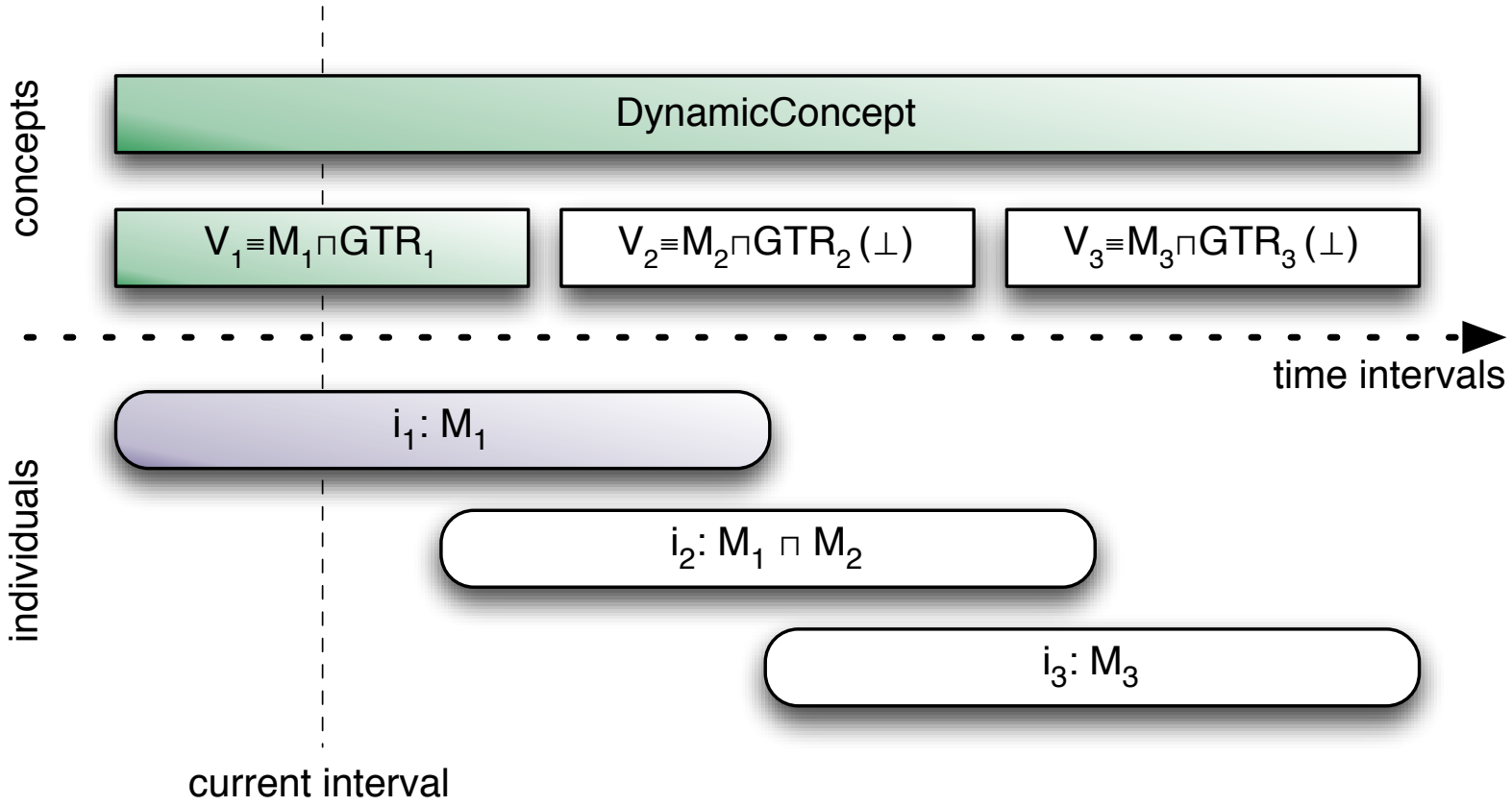
$$\text{DynamicConcept} \equiv \text{Variant}_1 \sqcup \ldots \sqcup \text{Variant}_m$$

Each variant is an intersection of its *meaning* and a GeneralTRestriction:

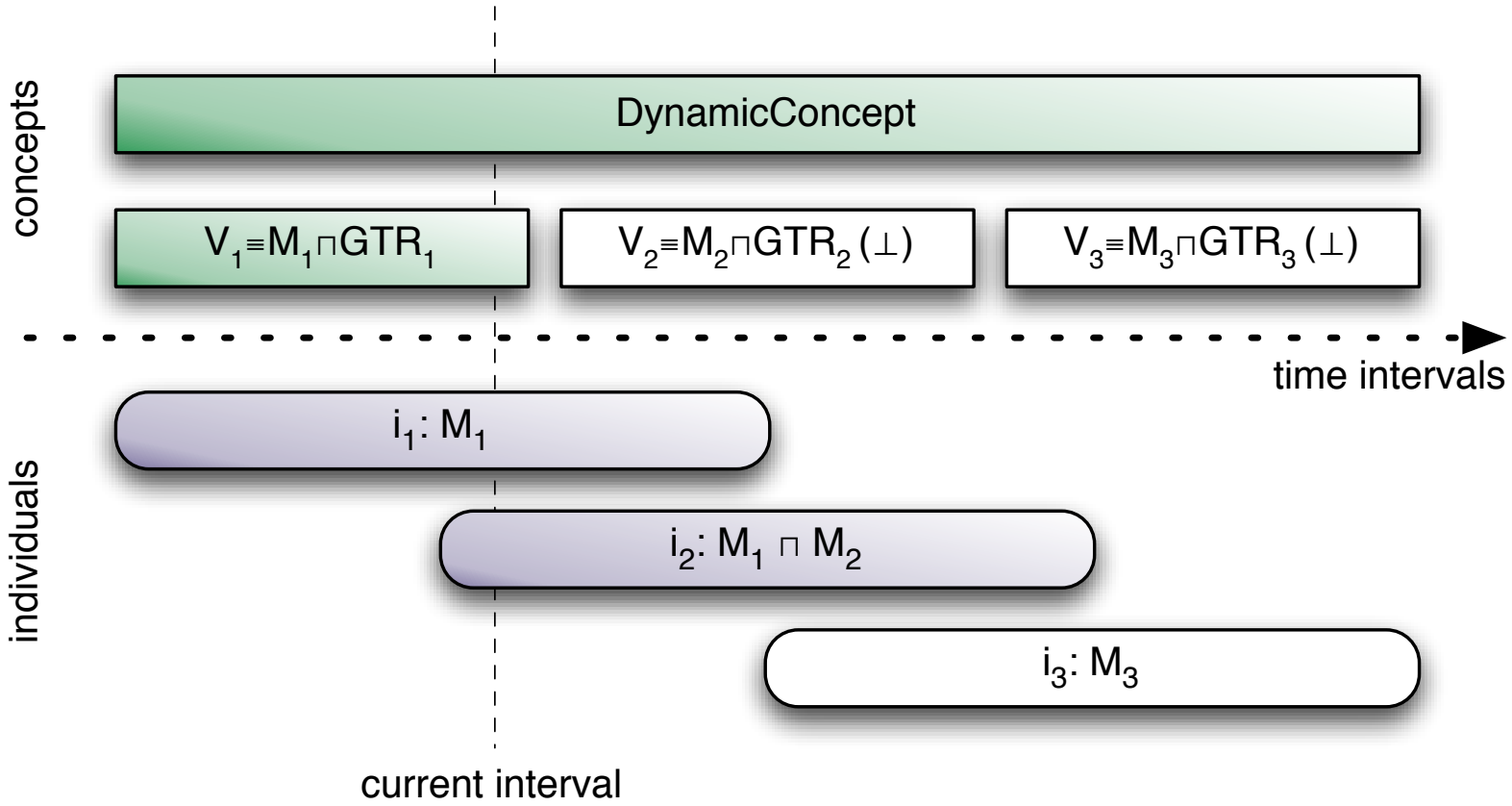
$$\text{Variant}_k \equiv \text{Meaning}_k \sqcap \text{GeneralTRestriction}_k$$

The variants exclusively, and exhaustively cover the time axis.
An individual can be **classified** as an instance of a variant in the current interval only if the **individual exists** and the **variant is valid** in that interval.

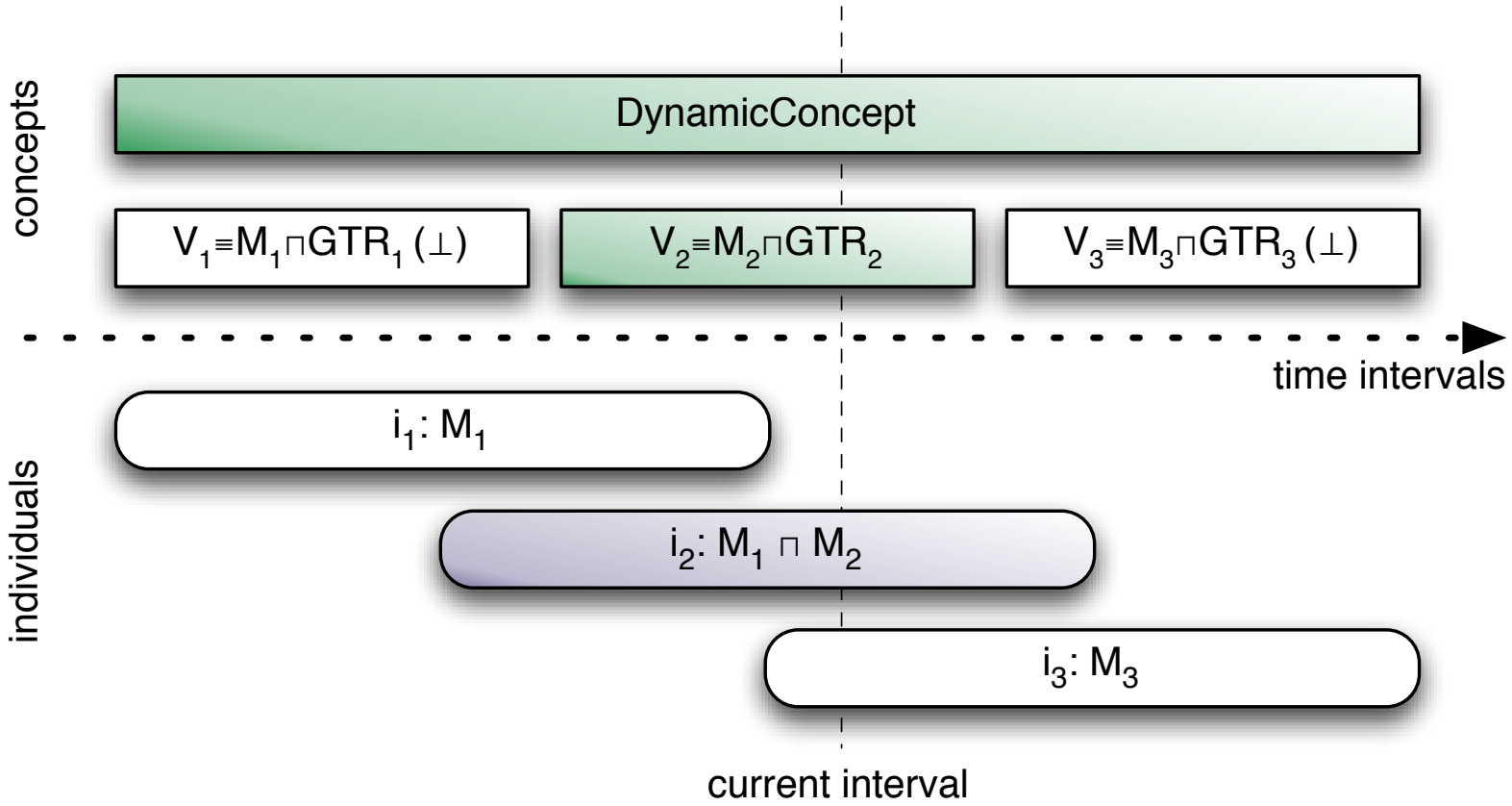
Walkthrough



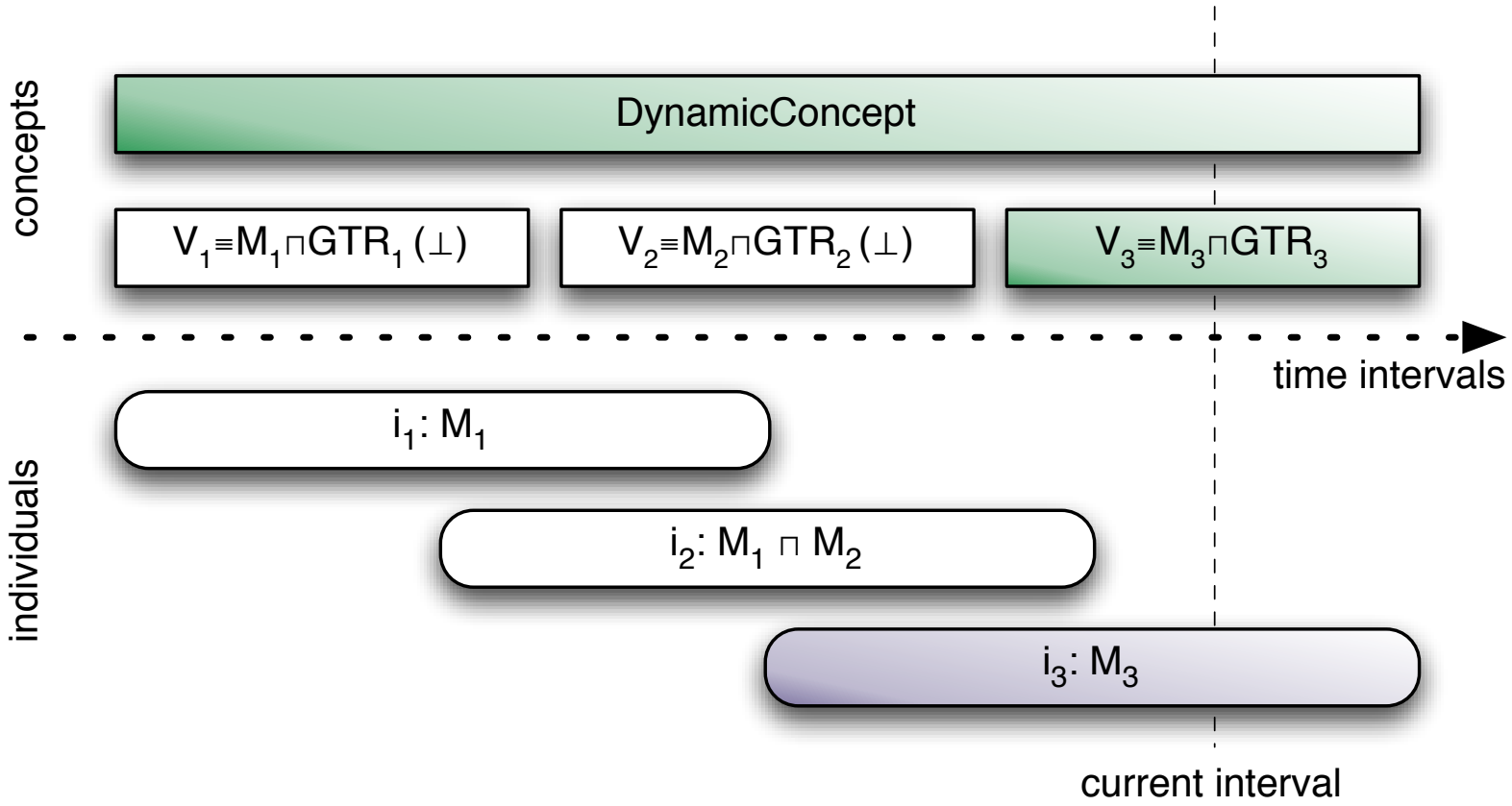
Walkthrough



Walkthrough



Walkthrough



Activity, Applicability and Validity

- A *legal* dynamic concepts, requires interaction between multiple intervals:

$$\mathsf{AppInterval} \equiv \exists \mathsf{afterEq}.\{\mathsf{interval}_i\} \sqcap \exists \mathsf{beforeEq}.\{\mathsf{interval}_j\}$$

$$\mathsf{ActInterval} \equiv \exists \mathsf{afterEq}.(\exists \mathsf{from}^{-}.\{\mathsf{norm}\}) \sqcap \exists \mathsf{beforeEq}.(\exists \mathsf{to}^{-}.\{\mathsf{norm}\})$$

  where norm is a time-stamped individual representing some legal text.

- A concept is valid for some the current interval only if it is both **applicable** and the norm that defines it is **active** within that interval.
  Every GeneralTRestriction contains at least the CurrentInterval and some AppInterval and ActInterval

Summary

Representation of **definitional changes**, implementable using $\mathcal{SHOIN}$:

- supported directly by standard reasoners
- Supports incremental changes (reuse, maintenance)
- Backward compatibility (changes are handled monotonically)
- Easy access to all represented versions
- No update snowball effect
- Allows non-versioned concepts
- Space-efficient (performance cost)

Future work

- Performance gains:
  - CurrentInterval as individual vs. nominal (2x faster)
- Introduction of last_interval, to 'close' time scale,
- More complex interplay between intervals (retroactivity)
- Extension to jurisdiction:
  - Spatial
  - Authority