Consequence-based Reasoning for Lightweight Description Logics
Problems with Tableau-based Reasoning

Consequence-based Reasoning for $\mathcal{EL}$

Practical Consequence-based $\mathcal{EL}$ Reasoning
Key Features of Tableau Reasoning

Builds a counter model to test entailments

- Prove $\mathcal{O} \models C \subseteq D$
- Try to build an interpretation of $\mathcal{O}$ that satisfies $C \sqcap \neg D$
Key Features of Tableau Reasoning

Builds a counter model to test entailments

- Prove $\mathcal{O} \models C \sqsubseteq D$
- Try to build an interpretation of $\mathcal{O}$ that satisfies $C \sqcap \neg D$

Sound and complete for expressive DLs (up to $\mathbf{SROIQ}$)
Key Features of Tableau Reasoning

Builds a counter model to test entailments

- Prove $\mathcal{O} \models C \subseteq D$
- Try to build an interpretation of $\mathcal{O}$ that satisfies $C \cap \neg D$

Sound and complete for expressive DLs (up to $\mathbf{SROIQ}$)

Often practical despite the high complexity
However...

Focuses on individual (non) entailments

- Test each $A \sqcap B$ to classify an ontology
- 99.9% entailments do not hold
- Optimizations rectify this but…
However...

Focuses on individual (non) entailments

- Test each $A \sqsubseteq B$ to classify an ontology
- 99.9% entailments do not hold
- Optimizations rectify this but...

Models explode (completion graphs exponential even for $\mathcal{ALC}$)
However...

Focuses on individual (non) entailments

- Test each $A \sqsubseteq B$ to classify an ontology
- 99.9% entailments do not hold
- Optimizations rectify this but...

Models explode (completion graphs exponential even for $\mathcal{ALC}$)

Non-determinism where it shouldn’t be (GCIs)
- Example: $\exists R. \ C \sqsubseteq D$
However...

Focuses on individual (non) entailments

- Test each $A \sqsubseteq B$ to classify an ontology
- 99.9% entailments do not hold
- Optimizations rectify this but...

Models explode (completion graphs exponential even for $\mathcal{ALC}$)

Non-determinism where it shouldn’t be (GCIs)

- Example: $\exists R. C \sqsubseteq D$
  
- General GCI rule: add $\neg \exists R. C \uplus D$, horrible
However...

Focuses on individual (non) entailments

- Test each $A \sqsubseteq B$ to classify an ontology
- 99.9% entailments do not hold
- Optimizations rectify this but...

Models explode (completion graphs exponential even for $\mathcal{ALC}$)

Non-determinism where it shouldn’t be (GCIs)

- Example: $\exists R. C \sqsubseteq D$
- General GCI rule: add $\neg \exists R. C \sqcup D$, horrible
- Absorption: rewrite to $\exists R. \top \sqsubseteq D \sqcup \forall R. \neg C$, still bad
Issues Can Be Addressed

Via optimizations

- Smarter consistency algorithms
  (extending absorption, etc.)

- Smarter classification algorithms
  (reduce the number of consistency checks)

- Share information across consistency checks
  (pseudo-model merging, etc.)
Issues Can Be Addressed

Via optimizations

- Smarter \textit{consistency} algorithms (extending absorption, etc.)
- Smarter \textit{classification} algorithms (reduce the number of consistency checks)
- Share information \textit{across} consistency checks (pseudo-model merging, etc.)

Via alternative reasoning approaches

- Consequence-based reasoning
Problems with Tableau-based Reasoning

Consequence-based Reasoning for \( \mathcal{EL} \)

Practical Consequence-based \( \mathcal{EL} \) Reasoning
Consequence-based Reasoning

Goal-directed classification procedure for $\mathcal{EL}$ (and extensions)
Consequence-based Reasoning

Goal-directed classification procedure for $EL$ (and extensions)

Derives subsumptions, does not check non-subsumptions
Consequence-based Reasoning

Goal-directed classification procedure for \( \mathcal{EL} \) (and extensions)

Derives subsumptions, does not check non-subsumptions

Natural deduction calculus instead of model building tableau
Consequence-based Reasoning

Goal-directed classification procedure for \( \mathcal{EL} \) (and extensions)

Derives subsumptions, does not check non-subsumptions

Natural deduction calculus instead of model building tableau

Brief history

► Long time ago: used in logic programming
► 2005: \( \textcolor{red}{\mathcal{EL}^+} \) procedure by F. Baader, C. Lutz, and S. Brandt
► 2009: Full GALEN classified in a few seconds (Y. Kazakov)
► 2011: Extended to non-Horn logics
What is $\mathcal{EL}$?

$\mathcal{EL}$ is a lightweight Description Logic

Basis of OWL EL ontology language
What is $\mathcal{EL}$?

$\mathcal{EL}$ is a **lightweight** Description Logic

Basis of OWL EL ontology language

From its W3C spec:

- suitable for applications employing ontologies that define very large numbers of concepts and/or roles,
What is $\mathcal{EL}$?

$\mathcal{EL}$ is a lightweight Description Logic

Basis of OWL EL ontology language

From its W3C spec:

- suitable for applications employing ontologies that define very large numbers of concepts and/or roles,
- captures the expressivity of many existing ontologies,
What is $\mathcal{EL}$?

$\mathcal{EL}$ is a lightweight Description Logic

Basis of OWL EL ontology language

From its W3C spec:

- suitable for applications employing ontologies that define very large numbers of concepts and/or roles,
- captures the expressivity of many existing ontologies,
- ontology consistency, concept subsumption, and instance checking can be decided in \textit{polynomial} time
What is \( \mathcal{EL} \)?

Lightweight means limited expressive power.
What is $\mathcal{EL}$?

Lightweight means limited expressive power

Can express:

- Top concept $\top$
- Concept intersections: $\text{ContagiousDisease} \sqcap \text{AirborneDisease}$
What is $\mathcal{EL}$?

**Lightweight** means limited expressive power

Can express:

- Top concept $\top$
- Concept intersections: $\text{ContagiousDisease} \sqcap \text{AirborneDisease}$
- Existentials: $\exists \text{hasPart}.\text{Heart}$
What is $\mathcal{EL}$?

Lightweight means limited expressive power

Can express:

- Top concept $\top$
- Concept intersections: $\text{ContagiousDisease} \sqcap \text{AirborneDisease}$
- Existentials: $\exists \text{hasPart}.\text{Heart}$
- General concept inclusions (GCIs):
  $\text{Disease} \sqcap \exists \text{transmittedThrough}.\text{Air} \sqsubseteq \text{AirborneDisease}$
- Concept equivalences:
  $\text{InfectiousDisease} \equiv \text{Disease} \sqcap \exists \text{resultsFrom}.\text{Infection}$
What is $£L$?

Lightweight means limited expressive power

Can express:

- Top concept $\top$
- Concept intersections: $\text{ContagiousDisease} \sqcap \text{AirborneDisease}$
- Existentials: $\exists \text{hasPart.Heart}$
- General concept inclusions (GCIs):
  $\text{Disease} \sqcap \exists \text{transmittedThrough.Air} \sqsubseteq \text{AirborneDisease}$
- Concept equivalences:
  $\text{InfectiousDisease} \equiv \text{Disease} \sqcap \exists \text{resultsFrom.Infection}$
- Property inclusions: $\text{hasProperPart} \sqsubseteq \text{hasPart}$
- Property equivalences: $\text{hasDisease} \equiv \text{suffersFrom}$
What is $\mathcal{EL}$?

Lightweight means limited expressive power

Can express:

- Top concept $\top$
- Concept intersections: $\text{ContagiousDisease} \sqcap \text{AirborneDisease}$
- Existentials: $\exists \text{hasPart.Heart}$
- General concept inclusions (GCIs):
  $\text{Disease} \sqcap \exists \text{transmittedThrough.Air} \sqsubseteq \text{AirborneDisease}$
- Concept equivalences:
  $\text{InfectiousDisease} \equiv \text{Disease} \sqcap \exists \text{resultsFrom.Infection}$
- Property inclusions: $\text{hasProperPart} \sqsubseteq \text{hasPart}$
- Property equivalences: $\text{hasDisease} \equiv \text{suffersFrom}$
- Subproperty chains: $\text{hasProperPart} \circ \text{hasPart} \sqsubseteq \text{hasProperPart}$
What is $\mathcal{EL}$?

Lightweight means limited expressive power

Can not express:

- Disjunctions: BacterialDisease $\sqcup$ ViralDisease
- Universal role restrictions: $\forall$ isCausedBy Virus
- Question: negation? $\neg$ ViralDisease

No (recall De Morgan Laws)

ABoxes and $\bot$ (disjointness) can be added

Semantics (model theory) is the same as for ALC: $I = (\Delta I, \cdot I)$
What is \( \mathcal{EL} \)?

Lightweight means limited expressive power

Can not express:
- Disjunctions: \( \text{BacterialDisease} \sqcup \text{ViralDisease} \)
What is $\mathcal{EL}$?

Lightweight means limited expressive power

Can not express:

- Disjunctions: $\text{BacterialDisease} \sqcup \text{ViralDisease}$
- Universal role restrictions: $\forall \text{isCausedBy}.\text{Virus}$
What is \( \mathcal{EL} \)?

**Lightweight** means limited expressive power

Can **not** express:

- **Disjunctions:** \( \text{BacterialDisease} \sqcup \text{ViralDisease} \)
- **Universal role restrictions:** \( \forall \text{isCausedBy} \cdot \text{Virus} \)
- **Question:** negation? \( \neg \text{ViralDisease} \)
What is $\mathcal{EL}$?

Lightweight means limited expressive power

Can not express:

- Disjunctions: $\text{BacterialDisease} \sqcup \text{ViralDisease}$
- Universal role restrictions: $\forall \text{isCausedBy}.\text{Virus}$
- Question: negation? $\neg \text{ViralDisease}$
  
  No (recall De Morgan Laws)
What is $\mathcal{EL}$?

Lightweight means limited expressive power

Can not express:

- Disjunctions: $\text{BacterialDisease} \sqcup \text{ViralDisease}$
- Universal role restrictions: $\forall \text{isCausedBy} . \text{Virus}$
- Question: negation? $\neg \text{ViralDisease}$

No (recall De Morgan Laws)

ABoxes and $\bot$ (disjointness) can be added
What is $\mathcal{EL}$?

Lightweight means limited expressive power

Can not express:

- Disjunctions: $\text{BacterialDisease} \sqcup \text{ViralDisease}$
- Universal role restrictions: $\forall \text{isCausedBy}. \text{Virus}$
- Question: negation? $\neg \text{ViralDisease}$

No (recall De Morgan Laws)

ABoxes and $\perp$ (disjointness) can be added

Semantics (model theory) is the same as for $\mathcal{ALC}$: $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$
Tractability of $\mathcal{EL}$

$\mathcal{EL}$ is one of the few DLs for which standard reasoning tasks, such as ontology classification or subsumption, are **tractable**

**Question:** What do we mean by **tractable**?
Tractability of $\mathcal{EL}$

$\mathcal{EL}$ is one of the few DLs for which standard reasoning tasks, such as ontology classification or subsumption, are tractable.

Question: What do we mean by tractable?

Answer: Complexity is polynomial in the size of $\mathcal{O}$

Each $A \sqsubseteq B$ can be decided in polynomial time.
Tractability vs Practicality

Does tractable always imply practical?
Tractability vs Practicality

Does tractable always imply practical?

Let’s make a simple calculation:

- SNOMED CT contains roughly 300,000 medical terms
- We can build a tableau and check all pairwise subsumptions
- Every test is tractable, thus so is an $O(n^2)$ algo
- Suppose each test takes just 1 millisecond
- Then we classify SNOMED CT just in $300,000 \times 300,000$ ms
  $\approx 2.8$ years
Tractability vs Practicality

Does tractable always imply practical?

Let’s make a simple calculation:

- **SNOMED CT** contains roughly 300,000 medical terms
- We can build a tableau and check all pairwise subsumptions
- Every test is tractable, thus so is an $O(n^2)$ algo
- Suppose each test takes just 1 millisecond
- Then we classify **SNOMED CT just** in $300,000 \times 300,000$ ms
  $\approx 2.8$ years

This is **NOT** practical, need a goal-oriented approach
Abstract Idea of Consequence-based Reasoning

Use inference rules to derive consequences of existing axioms
Abstract Idea of Consequence-based Reasoning

Use *inference rules* to derive *consequences* of existing axioms

Inference rule: \[ R_{\text{name}} \frac{\alpha_1 \ldots \alpha_n}{\eta} : \gamma \]

- \( \alpha_1, \ldots, \alpha_n \) are premises
- \( \gamma \) is a boolean *side condition*
- \( \eta \) is the *conclusion*
Abstract Idea of Consequence-based Reasoning

Use *inference rules* to derive *consequences* of existing axioms

Inference rule: $R_{\text{name}} \frac{\alpha_1 \ldots \alpha_n}{\eta} : \gamma$

- $\alpha_1, \ldots, \alpha_n$ are premises
- $\gamma$ is a boolean *side condition*
- $\eta$ is the conclusion

$\mathit{Exp}$ is a set of *expressions*

$R_{\text{name}}$ is *applicable* if $\alpha_1, \ldots, \alpha_n \in \mathit{Exp}$ and $\gamma$ is *true*
Abstract Idea of Consequence-based Reasoning

Use **inference rules** to derive **consequences** of existing axioms

Inference rule: \( R_{\text{name}} \frac{\alpha_1 \ldots \alpha_n}{\eta} : \gamma \)

- \( \alpha_1, \ldots, \alpha_n \) are premises
- \( \gamma \) is a boolean side condition
- \( \eta \) is the conclusion

**Exp** is a set of **expressions**

**R_{\text{name}}** is applicable if \( \alpha_1, \ldots, \alpha_n \in \text{Exp} \) and \( \gamma \) is true

**Closure** of **Exp** w.r.t. \( R_1, \ldots, R_k \) is the minimal set which contains all conclusions of all applicable rules.
Abstract Idea of Consequence-based Reasoning

Use inference rules to derive consequences of existing axioms

Inference rule: $R_{name} \frac{\alpha_1 \ldots \alpha_n}{\eta} : \gamma$

- $\alpha_1, \ldots, \alpha_n$ are premises
- $\gamma$ is a boolean side condition
- $\eta$ is the conclusion

Exp is a set of expressions

$R_{name}$ is applicable if $\alpha_1, \ldots, \alpha_n \in Exp$ and $\gamma$ is true

Closure of Exp w.r.t. $R_1, \ldots, R_k$ is the minimal set which contains all conclusions of all applicable rules.

$\mathcal{EL} : Exp = \text{axioms, Closure contains the inferred taxonomy}$
Inference Rules for $\mathcal{EL}$

$R_0 \quad \frac{}{C \sqsubseteq C}$
Inference Rules for $\mathcal{EL}$

$\mathcal{R}_0 \quad \frac{}{C \sqsubseteq C}$

$\mathcal{R}_T \quad \frac{}{C \sqsubseteq \top}$
Inference Rules for $\mathcal{EL}$

$R_0 \quad C \sqsubseteq C$

$R_T \quad C \sqsubseteq \top$

$R_\sqsubseteq \quad \frac{C \sqsubseteq D \quad D \sqsubseteq E \in O}{C \sqsubseteq E}$
Inference Rules for $\mathcal{EL}$

\[ R_0 \quad \frac{C \sqsubseteq C}{\top} \]

\[ R_{\top} \quad \frac{C \sqsubseteq \top}{\cdot} \]

\[ R_{\sqsubseteq} \quad \frac{C \sqsubseteq D}{D \sqsubseteq E \in \mathcal{O}} \]

\[ R_{\exists} \quad \frac{E \sqsubseteq R \cdot C \quad C \sqsubseteq D}{\exists R \cdot D \text{ occurs in } \mathcal{O}} \]
Inference Rules for $\mathcal{EL}$

$R_0$ \hspace{1cm} $\frac{C \sqsubseteq C}{\;}$

$R_\sqsubseteq$ \hspace{1cm} $\frac{C \sqsubseteq D \quad D \sqsubseteq E \in \mathcal{O}}{C \sqsubseteq E}$

$R_\exists$ \hspace{1cm} $\frac{E \sqsubseteq R.C \quad C \sqsubseteq D}{E \sqsubseteq R.D \quad \exists R.D \text{ occurs in } \mathcal{O}}$

$R_\cap$ \hspace{1cm} $\frac{C \sqsubseteq D_1 \cap D_2}{C \sqsubseteq D_1 \quad C \sqsubseteq D_2}$
Inference Rules for $\mathcal{EL}$

$\mathcal{R}_0 \quad \frac{}{C \sqsubseteq C}$

$\mathcal{R}_\sqsubseteq \quad \frac{C \sqsubseteq D \quad D \sqsubseteq E}{C \sqsubseteq E} \quad : \quad D \sqsubseteq E \in \mathcal{O}$

$\mathcal{R}_\exists \quad \frac{E \sqsubset \exists R.C}{E \sqsubset \exists R.D} \quad : \quad \exists R.D \text{ occurs in } \mathcal{O}$

$\mathcal{R}_\cap^- \quad \frac{C \sqsubseteq D_1 \cap D_2}{C \sqsubseteq D_1 \quad C \sqsubseteq D_2}$

$\mathcal{R}_\cap^+ \quad \frac{C \sqsubseteq D_1 \cap D_2}{C \sqsubseteq D_1 \quad C \sqsubseteq D_2} \quad : \quad D_1 \cap D_2 \text{ occurs in } \mathcal{O}$

Facts about this rule system:

- Side conditions ensure all concepts in conclusions occur in $\mathcal{O}$
- Question: why is this important?
- Answer: ensures termination and polynomiality
- Provably complete, derives all entailed subsumptions between concept names
Inference Rules for $\mathcal{EL}$

\[ R_0 \quad \frac{C \sqsubseteq C}{\text{}} \]

\[ R_\top \quad \frac{C \sqsubseteq \top}{\text{}} \]

\[ R_\sqsubseteq \quad \frac{C \sqsubseteq D \sqsubseteq E}{D \sqsubseteq E \in \mathcal{O}} \]

\[ R_\exists \quad \frac{E \sqsubseteq \exists R.C \quad C \sqsubseteq D}{\exists R.D \text{ occurs in } \mathcal{O}} \]

\[ R_\sqcap^- \quad \frac{C \sqsubseteq D_1 \sqcap D_2}{C \sqsubseteq D_1 \sqcap D_2} \]

\[ R_\sqcap^+ \quad \frac{C \sqsubseteq D_1 \sqcap D_2}{D_1 \sqcap D_2 \text{ occurs in } \mathcal{O}} \]

Facts about this rule system:

- Side conditions ensure all concepts in conclusions occur in $\mathcal{O}$
- **Question:** why is this important?
Inference Rules for $\mathcal{EL}$

\[
\begin{align*}
\text{R}_0 & \quad \frac{C \sqsubseteq C}{\text{R}^+} \\
\text{R}_\sqsubseteq & \quad \frac{C \sqsubseteq D}{C \sqsubseteq E} : D \sqsubseteq E \in \mathcal{O} \\
\text{R}_\exists & \quad \frac{E \sqsubseteq \exists R.C}{E \sqsubseteq \exists R.D} : \exists R.D \text{ occurs in } \mathcal{O} \\
\text{R}_\sqcap & \quad \frac{C \sqsubseteq D_1 \sqcap D_2}{C \sqsubseteq D_1, C \sqsubseteq D_2} : D_1 \sqcap D_2 \text{ occurs in } \mathcal{O}
\end{align*}
\]

Facts about this rule system:

- Side conditions ensure all concepts in conclusions occur in $\mathcal{O}$
- **Question**: why is this important?
- **Answer**: ensures termination and polynomiality
Inference Rules for $\mathcal{EL}$

$R_0$ \quad $C \sqsubseteq C$

$R_{\sqsubseteq}$ \quad $\frac{C \sqsubseteq D}{C \sqsubseteq E} : D \sqsubseteq E \in \mathcal{O}$

$R_{\exists}$ \quad $\frac{E \sqsubseteq R.C}{E \sqsubseteq \exists R.D} : \exists R.D \text{ occurs in } \mathcal{O}$

$R_{\sqcap}$ \quad $\frac{C \sqsubseteq D_1 \sqcap D_2}{C \sqsubseteq D_1}$

$R^{+}$ \quad $\frac{C \sqsubseteq D_1 \sqcap D_2}{C \sqsubseteq D_1 \sqcap D_2} : D_1 \sqcap D_2 \text{ occurs in } \mathcal{O}$

Facts about this rule system:

- Side conditions ensure all concepts in conclusions occur in $\mathcal{O}$
- **Question**: why is this important?
- **Answer**: ensures termination and polynomialsity
- **Provably** complete, derives all entailed subsumptions between concept names
\( \mathcal{EL} \) Classification, Example

Ontology \( \mathcal{O} \):

1. \( \text{A} \sqsubseteq \exists R.\text{B} \)
2. \( \text{B} \sqsubseteq \text{C} \)
3. \( \exists R.\text{C} \sqsubseteq \text{D} \)

\[ \begin{align*}
R_0 & \quad \frac{C \sqsubseteq C}{-} \\
R_\subseteq & \quad \frac{C \sqsubseteq D \quad C \sqsubseteq E}{D \sqsubseteq E \in \mathcal{O}} \\
R_\exists & \quad \frac{E \sqsubseteq \exists R.\text{C} \quad C \sqsubseteq \text{D}}{\exists R.\text{D} \text{ occurs in } \mathcal{O}} \\
R_\land & \quad \frac{C \sqsubseteq D_1 \land D_2}{C \sqsubseteq D_1 \land D_2} \\
R_\lor & \quad \frac{C \sqsubseteq D_1 \land D_2}{C \sqsubseteq D_1 \lor D_2} \quad : D_1 \lor D_2 \text{ occurs in } \mathcal{O}
\end{align*} \]
**EL Classification, Example**

Ontology $\mathcal{O}$:
1. $A \sqsubseteq \exists R.B$
2. $B \sqsubseteq C$
3. $\exists R.C \sqsubseteq D$

\[ \begin{array}{c}
\mathcal{R}_0 & A \sqsubseteq A \\
\mathcal{R}_1 & C \sqsubseteq C \\
\mathcal{R}_\sqsubseteq & C \sqsubseteq D \quad : \quad D \sqsubseteq E \in \mathcal{O} \\
\mathcal{R}_\exists & E \sqsubseteq \exists R.C \quad E \sqsubseteq \exists R.D \quad : \quad \exists R.D \text{ occurs in } \mathcal{O} \\
\mathcal{R}_\sqcap & C \sqsubseteq D_1 \sqcap D_2 \\
\mathcal{R}_\sqcup & C \sqsubseteq D_1 \sqcup D_2 \\
\mathcal{R}_\sqcap & C \sqsubseteq D_1 \sqcap D_2 \quad : \quad D_1 \sqcap D_2 \text{ occurs in } \mathcal{O}
\end{array} \]
**$\mathcal{EL}$ Classification, Example**

- $A \sqsubseteq A$

**Ontology $\mathcal{O}$:**
1. $A \sqsubseteq \exists R.B$
2. $B \sqsubseteq C$
3. $\exists R.C \sqsubseteq D$

\[
\begin{array}{c}
\mathcal{R}_0 \\
\hline
A \sqsubseteq A
\end{array}
\]

\[
\begin{array}{ll}
\mathcal{R}_0 & C \sqsubseteq C \\
\mathcal{R}_\sqsubseteq & \frac{C \sqsubseteq D}{C \sqsubseteq E} : D \sqsubseteq E \in \mathcal{O} \\
\mathcal{R}_\exists & \frac{E \sqsubseteq \exists R.C}{E \sqsubseteq \exists R.D} : \exists R.D \text{ occurs in } \mathcal{O} \\
\mathcal{R}_\sqcap^- & \frac{C \sqsubseteq D_1 \sqcap D_2}{C \sqsubseteq D_1} \\
\mathcal{R}_\sqcap^+ & \frac{C \sqsubseteq D_1 \sqcap D_2}{C \sqsubseteq D_2}
\end{array}
\]
**EL Classification, Example**

- $A \sqsubseteq A$

Ontology $\mathcal{O}$:

1. $A \sqsubseteq \exists R.B$
2. $B \sqsubseteq C$
3. $\exists R.C \sqsubseteq D$

\[
\begin{array}{c}
\text{R}_0 \quad B \sqsubseteq B \\
\begin{array}{c}
\text{R}_0 \quad C \sqsubseteq C \\
\end{array}
\begin{array}{c}
\text{R}_E \quad C \sqsubseteq D \quad D \sqsubseteq E \in \mathcal{O} \\
\end{array}
\begin{array}{c}
\text{R}_R \quad E \sqsubseteq \exists R.C \quad C \sqsubseteq D \quad \exists R.D \text{ occurs in } \mathcal{O} \\
\end{array}
\begin{array}{c}
\text{R}_R^+ \quad C \sqsubseteq D \quad D \sqsubseteq E \in \mathcal{O} \\
\end{array}
\begin{array}{c}
\text{R}_R^- \quad C \sqsubseteq D_1 \cap D_2 \\
\text{R}_R^+ \quad C \sqsubseteq D \quad D \sqsubseteq E \in \mathcal{O} \\
\end{array}
\end{array}
\]
**EL Classification, Example**

- $A \sqsubseteq A$
- $B \sqsubseteq B$

**Ontology $\mathcal{O}$:**
1. $A \sqsubseteq \exists R.B$
2. $B \sqsubseteq C$
3. $\exists R.C \sqsubseteq D$

\[
\begin{array}{c}
R_0 \\
\hline
C \sqsubseteq C \\
\hline
R_\square \\
\hline
C \sqsubseteq D & D \sqsubseteq E \in \mathcal{O} \\
\hline
R_\exists \\
\hline
E \sqsubseteq \exists R.C \quad C \sqsubseteq D \\
\hline
R_\sqcap \\
\hline
C \sqsubseteq D_1 \sqcap D_2 & C \sqsubseteq D_2 \\
\hline
R_\sqcap^+ \\
\hline
C \sqsubseteq D_1 \sqcap D_2 \\
\hline
\end{array}
\]

$D_1 \sqcap D_2$ occurs in $\mathcal{O}$
\[ \mathcal{EL} \] Classification, Example

- \( A \sqsubseteq A \)
- \( B \sqsubseteq B \)

Ontology \( \mathcal{O} \):
1. \( A \sqsubseteq \exists R . B \)
2. \( B \sqsubseteq C \)
3. \( \exists R . C \sqsubseteq D \)

\[ \mathcal{R} \] \( \mathcal{R}_0 \) \( C \sqsubseteq C \) \( \mathcal{R}_T \) \( C \sqsubseteq T \)

\[ \mathcal{R} \] \( \mathcal{R}_\sqsubseteq \) \( C \sqsubseteq D \) \( D \sqsubseteq E \in \mathcal{O} \) \( \mathcal{R}_\exists \) \( E \sqsubseteq \exists R . C \) \( \exists R . D \) occurs in \( \mathcal{O} \)

\[ \mathcal{R} \]
- \( \mathcal{R}_\sqcap \) \( C \sqsubseteq D_1 \cap D_2 \) \( \mathcal{R}_\sqcup \) \( C \sqsubseteq D_1 \sqcup D_2 \)
- \( C \sqsubseteq D_1 \sqcap D_2 \)

\( \mathcal{R} \) \( \mathcal{R}_\sqsubseteq \) \( C \sqsubseteq D_1 \) \( C \sqsubseteq D_2 \) \( \mathcal{R}_\exists \) \( E \sqsubseteq \exists R . C \) \( \exists R . D \) occurs in \( \mathcal{O} \)

\( \mathcal{R} \) \( \mathcal{R}_\sqcap \) \( C \sqsubseteq D_1 \cap D_2 \) \( \mathcal{R}_\sqcup \) \( C \sqsubseteq D_1 \sqcup D_2 \)

\( \mathcal{R} \) \( \mathcal{R}_\sqsubseteq \) \( C \sqsubseteq D_1 \) \( C \sqsubseteq D_2 \)
**EL Classification, Example**

- $A \sqsubseteq A$
- $B \sqsubseteq B$
- $A \sqsubseteq \exists R.B$

**Ontology $\mathcal{O}$:**
1. $A \sqsubseteq \exists R.B$
2. $B \sqsubseteq C$
3. $\exists R.C \sqsubseteq D$

$$R \sqsubseteq \frac{A \sqsubseteq A}{A \sqsubseteq \exists R.B} : A \sqsubseteq \exists R.B \in \mathcal{O}$$

**Rules:**

- $R_0$  \hfill $C \sqsubseteq C$
- $R_T$  \hfill $C \sqsubseteq T$
- $R_\sqsubseteq$  \hfill $\frac{C \sqsubseteq D}{C \sqsubseteq E} : D \sqsubseteq E \in \mathcal{O}$
- $R_\exists$  \hfill $\frac{E \sqsubseteq \exists R.C}{E \sqsubseteq \exists R.D} : \exists R.D \text{ occurs in } \mathcal{O}$
- $R_\sqcap$  \hfill $\frac{C \sqcap D_1 \sqcap D_2}{C \sqsubseteq D_1 \sqcap D_2}$
- $R_\sqcup$  \hfill $\frac{C \sqcup D_1 \sqcup D_2}{C \sqsubseteq D_1 \sqcup D_2} : D_1 \sqcup D_2 \text{ occurs in } \mathcal{O}$
\[ \mathcal{EL} \] Classification, Example

- A \sqsubseteq A
- A \sqsubseteq \exists R.B
- B \sqsubseteq B

Ontology \( \mathcal{O} \):
1. A \sqsubseteq \exists R.B
2. B \sqsubseteq C
3. \exists R.C \sqsubseteq D

\[
\begin{array}{c}
\frac{B \sqsubseteq B}{B \sqsubseteq C} \because B \sqsubseteq C \in \mathcal{O}
\end{array}
\]

\[
\begin{array}{l}
\mathcal{R}_0 \qquad C \sqsubseteq C \\
\mathcal{R}_\sqsubseteq \qquad \frac{C \sqsubseteq D}{C \sqsubseteq E} \because D \sqsubseteq E \in \mathcal{O} \\
\mathcal{R}_\exists \qquad \frac{E \sqsubseteq \exists R.C}{E \sqsubseteq \exists R.D} \because \exists R.D \text{ occurs in } \mathcal{O} \\
\mathcal{R}_\sqcap \qquad \frac{C \sqsubseteq D_1 \sqcap D_2}{C \sqsubseteq D_1} \because C \sqsubseteq D_2 \\
\mathcal{R}_\sqcup \qquad \frac{C \sqsubseteq D_1 \sqcup D_2}{C \sqsubseteq D_1 \sqcup D_2} \because D_1 \sqcup D_2 \text{ occurs in } \mathcal{O}
\end{array}
\]
**EL Classification, Example**

- ▶ A ⊑ A
- ▶ A ⊑ ∃R.B
- ▶ B ⊑ B
- ▶ B ⊑ C

**Ontology \( \mathcal{O} \):**
1. A ⊑ ∃R.B
2. B ⊑ C
3. ∃R.C ⊑ D

\[
\begin{align*}
\mathsf{R_0} & \quad \frac{C \sqsubseteq C}{C \sqsubseteq C} \\
\mathsf{R_1} & \quad \frac{C \sqsubseteq D \quad C \sqsubseteq E}{D \sqsubseteq E} \\
\mathsf{R_2} & \quad \frac{C \sqsubseteq D_1 \cap D_2}{C \sqsubseteq D_1 \cap C \sqsubseteq D_2} \\
\mathsf{R_3} & \quad \frac{E \sqsubseteq \exists R.C \quad C \sqsubseteq D}{\exists R.D \quad C \sqsubseteq D} \\
\mathsf{R_T} & \quad \frac{C \sqsubseteq T}{C \sqsubseteq T}
\end{align*}
\]

\(D_1 \sqcap D_2 \) occurs in \( \mathcal{O} \)
\[ \mathcal{EL} \text{ Classification, Example} \]

- \( A \sqsubseteq A \)
- \( A \sqsubseteq \exists R.B \)
- \( B \sqsubseteq B \)
- \( B \sqsubseteq C \)

**Ontology** \( \mathcal{O} \):
1. \( A \sqsubseteq \exists R.B \)
2. \( B \sqsubseteq C \)
3. \( \exists R.C \sqsubseteq D \)

\[
\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C}{A \sqsubseteq \exists R.C} : \exists R.C \text{ occurs in } \mathcal{O}
\]

\[
\begin{align*}
R_0 & \quad C \sqsubseteq C \\
R_\sqsubseteq & \quad \frac{C \sqsubseteq D}{C \sqsubseteq E} : D \sqsubseteq E \in \mathcal{O} \\
R_\exists & \quad \frac{E \sqsubseteq \exists R.C \quad C \sqsubseteq D}{C \sqsubseteq E \sqsubseteq \exists R.D} : \exists R.D \text{ occurs in } \mathcal{O} \\
R_\cap & \quad \frac{C \sqsubseteq D_1 \cap D_2}{C \sqsubseteq D_1} \quad \frac{C \sqsubseteq D_2}{} \\
R_\cup & \quad \frac{C \sqsubseteq D_1 \cup D_2}{C \sqsubseteq D_1} \quad \frac{C \sqsubseteq D_2}{} \\
R_+ & \quad \frac{C \sqsubseteq D_1 \cap D_2}{C \sqsubseteq D_1} \quad \frac{C \sqsubseteq D_2}{} \\
\end{align*}
\]
**EL Classification, Example**

- A ⊑ A
- A ⊑ ∃R.B
- A ⊑ ∃R.C
- B ⊑ B
- B ⊑ C

Ontology O:
1. A ⊑ ∃R.B
2. B ⊑ C
3. ∃R.C ⊑ D

\[
\begin{array}{c}
R_3 \quad A ⊑ ∃R.B \\
B ⊑ C \\
A ⊑ ∃R.C
\end{array}
\quad : \exists R.C \text{ occurs in } O
\]

\[
\begin{array}{c}
R_0 \quad C ⊑ C \\
R_T \quad C ⊑ T
\end{array}
\]

\[
\begin{array}{c}
R_≤ \quad C ⊑ D \\
C ⊑ E \\
D ⊑ E \in O
\end{array}
\quad : \exists R.D \text{ occurs in } O
\]

\[
\begin{array}{c}
R_\cap \quad C ⊑ D_1 \cap D_2 \\
C ⊑ D_1 \\
C ⊑ D_2
\end{array}
\quad : D_1 \cap D_2 \text{ occurs in } O
\]
\[ \mathcal{EL} \] Classification, Example

- \( A \sqsubseteq A \)
- \( A \sqsubseteq \exists R.B \)
- \( A \sqsubseteq \exists R.C \)
- \( B \sqsubseteq B \)
- \( B \sqsubseteq C \)

Ontology \( \mathcal{O} \):

1. \( A \sqsubseteq \exists R.B \)
2. \( B \sqsubseteq C \)
3. \( \exists R.C \sqsubseteq D \)

\[
\begin{align*}
\mathcal{R} & \in \mathcal{A} \sqsubseteq \exists R.C \implies \exists R.C \sqsubseteq D \in \mathcal{O} \\
\mathcal{R}_0 & \implies C \sqsubseteq C \\
\mathcal{R}_T & \implies C \sqsubseteq T \\
\mathcal{R}_E & \implies C \sqsubseteq D : D \sqsubseteq E \in \mathcal{O} \\
\mathcal{R}_E & \implies E \sqsubseteq \exists R.C, C \sqsubseteq D : \exists R.D \text{ occurs in } \mathcal{O} \\
\mathcal{R}_E & \implies C \sqsubseteq D_1 \cap D_2 : D_1 \sqsubseteq D_2 \text{ occurs in } \mathcal{O}
\end{align*}
\]
\[ \mathcal{EL} \] Classification, Example

\[ \begin{align*}
&\rightarrow A \sqsubseteq A \\
&\rightarrow A \sqsubseteq \exists R.B \\
&\rightarrow A \sqsubseteq \exists R.C \\
&\rightarrow A \sqsubseteq D
\end{align*} \]

\[ \begin{align*}
&\rightarrow B \sqsubseteq B \\
&\rightarrow B \sqsubseteq C
\end{align*} \]

Ontology \( \mathcal{O} \) :

1. \( A \sqsubseteq \exists R.B \)
2. \( B \sqsubseteq C \)
3. \( \exists R.C \sqsubseteq D \)

\[ \begin{align*}
R \in \mathcal{O} : \exists R.C \sqsubseteq D & \quad \text{entails} \\
A \sqsubseteq \exists R.C & \\
A \sqsubseteq D & \quad \text{from}\ A \sqsubseteq \exists R.C
\end{align*} \]

\[ \begin{align*}
R_0 & \quad C \sqsubseteq C \\
R_T & \quad C \sqsubseteq T \\
R \in & \quad C \sqsubseteq D \quad : \quad D \sqsubseteq E \in \mathcal{O} \\
R_3 & \quad E \sqsubseteq R.C \quad \frac{C \sqsubseteq D \quad C \sqsubseteq D}{C \sqsubseteq E} \quad : \quad \exists R.D \quad \text{occurs in} \quad \mathcal{O} \\
R \in^n & \quad C \sqsubseteq D_1 \cap D_2 \quad : \quad D_1 \sqsubseteq D_2 \\
R \in^+ & \quad C \sqsubseteq D_1 \quad C \sqsubseteq D_2 \quad : \quad D_1 \sqsubseteq D_2 \quad \text{occurs in} \quad \mathcal{O}
\end{align*} \]
\( \mathcal{EL} \) Classification, Example

- \( A \sqsubseteq A \)
- \( A \sqsubseteq \exists R.B \)
- \( A \sqsubseteq \exists R.C \)
- \( A \sqsubseteq D \)
- \( B \sqsubseteq B \)
- \( B \sqsubseteq C \)

Ontology \( \mathcal{O} \):
1. \( A \sqsubseteq \exists R.B \)
2. \( B \sqsubseteq C \)
3. \( \exists R.C \sqsubseteq D \)

These are all entailed atomic subsumptions

\[
\begin{align*}
R_0 \quad & C \sqsubseteq C \\
R_{\sqsubseteq} \quad & \frac{C \sqsubseteq D}{C \sqsubseteq E} \quad : \quad D \sqsubseteq E \in \mathcal{O} \\
R_{\exists} \quad & \frac{E \sqsubseteq \exists R.C \quad C \sqsubseteq D}{C \sqsubseteq \exists R.D} \quad : \quad \exists R.D \text{ occurs in } \mathcal{O} \\
R_{\neg \sqsubseteq} \quad & \frac{C \sqsubseteq D_1 \cap D_2}{C \sqsubseteq D_1, C \sqsubseteq D_2} \\
R_{\neg \exists} \quad & \frac{C \sqsubseteq D_1 \cap D_2}{C \sqsubseteq D_1, C \sqsubseteq D_2} \quad : \quad D_1 \sqsubseteq D_2 \text{ occurs in } \mathcal{O}
\end{align*}
\]
Consequence vs. Tableau-based Procedures

Advantages of consequence-based procedures over subsumption testing procedures (tableau)
Consequence vs. Tableau-based Procedures

Advantages of consequence-based procedures over subsumption testing procedures (tableau):

1. They never consider subsumptions that are not entailed number of entailed subsumptions in SNOMED CT is <0.01%
Consequence vs. Tableau-based Procedures

Advantages of consequence-based procedures over subsumption testing procedures (tableau):

1. They never consider subsumptions that are not entailed. The number of entailed subsumptions in SNOMED CT is $<0.01\%$.

2. They can derive all subsumptions in one pass. The average subsumption time is much smaller than 1 millisecond.
Consequence vs. Tableau-based Procedures

Advantages of consequence-based procedures over subsumption testing procedures (tableau):

1. They never consider subsumptions that are not entailed. The number of entailed subsumptions in SNOMED CT is <0.01%.

2. They can derive all subsumptions in one pass. The average subsumption time is much smaller than 1 millisecond.

Modern tableau-based reasoners have optimizations that reduce the number of subsumption tests and reuse results between the tests.
Basic Implementation

A well-known procedure which can be used with any rules
Basic Implementation

A well-known procedure which can be used with any rules

Use two collections of expressions:

- **Closure**: expressions between which all rules are applied (initially empty)
- **Todo**: expressions to which rules are yet to be applied (initialized with input expressions)
Basic Implementation

A well-known procedure which can be used with any rules

Use two collections of expressions:

- **Closure**: expressions between which all rules are applied (initially empty)
- **Todo**: expressions to which rules are yet to be applied (initialized with input expressions)

Apply inferences:

- Take the next element from **Todo**
- Insert into **Closure**
- If it is new, apply all inferences with elements from **Closure**
- Add the result into **Todo**
Basic Implementation

A well-known procedure which can be used with any rules

Use two collections of expressions:

- **Closure**: expressions between which all rules are applied (initially empty)
- **Todo**: expressions to which rules are yet to be applied (initialized with input expressions)

Apply inferences:

- Take the next element from Todo
- Insert into Closure
- If it is new, apply all inferences with elements from Closure
- Add the result into Todo
Basic Implementation

A well-known procedure which can be used with any rules

Use two collections of expressions:

- **Closure**: expressions between which all rules are applied (initially empty)
- **Todo**: expressions to which rules are yet to be applied (initialized with input expressions)

Apply inferences:

- Take the next element from Todo
- Insert into Closure
- If it is new, apply all inferences with elements from Closure
- Add the result into Todo
Basic Implementation

A well-known procedure which can be used with **any rules**

- Use two collections of expressions:
  - **Closure**: expressions between which all rules are applied (initially empty)
  - **Todo**: expressions to which rules are yet to be applied (initialized with input expressions)

- Apply inferences:
  - Take the next element from **Todo**
  - Insert into **Closure**
  - If it is new, apply all inferences with elements from **Closure**
  - Add the result into **Todo**
Granularity of $\mathcal{EL}$ Reasoning

Subsumers for $C$ and $D$ can be computed (semi) independently

Limited interaction:

- Derived $C \sqsubseteq D$
- Derived $C \sqsubseteq \exists R.D$
Granularity of $\mathcal{EL}$ Reasoning

Subsumers for $C$ and $D$ can be computed (semi) independently

Limited interaction:

- Derived $C \sqsubseteq D$
- Derived $C \sqsubseteq \exists R. D$

Limited interaction enables the reasoner do:

- incremental reasoning: changes for $C$ do not affect $D$
Granularity of $\mathcal{EL}$ Reasoning

Subsumers for $C$ and $D$ can be computed (semi) independently

Limited interaction:
- Derived $C \sqsubseteq D$
- Derived $C \sqsubseteq \exists R. D$

Limited interaction enables the reasoner do:
- incremental reasoning: changes for $C$ do not affect $D$
- parallel reasoning: $C$ and $D$ saturated in parallel
Granularity of $\mathcal{EL}$ Reasoning

Subsumers for $C$ and $D$ can be computed (semi) independently

Limited interaction:

- Derived $C \sqsubseteq D$
- Derived $C \sqsubseteq \exists R. D$

Limited interaction enables the reasoner to:

- **incremental reasoning**: changes for $C$ do not affect $D$
- **parallel reasoning**: $C$ and $D$ saturated in parallel
- **distributed reasoning**: $C$ and $D$ may come from different ontologies!
Problems with Tableau-based Reasoning

Consequence-based Reasoning for $\mathcal{EL}$

Practical Consequence-based $\mathcal{EL}$ Reasoning
What is Practical?

You want your reasoner to be:

- Fast on current inputs (performance)
- Handle growing inputs (scalability)
What is Practical?

You want your reasoner to be:

- Fast on current inputs (performance)
- Handle growing inputs (scalability)

Question: how do you know that the reasoner is practical?
What is Practical?

You want your reasoner to be:

- Fast on current inputs (performance)
- Handle growing inputs (scalability)

**Question:** how do you know that the reasoner is practical?

**Answer:** evaluation!
What is Practical?

You want your reasoner to be:

- Fast on current inputs *(performance)*
- Handle growing inputs *(scalability)*

**Question:** how do you know that the reasoner is practical?

**Answer:** evaluation!

**Sad truth:** early implementations are almost always *not* practical
Evaluation Goals

Measure performance and scalability
Evaluation Goals

Measure performance and scalability

Find room for improvement

- Reveals performance bottlenecks (where the program spends most time)
- Scalability obstacles (which parameters’ growth hits performance most)
Evaluation Goals

Measure performance and scalability

Find room for improvement

- Reveals performance **bottlenecks**
  (where the program spends most time)

- Scalability obstacles
  (which parameters’ growth hits performance most)

Without evaluation optimization is like **shooting in the dark**
Stages of $\mathcal{EL}$ Reasoning

Consider one-time classification

- Indexing
- Saturation
- Taxonomy Construction
Stages of $\mathcal{EL}$ Reasoning

Consider one-time classification

We focus on saturation and indexing
EL Saturation Statistics

Useful saturation metrics:

- Number of rule applications
- Time spent applying rules
- Time spent selecting applicable rules
\textbf{\textit{EL} Saturation Statistics}

Useful saturation metrics:

- Number of rule applications
- Time spent \textit{applying} rules
- Time spent \textit{selecting} applicable rules

Collection mechanism \textbf{must} ensure:

- Can be turned on/off any time
- No need to change the rules
- Extensibility (w.r.t. new rules or new stats)
Statistics Collection in ELK

ELK rules represented as a class hierarchy

```
Rule
apply(α, Closure, O)
```

![Diagram showing class hierarchy with Rule as the root, Subsumption Rule, ... (ellipsis), and Existential Rule as children.]

```
Subsumption Rule
```
```
Existential Rule
```
Statistics Collection in ELK

ELK rules represented as a class hierarchy

Algorithm steps:
- Take $\alpha$ from Todo
- Pick some rule $R$
- $\text{visitor.visit}(\alpha, R, \text{Closure}, \mathcal{O})$
Statistics Collection in ELK

ELK rules represented as a class hierarchy

Algorithm steps:
- Take $\alpha$ from Todo
- Pick some rule $R$
- $\text{visitor}.\text{visit}(\alpha, R, \text{Closure}, \emptyset)$

Default visitor:
- $R.\text{apply}(\alpha, \text{Closure}, \emptyset)$
Statistics Collection in ELK

ELK rules represented as a class hierarchy

Algorithm steps:
- Take $\alpha$ from Todo
- Pick some rule $R$
- visitor.visit($\alpha$, $R$, Closure, $O$)

Rule counting visitor:
1. $\text{counter}_R++$
2. basic visit
Statistics Collection in ELK

ELK rules represented as a class hierarchy

Algorithm steps:
- Take $\alpha$ from Todo
- Pick some rule $R$
- `visitor.visit($\alpha, R, \text{Closure}, \mathcal{O}$)`

Rule timing visitor:
1. $t = \text{(current time)}$
2. basic visit
3. $t = \text{(current time)} - t$
Interpretation of Statistics

Rule application statistics collected, what is it telling me?
Interpretation of Statistics

Rule application statistics collected, what is it telling me?

Many (specific) rule applications:

- Duplicate inferences?
- Redundant inferences (not needed for inferring $A \sqsubseteq B$)?
Interpretation of Statistics

Rule application statistics collected, what is it telling me?

Many (specific) rule applications:
  ▶ Duplicate inferences?
  ▶ Redundant inferences (not needed for inferring $A \sqsubseteq B$)?

Much time spent selecting rules
  ▶ Poor saturation algorithm design
Interpretation of Statistics

Rule application statistics collected, what is it telling me?

Many (specific) rule applications:
  - Duplicate inferences?
  - Redundant inferences (not needed for inferring $A \sqsubseteq B$)?

Much time spent selecting rules
  - Poor saturation algorithm design

Much time spent applying rules:
  - Poor implementation of rules
  - This is mostly engineering
Interpretation of Statistics

Rule application statistics collected, what is it telling me?

Many (specific) rule applications:
- Duplicate inferences?
- Redundant inferences (not needed for inferring \(A \sqsubseteq B\))?

Much time spent selecting rules
- Poor saturation algorithm design

Much time spent applying rules:
- Poor implementation of rules
- This is mostly engineering

We consider the first two issues
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:

1. $A \sqsubseteq B$
2. $A \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqsubseteq B \cap C$
5. $B \cap C \sqsubseteq E$

Derived superclasses of $A$:
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:

1. $A \sqsubseteq B$
2. $A \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqsubseteq B \cap C$
5. $B \cap C \sqsubseteq E$

Derived superclasses of $A$:

- $A \sqsubseteq B$ by $R \sqsubseteq$
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:

1. $A \sqsubseteq B$
2. $A \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqsubseteq B \cap C$
5. $B \cap C \sqsubseteq E$

Derived superclasses of $A$:

- $A \sqsubseteq B$ by $R\sqsubseteq$
- $A \sqsubseteq C$ by $R\sqsubseteq$

Question: Does it break the termination property?

Answer: No, duplicate inferences are not inserted into Closure.

Lesson: $R\sqsubseteq$ should not apply to conclusions of $R\sqcap$.
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:
1. $A \sqsubseteq B$
2. $A \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqsubseteq B \cap C$
5. $B \cap C \sqsubseteq E$

Derived superclasses of $A$:
- $A \sqsubseteq B$ by $R_{\sqsubseteq}$
- $A \sqsubseteq C$ by $R_{\sqsubseteq}$
- $A \sqsubseteq B \cap C$ by $R_{\sqcap}$
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:

1. $A \sqsubseteq B$
2. $A \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqsubseteq B \sqcap C$
5. $B \sqcap C \sqsubseteq E$

Derived superclasses of $A$:

- $A \sqsubseteq B$ by $R_\sqsubseteq$
- $A \sqsubseteq C$ by $R_\sqsubseteq$
- $A \sqsubseteq B \sqcap C$ by $R_+^\sqcap$
- $A \sqsubseteq B$ by $R_\sqcap^-$
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:

1. $A \sqsubseteq B$
2. $A \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqsubseteq B \cap C$
5. $B \cap C \sqsubseteq E$

Derived superclasses of $A$:

- $A \sqsubseteq B$ by $R_{\sqsubseteq}$
- $A \sqsubseteq C$ by $R_{\sqsubseteq}$
- $A \sqsubseteq B \cap C$ by $R_{+}$
- $A \sqsubseteq B$ by $R_{\cap}$
- $A \sqsubseteq C$ by $R_{\cap}$
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:

1. $A \sqsubseteq B$
2. $A \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqsubseteq B \cap C$
5. $B \cap C \sqsubseteq E$

Derived superclasses of $A$:

- $A \sqsubseteq B$ by $R_{\sqsubseteq}$
- $A \sqsubseteq C$ by $R_{\sqsubseteq}$
- $A \sqsubseteq B \cap C$ by $R^+_\sqcap$
- $A \sqsubseteq B$ by $R^-_{\sqcap}$
- $A \sqsubseteq C$ by $R^-_{\sqcap}$
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $O$:

1. $A \sqsubseteq B$
2. $A \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqsubseteq B \cap C$
5. $B \cap C \sqsubseteq E$

Derived superclasses of $A$:

- $A \sqsubseteq B$ by $R\sqsubseteq$
- $A \sqsubseteq C$ by $R\sqsubseteq$
- $A \sqsubseteq B \cap C$ by $R^+\sqcap$
- $A \sqsubseteq B$ by $R\sqcap$
- $A \sqsubseteq C$ by $R\sqcap$

**Question:** Does it break the termination property?
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:

1. $A \sqsubseteq B$
2. $A \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqsubseteq B \cap C$
5. $B \cap C \sqsubseteq E$

Derived superclasses of $A$:

- $A \sqsubseteq B$ by $R_{\sqsubseteq}$
- $A \sqsubseteq C$ by $R_{\sqsubseteq}$
- $A \sqsubseteq B \cap C$ by $R_{\sqcap}^+$
- $A \sqsubseteq B$ by $R_{\sqcap}$
- $A \sqsubseteq C$ by $R_{\sqcap}$

Question: Does it break the termination property?

Answer: No, duplicate inferences are not inserted into Closure
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:

1. $A \sqsubseteq B$
2. $A \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqsubseteq B \cap C$
5. $B \cap C \sqsubseteq E$

Derived superclasses of $A$:

- $A \sqsubseteq B$ by $R_\sqsubseteq$
- $A \sqsubseteq C$ by $R_\sqsubseteq$
- $A \sqsubseteq B \cap C$ by $R_\sqcap^+$
- $A \sqsubseteq B$ by $R_\sqcap^-$
- $A \sqsubseteq C$ by $R_\sqcap^-$

Question: Does it break the termination property?

Answer: No, duplicate inferences are not inserted into Closure

Lesson: $R_\sqcap^-$ should not apply to conclusions of $R_\sqcap^+$
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:

1. $A \sqsubseteq \exists R. B$
2. $B \sqsubseteq C$
3. $\exists R. C \sqsubseteq D$
4. $C \sqsubseteq E$
5. $\exists R. E \sqsubseteq F$

Derived superclasses of $A$:
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:

1. $A \sqsubseteq \exists R . B$
2. $B \sqsubseteq C$
3. $\exists R . C \sqsubseteq D$
4. $C \sqsubseteq E$
5. $\exists R . E \sqsubseteq F$

Derived superclasses of $A$:

- $B \sqsubseteq E$ by $R$
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:
1. $A \sqsubseteq \exists R.B$
2. $B \sqsubseteq C$
3. $\exists R.C \sqsubseteq D$
4. $C \sqsubseteq E$
5. $\exists R.E \sqsubseteq F$

Derived superclasses of $A$:
- $B \sqsubseteq E$ by $R_\sqsubseteq$
- $A \sqsubseteq \exists R.E$ by $R_\exists$:
  \[
  \begin{array}{c}
  A \sqsubseteq \exists R.B \\
  B \sqsubseteq E \\
  \hline
  A \sqsubseteq \exists R.E
  \end{array}
  \]
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:

1. $A \sqsubseteq \exists R. B$
2. $B \sqsubseteq C$
3. $\exists R. C \sqsubseteq D$
4. $C \sqsubseteq E$
5. $\exists R. E \sqsubseteq F$

Derived superclasses of $A$:

- $B \sqsubseteq E$ by $R$\sqsubseteq
- $A \sqsubseteq \exists R. E$ by $R$\exists : $\frac{A \sqsubseteq R. B}{A \sqsubseteq \exists R. E}$  $\frac{B \sqsubseteq E}{B \sqsubseteq \exists R. E}$
- $A \sqsubseteq \exists R. C$ by $R$\exists : $\frac{A \sqsubseteq R. B}{A \sqsubseteq \exists R. C}$  $\frac{B \sqsubseteq C}{B \sqsubseteq \exists R. C}$
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:

1. $A \sqsubseteq \exists R. B$
2. $B \sqsubseteq C$
3. $\exists R. C \sqsubseteq D$
4. $C \sqsubseteq E$
5. $\exists R. E \sqsubseteq F$

Derived superclasses of $A$:

- $B \sqsubseteq E$ by $R_{\sqsubseteq}$
- $A \sqsubseteq \exists R. E$ by $R_{\exists}$:
  - $\exists R. B \sqsubseteq B \sqsubseteq E$
  - $\exists R. C \sqsubseteq C \sqsubseteq E$

Lesson: $R_{\exists}$ should not apply to conclusions of $R_{\sqsubseteq}$.
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology $\mathcal{O}$:
1. $A \sqsubseteq \exists R.B$
2. $B \sqsubseteq C$
3. $\exists R.C \sqsubseteq D$
4. $C \sqsubseteq E$
5. $\exists R.E \sqsubseteq F$

Derived superclasses of $A$:
- $B \sqsubseteq E$ by $R_{\sqsubseteq}$
- $A \sqsubseteq \exists R.E$ by $R_{\exists}$:
  - $AC \exists R.B \ B \sqsubseteq E$
  - $AC \exists R.E$
- $A \sqsubseteq \exists R.C$ by $R_{\exists}$:
  - $AC \exists R.B \ B \sqsubseteq C$
  - $AC \exists R.C$
- $A \sqsubseteq \exists R.E$ by $R_{\exists}$:
  - $AC \exists R.C \ C \sqsubseteq E$
  - $AC \exists R.E$
Duplicate Inferences

Sometimes the same inference(s) can be derived more than once.

Ontology \( \mathcal{O} \):

1. \( A \sqsubseteq \exists R.B \)
2. \( B \sqsubseteq C \)
3. \( \exists R.C \sqsubseteq D \)
4. \( C \sqsubseteq E \)
5. \( \exists R.E \sqsubseteq F \)

Derived superclasses of \( A \):

- \( B \sqsubseteq E \) by \( R\sqsubseteq \)
- \( A \sqsubseteq \exists R.E \) by \( R\exists : \frac{AC\exists R.B}{A\exists R.E} \frac{B\sqsubseteq E}{B\sqsubseteq E} \)
- \( A \sqsubseteq \exists R.C \) by \( R\exists : \frac{AC\exists R.C}{A\exists R.E} \frac{B\sqsubseteq C}{B\sqsubseteq C} \)
- \( A \sqsubseteq \exists R.E \) by \( R\exists : \frac{AC\exists R.C}{A\exists R.E} \frac{C\sqsubseteq E}{C\sqsubseteq E} \)

Lesson: \( R\exists \) should not apply to conclusions of \( R\exists \)
Duplicate Inferences: Experiment

What reducing duplicate inferences mean in practice?
Duplicate Inferences: Experiment

What reducing duplicate inferences mean in practice?

Let $O_\sqcap$, $O_\exists$ be optimizations of $R_\sqcap$, $R_\exists$

<table>
<thead>
<tr>
<th></th>
<th>time</th>
<th>derived $C \sqsubseteq D$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SNOMED CT</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no optimization</td>
<td>26.31</td>
<td>47,435,318</td>
</tr>
<tr>
<td>with $O_\sqcap$</td>
<td>25.48</td>
<td>41,770,050</td>
</tr>
<tr>
<td>with $O_\exists$</td>
<td>19.75</td>
<td>28,438,072</td>
</tr>
<tr>
<td>with $O_\sqcap$, $O_\exists$</td>
<td>18.71</td>
<td>22,772,804</td>
</tr>
<tr>
<td><strong>GALEN8</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no optimization</td>
<td>50.39</td>
<td>69,138,922</td>
</tr>
<tr>
<td>with $O_\sqcap$</td>
<td>48.63</td>
<td>62,822,068</td>
</tr>
<tr>
<td>with $O_\exists$</td>
<td>25.21</td>
<td>37,267,987</td>
</tr>
<tr>
<td>with $O_\sqcap$, $O_\exists$</td>
<td>20.65</td>
<td>26,111,096</td>
</tr>
</tbody>
</table>
Duplicate Inferences: Experiment

What reducing duplicate inferences mean in practice?

Let $O_\sqcap$, $O_\exists$ be optimizations of $R_\sqcap$, $R_\exists$

<table>
<thead>
<tr>
<th></th>
<th>time</th>
<th>derived $C \sqsubseteq D$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SNOMED CT</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no optimization</td>
<td>26.31</td>
<td><strong>47,435,318</strong></td>
</tr>
<tr>
<td>with $O_\sqcap$</td>
<td>25.48</td>
<td>41,770,050</td>
</tr>
<tr>
<td>with $O_\exists$</td>
<td>19.75</td>
<td>28,438,072</td>
</tr>
<tr>
<td>with $O_\sqcap$, $O_\exists$</td>
<td>18.71</td>
<td><strong>22,772,804</strong></td>
</tr>
<tr>
<td><strong>GALEN8</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no optimization</td>
<td>50.39</td>
<td>69,138,922</td>
</tr>
<tr>
<td>with $O_\sqcap$</td>
<td>48.63</td>
<td>62,822,068</td>
</tr>
<tr>
<td>with $O_\exists$</td>
<td>25.21</td>
<td>37,267,987</td>
</tr>
<tr>
<td>with $O_\sqcap$, $O_\exists$</td>
<td>20.65</td>
<td>26,111,096</td>
</tr>
</tbody>
</table>
Duplicate Inferences: Experiment

What reducing duplicate inferences mean in practice?

Let $O_\cap$, $O_\exists$ be optimizations of $R_\cap$, $R_\exists$

<table>
<thead>
<tr>
<th></th>
<th>time</th>
<th>derived $C \sqsubseteq D$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SNOMED CT</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no optimization</td>
<td>26.31</td>
<td>47,435,318</td>
</tr>
<tr>
<td>with $O_\cap$</td>
<td>25.48</td>
<td>41,770,050</td>
</tr>
<tr>
<td>with $O_\exists$</td>
<td>19.75</td>
<td>28,438,072</td>
</tr>
<tr>
<td>with $O_\cap$, $O_\exists$</td>
<td>18.71</td>
<td>22,772,804</td>
</tr>
<tr>
<td><strong>GALEN8</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no optimization</td>
<td>50.39</td>
<td><strong>69,138,922</strong></td>
</tr>
<tr>
<td>with $O_\cap$</td>
<td>48.63</td>
<td>62,822,068</td>
</tr>
<tr>
<td>with $O_\exists$</td>
<td>25.21</td>
<td>37,267,987</td>
</tr>
<tr>
<td>with $O_\cap$, $O_\exists$</td>
<td>20.65</td>
<td><strong>26,111,096</strong></td>
</tr>
</tbody>
</table>
Last Bit on Duplicate Inferences

**Question:** which subsumption rule generates more duplicates?

\[ \text{R} \quad \frac{C \sqsubseteq D}{C \sqsubseteq E} : \quad D \sqsubseteq E \in \mathcal{O} \quad \text{or} \quad \text{R'} \quad \frac{C \sqsubseteq D}{C \sqsubseteq E} \]

- premise from **Closure**
- + side condition
- both premises from **Closure**
Last Bit on Duplicate Inferences

Question: which subsumption rule generates more duplicates?

\[
R \frac{C \sqsubseteq D}{C \sqsubseteq E} : D \sqsubseteq E \in \mathcal{O} \quad \text{or} \quad R' \frac{C \sqsubseteq D}{C \sqsubseteq E} \quad \text{both premises from Closure}
\]

premise from Closure + side condition

Ontology: \( A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq D \)
Last Bit on Duplicate Inferences

**Question:** which subsumption rule generates more duplicates?

\[
\text{R} \subseteq \frac{C \sqsubseteq D}{C \sqsubseteq E} \quad \text{or} \quad \text{R}' \subseteq \frac{C \sqsubseteq D}{C \sqsubseteq E} \quad \text{both premises from Closure}
\]

Premise from Closure + side condition

Ontology: \(A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq D\)

Inferences using \(\text{R} \subseteq\):

1. \(A \sqsubseteq C\)
2. \(A \sqsubseteq D\)
3. \(B \sqsubseteq D\)

Inferences using \(\text{R}' \subseteq\):

1. \(A \sqsubseteq D\) by \(\text{R}' \subseteq\)
Last Bit on Duplicate Inferences

Question: which subsumption rule generates more duplicates?

\[
\frac{C \sqsubseteq D}{\text{premise from Closure}} : D \sqsubseteq E \in \mathcal{O} \quad \text{or} \quad \frac{C \sqsubseteq E}{\text{both premises from Closure}}
\]

+ side condition

Ontology: \( A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq D \)

Inferences using \( R \sqsubseteq \):

1. \( A \sqsubseteq C \)
2. \( A \sqsubseteq D \)
3. \( B \sqsubseteq D \)

Inferences using \( R' \sqsubseteq \):

1. \( A \sqsubseteq C \)

Last Bit on Duplicate Inferences

Question: which subsumption rule generates more duplicates?

\[
R \subseteq \frac{C \sqsubseteq D}{C \sqsubseteq E} : D \sqsubseteq E \in \mathcal{O} \quad \text{or} \quad R' \subseteq \frac{C \sqsubseteq D \quad D \sqsubseteq E}{C \sqsubseteq E}
\]

premise from Closure

+ side condition

both premises from Closure

Ontology: \( A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq D \)

Inferences using \( R \subseteq \):

1. \( A \sqsubseteq C \)
2. \( A \sqsubseteq D \)
3. \( B \sqsubseteq D \)

Inferences using \( R' \subseteq \):

1. \( A \sqsubseteq C \)
2. \( A \sqsubseteq D \)
Last Bit on Duplicate Inferences

Question: which subsumption rule generates more duplicates?

\[ R \equiv \frac{C \sqsubseteq D}{D \sqsubseteq E} \quad \text{or} \quad R' \equiv \frac{C \sqsubseteq D}{C \sqsubseteq E} \]

premise from Closure + side condition

both premises from Closure

Ontology: \( A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq D \)

Inferences using \( R \sqsubseteq \):

1. \( A \sqsubseteq C \)
2. \( A \sqsubseteq D \)
3. \( B \sqsubseteq D \)

Inferences using \( R' \sqsubseteq \):

1. \( A \sqsubseteq C \)
2. \( A \sqsubseteq D \)
3. \( B \sqsubseteq D \)
Last Bit on Duplicate Inferences

**Question:** which subsumption rule generates more duplicates?

\[ R \subseteq \frac{C \sqsubseteq D}{C \sqsubseteq E} \quad : \quad D \sqsubseteq E \in \mathcal{O} \quad \text{or} \quad R' \subseteq \frac{C \sqsubseteq D}{C \sqsubseteq E} \quad \text{both premises from Closure} \]

+ side condition

Ontology: \( \text{A} \sqsubseteq \text{B}, \text{B} \sqsubseteq \text{C}, \text{C} \sqsubseteq \text{D} \)

Inferences using \( R \subseteq \):

1. \( \text{A} \sqsubseteq \text{C} \)
2. \( \text{A} \sqsubseteq \text{D} \)
3. \( \text{B} \sqsubseteq \text{D} \)

Inferences using \( R' \subseteq \):

1. \( \text{A} \sqsubseteq \text{C} \)
2. \( \text{A} \sqsubseteq \text{D} \)
3. \( \text{B} \sqsubseteq \text{D} \)
4. \( \text{A} \sqsubseteq \text{D} \) by \( R' \subseteq \frac{\text{A} \sqsubseteq \text{B}}{\text{B} \sqsubseteq \text{D}} \)
Last Bit on Duplicate Inferences

Question: which subsumption rule generates more duplicates?

\[
\begin{align*}
\frac{C \sqsubseteq D}{C \sqsubseteq E} : D \sqsubseteq E \in \mathcal{O} \quad \text{or} \quad \frac{C \sqsubseteq D}{C \sqsubseteq E} \quad \text{both premises from Closure} \\
+ \text{ side condition}
\end{align*}
\]

Ontology: \( A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq D \)

Inferences using \( \mathcal{R} \sqsubseteq \):

1. \( A \sqsubseteq C \)
2. \( A \sqsubseteq D \)
3. \( B \sqsubseteq D \)

Inferences using \( \mathcal{R'} \sqsubseteq \):

1. \( A \sqsubseteq C \)
2. \( A \sqsubseteq D \)
3. \( B \sqsubseteq D \)
4. \( A \sqsubseteq D \) by \( \frac{A \sqsubseteq B}{A \sqsubseteq D} \)
Redundant Inferences

Also not all unique inferences are essential for classification

\( \mathcal{O} : A \sqsubseteq \exists R . B, B \sqsubseteq C, D \sqsubseteq \exists R . C \)
Redundant Inferences

Also not all unique inferences are essential for classification

\( \mathcal{O} : A \sqsubseteq \exists R.B, \ B \sqsubseteq C, \ D \sqsubseteq \exists R.C \)

\( \mathcal{O} \) entails no non asserted atomic subsumptions

But we derive \( A \sqsubseteq \exists R.C \) using

\( R_{\exists} : \begin{array}{c}
A \sqsubseteq \exists R.B \\
B \sqsubseteq C \\
A \sqsubseteq \exists R.C
\end{array} \implies \exists R.C \text{ occurs in } \mathcal{O} \)
Redundant Inferences

Also not all unique inferences are essential for classification

\[ \mathcal{O} : A \sqsubseteq \exists R. B, B \sqsubseteq C, D \sqsubseteq \exists R. C \]
\[ \mathcal{O} \] entails no non asserted atomic subsumptions

But we derive \( A \sqsubseteq \exists R. C \) using

\[ R_{\exists} : \frac{A \sqsubseteq \exists R. B \quad B \sqsubseteq C}{A \sqsubseteq \exists R. C} : \exists R. C \text{ occurs in } \mathcal{O} \]

Common in HCLS ontologies, many existentials only on the right:

SomeOrgan \( \sqsubseteq \exists \text{hasRole}. \text{SomeRole} \)
Redundant Inferences

Also not all unique inferences are essential for classification

\( \mathcal{O} : A \sqsubseteq \exists R.B, B \sqsubseteq C, D \sqsubseteq \exists R.C \)

\( \mathcal{O} \) entails no non asserted atomic subsumptions

But we derive \( A \sqsubseteq \exists R.C \) using

\( R_\exists : \frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C}{A \sqsubseteq \exists R.C} : \exists R.C \) occurs in \( \mathcal{O} \)

Common in HCLS ontologies, many existentials only on the right: SomeOrgan \( \sqsubseteq \exists \)hasRole.SomeRole

Can be proved that \( R_\exists : \frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C}{A \sqsubseteq \exists R.C} \) doesn’t have to apply if \( \exists R.C \) doesn’t occur on the left
Redundant Inferences

Also not all unique inferences are essential for classification

\( \mathcal{O} : A \sqsubseteq \exists R.B, B \sqsubseteq C, D \sqsubseteq \exists R.C \)

\( \mathcal{O} \) entails no non asserted atomic subsumptions

But we derive \( A \sqsubseteq \exists R.C \) using

\( R_\exists : \frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C}{A \sqsubseteq \exists R.C} : \exists R.C \text{ occurs in } \mathcal{O} \)

Common in HCLS ontologies, many existentials only on the right:

\textbf{SomeOrgan} \sqsubseteq \exists \text{hasRole}.\text{SomeRole}

Can be proved that \( R_\exists : \frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C}{A \sqsubseteq \exists R.C} \) doesn’t have to apply if \( \exists R.C \) doesn’t occur on the left

Also true for \( R^+_\sqcap : \frac{C \sqsubseteq D_1 \quad C \sqsubseteq D_2}{C \sqsubseteq D_1 \sqcap D_2}, D_1 \sqcap D_2 \text{ must occur on the left} \)
Testing for Duplicate and Redundant Inferences

Some duplicate and redundant inferences are inevitable

How do you know if you have a problem?
Testing for Duplicate and Redundant Inferences

Some duplicate and redundant inferences are inevitable

How do you know if you have a problem?

Manually! Collect statistics and compare numbers:

- \# produced inferences \(\gg\) \# unique inferences \(\leadsto\) potential problem with duplicates
- some rules apply much more often than others \(\leadsto\) potentially redundant inferences
Testing for Duplicate and Redundant Inferences

Some duplicate and redundant inferences are inevitable

How do you know if you have a problem?

Manually! Collect statistics and compare numbers:

- \( \# \text{ produced inferences} \gg \# \text{ unique inferences} \) \( \Rightarrow \) potential problem with duplicates

- some rules apply much more often than others \( \Rightarrow \) potentially redundant inferences

Definitions of \( \gg \) and “much more” depend on ontology
Slow Rule Selection

Finding applicable rules is non-trivial

---

**Input:** Set of named classes $CN$

**Result:** $Closure$, a set containing all atomic subsumptions

$Closure, Todo \leftarrow \emptyset$

for $C \in CN$ do

  Todo.add($\{C \sqsubseteq C, C \sqsubseteq T\}$)
Slow Rule Selection

Finding applicable rules is non-trivial

---

**Input:** Set of named classes $CN$

**Result:** Closure, a set containing all atomic subsumptions

Closure, Todo $\leftarrow \emptyset$;

for $C \in CN$ do

  Todo.add($\{C \sqsubseteq C, C \sqsubseteq \top\}$)

while ($\alpha \leftarrow$ Todo.poll()) $\neq$ null do

  if $\alpha \notin$ Closure then

    Closure.add($\alpha$)

    for $R \in$ select-rules($\alpha$, Closure) do

      Todo.add(conclusions of $R$)

return Closure
Slow Rule Selection

Finding applicable rules is non-trivial

---

**Input:** Set of named classes $CN$

**Result:** Closure, a set containing all atomic subsumptions

Closure, Todo ← $\emptyset$

for $C \in CN$ do

    Todo.add($\{C \sqsubseteq C, C \sqsubseteq T\}$)

while $(\alpha \leftarrow$ Todo.poll()) $\neq$ null do

    if $\alpha \notin$ Closure then

        Closure.add($\alpha$)

        for $R \in select-rules(\alpha, Closure)$ do

            Todo.add(conclusions of $R$)

return Closure

---

In the worst-case $select-rules(\ldots)$ requires $O(|Closure|)$

Need efficient rule lookups
Rule Lookups

Assume that the initialization rules are applied eagerly

Fast processing of each new axiom $\alpha$ requires
Rule Lookups

Assume that the initialization rules are applied eagerly

Fast processing of each new axiom $\alpha$ requires:

- Looking up all unary rules $R^{\alpha}_{\eta} : \gamma$ occurs in $O$
- Looking up all binary rules $R^{\alpha \beta}_{\eta} : \gamma$ occurs in $O$
- Looking up all binary rules $R^{\beta \alpha}_{\eta} : \gamma$ occurs in $O$
Rule Lookups

Assume that the initialization rules are applied **eagerly**

Fast processing of each new axiom $\alpha$ requires:

- Looking up all **unary** rules $R_{\alpha}^{\eta} : \gamma$ occurs in $\mathcal{O}$
- Looking up all **binary** rules $R_{\alpha \beta}^{\eta} : \gamma$ occurs in $\mathcal{O}$
- Looking up all **binary** rules $R_{\beta \alpha}^{\eta} : \gamma$ occurs in $\mathcal{O}$

$\alpha$ is given, $\beta$ and $\gamma$ need to be found **really** fast
Rule Lookups

Assume that the initialization rules are applied \(\text{eagerly}\).

Fast processing of each new axiom \(\alpha\) requires:

- Looking up all \textit{unary} rules \(\frac{\alpha}{\eta}: \gamma\) occurs in \(O\).
- Looking up all \textit{binary} rules \(\frac{\alpha}{\eta} \frac{\beta}{\eta}: \gamma\) occurs in \(O\).
- Looking up all \textit{binary} rules \(\frac{\beta}{\eta} \frac{\alpha}{\eta}: \gamma\) occurs in \(O\).

\(\alpha\) is given, \(\beta\) and \(\gamma\) need to be found \textit{really} fast.

Requires indexing of both \textit{Closure} and \(O\).
Rules as Functions

\[ R_{\eta} : \gamma \text{ occurs in } \mathcal{O} \rightarrow \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases} \]
Rules as Functions

\[ R_{\eta} : \gamma \text{ occurs in } \mathcal{O} \mapsto \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases} \]

\[ R_{\alpha \eta} : \gamma \text{ occurs in } \mathcal{O} \mapsto R : \alpha \mapsto \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases} \]
Rules as Functions

\[ R_{\eta}^{\gamma} : \gamma \text{ occurs in } \mathcal{O} \rightsquigarrow \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset & \text{otherwise} \end{cases} \]

\[ R_{\eta}^{\alpha} : \gamma \text{ occurs in } \mathcal{O} \rightsquigarrow R : \alpha \mapsto \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset & \text{otherwise} \end{cases} \]

\[ R_{\eta}^{\alpha \beta} : \gamma \text{ occurs in } \mathcal{O} \rightsquigarrow R : \alpha, \beta \mapsto \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset & \text{otherwise} \end{cases} \]
Rules as Functions

\[ R_\eta : \gamma \text{ occurs in } \mathcal{O} \leadsto \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases} \]

\[ R_\alpha^\eta : \gamma \text{ occurs in } \mathcal{O} \leadsto R : \alpha \mapsto \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases} \]

\[ R_\alpha^\beta_\eta : \gamma \text{ occurs in } \mathcal{O} \leadsto R : \alpha, \beta \mapsto \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases} \]

Currying unifies the last two cases

\[ R(\alpha, \beta) = R' : \alpha \mapsto (\beta \mapsto \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases}) \]
Rules as Functions

\[ R_{\eta} : \gamma \text{ occurs in } \mathcal{O} \leadsto \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases} \]

\[ R_{\alpha \eta} : \gamma \text{ occurs in } \mathcal{O} \leadsto R : \alpha \mapsto \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases} \]

\[ R_{\alpha \beta \eta} : \gamma \text{ occurs in } \mathcal{O} \leadsto R : \alpha, \beta \mapsto \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases} \]

Currying unifies the last two cases

\[ R(\alpha, \beta) = R' : \alpha \mapsto \left( \beta \mapsto \begin{cases} \eta, & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases} \right) \]

Rules can be indexed as: \( \alpha \mapsto (\beta \mapsto (\gamma \mapsto \eta)) \)
EL Rules as Functions

\( R_0 : \{ C \sqsubseteq C \} \)

\( R_T : \{ C \sqsubseteq T \} \)
\( \mathcal{EL} \) Rules as Functions

\[ R_0 : \{ C \sqsubseteq C \} \]
\[ R_\top : \{ C \sqsubseteq \top \} \]
\[ R_\land : C \sqsubseteq D_1 \land D_2 \mapsto \{ C \sqsubseteq D_1, C \sqsubseteq D_2 \} \]
**EL** Rules as Functions

\[ R_0 : \{ C \sqsubseteq C \} \]
\[ R_\top : \{ C \sqsubseteq \top \} \]
\[ R_\sqcap : C \sqsubseteq D_1 \sqcap D_2 \mapsto \{ C \sqsubseteq D_1, C \sqsubseteq D_2 \} \]
\[ R_\sqsubseteq : C \sqsubseteq D \mapsto \begin{cases} C \sqsubseteq E, & D \sqsubseteq E \in \mathcal{O} \\ \emptyset & \text{otherwise} \end{cases} \]
**EL Rules as Functions**

\[ R_0 : \{ C \sqsubseteq C \} \]

\[ R_T : \{ C \sqsubseteq \top \} \]

\[ R_{\cap} : C \sqsubseteq D_1 \cap D_2 \mapsto \{ C \sqsubseteq D_1, C \sqsubseteq D_2 \} \]

\[ R_{\sqsubseteq} : C \sqsubseteq D \mapsto \begin{cases} C \sqsubseteq E, & D \sqsubseteq E \in \mathcal{O} \\ \emptyset & \text{otherwise} \end{cases} \]

\[ R_{\sqcup}^+ : C \sqsubseteq D_1, C \sqsubseteq D_2 \mapsto \begin{cases} C \sqsubseteq D_1 \cap D_2, & D_1 \cap D_2 \text{ occurs in } \mathcal{O} \\ \emptyset & \text{otherwise} \end{cases} \]
\[EL\] Rules as Functions

\[R_0 : \{ C \sqsubseteq C \}\]

\[R_T : \{ C \sqsubseteq T \}\]

\[R_\cap : C \sqsubseteq D_1 \cap D_2 \mapsto \{ C \sqsubseteq D_1, C \sqsubseteq D_2 \}\]

\[R_\subseteq : C \sqsubseteq D \mapsto \begin{cases} C \sqsubseteq E, & D \sqsubseteq E \in \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases}\]

\[R_\cup^+ : C \sqsubseteq D_1, C \sqsubseteq D_2 \mapsto \begin{cases} C \sqsubseteq D_1 \cap D_2, & D_1 \cap D_2 \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases}\]

\[R_\exists : E \sqsubseteq \exists R.C, C \sqsubseteq D \mapsto \begin{cases} E \sqsubseteq \exists R.D, & \exists R.D \text{ occurs in } \mathcal{O} \\ \emptyset, & \text{otherwise} \end{cases}\]
\( \mathcal{EL} \) Rule Indexing

\[ R(\alpha, \beta) = R' : \alpha \mapsto \begin{cases} \beta & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset & \text{otherwise} \end{cases} \]

Implement \( \alpha \mapsto (\beta \mapsto (\gamma \mapsto \eta)) \) or \( \alpha \mapsto (\gamma \mapsto \eta) \) for \( \mathcal{EL} \)
**\( \mathcal{EL} \)** Rule Indexing

\[
R(\alpha, \beta) = R' : \alpha \mapsto \begin{cases} 
\beta \mapsto \begin{cases} 
\eta & \gamma \text{ occurs in } O \\
\emptyset & \text{otherwise} 
\end{cases} 
\end{cases}
\]

Implement \( \alpha \mapsto (\beta \mapsto (\gamma \mapsto \eta)) \) or \( \alpha \mapsto (\gamma \mapsto \eta) \) for \( \mathcal{EL} \)

Trivial for \( R_0, R_\top, \) and \( R_\bot \) (no side conditions)
\textbf{EL} Rule Indexing

\[ \mathbf{R}(\alpha, \beta) = \mathbf{R'} : \alpha \mapsto \left( \beta \mapsto \begin{cases} \eta & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset & \text{otherwise} \end{cases} \right) \]

Implement \( \alpha \mapsto (\beta \mapsto (\gamma \mapsto \eta)) \) or \( \alpha \mapsto (\gamma \mapsto \eta) \) for \textbf{EL}

\[ \mathbf{R} \vdash \frac{C \sqsupseteq D}{C \sqsubseteq E} : D \sqsubseteq E \in \mathcal{O} \]

told-subsumers \((\alpha \mapsto \gamma) : D \mapsto \{ E \mid D \sqsubseteq E \in \mathcal{O} \}\)
\( \mathcal{EL} \) Rule Indexing

\[
R(\alpha, \beta) = R' : \alpha \mapsto \begin{cases} \eta & \gamma \text{ occurs in } \mathcal{O} \\ \emptyset & \text{otherwise} \end{cases}
\]

Implement \( \alpha \mapsto (\beta \mapsto (\gamma \mapsto \eta)) \) or \( \alpha \mapsto (\gamma \mapsto \eta) \) for \( \mathcal{EL} \)

\[
R \subseteq \frac{C \sqsubseteq D}{C \sqsubseteq E} : D \sqsubseteq E \in \mathcal{O}
\]

told-subsumers \( (\alpha \mapsto \gamma) : D \mapsto \{ E \mid D \sqsubseteq E \in \mathcal{O} \} \)

When processing \( C \sqsubseteq D \) all \( \{ C \sqsubseteq E \} \) are derived with one look-up

This is rule grouping
Indexing $R^+_\sqcap$

\[
R^+_\sqcap \frac{C \sqsubseteq D_1 \quad C \sqsubseteq D_2}{C \sqsubseteq D_1 \sqcap D_2} : D_1 \sqcap D_2 \text{ occurs in } \mathcal{O}
\]

\[
\alpha = C \sqsubseteq D_1 \quad \beta = C \sqsubseteq D_2 \\
\gamma = D_1 \sqcap D_2 \text{ occurs in } \mathcal{O} \quad \eta = C \sqsubseteq D_1 \sqcap D_2
\]
Indexing $R^+_\sqcap$

$$R^+_\sqcap \frac{C\sqsubseteq D_1 \quad C\sqsubseteq D_2}{C\sqsubseteq D_1 \sqcap D_2} : D_1 \sqcap D_2 \text{ occurs in } \mathcal{O}$$

$$\alpha = C \sqsubseteq D_1 \quad \beta = C \sqsubseteq D_2$$
$$\gamma = D_1 \sqcap D_2 \text{ occurs in } \mathcal{O} \quad \eta = C \sqsubseteq D_1 \sqcap D_2$$

subsumers $\alpha \mapsto \beta$ $C \mapsto \{D_2 \mid C \sqsubseteq D_2 \in \text{Closure}\}$
Indexing $R_\cap^+$

$R_\cap^+ \frac{C \sqsubseteq D_1 \quad C \sqsubseteq D_2}{C \sqsubseteq D_1 \cap D_2} : D_1 \cap D_2 \text{ occurs in } \mathcal{O}$

$\alpha = C \sqsubseteq D_1 \quad \quad \quad \beta = C \sqsubseteq D_2$
$\gamma = D_1 \cap D_2 \text{ occurs in } \mathcal{O} \quad \eta = C \sqsubseteq D_1 \cap D_2$

subsumers $\alpha \mapsto \beta$  $C \mapsto \{D_2 \mid C \sqsubseteq D_2 \in \text{Closure} \}$
conjunctions $\beta \mapsto \gamma$  $D_2 \mapsto \{D_1 \mapsto D_1 \cap D_2 \mid D_1 \cap D_2 \text{ occurs in } \mathcal{O} \}$
Indexing $R^+_\sqcap$

$R^+_\sqcap \frac{C \sqsubseteq D_1}{C \sqsubseteq D_1 \sqcap D_2} : D_1 \sqcap D_2 \text{ occurs in } \mathcal{O}$

$\alpha = C \sqsubseteq D_1$
$\beta = C \sqsubseteq D_2$
$\gamma = D_1 \sqcap D_2 \text{ occurs in } \mathcal{O}$
$\eta = C \sqsubseteq D_1 \sqcap D_2$

subsumers $\alpha \leftrightarrow \beta$
conjunctions $\beta \leftrightarrow \gamma$

Result: $\eta \in \{ C \sqsubseteq D_1 \sqcap D_2 | D_2 \in \text{subsumers}(C), D_1 \in \text{conjunctions}(D_2) \}$
Selecting and Applying $R^+\sqcap$, Example

$R^+\sqcap \frac{C \sqsubseteq D_1 \quad C \sqsubseteq D_2}{C \sqsubseteq D_1 \sqcap D_2} : D_1 \sqcap D_2$ occurs in $\mathcal{O}$

Ontology $\mathcal{O}$  
1. $A \sqsubseteq B$
2. $B \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqcap C \sqsubseteq E$
Selecting and Applying $\mathbf{R}_\sqcap^+$, Example

\[
\begin{align*}
\mathbf{R}_\sqcap^+ & \frac{C \sqsubseteq D_1 \quad C \sqsubseteq D_2}{C \sqsubseteq D_1 \cap D_2} : D_1 \sqcap D_2 \text{ occurs in } \mathcal{O}
\end{align*}
\]

Ontology $\mathcal{O}$

1. $A \sqsubseteq B$
2. $B \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqcap C \sqsubseteq E$

Inferences

$\triangleright$ $A \sqsubseteq B$ by $\mathbf{R}_\sqsubseteq$
Selecting and Applying $\mathcal{R}^+\sqcap$, Example

\[
\mathcal{R}^+\sqcap \frac{C \sqsubseteq D_1 \quad C \sqsubseteq D_2}{C \sqsubseteq D_1 \sqcap D_2} : D_1 \sqcap D_2 \text{ occurs in } \mathcal{O}
\]

Ontology $\mathcal{O}$

1. $A \sqsubseteq B$
2. $B \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqcap C \sqsubseteq E$

Inferences

- $A \sqsubseteq B$ by $\mathcal{R}\sqsubseteq$
- $A \sqsubseteq D$ by $\mathcal{R}\sqsubseteq$
- $\{ C \mapsto C \sqcap D \} \in \text{conjunctions}(D)$
- but $C \not\in \text{subsumers}(A)$, $\mathcal{R}^+\sqcap$ doesn’t apply
Selecting and Applying $R^+_\sqcap$, Example

\[ R^+_\sqcap \frac{C \sqsubseteq D_1 \quad C \sqsubseteq D_2}{C \sqsubseteq D_1 \sqcap D_2} : D_1 \sqcap D_2 \text{ occurs in } O \]

Ontology $O$

1. $A \sqsubseteq B$
2. $B \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqcap C \sqsubseteq E$

Inferences

- $A \sqsubseteq B$ by $R_\sqsubseteq$
- $A \sqsubseteq D$ by $R_\sqsubseteq$
- $\{C \mapsto C \sqcap D\} \in \text{conjunctions}(D)$
  but $C \notin \text{subsumers}(A)$, $R^+_\sqcap$ doesn't apply
- $A \sqsubseteq C$ by $R_\sqsubseteq$
  $\{D \mapsto C \sqcap D\} \in \text{conjunctions}(C)$
  and $D \in \text{subsumers}(A)$, sooo
Selecting and Applying $R^+_n$, Example

$R^+_n \frac{C \sqsubseteq D_1 \land C \sqsubseteq D_2}{C \sqsubseteq D_1 \cap D_2} : D_1 \sqcap D_2$ occurs in $O$

Ontology $O$

1. $A \sqsubseteq B$
2. $B \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqcap C \sqsubseteq E$

Inferences

- $A \sqsubseteq B$ by $R^<_n$
- $A \sqsubseteq D$ by $R^<_n$
- $\{C \mapsto C \cap D\} \in \text{conjunctions}(D)$
  but $C \notin \text{subsumers}(A)$, $R^+_n$ doesn’t apply
- $A \sqsubseteq C$ by $R^<_n$
  $\{D \mapsto C \cap D\} \in \text{conjunctions}(C)$
  and $D \in \text{subsumers}(A)$, sooo
- $A \sqsubseteq C \cap D$ by $R^+_n$
Selecting and Applying $\mathbf{R}^+_\sqcap$, Example

\[ \mathbf{R}^+_\sqcap \frac{C \sqsubseteq D_1 \quad C \sqsubseteq D_2}{C \sqsubseteq D_1 \sqcap D_2} : D_1 \sqcap D_2 \text{ occurs in } \mathcal{O} \]

Ontology $\mathcal{O}$

1. $A \sqsubseteq B$
2. $B \sqsubseteq C$
3. $A \sqsubseteq D$
4. $D \sqcap C \sqsubseteq E$

Inferences

- $A \sqsubseteq B$ by $\mathbf{R}^+_\sqsubseteq$
- $A \sqsubseteq D$ by $\mathbf{R}^+_\sqsubseteq$
- \( \{ C \mapsto C \sqcap D \} \in \text{conjunctions}(D) \)
- but $C \not\in \text{subsumers}(A)$, $\mathbf{R}^+_\sqcap$ doesn’t apply
- $A \sqsubseteq C$ by $\mathbf{R}^+_\sqsubseteq$
- \( \{ D \mapsto C \sqcap D \} \in \text{conjunctions}(C) \)
- and $D \in \text{subsumers}(A)$, sooo
- $A \sqsubseteq C \sqcap D$ by $\mathbf{R}^+_\sqcap$
- $A \sqsubseteq E$ by $\mathbf{R}^+_\sqsubseteq$
Existential-based Indexing $R_\exists$

$R_\exists \frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C}{A \sqsubseteq \exists R.C} : \exists R.C \text{ occurs in } O$

Processing $E \sqsubseteq \exists R.C$

\[
\begin{align*}
\alpha &= A \sqsubseteq \exists R.B \\
\beta &= B \sqsubseteq C \\
\gamma &= \exists R.C \text{ occurs in } O \\
\eta &= A \sqsubseteq \exists R.C
\end{align*}
\]
Existential-based Indexing $R_\exists$

$$R_\exists \frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C}{A \sqsubseteq \exists R.C} : \exists R.C \text{ occurs in } \mathcal{O}$$

Processing $E \sqsubseteq \exists R.C$

$$\begin{align*}
\alpha &= A \sqsubseteq \exists R.B \\
\beta &= B \sqsubseteq C \\
\gamma &= \exists R.C \text{ occurs in } \mathcal{O} \\
\eta &= A \sqsubseteq \exists R.C
\end{align*}$$

subsumptions $\alpha \mapsto \beta \quad B \mapsto \{ C \mid B \sqsubseteq C \in \text{Closure} \}$
Existential-based Indexing $R_\exists$

$$R_\exists \quad \frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C}{A \sqsubseteq \exists R.C} : \exists R.C \text{ occurs in } \mathcal{O}$$

Processing $E \sqsubseteq \exists R.C$

$$\alpha = A \sqsubseteq \exists R.B \quad \beta = B \sqsubseteq C$$

$$\gamma = \exists R.C \text{ occurs in } \mathcal{O} \quad \eta = A \sqsubseteq \exists R.C$$

subsumptions $\alpha \mapsto \beta$ $B \mapsto \{ C \mid B \sqsubseteq C \in \text{Closure} \}$

existentials $\beta \mapsto \gamma$ $C \mapsto \{ R \mid \exists R.C \text{ occurs in } \mathcal{O} \}$
Existential-based Indexing $R_\exists$

\[
R_\exists \quad \frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C}{A \sqsubseteq \exists R.C} \quad : \quad \exists R.C \text{ occurs in } \mathcal{O}
\]

Processing $E \sqsubseteq \exists R.C$

\[
\alpha = A \sqsubseteq \exists R.B \quad \beta = B \sqsubseteq C
\]
\[
\gamma = \exists R.C \text{ occurs in } \mathcal{O} \quad \eta = A \sqsubseteq \exists R.C
\]

Subsumptions $\alpha \mapsto \beta \quad B \mapsto \{ C \mid B \sqsubseteq C \in \text{Closure} \}$

Existentials $\beta \mapsto \gamma \quad C \mapsto \{ R \mid \exists R.C \text{ occurs in } \mathcal{O} \}$

Result $\eta \in \{ A \sqsubseteq \exists R.C \mid C \in \text{subsumptions}(B), \quad R \in \text{existentials}(C) \}$
Subsumption-based Indexing $R_\exists$

$$R_\exists \frac{B \sqsubseteq C \quad A \sqsubseteq \exists R.B}{A \sqsubseteq \exists R.C} : \exists R.C \text{ occurs in } \mathcal{O}$$

Processing $B \sqsubseteq C$

$$\alpha = B \sqsubseteq C \quad \beta = A \sqsubseteq \exists R.B \quad \gamma = \exists R.C \text{ occurs in } \mathcal{O} \quad \eta = A \sqsubseteq \exists R.C$$
Subsumption-based Indexing $R_\exists$

$R_\exists \frac{B \sqsubseteq C \quad A \sqsubseteq \exists R.B}{A \sqsubseteq \exists R.C} : \exists R.C \text{ occurs in } \mathcal{O}$

Processing $B \sqsubseteq C$

$\alpha = B \sqsubseteq C \quad \beta = A \sqsubseteq \exists R.B$

$\gamma = \exists R.C \text{ occurs in } \mathcal{O} \quad \eta = A \sqsubseteq \exists R.C$

backward-links $\alpha \hookrightarrow \beta \quad B \hookrightarrow \{ R \hookrightarrow A \mid A \sqsubseteq \exists R.B \in \text{Closure} \}$
Subsumption-based Indexing $R_\exists$

$R_\exists \begin{array}{c} B \sqsubseteq C \\ A \sqsubseteq \exists R.B \end{array} : \exists R.C \text{ occurs in } \mathcal{O}$

Processing $B \sqsubseteq C$

$$\alpha = B \sqsubseteq C \quad \beta = A \sqsubseteq \exists R.B$$

$$\gamma = \exists R.C \text{ occurs in } \mathcal{O} \quad \eta = A \sqsubseteq \exists R.C$$

backward-links $\alpha \mapsto \beta \quad B \mapsto \{ R \mapsto A \mid A \sqsubseteq \exists R.B \in \text{Closure} \}$

existentials $\beta \mapsto \gamma \quad C \mapsto \{ R \mid \exists R.C \text{ occurs in } \mathcal{O} \}$
Subsumption-based Indexing $R_\exists$

$$R_\exists \frac{B \sqsubseteq C}{A \sqsubseteq \exists R.C} : \exists R.C \text{ occurs in } \mathcal{O}$$

Processing $B \sqsubseteq C$

$$\alpha = B \sqsubseteq C \quad \beta = A \sqsubseteq \exists R.B$$
$$\gamma = \exists R.C \text{ occurs in } \mathcal{O} \quad \eta = A \sqsubseteq \exists R.C$$

backward-links  \hspace{1cm} \alpha \mapsto \beta \quad B \mapsto \{ R \mapsto A \mid A \sqsubseteq \exists R.B \in \text{Closure} \}
existentials  \hspace{1cm} \beta \mapsto \gamma \quad C \mapsto \{ R \mid \exists R.C \text{ occurs in } \mathcal{O} \}

Result  \hspace{1cm} \eta \in \{ A \sqsubseteq \exists R.C \mid \{ A \in \text{backward-links}(B, R) \}, R \in \text{existentials}(C) \}$$
Selecting and Applying $R_\exists$, Example

$R_\exists \frac{E \sqsubseteq \exists R.C \quad C \sqsubseteq D}{E \sqsubseteq \exists R.D}$: $E \sqsubseteq \exists R.D$ occurs in $\mathcal{O}$

Ontology $\mathcal{O}$: $A \sqsubseteq B$ $B \sqsubseteq \exists R.(C \cap D)$ $C \sqsubseteq E$ $\exists R.E \sqsubseteq X$

Closure:
Selecting and Applying $R_\exists$, Example

$R_\exists \frac{E \subseteq \exists R. C \quad C \subseteq D}{E \subseteq \exists R. D}$: $E \subseteq \exists R. D$ occurs in $\mathcal{O}$

Ontology $\mathcal{O}$: $A \sqsubseteq B$ $B \sqsubseteq \exists R.(C \cap D)$ $C \subseteq E$ $\exists R.E \subseteq X$

Closure:

$A \sqsubseteq B$ by $R_{\subseteq}$
Selecting and Applying $R_{\exists}$, Example

$R_{\exists} \frac{E \sqsubseteq \exists R.C}{E \sqsubseteq \exists R.D} : E \sqsubseteq \exists R.D$ occurs in $O$

Ontology $O$: $A \sqsubseteq B$ $B \sqsubseteq \exists R.(C \cap D)$ $C \sqsubseteq E$ $\exists R.E \sqsubseteq X$

Closure:

$A \sqsubseteq B$ by $R_{\sqsubseteq}$

$A \sqsubseteq \exists R.(C \cap D)$ by $R_{\sqsubseteq}$ using $B \sqsubseteq \exists R.(C \cap D)$
Selecting and Applying $R_\exists$, Example

\[
R_\exists \frac{E \subseteq \exists R.C \quad C \subseteq D}{E \subseteq \exists R.D} : E \subseteq \exists R.D \text{ occurs in } \mathcal{O}
\]

Ontology $\mathcal{O}$: \quad A \subseteq B \quad B \subseteq \exists R.(C \cap D) \quad C \subseteq E \quad \exists R.E \subseteq X

Closure:
\[
\begin{align*}
A & \subseteq B \quad \text{by } R_{\subseteq} \\
A & \subseteq \exists R.(C \cap D) \quad \text{by } R_{\subseteq} \text{ using } B \subseteq \exists R.(C \cap D) \\
& \quad \text{A added to backward-links}(C \cap D, R)
\end{align*}
\]
Selecting and Applying $R_\exists$, Example

$R_\exists \frac{E \sqsubseteq \exists R. C \quad C \sqsubseteq D}{E \sqsubseteq \exists R. D}$: $E \sqsubseteq \exists R. D$ occurs in $\mathcal{O}$

Ontology $\mathcal{O}$: $A \sqsubseteq B$ $B \sqsubseteq \exists R. (C \cap D)$ $C \sqsubseteq E$ $\exists R. E \sqsubseteq X$

Closure:

$A \sqsubseteq B$ by $R_\subseteq$

$A \sqsubseteq \exists R. (C \cap D)$ by $R_\subseteq$ using $B \sqsubseteq \exists R. (C \cap D)$

$A$ added to backward-links($C \cap D$, $R$)

$C \cap D \sqsubseteq C \cap D$ by $R_0$
Selecting and Applying $R_\exists$, Example

$R_\exists \frac{E \subseteq \exists R.C}{E \subseteq \exists R.D} : E \subseteq \exists R.D$ occurs in $\mathcal{O}$

Ontology $\mathcal{O}$: $A \subseteq B$ $B \subseteq \exists R.(C \land D)$ $C \subseteq E$ $\exists R.E \subseteq X$

Closure:

$A \subseteq B$ by $R_\subseteq$

$A \subseteq \exists R.(C \land D)$ by $R_\subseteq$ using $B \subseteq \exists R.(C \land D)$

$A$ added to $\text{backward-links}(C \land D, R)$

$C \land D \subseteq C \land D$ by $R_0$

$C \land D \subseteq C$ by $R_\land$
Selecting and Applying $R_\exists$, Example

$R_\exists \hspace{1em} \frac{E \sqsubseteq \exists R.C \quad C \subseteq D}{E \sqsubseteq \exists R.D} : E \sqsubseteq \exists R.D$ occurs in $\mathcal{O}$

Ontology $\mathcal{O}$: $A \subseteq B \quad B \sqsubseteq \exists R.(C \cap D) \quad C \subseteq E \quad \exists R.E \subseteq X$

Closure:

- $A \subseteq B$ by $R_c$
- $A \subseteq \exists R.(C \cap D)$ by $R_c$ using $B \subseteq \exists R.(C \cap D)$
- $A$ added to $\text{backward-links}(\mathcal{C} \cap D, R)$
- $C \cap D \subseteq C \cap D$ by $R_0$
- $C \cap D \subseteq C$ by $R_\cap$
- $A \subseteq \exists R.C$ ???
Selecting and Applying $\mathbf{R}_\exists$, Example

$$\frac{E \subseteq \exists R.C \quad C \subseteq D}{E \subseteq \exists R.D} : E \subseteq \exists R.D \text{ occurs in } \mathcal{O}$$

Ontology $\mathcal{O}$: \quad A \subseteq B \quad B \subseteq \exists R.(C \cap D) \quad C \subseteq E \quad \exists R.E \subseteq X$

Closure:

- \quad A \subseteq B \quad \text{by} \quad \mathbf{R} \subseteq
- \quad A \subseteq \exists R.(C \cap D) \quad \text{by} \quad \mathbf{R} \subseteq \text{ using } B \subseteq \exists R.(C \cap D)
- \quad A \text{ added to backward-links}(C \cap D, R)
- \quad C \cap D \subseteq C \cap D \quad \text{by} \quad \mathbf{R}_0
- \quad C \cap D \subseteq C \quad \text{by} \quad \mathbf{R}_-$
- \quad A \subseteq \exists R.C \quad ??? \quad \text{No} \quad R \notin \text{existentials}(C)$
Selecting and Applying $R_\exists$, Example

$R_\exists \frac { E \subseteq \exists R \cdot C \quad C \subseteq D } { E \subseteq \exists R \cdot D }$: $E \subseteq \exists R \cdot D$ occurs in $O$

Ontology $O$: $A \subseteq B \quad B \subseteq \exists R \cdot (C \cap D) \quad C \subseteq E \quad \exists R \cdot E \subseteq X$

Closure:

- $A \subseteq B$ by $R_\subseteq$
- $A \subseteq \exists R \cdot (C \cap D)$ by $R_\subseteq$ using $B \subseteq \exists R \cdot (C \cap D)$
  - $A$ added to backward-links($C \cap D$, $R$)
- $C \cap D \subseteq C \cap D$ by $R_0$
- $C \cap D \subseteq C$ by $R_\cap$
- $A \subseteq \exists R \cdot C$ ??? No $R \notin$ existentials($C$)
- $C \cap D \subseteq E$ by $R_\subseteq$
Selecting and Applying $\mathbf{R}_\exists$, Example

\[
\begin{array}{c}
\text{R}_\exists \quad \frac{E \subseteq \exists R.C}{E \subseteq \exists R.D} : E \subseteq \exists R.D \text{ occurs in } \mathcal{O}
\end{array}
\]

Ontology $\mathcal{O}$:
\[
\begin{array}{llll}
A & \subseteq & B & \\
B & \subseteq & \exists R.(C \cap D) & \\
C & \subseteq & E & \\
\exists R.E & \subseteq & X
\end{array}
\]

Closure:
\[
\begin{array}{ll}
A & \subseteq B \\
A & \subseteq \exists R.(C \cap D) \\
C \cap D & \subseteq C \cap D \\
C \cap D & \subseteq C \\
A & \subseteq \exists R.C \\
C \cap D & \subseteq E \\
A & \subseteq \exists R.E
\end{array}
\]

by
\[
\begin{array}{ll}
R \subseteq \\
R \subseteq \text{ using } B \subseteq \exists R.(C \cap D) \\
R_0 \\
R_\cap \\
\exists \text{ existentials}(C) \\
R \subseteq
\end{array}
\]
Selecting and Applying $\mathbf{R}_\exists$, Example

$$\frac{E \sqsubseteq \exists R.C \quad C \sqsubseteq D}{E \sqsubseteq \exists R.D} : E \sqsubseteq \exists R.D \text{ occurs in } \mathcal{O}$$

Ontology $\mathcal{O}$: $A \sqsubseteq B$, $B \sqsubseteq \exists R.(C \cap D)$, $C \sqsubseteq E$, $\exists R.E \sqsubseteq X$

Closure:

- $A \sqsubseteq B$ by $\mathbf{R}_\subseteq$
- $A \sqsubseteq \exists R.(C \cap D)$ by $\mathbf{R}_\subseteq$ using $B \sqsubseteq \exists R.(C \cap D)$
- $A$ added to backward-links($C \cap D, R$)
- $C \cap D \sqsubseteq C \cap D$ by $\mathbf{R}_0$
- $C \cap D \sqsubseteq C$ by $\mathbf{R}_-$
- $A \sqsubseteq \exists R.C$ ??? No $\mathbf{R} \notin$ existentials($C$)
- $C \cap D \sqsubseteq E$ by $\mathbf{R}_\subseteq$
- $A \sqsubseteq \exists R.E$ ??? Yes $\mathbf{R} \in$ existentials($E$), $A \in$ backward-links($C \cap D, R$)
Selecting and Applying $\mathbf{R}_\exists$, Example

$$\frac{E \sqsubseteq \exists R.C \quad C \sqsubseteq D}{E \sqsubseteq \exists R.D} : E \sqsubseteq \exists R.D \text{ occurs in } \mathcal{O}$$

Ontology $\mathcal{O}$: $A \sqsubseteq B \quad B \sqsubseteq \exists R.(C \sqcap D) \quad C \sqsubseteq E \quad \exists R.E \sqsubseteq X$

Closure:

$A \sqsubseteq B$ by $\mathbf{R}_\subseteq$

$A \sqsubseteq \exists R.(C \sqcap D)$ by $\mathbf{R}_\subseteq$ using $B \sqsubseteq \exists R.(C \sqcap D)$

$A$ added to backward-links$(C \sqcap D, R)$

$C \sqcap D \sqsubseteq C \sqcap D$ by $\mathbf{R}_0$

$C \sqcap D \sqsubseteq C$ by $\mathbf{R}_\ominus$

$A \sqsubseteq \exists R.C$ ??? No $\mathbf{R} \notin$ existentials$(C)$

$C \sqcap D \sqsubseteq E$ by $\mathbf{R}_\subseteq$

$A \sqsubseteq \exists R.E$ ??? Yes $\mathbf{R} \in$ existentials$(E)$,

$A \in$ backward-links$(C \sqcap D, R)$

$A \sqsubseteq X$ by $\mathbf{R}_\subseteq$ using $\exists R.E \sqsubseteq X$
So Is It Practical?

Short answer: yes

- \(<10s\) to classify SNOMED CT (\(>200s\) for tableau)
- 10s for GALEN (\(\infty\) for tableau)
So Is It Practical?

Short answer: yes

- <10s to classify SNOMED CT (>200s for tableau)
- 10s for GALEN (∞ for tableau)
- 1hr to classify SNOMED CT on Google Nexus! 😊
So Is It Practical?

Short answer: yes

- <$10\text{s}$ to classify SNOMED CT ($>200\text{s}$ for tableau)
- $10\text{s}$ for GALEN ($\infty$ for tableau)
- $1\text{hr}$ to classify SNOMED CT on Google Nexus! 😊

There is still room for improvement

- around $23,000,000$ inferences made to classify SNOMED CT
- ... but only $300,000$ concepts, few subsumers per each
- even more economical classification might be possible
Take Home Message

Consequence-based reasoning is different from tableau reasoning

- Uses natural deduction (rules) instead of building a model
- Never tries to test a subsumption that doesn’t hold
Take Home Message

Consequence-based reasoning is different from tableau reasoning

- Uses natural deduction (rules) instead of building a model
- Never tries to test a subsumption that doesn’t hold

Sound and complete rule systems known for

- $\mathcal{EL}$, $\mathcal{EL}^+$, $\mathcal{EL}^{++}$ (this lecture)
- Horn-$\mathcal{SHIQ}$ (the language of Full GALEN)
- $\mathcal{ALCH}$ (non-deterministic language)
Take Home Message

Consequence-based reasoning is different from tableau reasoning

- Uses natural deduction (rules) instead of building a model
- Never tries to test a subsumption that doesn’t hold

Sound and complete rule systems known for

- $\mathcal{EL}$, $\mathcal{EL}^+$, $\mathcal{EL}^{++}$ (this lecture)
- Horn-$\mathcal{SHIQ}$ (the language of Full GALEN)
- $\mathcal{ALCH}$ (non-deterministic language)

Tractable does not necessarily mean practical!

- Even $O(n^2)$ is fatal if it is typical case
- Converse: intractable does not always mean impractical