Decidability of $\mathcal{SHIQ}$ with Complex Role Inclusion Axioms

Ian Horrocks and Ulrike Sattler
Computer Science Department, University of Manchester, UK

Extending $\mathcal{SHIQ}$ with expressive means for the propagation of properties along roles, involving surprising (un)decidability results

Acapulco, August, 2003

The Description Logic $\mathcal{SHIQ}$

$\mathcal{SHIQ}$ is a Description Logic which

✓ is underlying ontologies languages **OIL**, **DAML+OIL**, and **OWL**

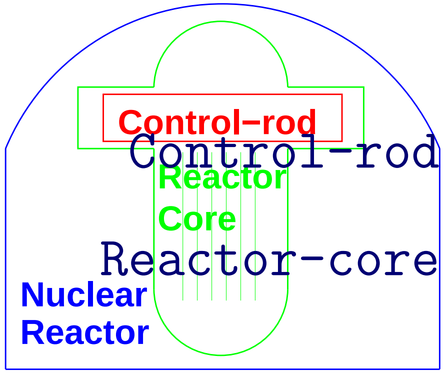
✓ is implemented in the successful DL reasoner **FaCT**, who

✓ behaves well despite reasoning in $\mathcal{SHIQ}$ being ExpTime-complete

✓ extends $\mathcal{ALC}$ (or multi modal K) with

- **general TBoxes** (sets of GCIs of the form $C \sqsubseteq D$)
- **number restrictions** (e.g. $(\geqslant 4\,\texttt{hasComp.Wheel})$ or $(\leqslant 4\,\texttt{hasComp.Wheel})$)
- **transitive** roles (e.g. `hasPart`, `hasAncestor`)
- **inverse** roles (e.g. both `hasPart` and `hasPart`)
- **role inclusions** (set of axioms of the form $r \sqsubseteq s$, e.g. $\texttt{hasDaughter} \sqsubseteq \texttt{hasChild}$)

Example: a $\mathcal{SHIQ}$ TBox



$$\mathtt{Control\text{-}rod} \doteq \mathtt{Device} \sqcap \exists \mathtt{part\text{-}of.Reactor\text{-}core}$$

$$\mathtt{Reactor\text{-}core} \doteq \mathtt{Device} \sqcap \exists \mathtt{has\text{-}part.Control\text{-}rod} \sqcap \exists \mathtt{part\text{-}of.N\text{-}reactor}$$

$$\mathtt{tor} \sqcap \exists \mathtt{has\_part.Faulty} \doteq \mathtt{Dangerous}$$

and an implied subsumption relationship:

$$\mathtt{Control\_rod} \sqcap \mathtt{Faulty} \quad \textbf{is subsumed by} \quad \exists \mathtt{part\text{-}of.Dangerous}$$

Inference problems: satisfiability and subsumption w.r.t. a TBox and an RBox

Tableau algorithm for $\mathcal{SHIQ}$ [HorrocksS_Tobies-LPAR99,HorrocksS_ECAI2002]

- **decides** satisfiability and subsumption of $\mathcal{SHIQ}$**-concepts w.r.t. TBoxes**
- is implemented in the DL reasoner **FaCT** [Horrocks-KR98]
- tries to **generate a model** of input concept C w.r.t. TBox $\mathcal{T}$ by
- **breaking down** C **and** $\mathcal{T}$ **syntactically**, thus inferring contraints on such a model
- uses a special **cycle detection** mechanism to ensure termination

  whose careful design is crucial for **correctness** of algorithm and
  performance of its implementation

The Tableau Algorithm for $\mathcal{SHIQ}$

More precisely: the tableau algorithm works on a **tree**, whose

nodes correspond to objects

edges edges indicate "direct" role-successorship, where

implied edges (transitivity!) have to be added in the model construction

Example: rules that are applied to the tree include

- if $C \sqcap D \in \mathcal{L}(x)$, then add C and D to $\mathcal{L}(x)$
- if $\exists r.C \in \mathcal{L}(x)$, then **create** a new r-successor y of x with $\mathcal{L}(y) = \{C\}$
- if $\forall r.C \in \mathcal{L}(x)$ and y is an r-**neighbour** of x, then
  - add C to $\mathcal{L}(y)$ and
  - if r is transitive, then add $\forall r.C$ to $\mathcal{L}(y)$ and
  - if r has a transitive sub-role s, then add $\forall s.C$ to $\mathcal{L}(y)$

$\mathcal{SHIQ}$ and the Propagation of Properties

Expressible in $\mathcal{SHIQ}$:

faultiness *propagates from a component to its aggregate*

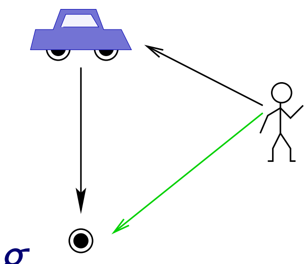
$$\texttt{Device} \sqsubseteq (\texttt{Faulty} \Rightarrow \forall \texttt{hasComp}^{-}.\texttt{Faulty})$$

colours *propagate from a segment to its aggregate*

$$\texttt{Thing} \sqsubseteq (\texttt{Green} \Rightarrow \forall \texttt{hasSegment}^{-}.\texttt{Green}) \sqcap (\texttt{Red} \Rightarrow \forall \texttt{hasSegment}^{-}.\texttt{Red}) \sqcap \ldots$$

Not expressible in $\mathcal{SHIQ}$ or other implemented DL

➠ **various questionable work-arounds:**



ownership *propagates from an aggregate to its parts, e.g.*
the owner of the car is also the owner of the car's parts

localisation *propagates from a division to its aggregate , e.g.*
a trauma located in a part of a body structure is a trauma of the body structure

Extending $\mathcal{SHIQ}$ with Complex RIAs

Solution: extend $\mathcal{SHIQ}$ with **role inclusion axioms (RIAs)** of the form $r \circ s \sqsubseteq t$, e.g.

$$\begin{aligned} \texttt{owns} \circ \texttt{has-part} &\sqsubseteq \texttt{owns}, \\ \texttt{hasLocation} \circ \texttt{divisionOf} &\sqsubseteq \texttt{hasLocation} \end{aligned}$$

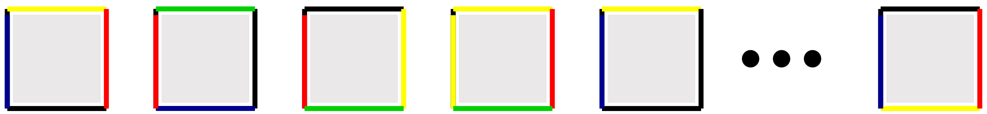
Result: known from Grammar Logic [Baldoni1998]:
$\mathcal{SHIQ}$ (or even $\mathcal{ALC}$) with such an extension becomes **undecidable**

Next Solution: investigate motivating examples more closely,
➠ axioms of the form $r \circ s \sqsubseteq s$ **or** $s \circ r \sqsubseteq s$ **suffice** for propagation
➠ undecidability result from [Baldoni1998] **not applicable**

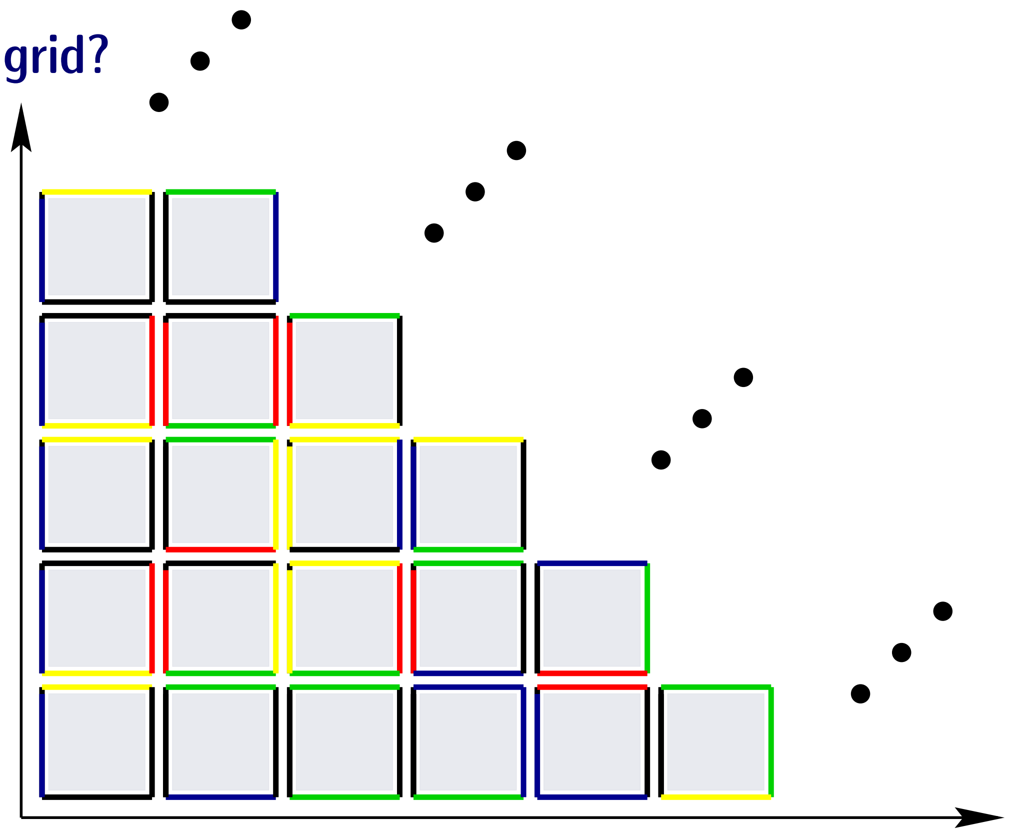
Next Result: $\mathcal{SHIQ}$ with such an extension is still **undecidable**
proof by reduction of the **Domino Problem**

An Undecidable Problem: the Domino Problem

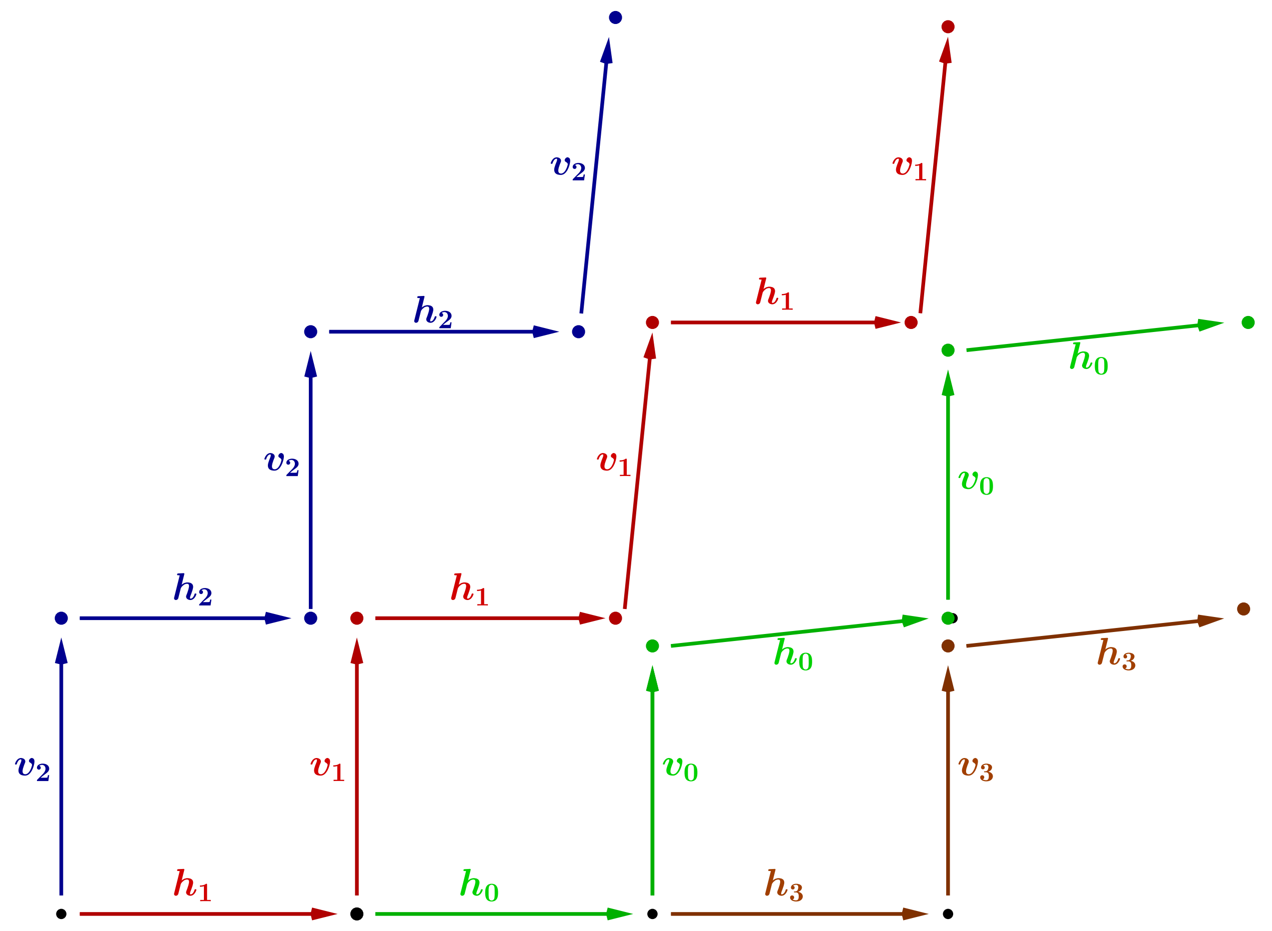
Given (an unbounded amount of) instances of finitely many **dominos types**



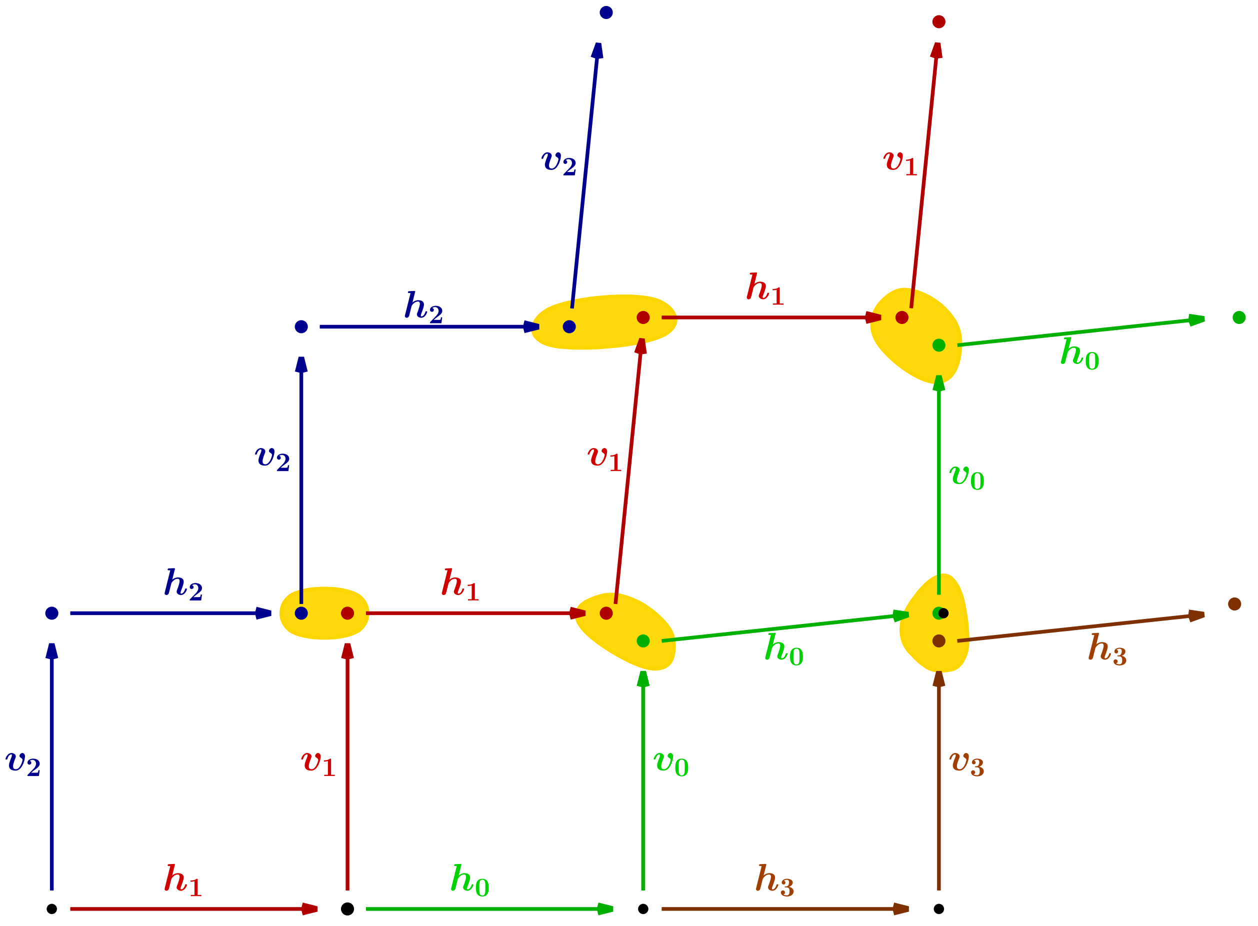
can we tile the (infinite) grid?



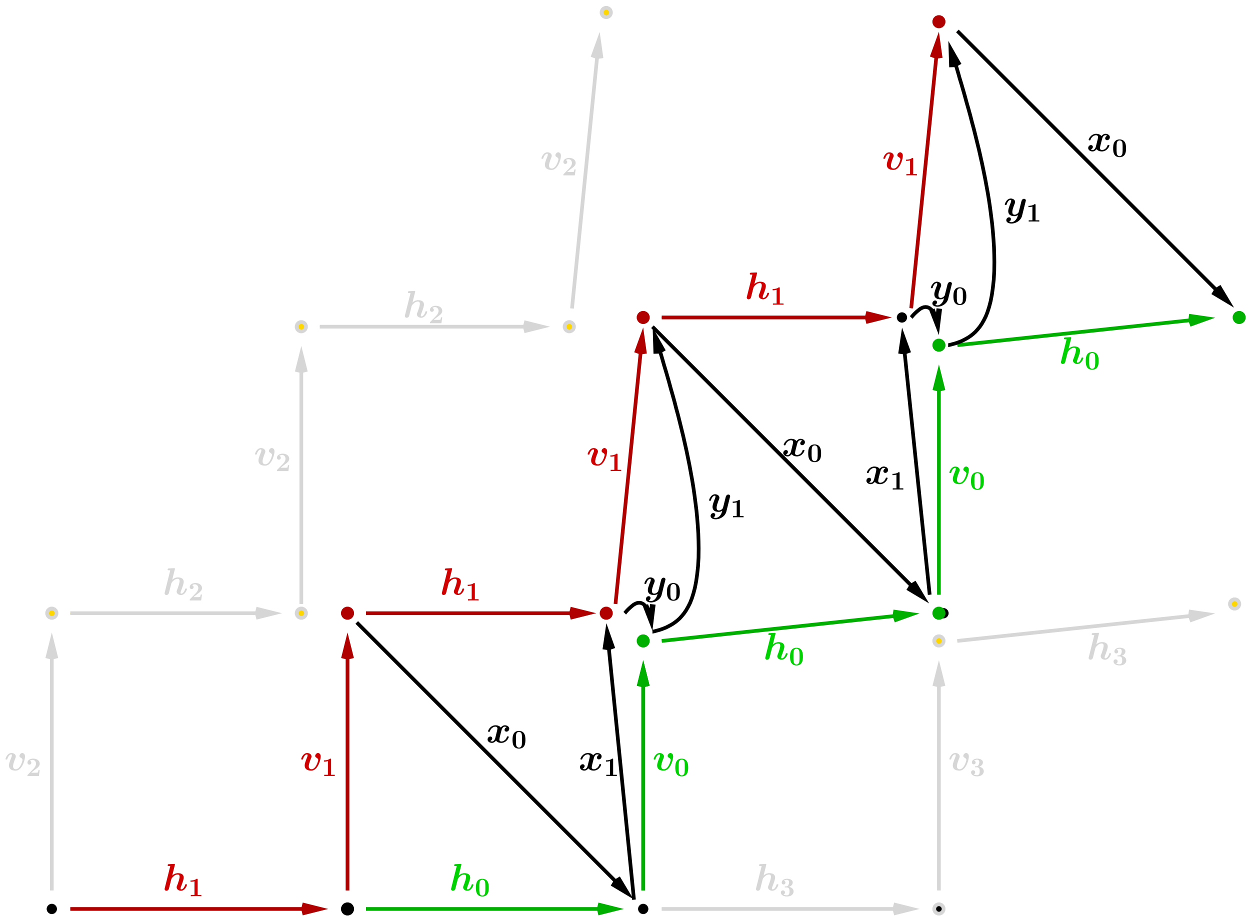
General idea: describe **staircases** – easy, possible in $\mathcal{SHIQ}$



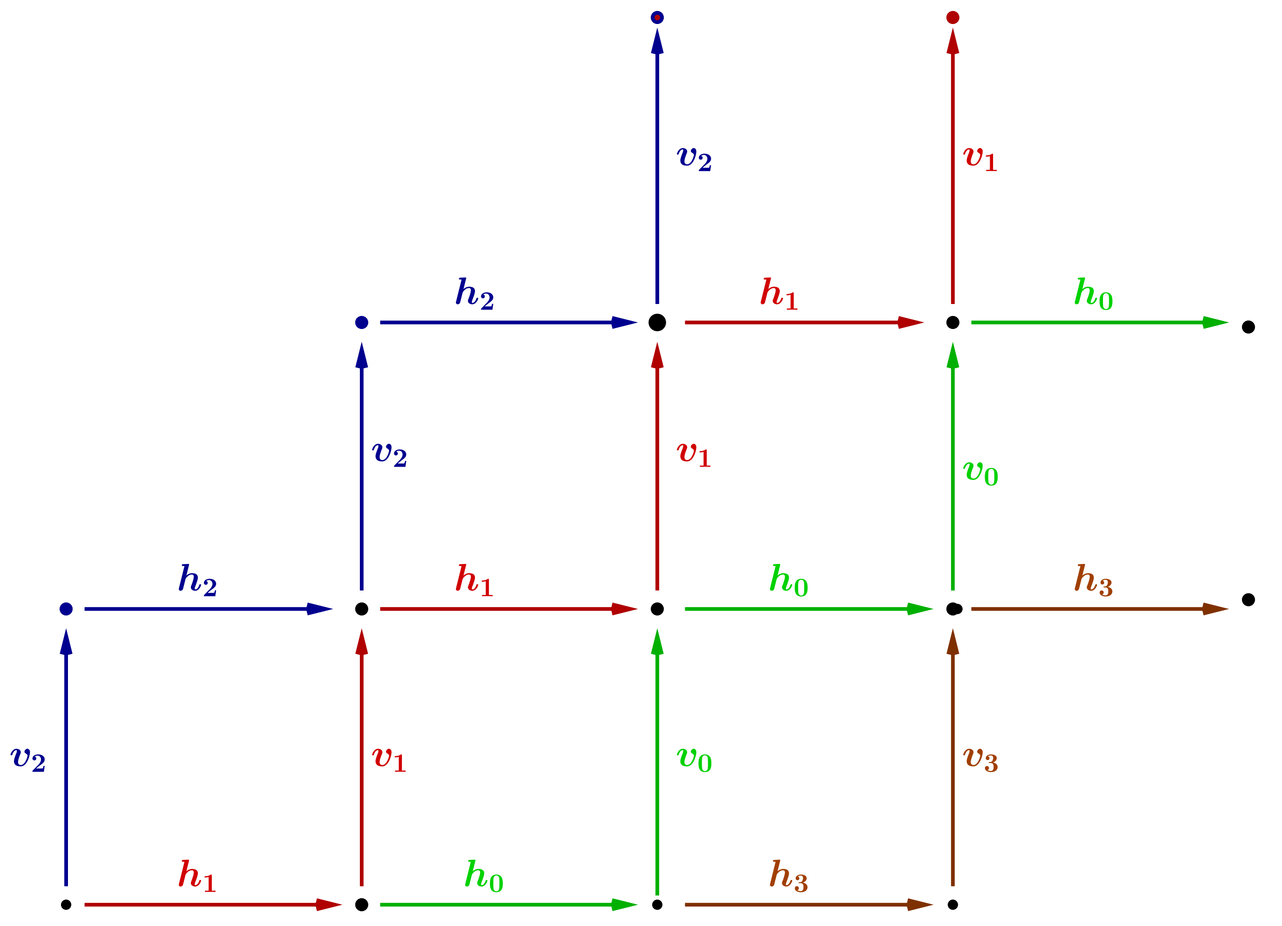
General idea: describe **staircases** – easy, possible in $\mathcal{SHIQ}$
use RIAs to **merge** staircases into **grid**



General idea: describe **staircases** – easy, possible in $\mathcal{SHIQ}$
use RIAs to **merge** staircases into **grid**



General idea: describe **staircases** – easy, possible in $\mathcal{SHIQ}$
use RIAs to **merge staircases into grid**



Source of complexity of RIAs: **inverse** roles and **cycles** in set of RIAs

Third solution: dissallow cycles, i.e., $\mathcal{RIQ}$ is the extensions of $\mathcal{SHIQ}$ with finite, **cycle-free** set of RIAs $\mathcal{R}$, where

$\mathcal{R}$ is cycle-free if **uses**$^+$ is cycle-free: s **uses** r_i **iff**

$$\textbf{expr}(r_1, \ldots, r_n) \sqsubseteq s \in \mathcal{R} \textbf{ and } r_i \neq s$$

In $\mathcal{RIQ}$, each RIA can be of the form

$$\begin{aligned} r_1 \cdots r_\ell s &\sqsubseteq s \quad \text{or} \\ s r_1 \cdots r_\ell &\sqsubseteq s \quad \text{or} \\ ss &\sqsubseteq s \end{aligned}$$

Acyclicity is not a serious restriction:
(1) motivating examples still expressible
(2) some ontology experts thinks that cycles indicate **modelling flaws**

Tableau algorihm for $\mathcal{RIQ}$ similar to the one for $\mathcal{SHIQ}$, but

- $\mathcal{R}$ is made explicit in a **finite automaton** A_r for each role r
- concepts $\forall r.C$ in node labels are replaced with $\forall A_r.C$
- automata A_r are then used to

1. **remember** roles paths along which $\forall r.C$ was "pushed":

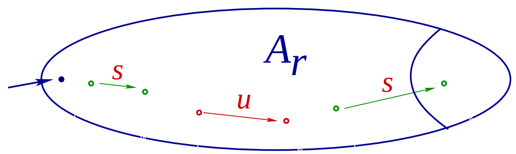
   if y is an s-neighbour of x and $\forall A.C \in \mathfrak{L}(x)$,
   then add $\forall A'.C$ to $\mathfrak{L}(y)$, where
   A' is the result of A reading s obtained by switching initial states

2. **decide** whether to add C to $\mathfrak{L}(y)$

   if $\forall A.C \in \mathfrak{L}(y)$ and A is in a **final state**,
   then **add** C to $\mathfrak{L}(y)$

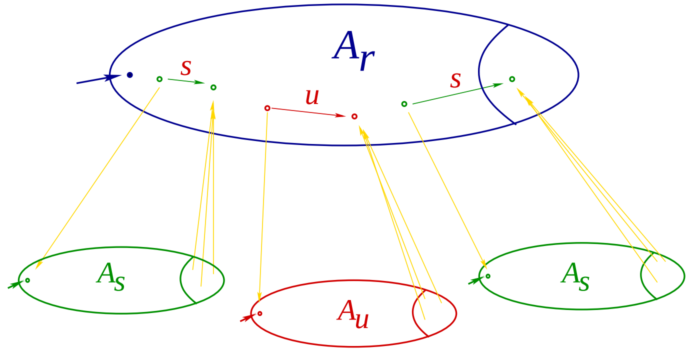
Construction of A_r: working up the cycle-free (!) uses relation, for each role r,

1. construct **automaton** A_r^1 for **expr**$(s_1, \ldots, s_n) \cup r$



Construction of A_r: working up the cycle-free (!) uses relation, for each role r,

1. construct **automaton** A_r^1 for **regexp**$(\mathsf{s_1}, \ldots, \mathsf{s_n}) \cup r$
2. add a **disjoint copy of** A_s for each $\bullet \xrightarrow{s} \bullet$ in A_r^1
3. add ε-transition from $\bullet \xrightarrow{s} \bullet$ to **init**(A_s)
4. add ε-transitions from **final**(A_s) to $\bullet \xrightarrow{s} \bullet$

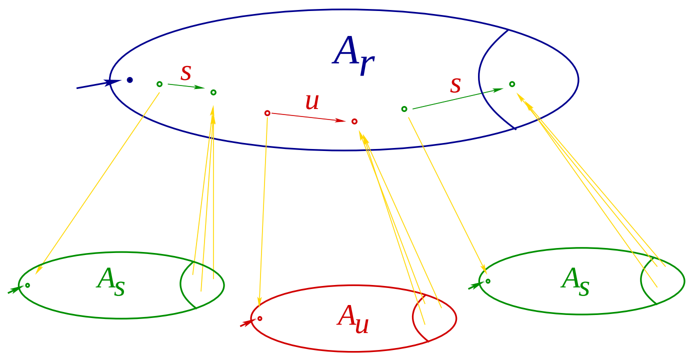


➠ **automaton** A_r **for** r — whose size is possibly exponential in $\mathcal{R}$: **unfolding**

Tableau Algorithm for $\mathcal{RIQ}$ II

Construction of A_r: working up the cycle-free (!) uses relation, for each role r,

1. construct **automaton** A_r^1 for **regexp**$(\mathsf{s_1}, \ldots, \mathsf{s_n}) \cup r$
2. add a **disjoint copy of** A_s for each $\bullet \xrightarrow{s} \bullet$ in A_r^1
3. add ε-transition from $\bullet \xrightarrow{s} \bullet$ to **init**(A_s)
4. add ε-transitions from **final**(A_s) to $\bullet \xrightarrow{s} \bullet$



➠ **automaton A_r for r** — whose size is possibly exponential in $\mathcal{R}$: **unfolding**

Lemma:

$\mathcal{I}$ **is a model of** $\mathcal{R}$

iff

for each $r_1 \cdots r_n \in L(A_r)$, for each $x, y \in \Delta^{\mathcal{I}}$,

if $\langle x, y\rangle \in r_1^{\mathcal{I}} \circ \ldots \circ r_n$, **then** $\langle x, y\rangle \in r^{\mathcal{I}}$

Implementing the Tableau Algorithm for $\mathcal{RIQ}$

FaCT was extended to $\mathcal{RIQ}$:

- each A_r is transformed into **minimal DFA** using AT&T FSM Library$^{\text{TM}}$ [MoPR98]
  - "only" pre-processing
  - yields fewer node labels in tableau algorithm $\Rightarrow$ smaller search space
- first tests on **Galen medical terminology KB** (2,740 named concepts, 413 roles, 26 transitive ones) is **promising:**
  - the (pre-)processing of RIAs takes some time: $+100\%$
  - but satisfiability algorithm shows **similar performance**: only $+3\%$
  - system can draw **useful, additional inferences**: e.g., w.r.t. the RIA

$\mathtt{hasLocation} \circ \mathtt{divisionOf} \sqsubseteq \mathtt{hasLocation}$, the concept

$$\mathtt{Fracture} \sqcap \exists \mathtt{hasLocation}.(\mathtt{Neck} \sqcap \exists \mathtt{isDivisionOf}.\mathtt{Femur})$$

is indeed subsumed by

$$\mathtt{Fracture} \sqcap \exists \mathtt{hasLocation}.\mathtt{Femur}$$

Summary and Outlook

Extending successful $\mathcal{SHIQ}$ with a saught-after constructor for propagation

Results:

1. a **naive** such extension leads to undecidability
2. a **semi-naive** such extension still leads to undecidability
3. a **careful** such extension, $\mathcal{RIQ}$, yields a DL with
   - elegant tableau algorithm
   - behaviour similar to the one for $\mathcal{SHIQ}$
   - being able to draw useful, additional inferences

Open questions:

1. is exponential blow-up **avoidable**?
2. how does implementation of $\mathcal{RIQ}$ behave on other knowledge bases?