Evolving Ontology Evolution

Dimitris Plexousakis
dp@ics.forth.gr

Institute of Computer Science, FORTH and

Department of Computer Science,

University of Crete, Greece

joint work with

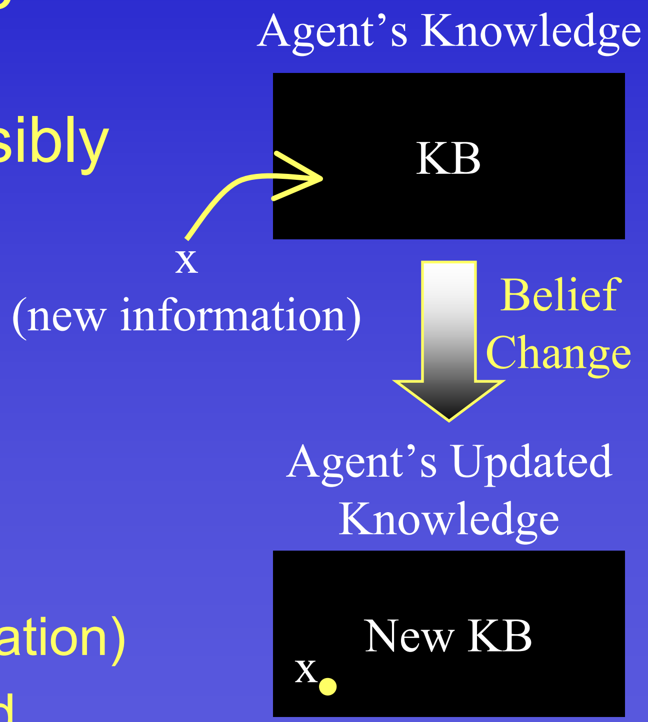
Giorgos Flouris and Grigoris Antoniou

Objective

- This work attempts to use results from the field of *belief change* in order to address problems related to *ontology evolution* on the Semantic Web
- Establishes a formal basis for studying properties of frameworks and languages for representing knowledge on the Semantic Web
  - especially in their ability to accommodate evolving knowledge

Belief Change
(a.k.a belief revision)

- The problem of belief change is the problem of updating an agent's knowledge in the face of new (possibly contradictory) information



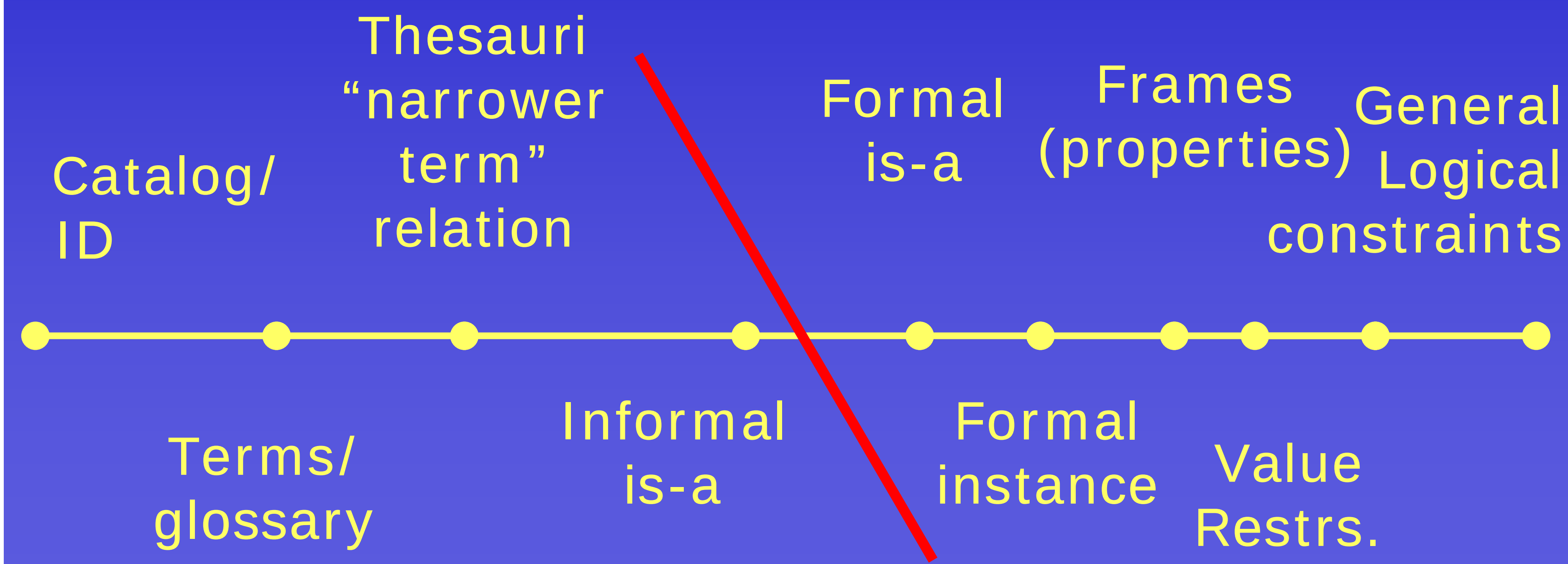
- Several reasons for that:
  - Mistakes during acquisition / input
  - New observations
  - New knowledge (e.g. classified information)
  - The world being modeled has changed
  - …

Semantic Web

- a web of meaning
- providing infrastructure for expressing information in a precise, humanly-readable, and machine-interpretable form
- enabling both syntactic and semantic interoperability among independently-developed Web applications, allowing them to efficiently perform sophisticated tasks for humans
- enabling Web resources (information & services) to be accessible by their meaning rather than by keywords and syntactic forms

Ontologies: what are they?

- Ontologies are shared, formal conceptualizations of particular domains

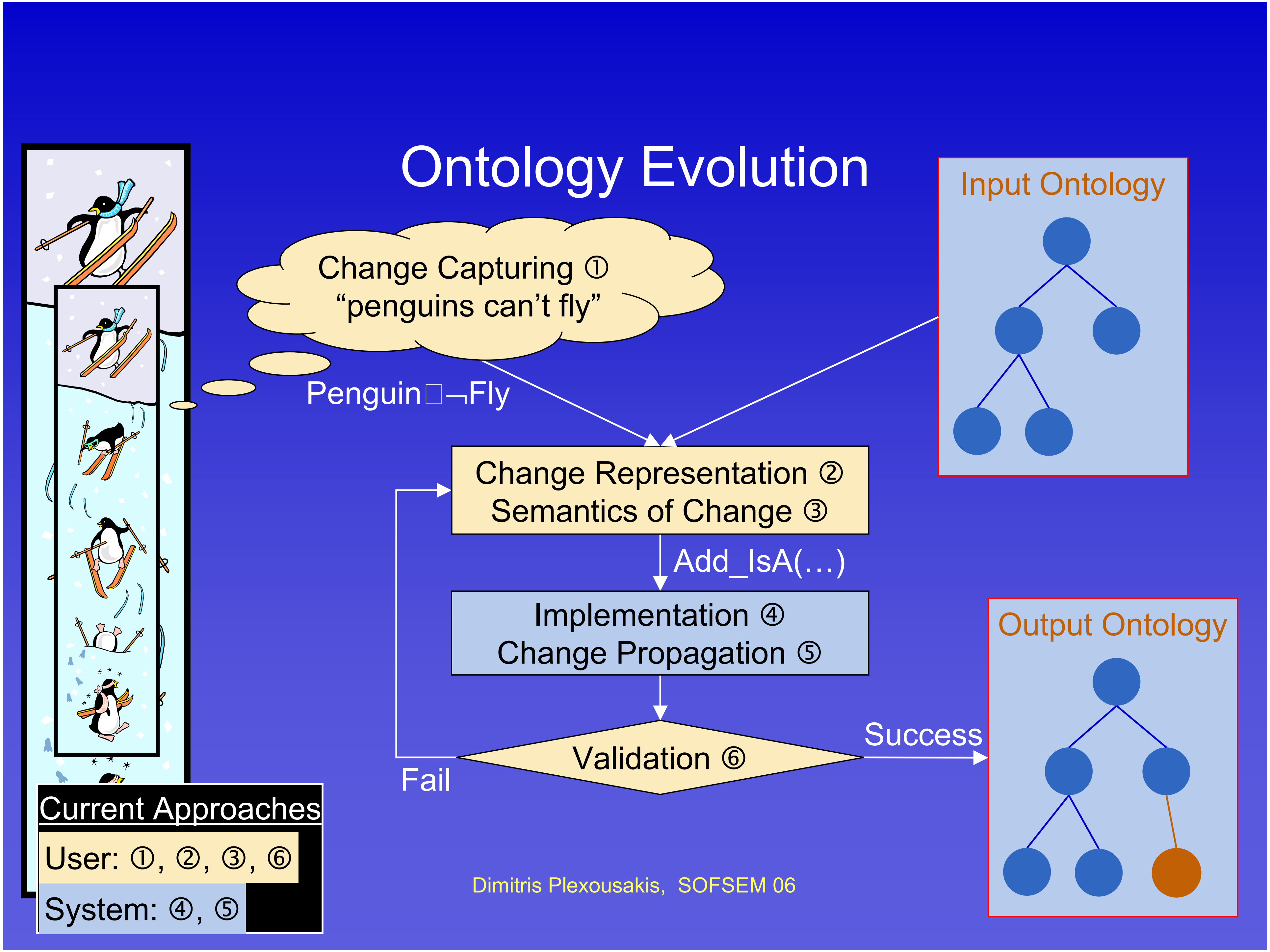


Ontologies: Design and Maintenance

- An ontology serves as a representation vocabulary of the concepts in the subject area, the relations among the terms and the way the terms can or cannot be related to each other (i.e., a reference model)
- Ontologies are useful for the SW (domain modeling, semantic integration, interoperability, etc.)
- Building an ontology is not enough; it must be maintained!
- One of the main problems related to ontology maintenance is ontology evolution

Ontology Evolution: Definition and Importance

- Ontology evolution is the process of modifying an ontology in response to a certain change in the domain or its conceptualization
- Main reasons for ontology evolution:
  - Dynamic domains
  - Change in users' needs or perspective
  - New information (previously unknown, classified or unavailable) that improves the conceptualization
  - Errors during original conceptualization
  - Ontology dependencies
  - …

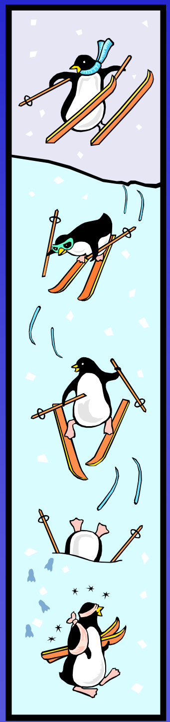


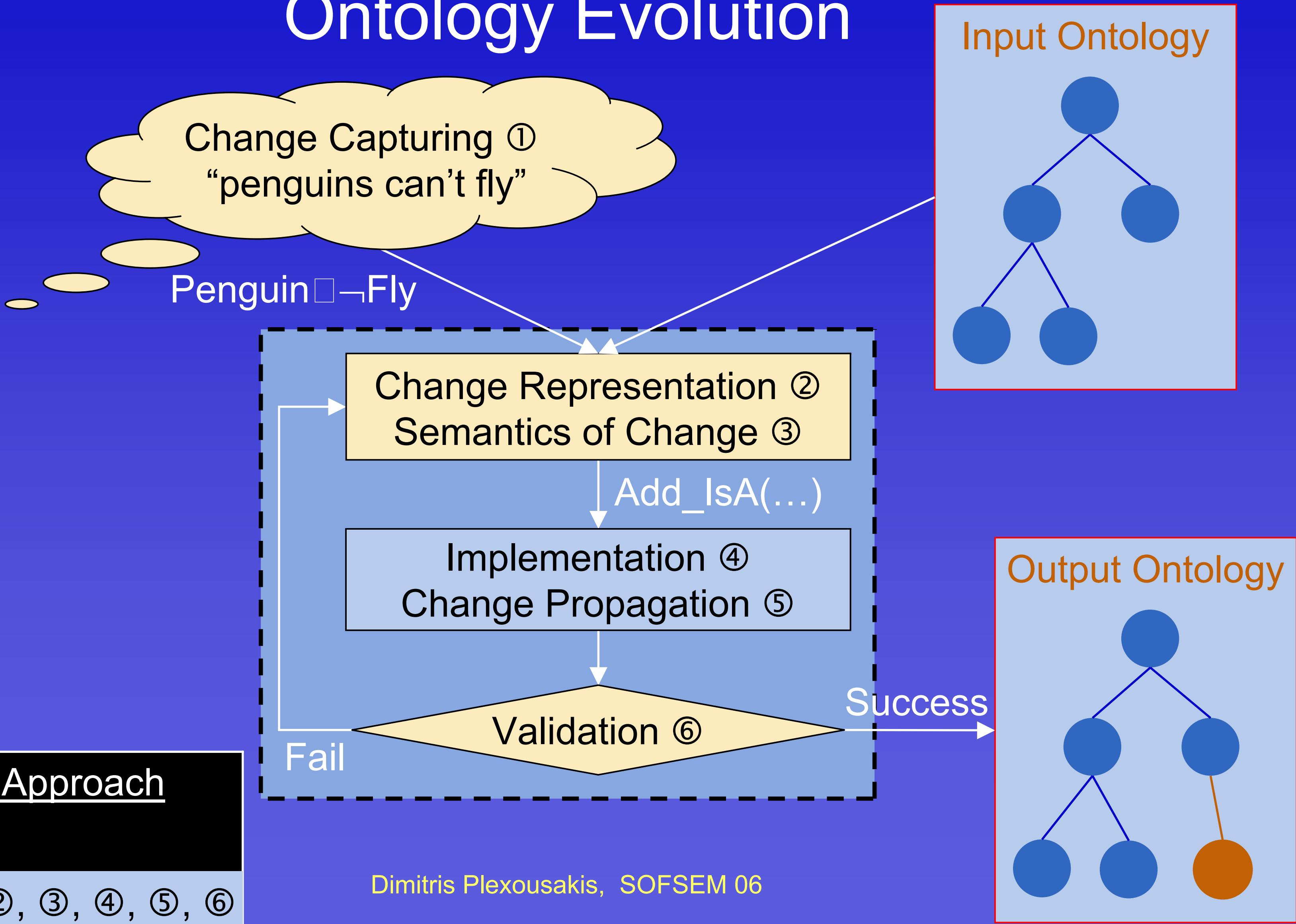
Ontology Evolution
Change Capturing ①
“penguins can’t fly”
Penguin□¬Fly
Input Ontology
Change Representation ②
Semantics of Change ③
Add_IsA(…)
Implementation ④
Change Propagation ⑤
Validation ⑥
Fail
Success
Output Ontology
Current Approaches
User: ①, ②, ③, ⑥
System: ④, ⑤
Dimitris Plexousakis, SOFSEM 06

Limitations

- Main limitations of current approaches:
  - Manual or semi-automatic approaches
  - Too many operators (complex and atomic)
  - No formal semantics
- Cause problems:
  - Automated agents and systems
  - Scalability
  - Formal properties unknown
  - Bottleneck for current research

Ontology Evolution





Proposed Approach

User: ①

System: ②, ③, ④, ⑤, ⑥

Belief Change and Ontology Evolution

- Ontologies and KBs contain a vast amount of knowledge, which must always be up-to-date
  - Keeping KBs up-to-date: belief change
  - Keeping ontologies up-to-date: ontology evolution

- Belief change:
  - Mature
  - Formal
  - Rich literature
  - Automatic

- Ontology evolution:
  - Not yet mature
  - Informal
  - New field
  - User-driven (manual or semi-automatic)

Motivating Idea

- It makes sense to migrate belief change techniques, intuitions, ideas, theories to ontology evolution:
  - Take advantage of 20+ years of research on belief change
  - View belief change techniques, ideas, intuitions, results, algorithms and methods under the prism of ontology evolution
  - Address ontology evolution using belief change

Belief Change Issues

- Belief Change addresses important issues that have not been considered in ontology evolution:
  - Foundational vs Coherence Theories
  - Postulations vs Explicit Constructions
  - Principle of Primacy of New Information
  - Principle of Irrelevance of Syntax
  - Principle of Consistency Maintenance
  - Principle of Minimal Change
  - Different operations: revision, update, contraction, erasure
  - …

Difficulties and Methodology

- Main problem: different representation languages
- Belief change techniques are generally targeted at classical logic:
  - Their assumptions fail for most languages used for ontology definition
  - Cannot be directly used for such logics
  - But: the underlying intuitions are applicable
- Belief change techniques need to be migrated to the ontology evolution context

Approach: Phase 1

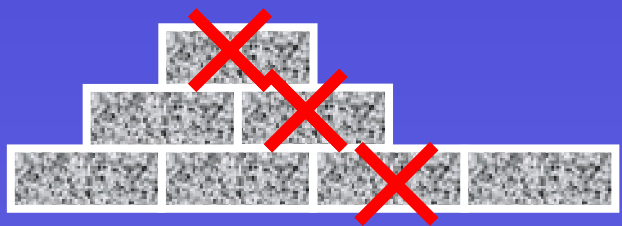
- Phase 1:
  - Set the foundations for future work on the subject
  - Very abstract, long-term and ambitious goal

Foundation and Coherence Belief Bases and Belief Sets (1)

- Foundational Viewpoint (pyramid):
  - KB consists of the explicitly represented knowledge
  - Only explicit knowledge can be changed
  - Implicit knowledge (implications) is affected indirectly, through the changes in the explicit knowledge (so that the resulting “pyramid” is “stable”)
  - Explicit knowledge forms the belief base

Supported Knowledge

Basic Knowledge

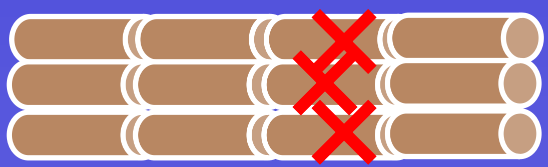


Basic: $\{a, b, c\}$

Supported: $\{a \wedge b, b \wedge c, \ldots\}$

Foundation and Coherence Belief Bases and Belief Sets (2)

- Coherence Viewpoint (raft):
  - No discrimination between implicit and explicit knowledge
  - Both explicit and implicit knowledge can be changed
  - Changes should be made coherently, so that the resulting KB makes sense (i.e., the “raft” is “stable”)
  - Explicit and implicit knowledge together form the belief set



Knowledge
$\{a, b, c, a \wedge b, b \wedge c, \ldots\}$

Foundation and Coherence Belief Bases and Belief Sets (3)

- Ontology evolution uses the foundational viewpoint
  - Implicit choice
  - No reasons given for this choice
- Belief change uses both
  - Most influential approaches use coherence model
- Foundational model seems more adequate for ontology evolution, but this is not a priori certain

Postulation and Explicit Construction for change determination

- Postulation:
  - What are the properties that a proper change (belief change or ontology evolution) algorithm should satisfy?
- Explicit Construction:
  - How can we construct a proper change (belief change or ontology evolution) algorithm?
- Ontology Evolution: only the latter
- Belief Change: both, in tandem

Primacy of New Information

- Should we accept new knowledge unconditionally?
- Usually yes:
  - New knowledge usually represents a newer and more accurate view of the world
- But there are cases where this is inappropriate:
  - Agent communication context
  - Unreliable and untrustworthy sources
- These cases appear often in the Semantic Web context so:
  - Non-prioritized belief revision: new information may be (partially or totally) rejected

Irrelevance of Syntax

- Is the result of the change affected by the syntactic formulation of the operands, or is it affected by its semantical properties only?
- Normally, semantic considerations should determine the result
- Fails in the foundational model: different justifications may result to equivalent KB closures
- In current ontology evolution, syntax is important (foundational approach; “irrelevance of syntax” ignored)

Consistency Maintenance (1)

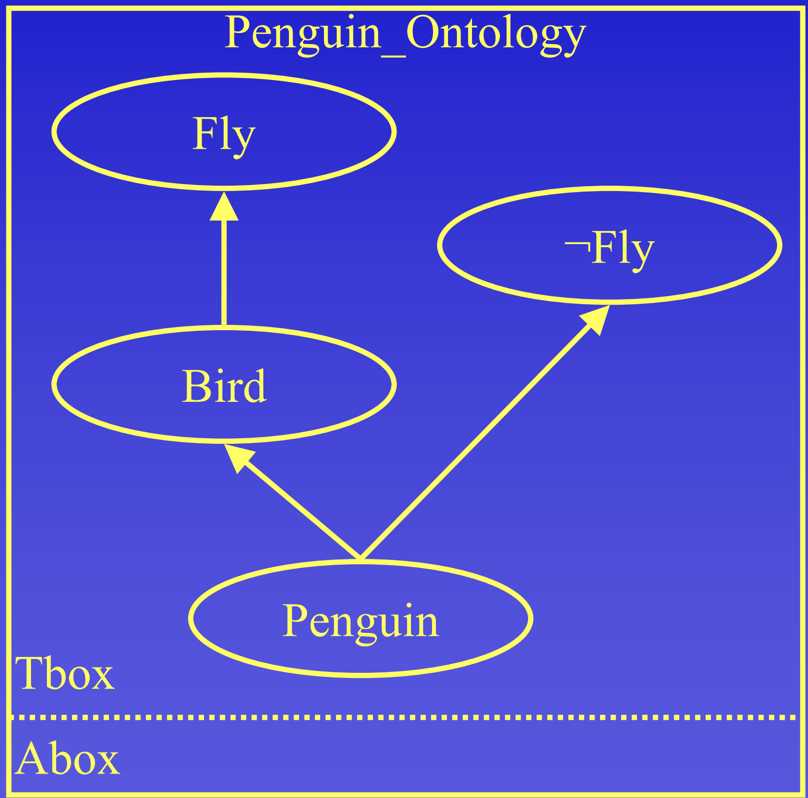
- The result of the change should be consistent
  - Obvious, but what does “consistent” mean?
- In ontology evolution, several definitions are used:
  - No models (i.e., explosive inference, anything is implied by the ontology)
  - Unsatisfiable concepts
  - Satisfies the restrictions of the “consistency model”
  - All entities are defined
  - Logical, structural, user-defined consistency
  - Semantical and syntactical consistency
  - …

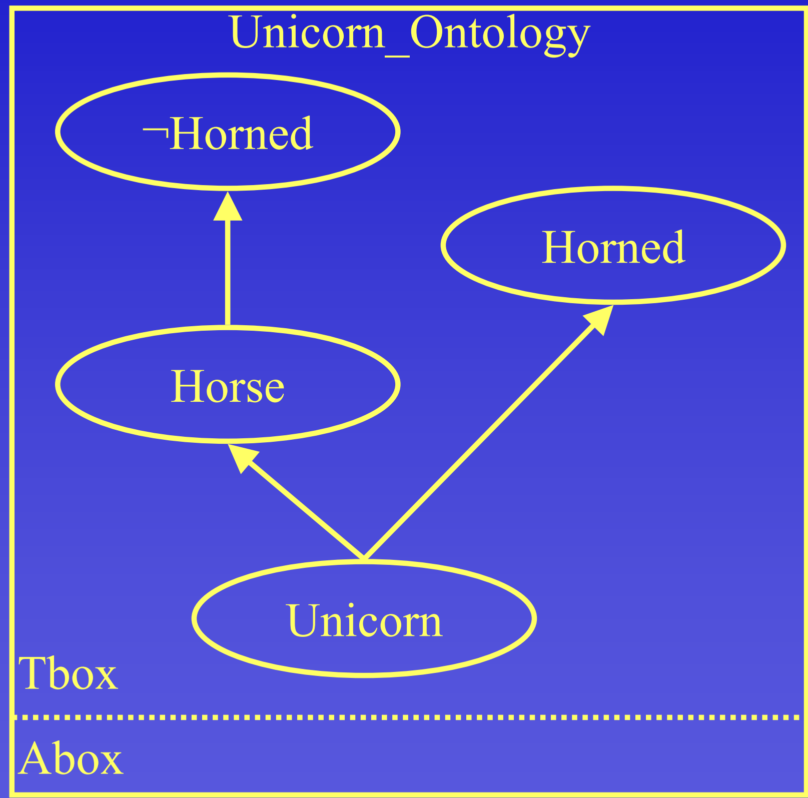
Consistency Maintenance (2)

- These different definitions are incompatible!
- Our approach:
  - Consistent ontology: it has at least one model
  - Coherent ontology: satisfies a set of conditions, constraints or invariants related to efficient ontology design (covers every other case of “inconsistency”)

- Only consistency maintenance should concern us when it comes to ontology evolution

Consistency Maintenance (3)





Principle of Minimal Change

- The most important principle:
  - The resulting knowledge (KB/ontology) should be as “close” as possible to the original knowledge (KB/ontology)
  - Change (information loss) is “minimal”
- Open problems:
  - How is “closeness” defined?
  - How is “information loss” counted?
- Several different (automatic and formal) approaches are used in belief change
- Human expertise is used in ontology evolution

Ontology Evolution Operations (1)

- Too many operations in ontology evolution literature:
  - Add_IsA, Remove_Concept, Move_SubTree etc
  - Operations are too "procedural": they indicate directly what change(s) should be made in the ontology
- Advantages:
  - Simple approach and implementation of each operation
- Drawbacks:
  - Changes must be known by the user a priori
  - Operations are too many to be implemented; some should be emulated (manually, by the user)
  - No general consensus on the interesting operations

Ontology Evolution Operations (2)

- Our approach is inspired from the belief change literature

Operation	State of the world (Static/Dynamic)	Type of Change (Addition/Retraction)
Revision	Static (conceptualization)	Addition
Contraction	Static (conceptualization)	Retraction
Update	Dynamic (world)	Addition
Erasure	Dynamic (world)	Retraction

Ontology Evolution Operators (3)

Example: a box of chess pieces contains a King

Revision: King is Black (observation)

Update: King is painted Black (action)

Contraction: the previous observation on King's black color is unreliable (unreliable observation)

Erasure: if King is black, paint it an arbitrary color (action with unknown effect)

Ontology Evolution Operations (4)

- The two approaches are not directly comparable
- They are based on a different view of the change:
  - “Fact-centered”: the change is a new fact that should be accommodated in the ontology (belief change / our approach)
  - “Modification-centered”: the fact itself is irrelevant; a change is a certain modification operation upon the ontology itself (ontology evolution)

Ontology Evolution Operations (5): the “fact-centered” view

- Advantages:
  - Changes need not be known a priori
  - Only four operations to consider
  - No user participation necessary (facts may be captured by sensors or other input devices)
  - Extra layer of abstraction
- Drawbacks:
  - Requires an extra step to determine the modifications
  - This extra step is very difficult (belief change deals with that)

Recasting the Problem

- All the issues (up to now) have not used any of the properties of the underlying knowledge representation formalism
- Ultimately, we will need a common formalism to be based upon:
  - Tarski's model

Results so far

- This analysis shows that there are several issues that have been studied in belief change but have not been considered in ontology evolution

- The ontology evolution field could benefit in many ways by using techniques, ideas, results etc from the belief change paradigm
  - Example: the definition of operations

A More Specific Approach: the AGM Theory

- Phase 1 dealt with ontology evolution very abstractly, not precisely specifying any direct solutions to the problem
- From this point on we restrict ourselves to deal with:
  - The most influential belief change theory (AGM theory)
  - The most fundamental operation (contraction)
  - The most popular languages for ontological representation (DLs and OWL)
- Phase 2:
  - Study the applicability of the AGM theory of contraction in DLs and OWL

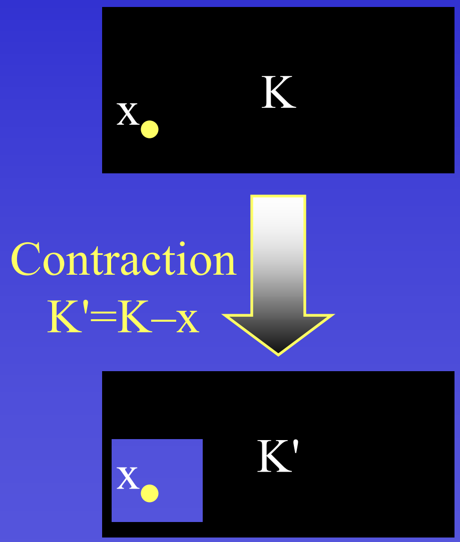
Rationale

- AGM theory (Alchourron, Gärdenfors, Makinson):
  - Mature, general and widely accepted method for belief change (most influential approach)
  - Its theoretical properties are well-understood
  - Captures the notion of “rationality”
  - “Rationality” is independent of the underlying logic

- Contraction is the most important operation for theoretical purposes (for practical purposes: revision)

Contraction

- We deal with contraction: the process of consistently removing some information from a KB
- Useful operation:
  - Malfunctioning instrument: all information acquired by this instrument should be removed from the KB when we discover the malfunction, because it is not reliable any more
- For KB K and information x: K'=K–x
- The new KB should not imply x



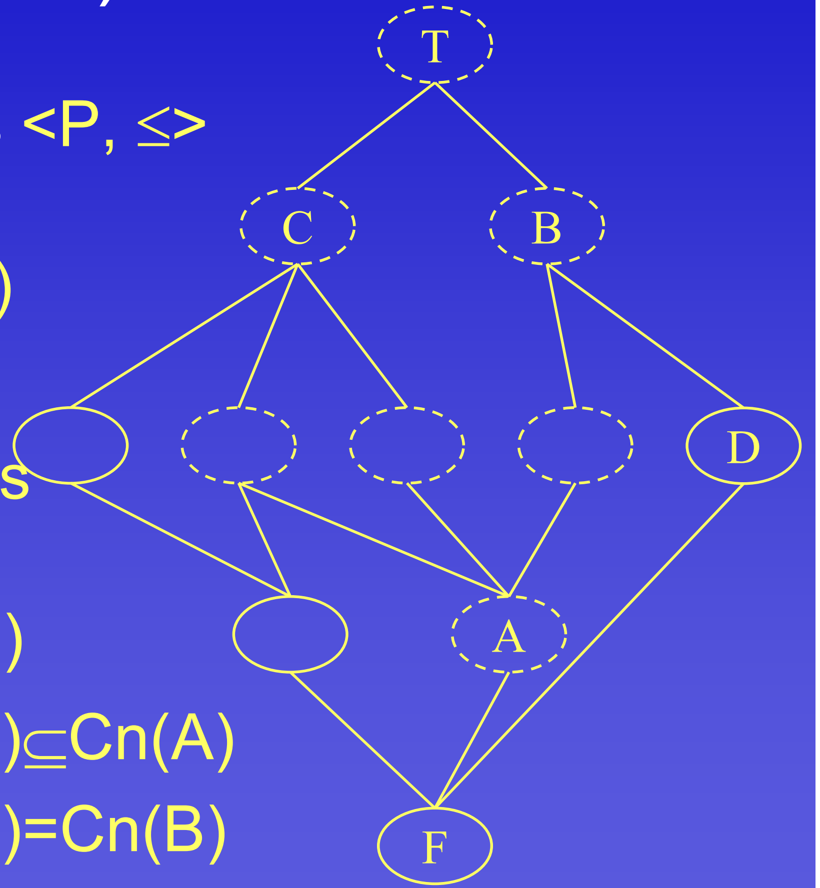
AGM Theory

- Main contribution: 6 AGM postulates that determine whether a contraction operator behaves “rationally”
- AGM theory is based on certain assumptions on the underlying logic, so, as usual:
  - Intuitions applicable in ontologies
  - Postulates and results not applicable in ontologies

Logics:Tarski's Model

- We use Tarski's model <L,Cn>
- <L,Cn>
  - L is a set (any set)
  - Cn is a consequence operator
    Cn(A) contains all the propositions implied by X
- Close interrelationship with inference relation:
  - $Cn(A)=\{x \in L: A \models x\}$
  - $A \models \{x\}$ iff $x \in Cn(A)$

Lattices and Logics (Visualization)



- Visualization: complete lattices $<P, \leq>$ can represent logics $<L, Cn>$
- $T \equiv Cn(\emptyset)$ $\quad\quad$ $Cn(T)=Cn(\emptyset)$
- $F \equiv L$ $\quad\quad$ $Cn(F)=L$
- $Cn(A)=\{X \mid A \leq X\}$: dashed nodes
- $D<B$ $\quad\quad$ $D \models B$, $B \not\models D$
  $Cn(B) \subset Cn(D)$
- $inf\{B,C\}=A$ $\quad\quad$ $Cn(B) \cup Cn(C) \subseteq Cn(A)$
- $sup\{A,D\}=B$ $\quad\quad$ $Cn(A) \cap Cn(D)=Cn(B)$

Assumptions on the Logic

AGM

- <L, Cn>: L is a set, Cn is a consequence operator, $Cn(A)=\{x \in L: A \models x\}$
- L: closed under usual operators ($\neg$, $\wedge$, $\vee$, $\rightarrow$ etc)
- Cn: satisfies Tarskian axioms (iteration, inclusion, monotony)
- Cn: includes classical tautological implication
- Cn: compact
- Cn: satisfies rule of introduction of disjunction in the premises

Generic

- <L, Cn>: L is a set, Cn is a consequence operator, $Cn(A)=\{x \in L: A \models x\}$
- L: no operators
- Cn: satisfies Tarskian axioms (iteration, inclusion, monotony)
- No further assumptions on Cn

Assumptions on the Contraction Operator

AGM

- Contraction operator '–' $K'=K-x$, where:
  - K is a theory ($K=Cn(K)$)
  - x is a proposition ($x \in L$)
- Satisfies the original AGM postulates:
  - Closure: $K-x=Cn(K-x)$
  - Inclusion: $K-x \subseteq K$
  - Vacuity: If $x \notin K$ then $K-x=K$
  - Success: If $x \notin Cn(\varnothing)$ then $x \notin K-x$
  - Preservation: If $Cn(\{x\})=Cn(\{y\})$ then $K-x=K-y$
  - Recovery: $K \subseteq Cn((K-x) \cup \{x\})$

Generic

- Contraction operator '–' $K'=K-A$, where:
  - K is a theory ($K=Cn(K)$)
  - A is a set ($A \subseteq L$)
- Satisfies the generalized AGM postulates:
  - $K-A=Cn(K-A)$
  - $K-A \subseteq K$
  - If $A \not\subseteq K$ then $K-A=K$
  - If $A \not\subseteq Cn(\varnothing)$ then $A \not\subseteq K-A$
  - If $Cn(A)=Cn(B)$ then $K-A=K-B$
  - $K \subseteq Cn((K-A) \cup A)$

Generalization: what do we gain?

- AGM assumptions are fairly general; include many interesting logics, such as Propositional Calculus (PC) and First-order Logic (FOL)
- Fail to accommodate equational logic, Description Logics (DLs), logics that describe semantic networks (e.g. those used in the Semantic Web)
- All the above logics are included in our model

Generalization: what do we lose?

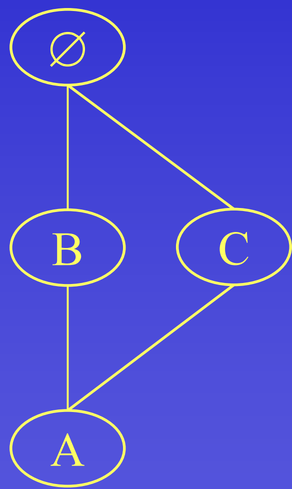
- AGM results no longer hold:
  - In any logic, we can define a whole family of contraction operators that satisfy the AGM postulates
- Noticed that only some of the logics in our class admit an operator satisfying the generalized postulates (i.e., a “rational” operator):
  - Termed AGM-compliant logics

Notions related to AGM-compliance

- Notion: Decomposability
  - A property that a set of expressions should satisfy
- Notion: Cuts
  - A special structure related to a set of expressions
  - Several cuts per set of expressions
  - Violating cuts: a special, “bad” type of cuts
- Notion: Max-cuts
  - A special type of cuts
  - Is unique per set of expressions, does not always exist
  - Violating max-cuts: a special, “bad” type of max-cuts

Decomposability Definition

- A set A is decomposable iff for all B such that $Cn(\emptyset) \subset Cn(B) \subset Cn(A)$, there is a C such that:
  - $Cn(C) \subset Cn(A)$
  - $Cn(A)=Cn(B \cup C)$



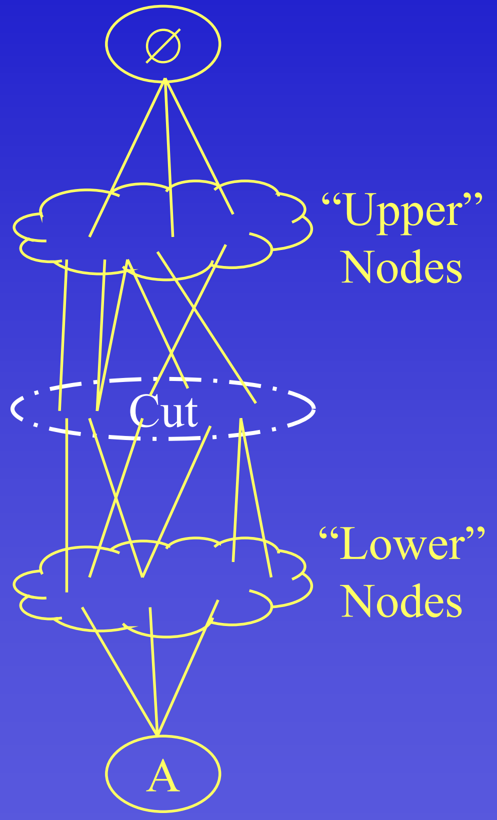
- Theorem: a logic is AGM-compliant iff all its sets are decomposable

- A–B=C

Cuts
Intuition

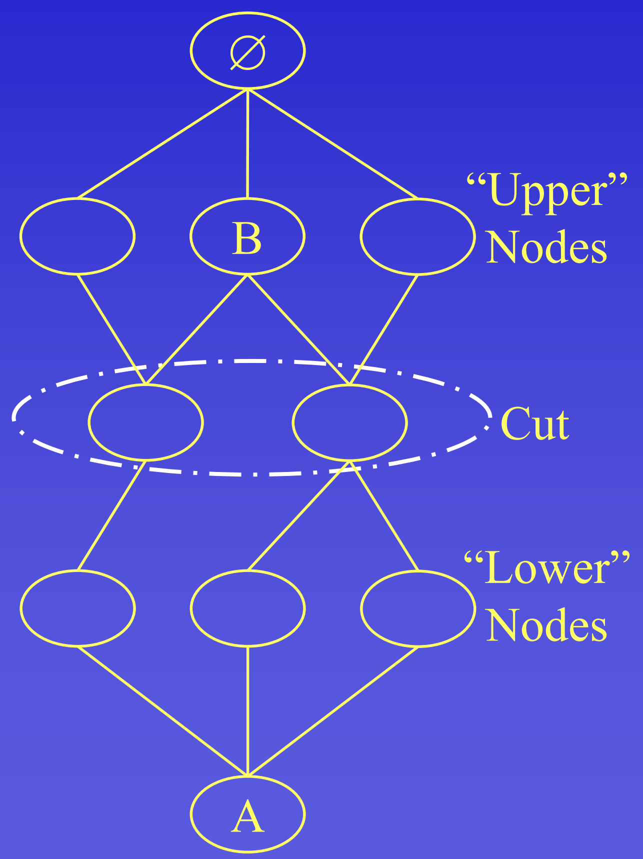
- A cut of a set A is a family of beliefs that "divides" the beliefs implied by A in two categories:
  - "Upper" nodes
  - "Lower" nodes

- Every belief implied by A either implies or is implied by a set in the cut



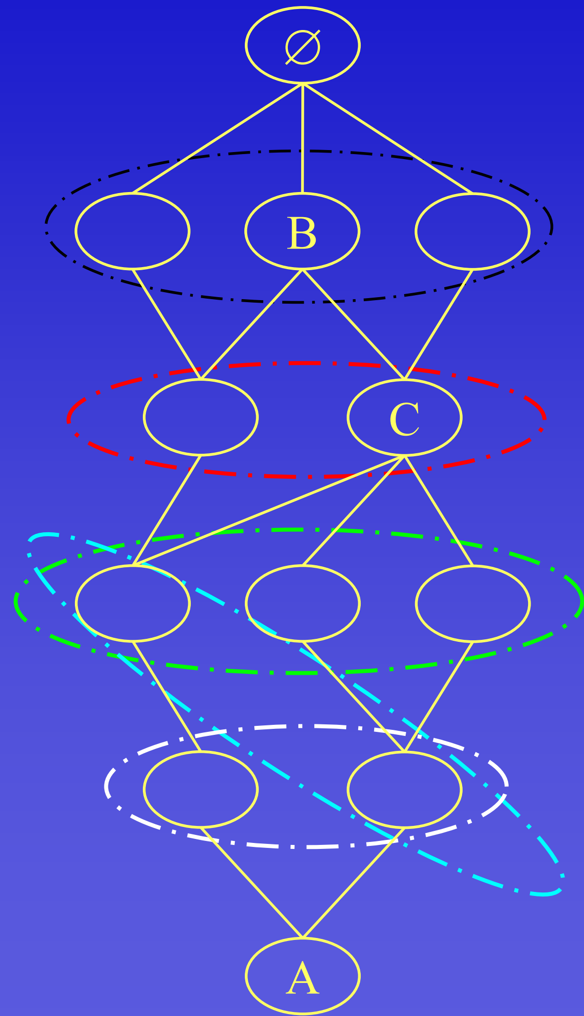
Cuts
Connection with Decomposability

- If B is implied by all the sets in a given cut then set C=A–B
- If C is an "upper" node, then recovery is not satisfied
- If C is a member of the cut or a "lower" node, then success is not satisfied
- So if Cn(B)≠Cn(∅), then A is not decomposable
- Theorem: A set A is decomposable iff there is no "violating cut" of A



Max-cuts
Intuition

- Cuts with “bigger” sets are more likely to be violating cuts
- Motivates us to look for the “biggest” cut
- A max-cut captures this notion
- A max-cut is unique, but it does not always exist
- If the max-cut is not a violating cut, then there is no violating cut
- Theorem: A set A is decomposable iff its max-cut is not violating



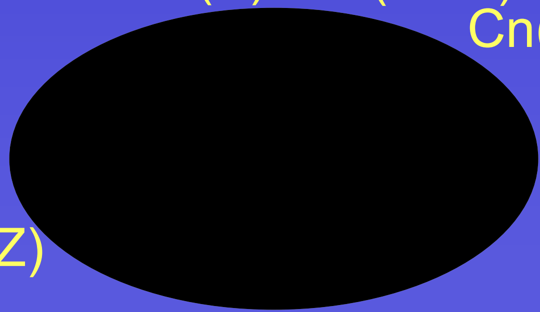
Results on AGM-compliance

- The following are equivalent:
  - A logic is AGM-compliant
  - All sets of a logic are decomposable
  - All cuts of all sets are non-violating
  - The max-cuts of all sets are non-violating

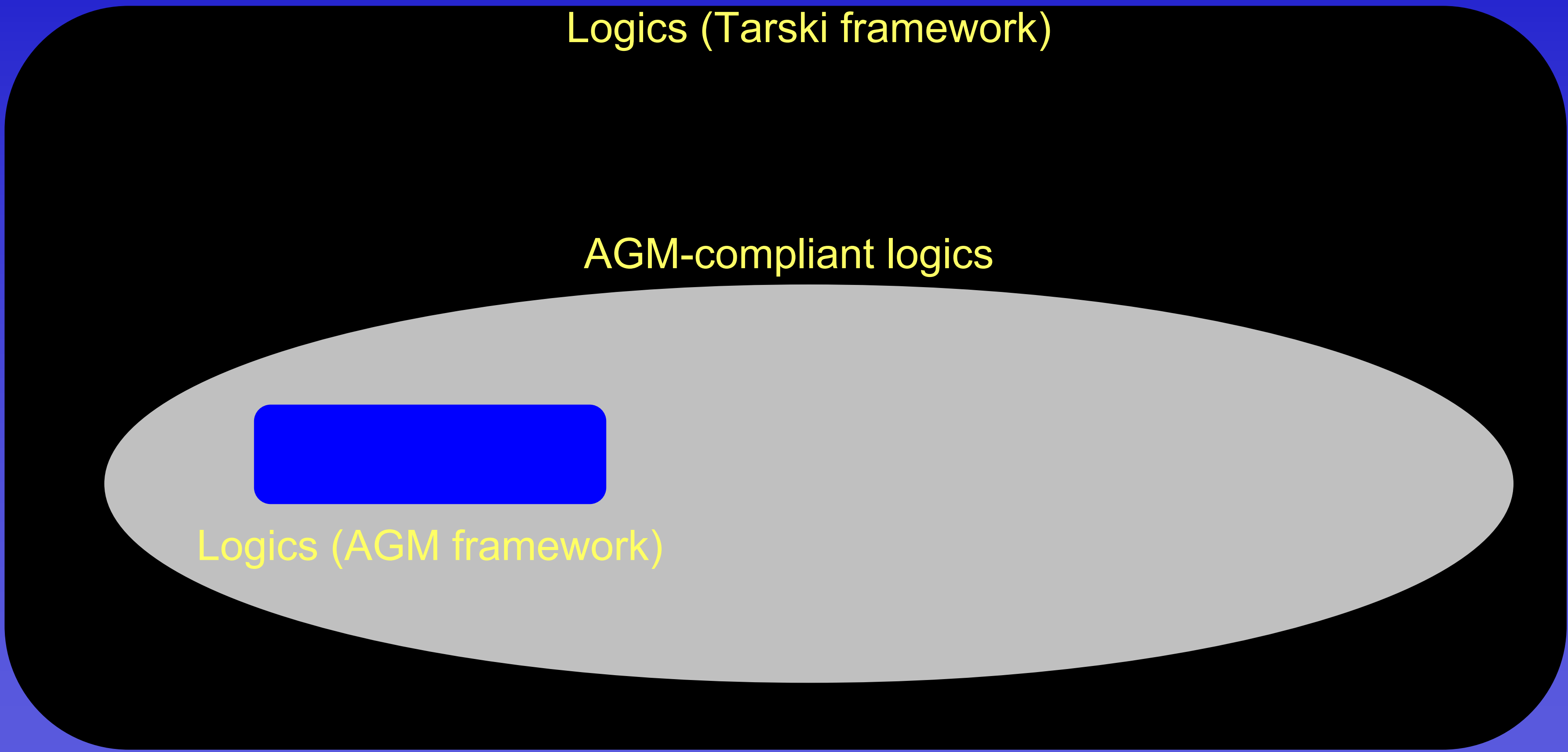
- Three equivalent characterizations of AGM-compliant logics

Decomposability

- A logic <L,Cn> is decomposable (equivalently: AGM-compliant) iff for every $X,Y \subseteq L$ such that $Cn(\emptyset) \subset Cn(Y) \subset Cn(X)$ there is a $Z \subseteq L$ such that:
  - $Cn(Z) \subset Cn(X)$
  - $Cn(Y \cup Z) = Cn(X)$
- Example (from Propositional Calculus):
  - $X=\{a \wedge b\}$
  - $Y=\{a\}$
  - $Z=\{a \rightarrow b\}$ or $Z=\{b\}$



The Situation



Equivalence Relation Lattice Theory

- Defined equivalence relation between logics
- Every logic can be mapped to a complete lattice
- Every complete lattice can be mapped to a logic
- Logics and lattices are isomorphic, modulo equivalence of lattices and logics

- Equivalent logics have the same status as far as AGM-compliance is concerned
- Thus, AGM-compliance can be determined by the lattice's structure

Belief Base Operations
Motivation

- AGM model deals with theories (sets closed under Cn), i.e., coherence model
- Problems with this approach:
  - Theories are (usually) infinite sets
  - Theories do not discriminate between explicit and implicit information
- Explicit information: facts, rules, observations etc
- Implicit information: deduced from explicit
- Solution: belief base operations (i.e., foundational model)

Belief Base Operations
Initial Observations

- A KB is a set (belief base) containing only the explicit facts
- Consequences of this viewpoint:
  - A KB is not necessarily a theory
  - The result of a contraction is not necessarily a theory
  - Contraction removes facts from the base only and not from the implied facts

Belief Base Operations and AGM Postulates

- There is no base contraction operator that satisfies the AGM postulates in the logics of the AGM framework
  - The base AGM postulates were rejected as a rationality test for belief base operations (no operation would pass the test)
- This result is no longer true in our more general class:
  - There may be logics that admit base-AGM-compliant operators
- Problem to solve: what properties must a logic satisfy in order to admit a base-AGM-compliant operator?

Notions
Base-AGM-compliance

- Close connection between the two types of compliance:

  Base-AGM-compliance = AGM-compliance + Subset constraint

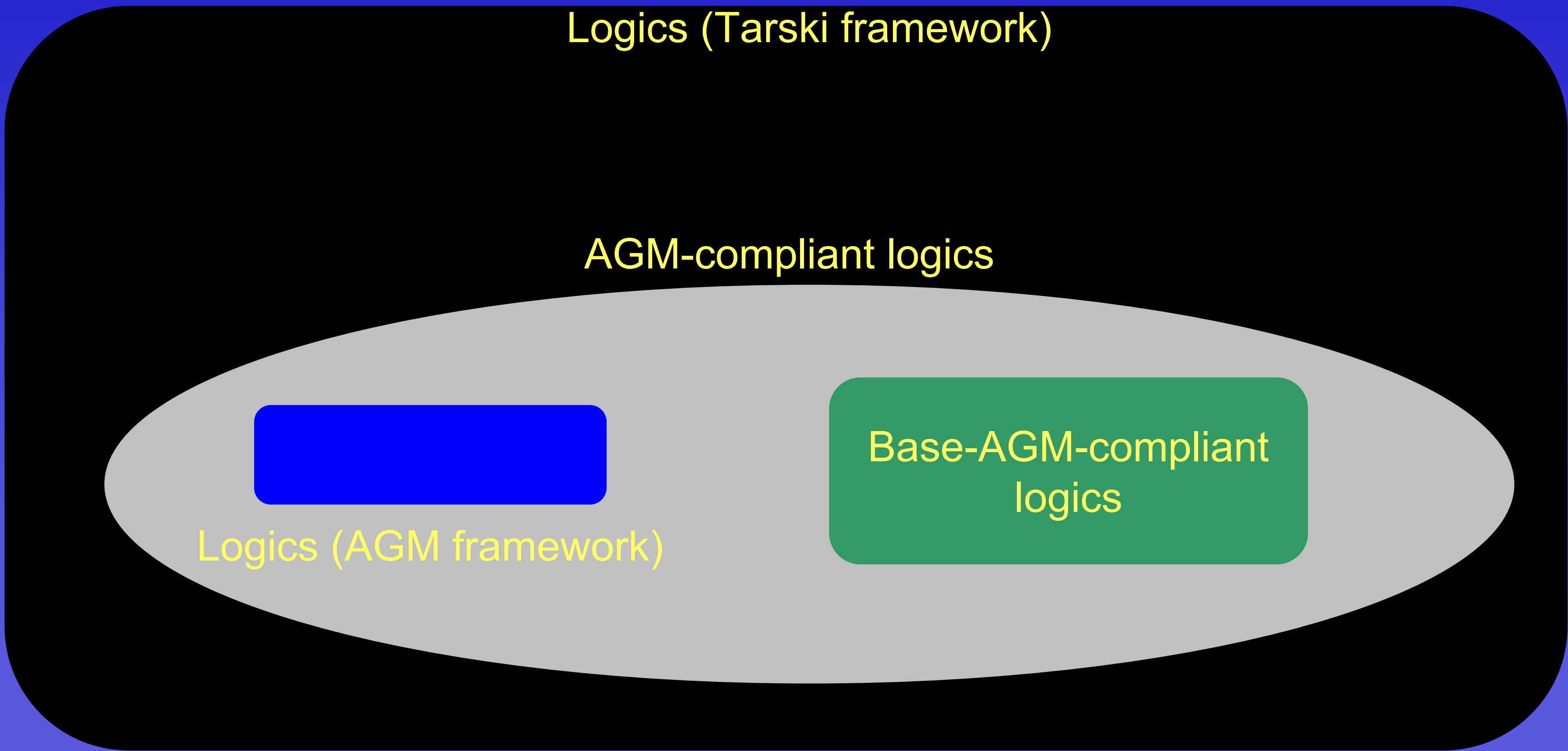
- Notion: Base Decomposability
  - A property that a set of expressions should satisfy
- Notion: Base Cuts
  - A special structure related to a set of expressions
  - Several base cuts per set of expressions
  - Violating base cuts: a special, “bad” type of base cuts

Results (2)
Base-AGM-compliance

- The following are equivalent:
  - A logic is base-AGM-compliant
  - All sets of a logic are base decomposable
  - All base cuts of all sets are not violating

- Two equivalent characterizations of base-AGM-compliant logics
- Base-AGM-compliance = AGM-compliance + Subset
  - The only difference between the standard case and the base case is the subset constraint, which is reflected in the definition of base decomposability and base cuts

The Situation



Description Logics (DLs)
Web Ontology Language (OWL)

- Knowledge representation formalisms that are constantly gaining popularity in the Semantic Web
- DLs: a family of languages
- OWL
  - Syntax: RDF
  - Semantics: DLs
- Research question:
  - Are these logics AGM-compliant or not?

Preliminaries
Description Logics (1)

- Primitive blocks: Classes, Roles, Individuals
- Used with operators to form terms
- Terms used with connectives to form axioms
  - Man $\equiv$ Male $\sqcap$ Human
  - Cat $\sqsubseteq \neg$ Dog
  - Cat(Garfield)
  - Human $\sqsupseteq \exists$has_offspring.Human
  - Man $\sqsupseteq \forall$has_wife.{Mary}
  - has_wife $\sqsubseteq$ has_spouse

Preliminaries
Description Logics (2)

- Operators: $\neg$, $\sqcap$, $\exists$, $\forall$, …
- Connectives: $\sqsubseteq$, $\equiv$, …
- A great variety of DLs:
  - Different properties
  - Different expressive power
  - Different reasoning complexity
- Model-theoretic reasoning based on interpretations
- Knowledge is stored in DL KBs: a set of DL axioms
  - An example of a DL KB: {Cat $\sqsubseteq \neg$ Dog, Cat(Garfield)}

Preliminaries
OWL

- OWL comes in three “flavours”
- OWL Full
  - Full expressive power, but undecidable
  - Complete integration with RDF
- OWL DL
  - Equivalent to the DL $SHOIN^{+}(D)$
  - Average expressive power and reasoning complexity, but decidable
- OWL Lite
  - Equivalent to the DL $SHIF^{+}(D)$
  - Least expressive and most efficient of the three flavours

CVA and OVA (1)

- It is often considered that only elements that appear in an ontology are "relevant" to the ontology
  - "Penguin" irrelevant to a University ontology
  - No reasoning is possible for axioms using the concept "Penguin"
- Thus:
  - Associate_Professor $\sqsubseteq$ Professor
    does not imply:
    Associate_Professor $\sqsubseteq$ Professor $\sqcup$ Penguin
- This is not the case for standard DL/OWL reasoning

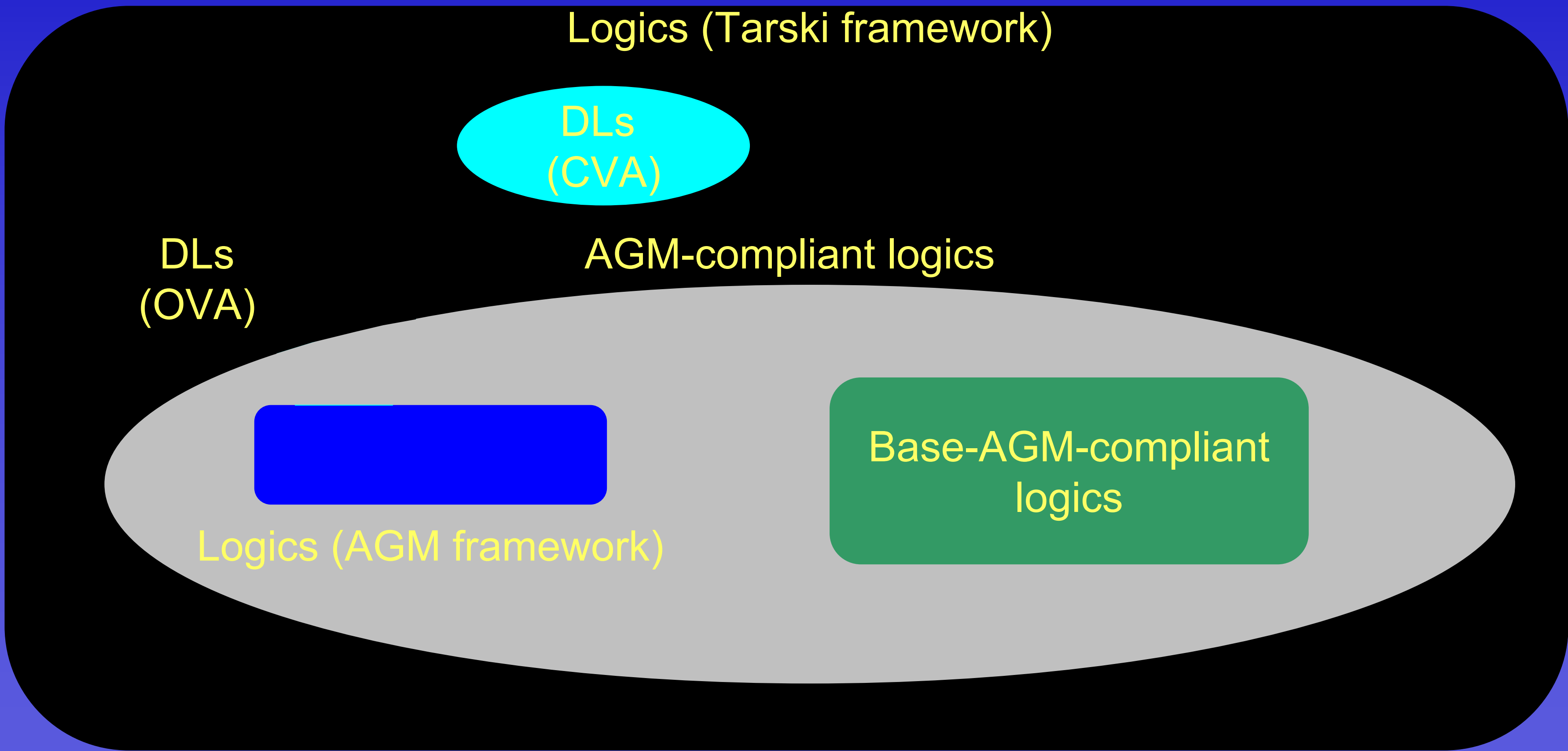
CVA and OVA (2)

- Two viewpoints on existence:
  - All elements exist, some with zero information
    All elements are “relevant”
    Everything is in the ontological signature
    No point in dynamically adding/removing elements
    Open Vocabulary Assumption – OVA
  - Only the elements that are “relevant” exist
    The “relevant” elements are exactly those that appear explicitly in the KB
    Only the “relevant” elements appear in the signature
    Can add/remove elements dynamically
    Closed Vocabulary Assumption – CVA

DLs and OWL under CVA/OVA

- DLs and OWL under CVA: non-AGM-compliant
- CVA is inherently non-AGM-compliant
- Under OVA, things are not so straightforward:
  - Some DLs are AGM-compliant
  - Some are not
  - OWL is not AGM-compliant

The Situation



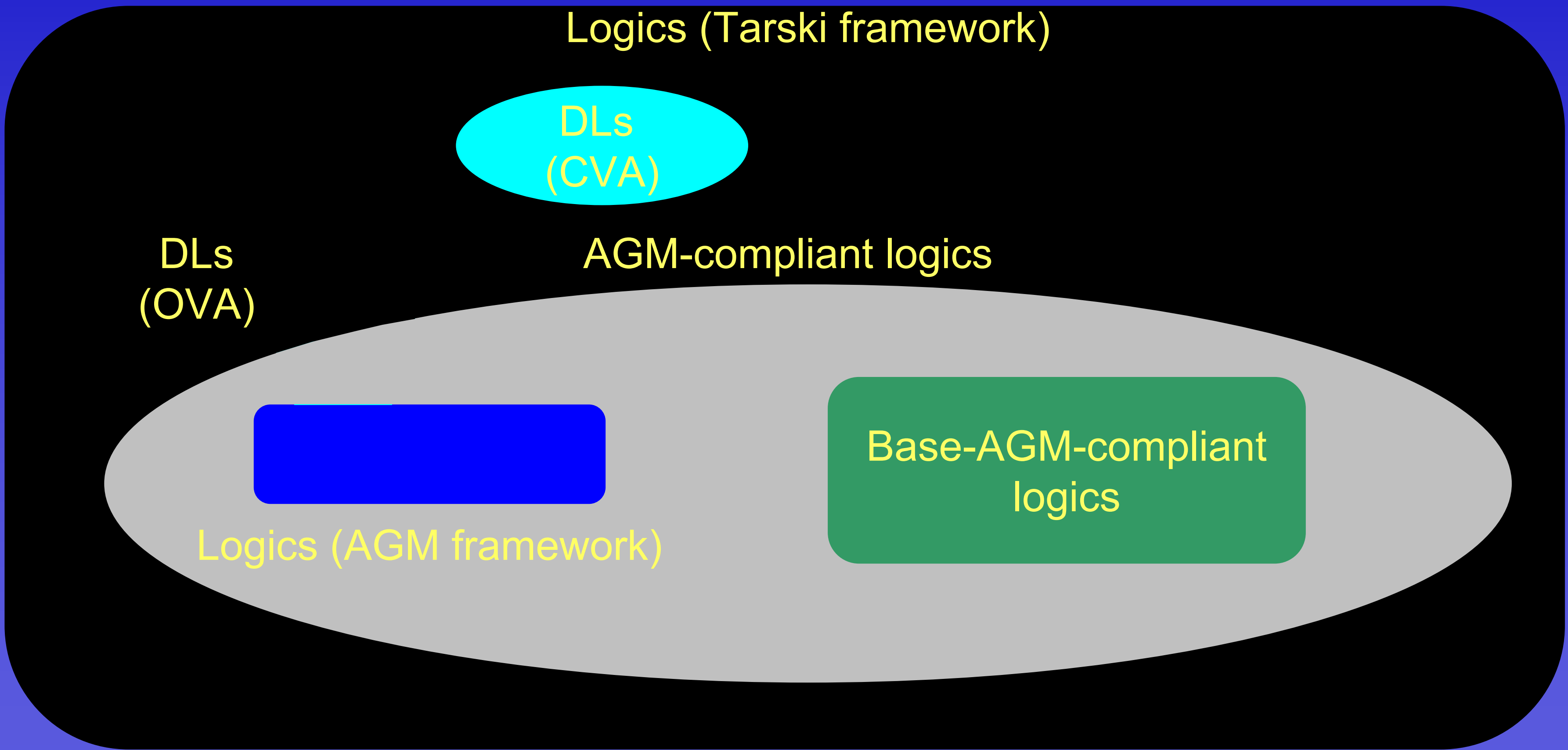
AGM-Compliant DLs

- Recall: A logic <L,Cn> is AGM-compliant iff for every $X,Y \subseteq L$ such that $Cn(\emptyset) \subset Cn(Y) \subset Cn(X)$ there is a $Z \subseteq L$ such that:
  - $Cn(Z) \subset Cn(X)$
  - $Cn(Y \cup Z) = Cn(X)$
- A DL can be shown AGM-compliant by finding two transformations such that:
- Each set $X \subseteq L$ is mapped to a set $X' \subseteq L$ such that:
  - $X' = \{A_j \sqsupseteq \top \mid j \in J\}$
  - $Cn(X') = Cn(X)$
- Each set $Y \subseteq L$ is mapped to a set $Y' \subseteq L$ such that:
  - $Y' = \{B \sqsupseteq \top\}$
  - $Cn(\emptyset) \subset Cn(Y') \subseteq Cn(Y)$
  - There is an interpretation I such that $B^I = \emptyset$

Discussion
AGM-Compliant DLs

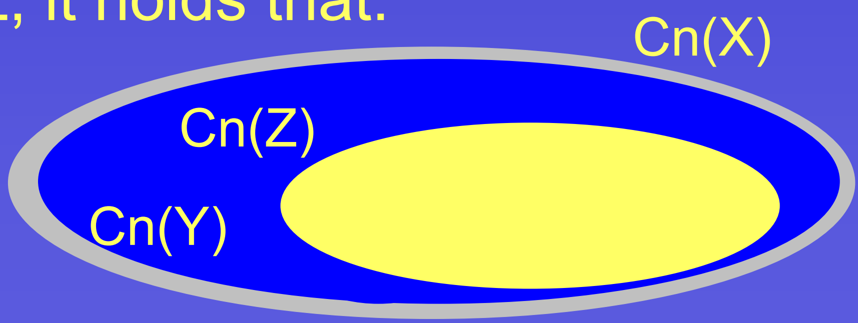
- Such transformations exist in several DLs
  - Depending on the available operators and connectives
- Several alternatives exist:
  - Different transformations
  - Equivalent operators
  - Necessary transformations depend on available axioms
  - Additional operators do not bar AGM-compliance
- The important point is:
  if such transformations exist, the DL is AGM-compliant

The Situation



Non-AGM-Compliance (General)

- Take a set of axioms X and the set Cn(X)
- Set $Y=\{x\in Cn(X) \mid Cn(\{x\})\subset Cn(X)\}$
- Suppose that $Cn(Y)\subset Cn(X)$
- For $X,Y\subseteq L$, we seek a $Z\subseteq L$ such that:
  - $Cn(Z)\subset Cn(X)$
  - $Cn(Y\cup Z)=Cn(X)$
- Lemma: For X, Y and any Z, it holds that: $Cn(Y\cup Z)=Cn(Y)\subset Cn(X)$



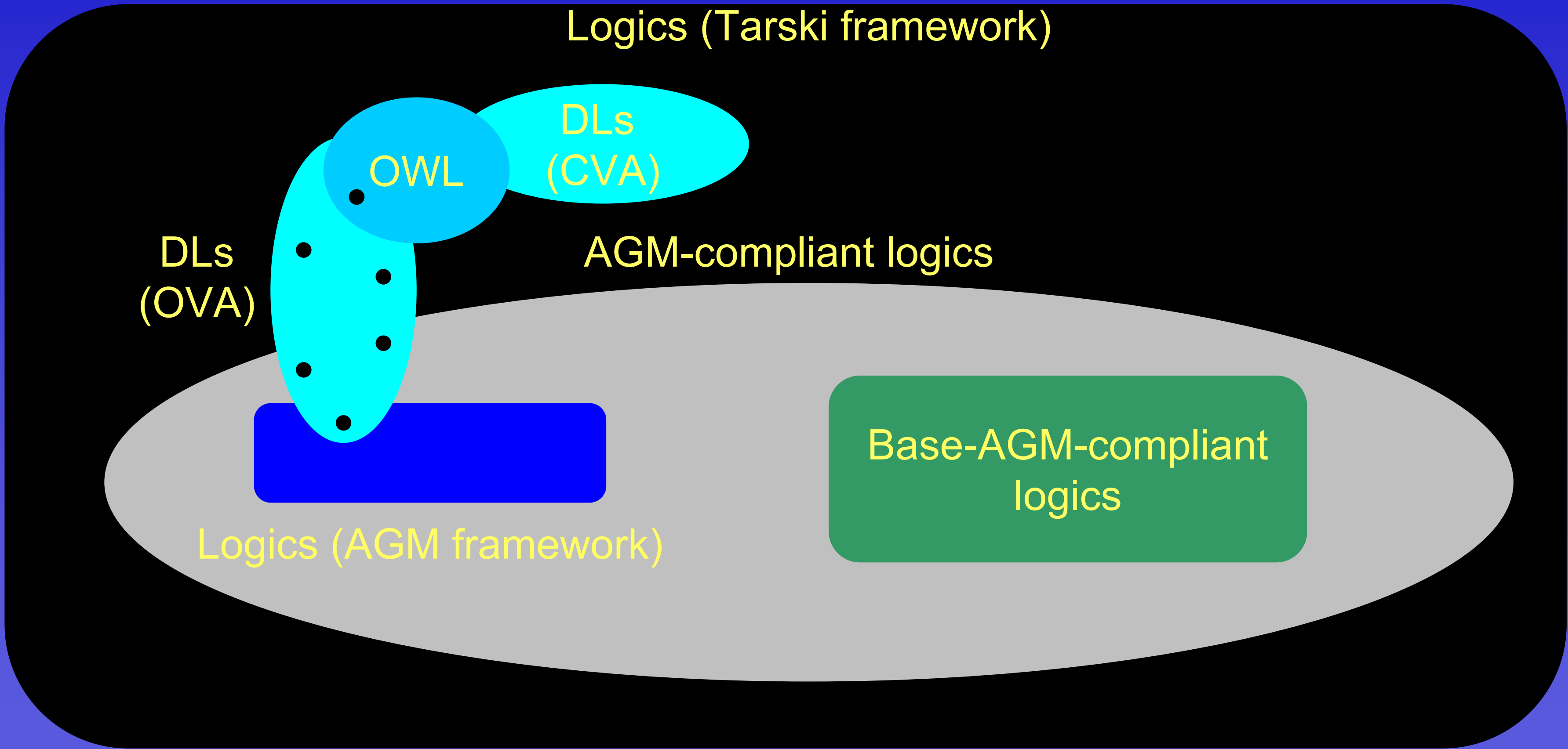
Non-AGM-Compliance (in DLs)

- This situation appears for sets of the form:
  - $X=\{R \equiv S\}$, for two roles R, S
  - $X=\{R \sqsubseteq S\}$, for two roles R, S
- Proper consequences of $X=\{R \equiv S\}$:
  - $\exists R.A \equiv \exists S.A$, $\forall R.A \equiv \forall S.A$, …
- Depending on the operators allowed in the DL, all these consequences combined may not imply X:
  - Role operators ($\neg$, $\sqcap$, $\sqcup$) seem necessary to imply X
- So: such DLs are not AGM-compliant

Discussion
Non-AGM-Compliant DLs

- Several DLs with role axioms ($R \equiv S$), but without role operators ($\neg$, $\sqcap$, $\sqcup$) are not AGM-compliant
  - Role operators rarely appear in the literature
  - We encourage research on DLs that contain these operators
- Rule of thumb:
  - If transformations can be found then AGM-compliant
  - If transformations cannot be found, try $X=\{R \equiv S\}$ or $X=\{R \sqsubseteq S\}$
- Conditions are not necessary and sufficient
- …but so far, they have worked in every DL we have tried

The Situation



DLs in the Literature (Partial List)

AGM-compliant:

- $ALCO^{\neg,\square,\square}$
- $ALC^{\neg,\square,\square}$ with no axioms involving individuals
- ALCO with no axioms involving roles
- ALC with no axioms involving individuals and no axioms involving roles
- All DLs with more operators but no more connectives (axiom types)

Non-AGM-compliant:

- SH, SHI, SHIN, SHOIN, SHOIN(D), $SHOIN^{+}$, $SHOIN^{+}$(D), SHIQ, SHIF, SHIF(D), $SHIF^{+}$, $SHIF^{+}$(D)
- FL_0, FL^{-} with role axioms
- All DLs between ALH and ALHCIOQ
- OWL Full, OWL DL, OWL Lite with annotations
- OWL DL, OWL Lite without annotations

Conclusion

- Phase 1:
  - Proposed the study of ontology evolution from a different perspective, using belief change ideas and terminology
- Phase 2:
  - Focused on the AGM theory of contraction
  - Determined its applicability to general logics
  - Focused on DLs and OWL, providing specialized conditions for these logics

Future Work

- Study other belief change approaches
- Connection of AGM-compliance with other AGM-related results:
  - The operation of revision
  - Levi identity
  - Representation theorems
- The development and/or implementation of a specific algorithm for integration into ontology evolution tools

Thank you!

Notes

- Some more details follow

A Naïve Approach

- Suppose that $X=\{A \sqsupseteq \top\}$, $Y=\{B \sqsupseteq \top\}$, such that:
  - $Cn(\emptyset) \subset Cn(Y) \subset Cn(X)$
- We seek a $Z \subseteq L$ such that:
  - $Cn(Z) \subset Cn(X)$
  - $Cn(Y \cup Z) = Cn(X)$
- Take $Z=\{A \sqsupseteq B\}$, then:
  - $X \models Z$, so $Cn(Z) \subseteq Cn(X)$
  - $X \models Z$, $X \models Y$ and $Y \cup Z \models X$, so $Cn(Y \cup Z) = Cn(X)$

CATCH
Should be $Cn(Z) \subset Cn(X)$

A Refined Approach

- Suppose that $X=\{A \sqsupseteq \top\}$, $Y=\{B \sqsupseteq \top\}$, such that:
  - $Cn(\emptyset) \subset Cn(Y) \subset Cn(X)$
  - There is an interpretation I such that $B^I=\emptyset$
- We seek a $Z \subseteq L$ such that:
  - $Cn(Z) \subset Cn(X)$
  - $Cn(Y \cup Z)=Cn(X)$
- Take $Z=\{A \sqsupseteq B\}$, then:
  - $X \models Z$ and $Z \not\models X$, so $Cn(Z) \subset Cn(X)$
  - $X \models Z$, $X \models Y$ and $Y \cup Z \models X$, so $Cn(Y \cup Z)=Cn(X)$

Generalizing the Approach

- What if X,Y are not of the desired form?
- Transformations (might) apply:
  - Find a X' such that $X'=\{A_j \sqsupseteq \top \mid j \in J\}$ and $Cn(X')=Cn(X)$
  - Find a Y' such that $Y'=\{B \sqsupseteq \top\}$, $Cn(\emptyset) \subset Cn(Y') \subseteq Cn(Y)$ and there is an interpretation I such that $B^I=\emptyset$

- Take $Z=\{A_j \sqsupseteq B \mid j \in J\}$, then it can be shown that:
  - $X \models Z$ and $Z \not\models X$, so $Cn(Z) \subset Cn(X)$
  - $X \models Z$, $X \models Y$ and $Y \cup Z \models X$, so $Cn(Y \cup Z)=Cn(X)$