Reasoning with Expressive Description Logics

Ian Horrocks

Department of Computer Science

University of Manchester, UK

horrocks@cs.man.ac.uk

Motivation

DLs have proved useful in wide range of applications

- Configuration
- DataBases
  - Schema design and query optimisation
  - Interoperability and federation
- Software information systems
- Ontological engineering

Wider acceptance in **realistic applications** hindered by conflict between

- **Expressiveness** of logic
  - Need powerful concept description language
- **Tractability** of subsumption reasoning
  - Generate large KBs and many 1,000s of subsumption tests

Example of such an application is the GALEN project

- Aims to promote sharing and re-use of medical data
- Using DL to construct large medical terminology KB
- KB can be used by applications as flexible and extensible classification schema

KB constructed using DL which supports:

- Transitive roles for anatomical partonomy, transitive causality etc.

  e.g. given axioms

$$\begin{aligned} \text{Smoking} &\sqsubseteq \exists \mathit{causes}.\text{Cancer} \\ \text{Cancer} &\sqsubseteq \exists \mathit{causes}.\text{Death} \end{aligned}$$

  want to infer

$$\text{Smoking} \sqsubseteq \exists \mathit{causes}.\text{Death}$$

- General Concept Inclusion axioms (GCIs) represent additional non-definitional knowledge

  e.g. want to *define*

$$\text{Stomach-Ulcer} \doteq \text{Ulcer} \sqcap \exists \textit{location}.\text{Stomach}$$

  but also add *knowledge* that stomach ulcers *always* occur in lining of stomach

$$\text{Stomach-Ulcer} \sqsubseteq \exists \textit{location}.\text{Lining-Of-Stomach}$$

- Hierarchy of sub and super roles

e.g. given a role *division*, would like to define more specific sub-roles

linear-division and *blind-pouch-division*

such that

$$\begin{aligned} \textit{linear-division} &\sqsubseteq \textit{division} \\ \textit{blind-pouch-division} &\sqsubseteq \textit{division} \end{aligned}$$

so we can state

$$\begin{aligned} \text{Transverse-Colon} &\sqsubseteq \exists \textit{linear-division}.\text{Large-Intestine} \\ \text{Appendix} &\sqsubseteq \exists \textit{blind-pouch-division}.\text{Large-Intestine} \end{aligned}$$

and still infer that both are *division*s of Large-Intestine

Would also *like* to have:

- Inverse (converse) roles

e.g. given role *caused-by*, such that

$$\textit{caused-by} \doteq \textit{causes}^{-}$$

and axiom

$$\text{Death} \sqcap \exists \textit{caused-by}.\text{Smoking} \sqsubseteq \text{Premature-Death}$$

want to infer

$$\text{Smoking} \sqsubseteq \exists \textit{causes}.\text{Premature-Death}$$

- Cardinality restrictions on roles

  e.g. given axioms

$$\begin{aligned} \text{Blood-Pressure} &\sqsubseteq \exists \mathit{value}.(\text{High} \sqcup \text{Low}) \sqcap \leqslant 1 \mathit{value} \\ \text{High} &\sqsubseteq \neg \text{Low} \end{aligned}$$

  want to infer

$$\text{Blood-Pressure} \sqcap \exists \mathit{value}.\text{High} \sqcap \exists \mathit{value}.\text{Low} \quad \sqsubseteq \quad \bot$$

FaCT DL Classifier

Subsumption reasoning known to be highly intractable

- Existing classifier uses structural subsumption reasoning
  - Relatively efficient
  - Incomplete
- Goal of **FaCT** system to provide
  - Similar (preferably greater) expressive power
  - Sound and complete reasoning
  - Empirical tractability

- Logic used in FaCT is based on $\mathcal{ALC}_{R^+}$ DL
  - $\mathcal{ALC}$ concept forming operators ($\sqcap$, $\sqcup$, $\neg$, $\forall$, $\exists$)
    i.e., standard boolean connectives plus (multi-modal) $\Box$ and $\Diamond$
  - Transitive roles (**not** transitive closure operator)
    e.g., *ancestor* $\in \mathbf{R}_+ \Rightarrow$ *ancestor* is **always** transitive
  - Complexity of subsumption problem is in $\textsc{PSpace}$
- Augmented with role hierarchy to give $\mathcal{ALCH}_{R^+}$
  - Increases subsumption complexity to $\textsc{ExpTime}$

Standard Tarski model theoretic semantics

Syntax	Semantics
$\top$	$\Delta^{\mathcal{I}}$
$\bot$	$\emptyset$
C	$C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
$\neg C$	$\Delta^{\mathcal{I}} - C^{\mathcal{I}}$
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\exists R.C$	$\{d \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(d) \cap C^{\mathcal{I}} \neq \emptyset\}$
$\forall R.C$	$\{d \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(d) \subseteq C^{\mathcal{I}}\}$

KB is set of axioms of form:

Syntax	Semantics
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$C \doteq D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$
$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
$R \in \mathbf{R}_+$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^+$

- $C \sqsubseteq D$ and $C \doteq D$ axioms are concept definitions or GCIs
  - note $C \doteq D$ equivalent to $C \sqsubseteq D$ plus $D \sqsubseteq C$
- $R \sqsubseteq S$ axioms define partial order on roles
- $R \in \mathbf{R}_+$ axioms define set of transitive role names $\mathbf{R}_+$

Subsumption Reasoning

- Subsumption transformed to satisfiability in usual way
  - C subsumes D iff $D \sqcap \neg C$ not satisfiable
- Tableaux algorithm used to test satisfiability
  - Tries to build model of $D \sqcap \neg C$
  - Or demonstrate contradictions (there is no model)

- Model represented by tree $\mathbf{T}$
  - Nodes correspond to individuals in model
  - Nodes labeled with sets of subconcepts of $D \sqcap \neg C$
  - Edges labeled with role names
- Starts from root node labeled $\{D \sqcap \neg C\}$
- Applies expansion rules to node labels until
  - Expansion completed (tree represents valid model)
  - Contradictions prove there is no model (e.g., some individual must satisfy both C and $\neg C$ for some concept C)

Simple $\sqcap$ and $\sqcup$-rules deal with propositional reasoning:

if $C_1 \sqcap C_2 \in \mathcal{L}(x)$

then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{C_1, C_2\}$

if $C_1 \sqcup C_2 \in \mathcal{L}(x)$

then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{C\}$ for some $C \in \{C_1, C_2\}$

note: 1. negations internalised until they only apply to atoms

2. rules only applicable if they extend labeling

e.g. for $\sqcap$-rule, $\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$

basic $\exists$-rule extended to deal with hierarchy and cycles:

if 1. $\exists R.C \in \mathcal{L}(x)$

2. no S-successor y of x with $S \sqsubseteq R$ and $C \in \mathcal{L}(y)$

3. x is not **blocked**

then create new R-successor z of x with $\mathcal{L}(z) = \{C\}$

note x is **blocked** (cyclical model) if for some ancestor w of x:

$$\mathcal{L}(x) \subseteq \mathcal{L}(w)$$

basic $\forall$-rule extended to deal with hierarchy and transitivity:

if $\forall R.C \in \mathcal{L}(x)$, y is an S-successor of x, and $S \sqsubseteq R$

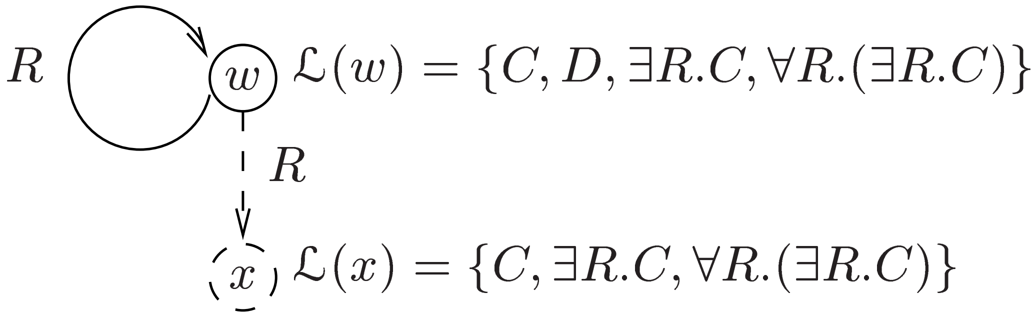
then 1. $\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{C\}$

2. $\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{\forall P.D \mid P \in \mathbf{R}_+,\ S \sqsubseteq P \text{ and } P \sqsubseteq R\}$

Simple non-transitive role ($R = S$, $R \notin \mathbf{R}_+$): $\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{C\}$

Simple transitive role ($R = S$, $R \in \mathbf{R}_+$): $\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{C, \forall R.C\}$

Example of blocking (for $R \in \mathbf{R}_+$):



Expansion of $\exists R.C \in \mathcal{L}(x)$ is blocked because $\mathcal{L}(x) \subseteq \mathcal{L}(w)$

$\Rightarrow$ cyclical model where x identified with w

Transitive Roles -v- Transitive Closure

- In $\mathcal{ALCH}_{R^+}$:
  - $\exists R.C$ deterministically expanded, even when $R \in \mathbf{R}_+$
  - Cycles **always** represent (part of) valid cyclical models
- In $\mathcal{ALC}^+$
  - $\exists R^+.C$ non-deterministically expanded to $\exists R.C$ *or* $\exists R.(\exists R^+.C)$
  - Cycles **either** represent valid cyclical models (*good cycles*)
  - **or** contradictions (*bad cycles*), e.g. $\exists R^+.C \sqcap \forall R^+.\neg C$

Although worst-case complexity in EXPTIME, $\mathcal{ALCH}_{R^+}$ algorithm:

- Simple to implement
- Amenable to optimization
- Behaves well in realistic applications

Would like to extend these benifits to more expressive logics

Converse Roles

Extend language with converse role forming operator:

- Role is now P or P^-, where P is a role name
- $(P^-)^\mathcal{I} = \{\langle x, y\rangle \mid \langle y, x\rangle \in P^\mathcal{I}\}$
- $P^{--} \equiv P$
- P^- is transitive iff P is transitive

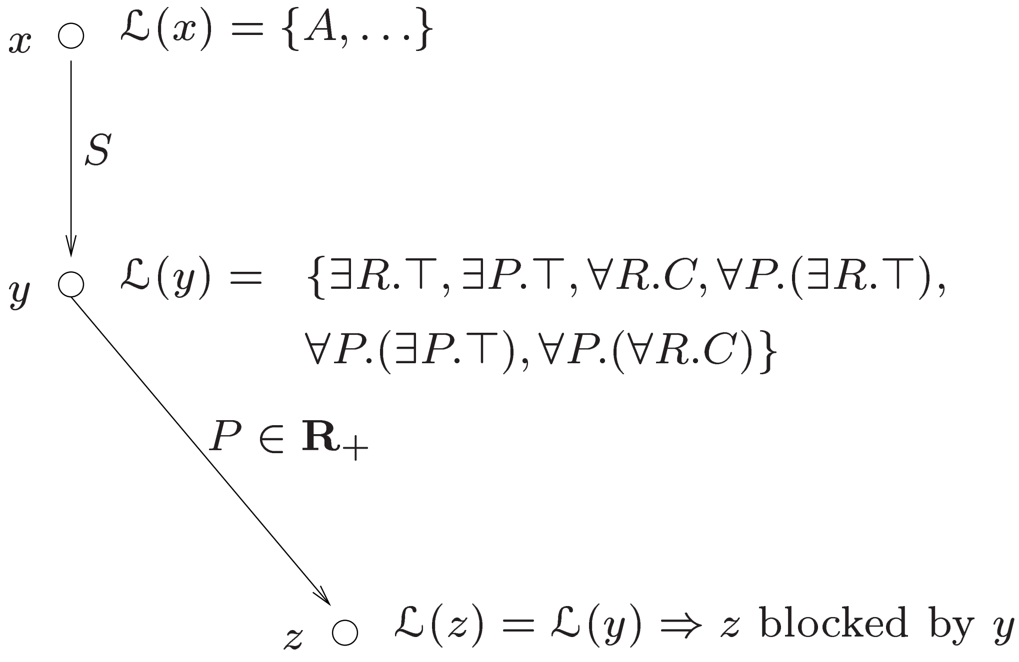
Reasoning with converse roles more problematical:

- Expanding node label can now affect predecessor
- Can no longer treat nodes as independent sub-problems
- Simple sub-set blocking no longer works
- Static (once and for all) blocking no longer works

e.g. tableau for $A \sqcap \exists S.(\exists R.\top \sqcap \exists P.\top \sqcap \forall R.C \sqcap \forall P.(\exists R.\top) \sqcap$
$\forall P.(\exists P.\top) \sqcap \forall P.(\forall R.C))$

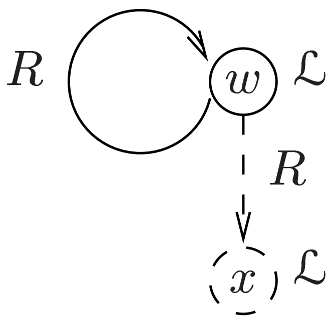
where $C \doteq \forall R^{-}.(\forall P^{-}.(\forall S^{-}.\neg A))$

and $P \in \mathbf{R}_{+}$


x
$\mathcal{L}(x) = \{A, \ldots\}$
S
y
$\mathcal{L}(y) = \{\exists R.\top, \exists P.\top, \forall R.C, \forall P.(\exists R.\top), \forall P.(\exists P.\top), \forall P.(\forall R.C)\}$
$P \in \mathbf{R}_+$
z
$\mathcal{L}(z) = \mathcal{L}(y) \Rightarrow z$ blocked by y

Solution is to use **dynamic** blocking:

- Blocks can be established, broken and re-established
- Continue to expand $\forall R.C$ concepts in blocked nodes
- Check that cycles satisfy $\forall R^{-}.C$ concepts:

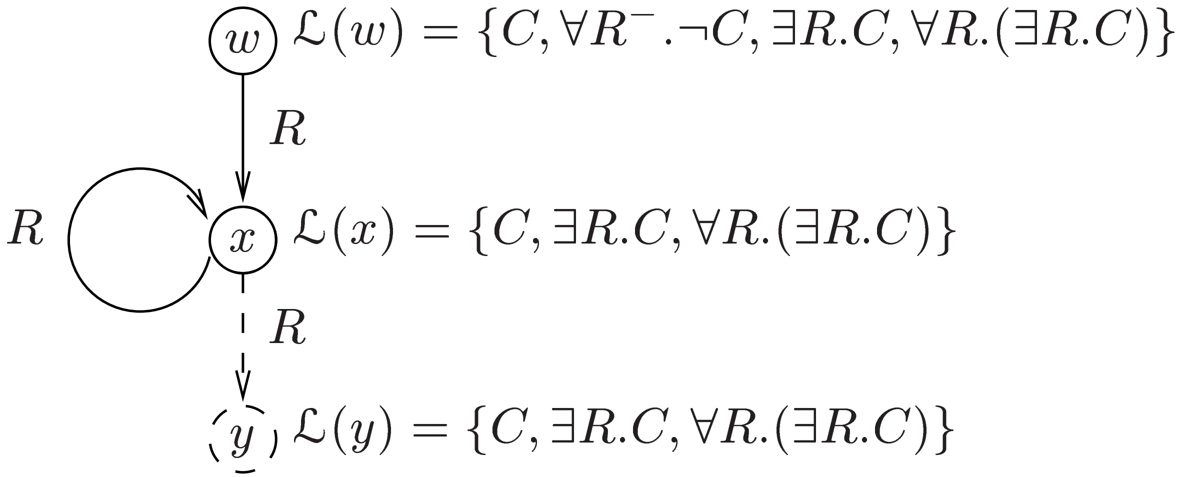


$$\mathcal{L}(w) = \{C, \forall R^{-}.\neg C, \exists R.C, \forall R.(\exists R.C)\}$$

$$\mathcal{L}(x) = \{C, \exists R.C, \forall R.(\exists R.C)\}$$

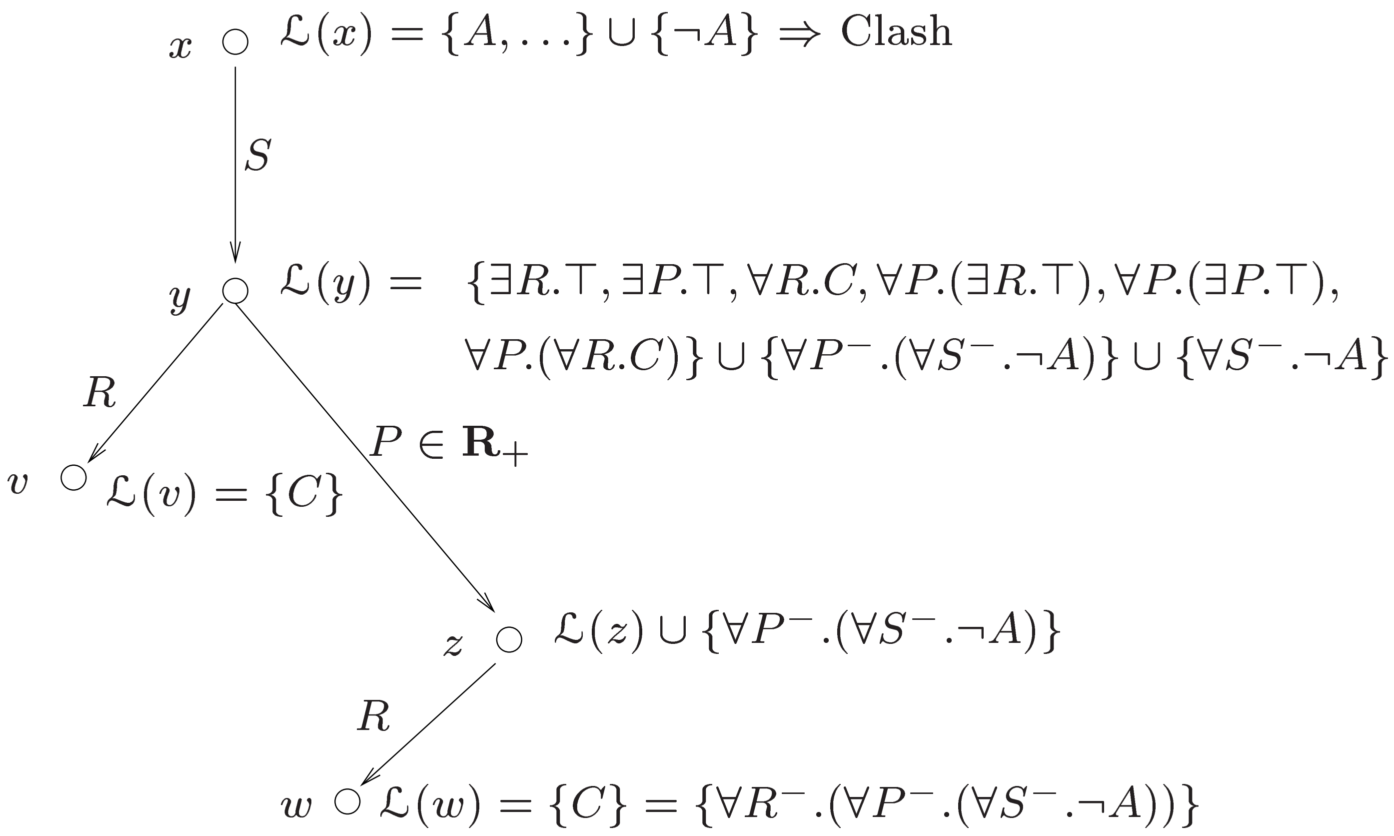
Cyclical model not valid because $\forall R^{-}\neg C \in \mathcal{L}(w) \longrightarrow \{C, \neg C\} \subseteq \mathcal{L}(w)$

$\forall R^{-}.C$ concepts in blocking node must be satisfied in blocked node:



Expansion continues and valid model discovered with $y = x$

In earlier example, z un-blocked by $\forall P^{-}.(\forall S^{-}.\neg A)$ added to $\mathcal{L}(y)$ from C in $\mathcal{L}(v)$



Clash now discovered due to C in $\mathcal{L}(w)$

Transitive Roles -v- Transitive Closure

- In $\mathcal{ALCHI}_{R^+}$:
  - $\exists R.C$ concepts have same deterministic expansion as $\mathcal{ALCH}_{R^+}$
  - Blocking condition slightly more complex
- In $\mathcal{ALCI}^+$:
  - $\exists R^+.C$ concepts non-deterministically expanded as in $\mathcal{ALC}^+$
  - Still have good and bad cycles
  - But now difficult to be sure cycle *really* bad

e.g. tableau for $\neg D \sqcap \exists R.(\exists R^{+}.C)$

where $C \doteq \forall R^{-}.(\forall R^{-}.D)$

$$\begin{tikzcd} w \arrow[d, "R"] & \mathcal{L}(w) = \{\neg D, \exists R.(\exists R^{+}.C)\} \\ x \arrow[d, "R"] & \mathcal{L}(x) = \{\exists R^{+}.C\} \\ y & \mathcal{L}(y) = \{\exists R^{+}.C\} \end{tikzcd}$$

$\mathcal{L}(y) = \mathcal{L}(x) \Rightarrow$ block, and $\exists R^{+}.C \in \mathcal{L}(x)$ never satisfied $\Rightarrow$ bad-cycle

but. . .

If expansion continued one more step:

$$
\begin{tikzcd}
w \arrow[d, "R"] & \mathcal{L}(w) = \{\neg D, \exists R.(\exists R^{+}.C)\} \\
x \arrow[d, "R"] & \mathcal{L}(x) = \{\exists R^{+}.C\} \cup \{D\} \\
y \arrow[d, "R"] & \mathcal{L}(y) = \{\exists R^{+}.C\} \cup \{\forall R^{-}.D\} \\
z & \mathcal{L}(z) = \{C\} = \{\forall R^{-}.(\forall R^{-}.D)\}
\end{tikzcd}
$$

valid model discovered

Solution (for Converse-PDL) is to use **cut-rule**:

- Past of blocking node must be made *explicit*
- Can only generate sub-concepts, so can restrict attention to Fisher-Ladner closure
- For each C in F-L closure, add $C \sqcup \neg C$ to label of *blocking* node
- Highly non-deterministic $\Rightarrow$ poor computational behaviour

Cardinality Restrictions on Roles

Can extend $\mathcal{ALCI}_{R^+}$ with cardinality restrictions on roles:

Syntax	Semantics
$\leqslant nR$	$\{d \in \Delta^{\mathcal{I}} \mid \lvert R^{\mathcal{I}}(d)\rvert \leqslant n\}$
$\geqslant nR$	$\{d \in \Delta^{\mathcal{I}} \mid \lvert R^{\mathcal{I}}(d)\rvert \geqslant n\}$

where R is *simple* role, i.e. no S such that $S \in \mathbf{R}_+$ and $S \sqsubseteq R$

Expansion rules straightforward — wont discuss in detail

- $\geqslant$-rule, create successor(s) when required
- $\leqslant$-rule, identify successors when required (non-deterministic)
- $\{\geqslant nR, \leqslant mS\}$, $R \sqsubseteq S$ and $n > m \Rightarrow$ contradiction

Problem now is that no longer have finite model property, e.g. for

$$\neg C \sqcap \exists F^{-}.C \sqcap (\leqslant 1\ F) \sqcap \forall R^{-}.(\exists F^{-}.(C \sqcap (\leqslant 1\ F)))$$

with $F \sqsubseteq R$ and $R \in \mathbf{R}_{+}$, all valid models have infinite chain of F^{-}-successors:

$$w \quad \mathcal{L}(w) = \{\neg C, \exists F^{-}.C, (\leqslant 1\ F), \forall R^{-}.(\exists F^{-}.(C \sqcap (\leqslant 1\ F)))\}$$

$$\big\downarrow F^{-}$$

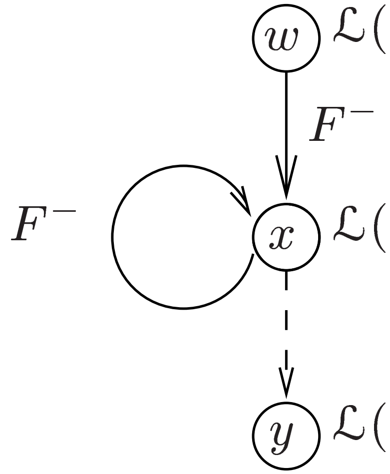
$$x \quad \mathcal{L}(x) = \{C, (\leqslant 1\ F), \forall R^{-}.(\exists F^{-}.(C \sqcap (\leqslant 1\ F)))\}$$

$$\big\downarrow F^{-}$$

$$y \quad \mathcal{L}(y) = \{C, (\leqslant 1\ F), \forall R^{-}.(\exists F^{-}.(C \sqcap (\leqslant 1\ F)))\}$$

$$\vdots$$

Cyclical interpretation of blocking would cause chain to collapse into w:



$$\mathcal{L}(w) = \{\neg C, \exists F^{-}.C, (\leqslant 1\ F), \forall R^{-}.(\exists F^{-}.(C \sqcap (\leqslant 1\ F)))\}$$

$$\mathcal{L}(x) = \{C, (\leqslant 1\ F), \forall R^{-}.(\exists F^{-}.(C \sqcap (\leqslant 1\ F)))\}$$

$$\mathcal{L}(y) = \{C, (\leqslant 1\ F), \forall R^{-}.(\exists F^{-}.(C \sqcap (\leqslant 1\ F)))\}$$

- Both w and x are F-successors of x
- $(\leqslant 1\ F) \in \mathcal{L}(x) \Rightarrow w = x$
- $\{C, \neg C\} \in \mathcal{L}(w) \Rightarrow$ contradiction

Solution is to use **pair-wise** blocking:

- Same basic conditions as dynamic blocking
- Block $\Rightarrow$ infinite model
  - Recursively replace blocked node with copy of tree rooted at blocking node
  - Infinite model not always required, but always valid
- Consider predecessors of blocked and blocking nodes to ensure replacement is valid
  - Check that $\exists R.C$ concepts satisfied by predecessor of blocking node also satisfied by predecessor of blocked node

e.g. tableau for $\neg C \sqcap (\leqslant 1\ F) \sqcap \exists F^{-}.D \sqcap \forall R^{-}.(\exists F^{-}.D)$

where $F \sqsubseteq R$, $R \in \mathbf{R}_{+}$ and $D \doteq C \sqcap (\leqslant 1\ F) \sqcap \exists F.\neg C$

$$
\begin{array}{ll}
x & \mathcal{L}(x) = \{\neg C, (\leqslant 1\ F), \exists F^{-}.D, \forall R^{-}.(\exists F^{-}.D)\} \\
\downarrow F^{-} & \\
y & \mathcal{L}(y) = \{D, \exists F^{-}.D, \forall R^{-}.(\exists F^{-}.D), C, (\leqslant 1\ F), \exists F.\neg C\} \\
\downarrow F^{-} & \\
z & \mathcal{L}(z) = \{D, \exists F^{-}.D, \forall R^{-}.(\exists F^{-}.D), C, (\leqslant 1\ F), \exists F.\neg C\}
\end{array}
$$

- $\mathcal{L}(z) = \mathcal{L}(y) \Rightarrow$ dynamic block
- Cyclical model not valid due to $(\leqslant 1\ F) \in \mathcal{L}(y)$
- Pair-wise condition not satisfied because $\exists F.\neg C \in \mathcal{L}(y)$, which was satisfied by x, not satisfied by y

Block not valid $\Rightarrow$ algorithm continues to expand $\mathcal{L}(z)$

$$
\begin{array}{ll}
x & \mathcal{L}(x) = \{\neg C, (\leqslant 1\ F), \exists F^-.D, \forall R^-.(\exists F^-.D)\} \\
\big\downarrow F^- & \\
y & \mathcal{L}(y) = \{D, \exists F^-.D, \forall R^-.(\exists F^-.D), C, (\leqslant 1\ F), \exists F.\neg C\} \\
\big\downarrow F^- & \\
z & \mathcal{L}(z) = \{D, \exists F^-.D, \forall R^-.(\exists F^-.D), C, (\leqslant 1\ F), \exists F.\neg C\} \\
\big\downarrow F & \\
w & \mathcal{L}(w) = \{\neg C\}
\end{array}
$$

- $\exists F.\neg C \in \mathcal{L}(z) \Rightarrow$ F-successor w of z with $\mathcal{L}(w) = \{\neg C\}$
- $(\leqslant 1\ F) \in \mathcal{L}(z) \Rightarrow w = y$
- $\{C, \neg C\} \in \mathcal{L}(y) \Rightarrow$ contradiction

Note on Decidability

If non-simple roles allowed in cardinality restrictions

- No longer have tree model property
- Can encode grid, and thus tiling problems
- No longer decidable

e.g., concept $A \sqcap \exists R.(B \sqcap \exists R.D) \sqcap \exists R.(C \sqcap \exists R.D) \sqcap \leqslant 3S$

where $S \in \mathbf{R}_+$, $R \sqsubseteq S$ and A, B, C and D are disjoint

$$\begin{tikzcd}
& w \arrow[dl, "R"'] \arrow[dr, "R"] \arrow[dd, dashed, "S"] & \\
x \arrow[dr, "R"'] & & y \arrow[dl, "R"] \\
& z &
\end{tikzcd}$$

$$\mathcal{L}(w) = \{A, \exists R.(B \sqcap \exists R.D), \exists R.(C \sqcap \exists R.D), \leqslant 3S\}$$

$$\mathcal{L}(x) = \{B, \exists R.D\} \qquad \mathcal{L}(y) = \{C, \exists R.D\}$$

$$\mathcal{L}(z) = \{D\}$$

Note on Complexity

$\mathcal{ALC}_{R^+}$ + inverse roles + cardinality restrictions $\Rightarrow$ still in PSPACE

- As in $\mathcal{ALCH}_{R^+}$, role hierarchy is what pushes logic into EXPTIME
- Encouraging result
  - Can stay in lower complexity class if role hierarchy not required
  - Suggests that new algorithm will also be well behaved in realistic applications

Transitive Roles -v- Transitive Closure

- No known *direct* algorithm for $\mathcal{CIN}$ (Converse-PDL with cardinality restrictions)
- Technique described in literature is to *encode* $\mathcal{CIN}$ concepts as equisatisfiable CPDL concepts
- Encoding introduces large number of general inclusion axioms
- Very high degree of non-determinism $\Rightarrow$ very poor computational behaviour
- This is in addition to already poor behaviour of CPDL algorithm

Discussion

- Transitive roles have better computational properties than transitive closure (although logics may be in same complexity class)
- FaCT system has demonstrated potential for implementation level optimisation and practical application
- Can extend expressiveness while (apparently) retaining similar computational properties
- In contrast, C-PDL and $\mathcal{CIN}$ algorithms look computationally disastrous
- New optimised implementation will be used to test empirical tractability of new algorithm