Blocking Mechanisms in Description Logics, A General Approach

Dmitry Tishkovsky

School of Computer Science
The University of Manchester
dmitry.tishkovsky@manchester.ac.uk

PRAGUE, CZECH REPUBLIC

11 JUNE 2008

Outline

1. Crash course on DLs
   - ALC
   - Universal modality
   - TBox
   - ABox and individuals
   - RBox
   - Problem statement
   - Tableau procedure
2. Blocking mechanisms
   - Subset blocking
   - Equality blocking
   - Pairwise blocking
   - Successor and anywhere blocking
3. General blocking mechanism
   - Unrestricted blocking rule
   - Simulating existing blocking mechanisms

$\mathcal{ALC}$ Syntax and Semantics

Atomic concepts: $p, p_0, p_1 \ldots$

Atomic roles: $r, r_0, r_1 \ldots$

Concepts: $C, D \stackrel{\text{def}}{=} p \mid \neg C \mid C \sqcup D \mid \exists r.C$

$$C \sqcap D \stackrel{\text{def}}{=} \neg(\neg C \sqcup \neg D), \qquad \forall r.C \stackrel{\text{def}}{=} \neg \exists r.\neg C.$$

$\mathcal{ALC}$ Syntax and Semantics

Atomic concepts: $p, p_0, p_1 \ldots$

Atomic roles: $r, r_0, r_1 \ldots$

Concepts: $C, D \stackrel{\text{def}}{=} p \mid \neg C \mid C \sqcup D \mid \exists r.C$

$$C \sqcap D \stackrel{\text{def}}{=} \neg(\neg C \sqcup \neg D), \qquad \forall r.C \stackrel{\text{def}}{=} \neg\exists r.\neg C.$$

Interpretation (model): $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ satisfying

$$p^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \qquad r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \qquad \ell^{\mathcal{I}} \in \Delta^{\mathcal{I}}$$

$$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$

$$(\exists r.C)^{\mathcal{I}} = \{x \mid \exists y \in C^{\mathcal{I}}\ (x, y) \in r^{\mathcal{I}}\}$$

MANCHESTER 1824 The University of Manchester

Universal Modality

- $\forall C$, $\exists C$.
- $(\forall C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \mid \forall y \in S\ y \in C^{\mathcal{I}}\}$.
- $(\exists C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \mid \exists y \in S\ y \in C^{\mathcal{I}}\}$.

Universal Modality

- $\forall C$, $\exists C$.
- $(\forall C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \mid \forall y \in S \; y \in C^{\mathcal{I}}\}$.
- $(\exists C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \mid \exists y \in S \; y \in C^{\mathcal{I}}\}$.

MANCHESTER 1824 The University of Manchester

Universal Modality

- $\forall C$, $\exists C$.
- $(\forall C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \mid \forall y \in S\ y \in C^{\mathcal{I}}\}$.
- $(\exists C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \mid \exists y \in S\ y \in C^{\mathcal{I}}\}$.

General TBox

- *Terminological axiom:* $C = D$.
- Interpretation: $C^{\mathcal{I}} = D^{\mathcal{I}}$.
- A *general TBox* is a set of terminological axioms.
- *Subsumption axiom* $C \sqsubseteq D$.
- Interpretation: $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- Every general TBox is equivalent to a set of subsumption axioms.
- If language contains a universal modality than TBox is representable as a set of concepts: $\forall(\neg C \sqcup D)$.

General TBox

- *Terminological axiom:* $C = D$.
- Interpretation: $C^{\mathcal{I}} = D^{\mathcal{I}}$.
- A *general TBox* is a set of terminological axioms.
- *Subsumption axiom* $C \sqsubseteq D$.
- Interpretation: $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- Every general TBox is equivalent to a set of subsumption axioms.
- If language contains a universal modality than TBox is representable as a set of concepts: $\forall(\neg C \sqcup D)$.

MANCHESTER 1824 The University of Manchester

General TBox

- *Terminological axiom:* $C = D$.
- Interpretation: $C^{\mathcal{I}} = D^{\mathcal{I}}$.
- A *general TBox* is a set of terminological axioms.
- *Subsumption axiom* $C \sqsubseteq D$.
- Interpretation: $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- Every general TBox is equivalent to a set of subsumption axioms.
- If language contains a universal modality than TBox is representable as a set of concepts: $\forall(\neg C \sqcup D)$.

General TBox

- *Terminological axiom:* $C = D$.
- Interpretation: $C^{\mathcal{I}} = D^{\mathcal{I}}$.
- A *general TBox* is a set of terminological axioms.
- *Subsumption axiom* $C \sqsubseteq D$.
- Interpretation: $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- Every general TBox is equivalent to a set of subsumption axioms.
- If language contains a universal modality than TBox is representable as a set of concepts: $\forall(\neg C \sqcup D)$.

The University of Manchester
MANCHESTER 1824

General TBox

- *Terminological axiom:* $C = D$.
- Interpretation: $C^{\mathcal{I}} = D^{\mathcal{I}}$.
- A *general TBox* is a set of terminological axioms.
- *Subsumption axiom* $C \sqsubseteq D$.
- Interpretation: $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- Every general TBox is equivalent to a set of subsumption axioms.
- If language contains a universal modality than TBox is representable as a set of concepts: $\forall(\neg C \sqcup D)$.

MANCHESTER 1824 The University of Manchester

General TBox

- *Terminological axiom:* $C = D$.
- Interpretation: $C^{\mathcal{I}} = D^{\mathcal{I}}$.
- A *general TBox* is a set of terminological axioms.
- *Subsumption axiom* $C \sqsubseteq D$.
- Interpretation: $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- Every general TBox is equivalent to a set of subsumption axioms.
- If language contains a universal modality than TBox is representable as a set of concepts: $\forall(\neg C \sqcup D)$.

MANCHESTER 1824 The University of Manchester

General TBox

- *Terminological axiom:* $C = D$.
- Interpretation: $C^{\mathcal{I}} = D^{\mathcal{I}}$.
- A *general TBox* is a set of terminological axioms.
- *Subsumption axiom* $C \sqsubseteq D$.
- Interpretation: $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- Every general TBox is equivalent to a set of subsumption axioms.
- If language contains a universal modality than TBox is representable as a set of concepts: $\forall(\neg C \sqcup D)$.

ABox and Individuals

- ABox A is a set of *concept assertions* $C(a)$ and *role assertions* $R(a, b)$ where a, b stand for an elements of a model.
- *Set of individual names:* $\ell, \ell_0, \ell_1, \ldots$
- *Singleton concepts:* $\{\ell\}$.
- Interpretation: $\ell^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and $\{\ell\}^{\mathcal{I}} \stackrel{\text{def}}{=} \{\ell^{\mathcal{I}}\}$.
- *Concept assertion:* $\ell : C$.
- Interpretation:

$$(\ell : C)^{\mathcal{I}} \stackrel{\text{def}}{=} \begin{cases} \Delta^{\mathcal{I}}, & \text{if } \ell^{\mathcal{I}} \in C^{\mathcal{I}}, \\ \varnothing, & \text{otherwise,} \end{cases}$$

- *Role assertion:* $(\ell, \ell') : R \stackrel{\text{def}}{=} \ell : \exists R.\{\ell'\}$.

ABox and Individuals

- ABox A is a set of *concept assertions* $C(a)$ and *role assertions* $R(a, b)$ where a, b stand for an elements of a model.
- *Set of individual names:* $\ell, \ell_0, \ell_1, \ldots$
- *Singleton concepts:* $\{\ell\}$.
- Interpretation: $\ell^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and $\{\ell\}^{\mathcal{I}} \stackrel{\text{def}}{=} \{\ell^{\mathcal{I}}\}$.
- *Concept assertion:* $\ell : C$.
- Interpretation:

$$(\ell : C)^{\mathcal{I}} \stackrel{\text{def}}{=} \begin{cases} \Delta^{\mathcal{I}}, & \text{if } \ell^{\mathcal{I}} \in C^{\mathcal{I}}, \\ \emptyset, & \text{otherwise,} \end{cases}$$

- *Role assertion:* $(\ell, \ell') : R \stackrel{\text{def}}{=} \ell : \exists R.\{\ell'\}$.

ABox and Individuals

- ABox A is a set of *concept assertions* $C(a)$ and *role assertions* $R(a, b)$ where a, b stand for an elements of a model.
- *Set of individual names:* $\ell, \ell_0, \ell_1, \ldots$
- *Singleton concepts:* $\{\ell\}$.
- Interpretation: $\ell^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and $\{\ell\}^{\mathcal{I}} \stackrel{\text{def}}{=} \{\ell^{\mathcal{I}}\}$.
- *Concept assertion:* $\ell : C$.
- Interpretation:

$$(\ell : C)^{\mathcal{I}} \stackrel{\text{def}}{=} \begin{cases} \Delta^{\mathcal{I}}, & \text{if } \ell^{\mathcal{I}} \in C^{\mathcal{I}}, \\ \varnothing, & \text{otherwise,} \end{cases}$$

- *Role assertion:* $(\ell, \ell') : R \stackrel{\text{def}}{=} \ell : \exists R.\{\ell'\}$.

ABox and Individuals

- ABox A is a set of *concept assertions* $C(a)$ and *role assertions* $R(a, b)$ where a, b stand for an elements of a model.
- *Set of individual names:* $\ell, \ell_0, \ell_1, \ldots$
- *Singleton concepts:* $\{\ell\}$.
- Interpretation: $\ell^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and $\{\ell\}^{\mathcal{I}} \stackrel{\text{def}}{=} \{\ell^{\mathcal{I}}\}$.
- *Concept assertion:* $\ell : C$.
- Interpretation:

$$(\ell : C)^{\mathcal{I}} \stackrel{\text{def}}{=} \begin{cases} \Delta^{\mathcal{I}}, & \text{if } \ell^{\mathcal{I}} \in C^{\mathcal{I}}, \\ \emptyset, & \text{otherwise,} \end{cases}$$

- *Role assertion:* $(\ell, \ell') : R \stackrel{\text{def}}{=} \ell : \exists R.\{\ell'\}$.

MANCHESTER 1824 The University of Manchester

ABox and Individuals

- ABox A is a set of *concept assertions* $C(a)$ and *role assertions* $R(a, b)$ where a, b stand for an elements of a model.
- *Set of individual names:* $\ell, \ell_0, \ell_1, \ldots$
- *Singleton concepts:* $\{\ell\}$.
- Interpretation: $\ell^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and $\{\ell\}^{\mathcal{I}} \stackrel{\text{def}}{=} \{\ell^{\mathcal{I}}\}$.
- *Concept assertion:* $\ell : C$.
- Interpretation:

$$(\ell : C)^{\mathcal{I}} \stackrel{\text{def}}{=} \begin{cases} \Delta^{\mathcal{I}}, & \text{if } \ell^{\mathcal{I}} \in C^{\mathcal{I}}, \\ \varnothing, & \text{otherwise,} \end{cases}$$

- *Role assertion:* $(\ell, \ell') : R \stackrel{\text{def}}{=} \ell : \exists R.\{\ell'\}$.

ABox and Individuals

- ABox A is a set of *concept assertions* $C(a)$ and *role assertions* $R(a,b)$ where a,b stand for an elements of a model.
- *Set of individual names:* $\ell, \ell_0, \ell_1, \ldots$
- *Singleton concepts:* $\{\ell\}$.
- Interpretation: $\ell^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and $\{\ell\}^{\mathcal{I}} \stackrel{\text{def}}{=} \{\ell^{\mathcal{I}}\}$.
- *Concept assertion:* $\ell : C$.
- Interpretation:

$$(\ell : C)^{\mathcal{I}} \stackrel{\text{def}}{=} \begin{cases} \Delta^{\mathcal{I}}, & \text{if } \ell^{\mathcal{I}} \in C^{\mathcal{I}}, \\ \varnothing, & \text{otherwise,} \end{cases}$$

- *Role assertion:* $(\ell, \ell') : R \stackrel{\text{def}}{=} \ell : \exists R.\{\ell'\}$.

ABox and Individuals

- ABox A is a set of *concept assertions* $C(a)$ and *role assertions* $R(a, b)$ where a, b stand for an elements of a model.
- *Set of individual names:* $\ell, \ell_0, \ell_1, \ldots$
- *Singleton concepts:* $\{\ell\}$.
- Interpretation: $\ell^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and $\{\ell\}^{\mathcal{I}} \stackrel{\text{def}}{=} \{\ell^{\mathcal{I}}\}$.
- *Concept assertion:* $\ell : C$.
- Interpretation:

$$(\ell : C)^{\mathcal{I}} \stackrel{\text{def}}{=} \begin{cases} \Delta^{\mathcal{I}}, & \text{if } \ell^{\mathcal{I}} \in C^{\mathcal{I}}, \\ \varnothing, & \text{otherwise,} \end{cases}$$

- *Role assertion:* $(\ell, \ell') : R \stackrel{\text{def}}{=} \ell : \exists R.\{\ell'\}$.

RBox

- RBox is a set of *role inclusion axioms* $R \sqsubseteq S$.
- possibly, other assumptions on roles are included, e.g. transitivity of roles.

Problem Statement

- *Knowledge base* KB is a tuple (ABox, TBox, RBox).
- The problem (KB $\models C$?):
  Given a knowledge base KB and a concept C, find a model $\mathcal{I}$ which validate all the axioms of the knowledge base and $C^{\mathcal{I}} \neq \varnothing$.
- In modern description logic, tableau decision algorithms are usually used for solving the problem.

MANCHESTER 1824 The University of Manchester

Problem Statement

- *Knowledge base* KB is a tuple $(\mathrm{ABox}, \mathrm{TBox}, \mathrm{RBox})$.
- The problem (KB $\models C$?):
  Given a knowledge base KB and a concept C, find a model $\mathcal{I}$ which validate all the axioms of the knowledge base and $C^{\mathcal{I}} \neq \varnothing$.
- In modern description logic, tableau decision algorithms are usually used for solving the problem.

MANCHESTER 1824 The University of Manchester

Problem Statement

- *Knowledge base* KB is a tuple $(\mathrm{ABox}, \mathrm{TBox}, \mathrm{RBox})$.
- The problem (KB $\models C$?):
  Given a knowledge base KB and a concept C, find a model $\mathcal{I}$ which validate all the axioms of the knowledge base and $C^{\mathcal{I}} \neq \varnothing$.
- In modern description logic, tableau decision algorithms are usually used for solving the problem.

Tableau Procedure
Common Tableau Rules

Standard rules for $\mathcal{ALC}$

$$(\bot)\ \frac{\ell : C,\quad \ell : \neg C}{\bot} \qquad (\neg\neg)\ \frac{\ell : \neg\neg C}{\ell : C}$$

$$(\neg\sqcup)\ \frac{\ell : \neg(C \sqcup D)}{\ell : \neg C,\quad \ell : \neg D} \qquad (\sqcup)\ \frac{\ell : (C \sqcup D)}{\ell : C \mid \ell : D}$$

$$(\exists)\ \frac{\ell : \exists R.C}{\ell : \exists R.\{\ell'\},\quad \ell' : C}(\ell' \text{ is new}) \qquad (\neg\exists)\ \frac{\ell : \neg\exists R.C,\quad \ell : \exists R.\{\ell'\}}{\ell' : \neg C}$$

Rules for individuals

$$(\text{sym})\ \frac{\ell : \{\ell'\}}{\ell' : \{\ell\}} \qquad (\neg\text{sym})\ \frac{\ell : \neg\{\ell'\}}{\ell' : \neg\{\ell\}} \qquad (\text{ref})\ \frac{\ell : C}{\ell : \{\ell\}}$$

$$(\text{mon})\ \frac{\ell : \{\ell'\},\quad \ell' : C}{\ell : C} \qquad (\text{canc})\ \frac{\ell : (\ell' : C)}{\ell' : C}$$

Tableau Procedure
Common Tableau Rules

Standard rules for $\mathcal{ALC}$

$$(\bot)\ \frac{\ell : C,\ \ \ell : \neg C}{\bot} \qquad (\neg\neg)\ \frac{\ell : \neg\neg C}{\ell : C}$$

$$(\neg\sqcup)\ \frac{\ell : \neg(C \sqcup D)}{\ell : \neg C,\ \ \ell : \neg D} \qquad (\sqcup)\ \frac{\ell : (C \sqcup D)}{\ell : C \mid \ell : D}$$

$$(\exists)\ \frac{\ell : \exists R.C}{\ell : \exists R.\{\ell'\},\ \ \ell' : C}(\ell' \text{ is new}) \qquad (\neg\exists)\ \frac{\ell : \neg\exists R.C,\ \ \ell : \exists R.\{\ell'\}}{\ell' : \neg C}$$

Rules for individuals

$$(\text{sym})\ \frac{\ell : \{\ell'\}}{\ell' : \{\ell\}} \qquad (\neg\text{sym})\ \frac{\ell : \neg\{\ell'\}}{\ell' : \neg\{\ell\}} \qquad (\text{ref})\ \frac{\ell : C}{\ell : \{\ell\}}$$

$$(\text{mon})\ \frac{\ell : \{\ell'\},\ \ \ell' : C}{\ell : C} \qquad (\text{canc})\ \frac{\ell : (\ell' : C)}{\ell' : C}$$

$\ell : \{\ell'\} \equiv \ell = \ell'$

Outline

1 Crash course on DLs
- ALC
- Universal modality
- TBox
- ABox and individuals
- RBox
- Problem statement
- Tableau procedure

2 Blocking mechanisms
- Subset blocking
- Equality blocking
- Pairwise blocking
- Successor and anywhere blocking

3 General blocking mechanism
- Unrestricted blocking rule
- Simulating existing blocking mechanisms

Blocking

- Blocking is a detection of repetitions in partially constructed models.
- If some conditions are true for given two individuals (labels) ℓ and ℓ' then ℓ is blocked by ℓ' (for application of individual generating rules).
- To avoid cyclic blocking we assume that all individuals in branch are linearly ordered by an ordering $<$ and given two nominals we always block the largest one w.r.t. the ordering.
- If blocks on individuals are never undone then blocking is *static*. Otherwise it is called *dynamic*.
- Blocking mechanisms usually require access to a set of concepts $\tau(\ell)$ associated with given individual ℓ and sometimes a set of role links $\tau(\ell, \ell')$ associated with two individuals within the same branch $\mathcal{B}$:

$$\tau(\ell) \stackrel{\text{def}}{=} \{C \mid \ell : C \in \mathcal{B}\}$$

$$\tau(\ell, \ell') \stackrel{\text{def}}{=} \{R \mid \ell : \exists R.\{\ell'\} \in \mathcal{B}\}$$

MANCHESTER 1824 The University of Manchester

Blocking

- Blocking is a detection of repetitions in partially constructed models.
- If some conditions are true for given two individuals (labels) ℓ and ℓ' then ℓ is blocked by ℓ' (for application of individual generating rules).
- To avoid cyclic blocking we assume that all individuals in branch are linearly ordered by an ordering $<$ and given two nominals we always block the largest one w.r.t. the ordering.
- If blocks on individuals are never undone then blocking is *static*. Otherwise it is called *dynamic*.
- Blocking mechanisms usually require access to a set of concepts $\tau(\ell)$ associated with given individual ℓ and sometimes a set of role links $\tau(\ell, \ell')$ associated with two individuals within the same branch $\mathcal{B}$:

$$\tau(\ell) \stackrel{\text{def}}{=} \{C \mid \ell : C \in \mathcal{B}\}$$

$$\tau(\ell, \ell') \stackrel{\text{def}}{=} \{R \mid \ell : \exists R.\{\ell'\} \in \mathcal{B}\}$$

MANCHESTER 1824 The University of Manchester

Blocking

- Blocking is a detection of repetitions in partially constructed models.
- If some conditions are true for given two individuals (labels) ℓ and ℓ' then ℓ is blocked by ℓ' (for application of individual generating rules).
- To avoid cyclic blocking we assume that all individuals in branch are linearly ordered by an ordering $<$ and given two nominals we always block the largest one w.r.t. the ordering.
- If blocks on individuals are never undone then blocking is *static*. Otherwise it is called *dynamic*.
- Blocking mechanisms usually require access to a set of concepts $\tau(\ell)$ associated with given individual ℓ and sometimes a set of role links $\tau(\ell, \ell')$ associated with two individuals within the same branch $\mathcal{B}$:

$$\tau(\ell) \stackrel{\text{def}}{=} \{C \mid \ell : C \in \mathcal{B}\}$$

$$\tau(\ell, \ell') \stackrel{\text{def}}{=} \{R \mid \ell : \exists R.\{\ell'\} \in \mathcal{B}\}$$

MANCHESTER 1824 The University of Manchester

Blocking

- Blocking is a detection of repetitions in partially constructed models.
- If some conditions are true for given two individuals (labels) ℓ and ℓ' then ℓ is blocked by ℓ' (for application of individual generating rules).
- To avoid cyclic blocking we assume that all individuals in branch are linearly ordered by an ordering $<$ and given two nominals we always block the largest one w.r.t. the ordering.
- If blocks on individuals are never undone then blocking is *static*. Otherwise it is called *dynamic*.
- Blocking mechanisms usually require access to a set of concepts $\tau(\ell)$ associated with given individual ℓ and sometimes a set of role links $\tau(\ell,\ell')$ associated with two individuals within the same branch $\mathcal{B}$:

$$\tau(\ell) \stackrel{\text{def}}{=} \{C \mid \ell : C \in \mathcal{B}\}$$
$$\tau(\ell,\ell') \stackrel{\text{def}}{=} \{R \mid \ell : \exists R.\{\ell'\} \in \mathcal{B}\}$$

MANCHESTER 1824 The University of Manchester

Blocking

- Blocking is a detection of repetitions in partially constructed models.
- If some conditions are true for given two individuals (labels) ℓ and ℓ' then ℓ is blocked by ℓ' (for application of individual generating rules).
- To avoid cyclic blocking we assume that all individuals in branch are linearly ordered by an ordering $<$ and given two nominals we always block the largest one w.r.t. the ordering.
- If blocks on individuals are never undone then blocking is *static*. Otherwise it is called *dynamic*.
- Blocking mechanisms usually require access to a set of concepts $\tau(\ell)$ associated with given individual ℓ and sometimes a set of role links $\tau(\ell, \ell')$ associated with two individuals within the same branch $\mathcal{B}$:

$$\tau(\ell) \stackrel{\text{def}}{=} \{C \mid \ell : C \in \mathcal{B}\}$$

$$\tau(\ell, \ell') \stackrel{\text{def}}{=} \{R \mid \ell : \exists R.\{\ell'\} \in \mathcal{B}\}$$

MANCHESTER 1824 The University of Manchester

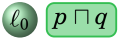
Subset Blocking

- If $\tau(\ell) \subseteq \tau(\ell')$ (and $\ell' < \ell$) then ℓ is blocked by ℓ'.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap q$.

MANCHESTER 1824 The University of Manchester

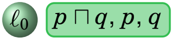
Subset Blocking

- If $\tau(\ell) \subseteq \tau(\ell')$ (and $\ell' < \ell$) then ℓ is blocked by ℓ'.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap q$.



Subset Blocking

- If $\tau(\ell) \subseteq \tau(\ell')$ (and $\ell' < \ell$) then ℓ is blocked by ℓ'.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap q$.



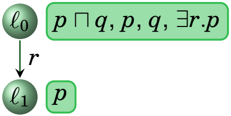
Subset Blocking

- If $\tau(\ell) \subseteq \tau(\ell')$ (and $\ell' < \ell$) then ℓ is blocked by ℓ'.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap q$.

ℓ_0 $p \sqcap q, p, q, \exists r.p$

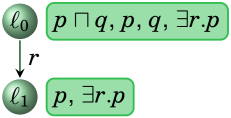
Subset Blocking

- If $\tau(\ell) \subseteq \tau(\ell')$ (and $\ell' < \ell$) then ℓ is blocked by ℓ'.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap q$.



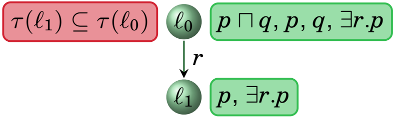
Subset Blocking

- If $\tau(\ell) \subseteq \tau(\ell')$ (and $\ell' < \ell$) then ℓ is blocked by ℓ'.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap q$.



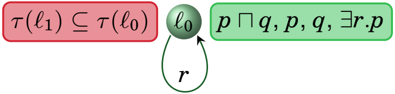
Subset Blocking

- If $\tau(\ell) \subseteq \tau(\ell')$ (and $\ell' < \ell$) then ℓ is blocked by ℓ'.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap q$.



Subset Blocking

- If $\tau(\ell) \subseteq \tau(\ell')$ (and $\ell' < \ell$) then ℓ is blocked by ℓ'.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap q$.



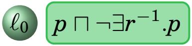
Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.

MANCHESTER 1824
The University of Manchester

Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.



Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.

ℓ_0: $p \sqcap \neg\exists r^{-1}.p, p, \neg\exists r^{-1}.p$

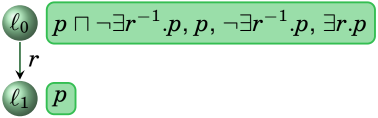
Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.

ℓ_0: $p \sqcap \neg\exists r^{-1}.p,\ p,\ \neg\exists r^{-1}.p,\ \exists r.p$

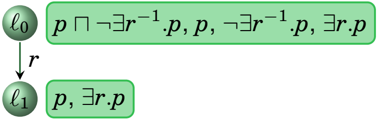
Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.



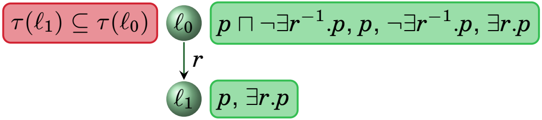
Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.



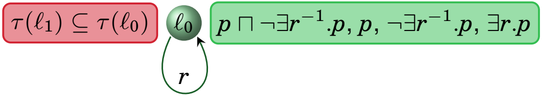
Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.



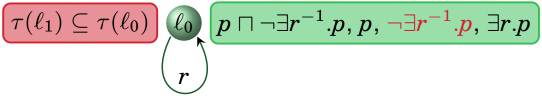
Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.



Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.



Equality Blocking

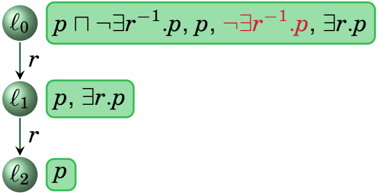
- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.

$$\ell_0 \quad p \sqcap \neg\exists r^{-1}.p,\ p,\ \neg\exists r^{-1}.p,\ \exists r.p$$

MANCHESTER 1824 The University of Manchester

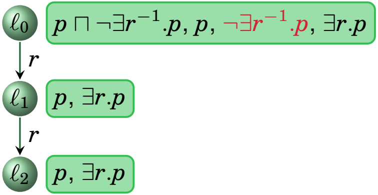
Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.



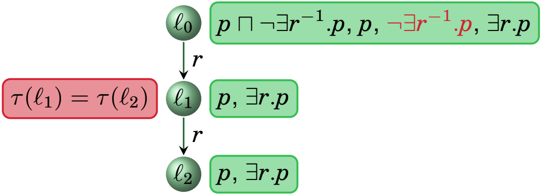
Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.



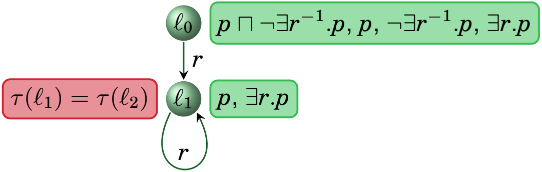
Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.



Equality Blocking

- If $\tau(\ell) = \tau(\ell')$ then ℓ is blocked by ℓ'.
- It is required with role inverse.
- Example: TBox $= \{\top \sqsubseteq \exists r.p\}$, a concept $p \sqcap \neg\exists r^{-1}.p$.



Pairwise Blocking

- A specific blocking mechanism required in extensions of $\mathcal{SHIF}$.
- Block ℓ by ℓ' only if for their predecessors ℓ_0 and ℓ'_0, respectively, the following conditions hold:
  - $\tau(\ell, \ell_0) = \tau(\ell', \ell'_0)$;
  - $\tau(\ell) = \tau(\ell')$;
  - $\tau(\ell_0) = \tau(\ell'_0)$.
- Example: RBox $= \{s \sqsubseteq r, r \in \text{Trans}\}$, concept $\neg p \sqcap \exists s.D \sqcap \forall r.\exists s.D$ where $D \stackrel{\text{def}}{=} p \sqcap (\leq 1\ s^{-1}) \sqcap \exists s^{-1}.\neg p$.

MANCHESTER 1824 The University of Manchester

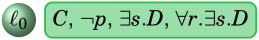
Pairwise Blocking

- A specific blocking mechanism required in extensions of $\mathcal{SHIF}$.
- Block ℓ by ℓ' only if for their predecessors ℓ_0 and ℓ'_0, respectively, the following conditions hold:
  - $\tau(\ell, \ell_0) = \tau(\ell', \ell'_0)$;
  - $\tau(\ell) = \tau(\ell')$;
  - $\tau(\ell_0) = \tau(\ell'_0)$.
- Example: RBox $= \{s \sqsubseteq r, r \in \text{Trans}\}$, concept $\neg p \sqcap \exists s.D \sqcap \forall r.\exists s.D$ where $D \stackrel{\text{def}}{=} p \sqcap (\leq 1\, s^{-1}) \sqcap \exists s^{-1}.\neg p$.

MANCHESTER 1824 The University of Manchester

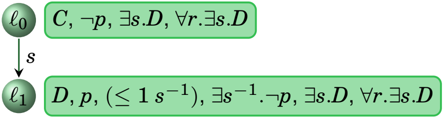
Pairwise Blocking

- A specific blocking mechanism required in extensions of $\mathcal{SHIF}$.
- Block ℓ by ℓ' only if for their predecessors ℓ_0 and ℓ'_0, respectively, the following conditions hold:
  - $\tau(\ell, \ell_0) = \tau(\ell', \ell'_0)$;
  - $\tau(\ell) = \tau(\ell')$;
  - $\tau(\ell_0) = \tau(\ell'_0)$.
- Example: RBox $= \{s \sqsubseteq r, r \in \text{Trans}\}$, concept $\neg p \sqcap \exists s.D \sqcap \forall r.\exists s.D$ where $D \stackrel{\text{def}}{=} p \sqcap (\leq 1\, s^{-1}) \sqcap \exists s^{-1}.\neg p$.



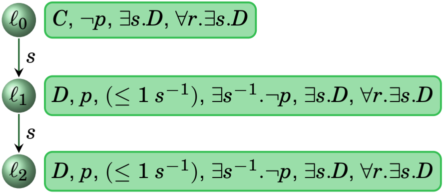
Pairwise Blocking

- A specific blocking mechanism required in extensions of $\mathcal{SHIF}$.
- Block ℓ by ℓ' only if for their predecessors ℓ_0 and ℓ'_0, respectively, the following conditions hold:
  - $\tau(\ell, \ell_0) = \tau(\ell', \ell'_0)$;
  - $\tau(\ell) = \tau(\ell')$;
  - $\tau(\ell_0) = \tau(\ell'_0)$.
- Example: RBox $= \{s \sqsubseteq r, r \in \text{Trans}\}$, concept $\neg p \sqcap \exists s.D \sqcap \forall r.\exists s.D$ where $D \stackrel{\text{def}}{=} p \sqcap (\leq 1\, s^{-1}) \sqcap \exists s^{-1}.\neg p$.



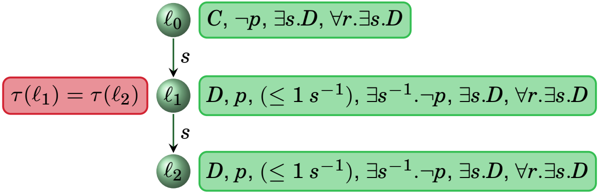
Pairwise Blocking

- A specific blocking mechanism required in extensions of $\mathcal{SHIF}$.
- Block ℓ by ℓ' only if for their predecessors ℓ_0 and ℓ'_0, respectively, the following conditions hold:
  - $\tau(\ell, \ell_0) = \tau(\ell', \ell'_0)$;
  - $\tau(\ell) = \tau(\ell')$;
  - $\tau(\ell_0) = \tau(\ell'_0)$.
- Example: RBox $= \{s \sqsubseteq r, r \in \text{Trans}\}$, concept $\neg p \sqcap \exists s.D \sqcap \forall r.\exists s.D$ where $D \stackrel{\text{def}}{=} p \sqcap (\leq 1\ s^{-1}) \sqcap \exists s^{-1}.\neg p$.



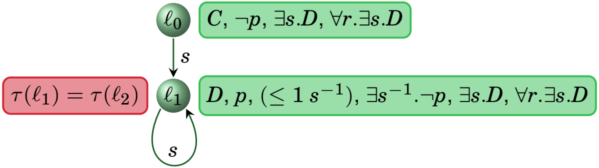
Pairwise Blocking

- A specific blocking mechanism required in extensions of $\mathcal{SHIF}$.
- Block ℓ by ℓ' only if for their predecessors ℓ_0 and ℓ'_0, respectively, the following conditions hold:
  - $\tau(\ell, \ell_0) = \tau(\ell', \ell'_0)$;
  - $\tau(\ell) = \tau(\ell')$;
  - $\tau(\ell_0) = \tau(\ell'_0)$.
- Example: RBox $= \{s \sqsubseteq r, r \in \text{Trans}\}$, concept $\neg p \sqcap \exists s.D \sqcap \forall r.\exists s.D$ where $D \stackrel{\text{def}}{=} p \sqcap (\leq 1\, s^{-1}) \sqcap \exists s^{-1}.\neg p$.



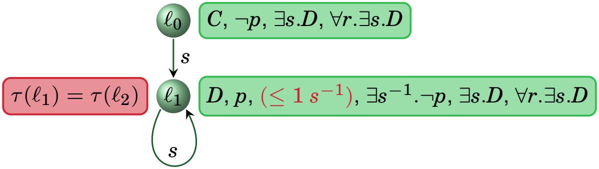
Pairwise Blocking

- A specific blocking mechanism required in extensions of $\mathcal{SHIF}$.
- Block ℓ by ℓ' only if for their predecessors ℓ_0 and ℓ'_0, respectively, the following conditions hold:
  - $\tau(\ell, \ell_0) = \tau(\ell', \ell'_0)$;
  - $\tau(\ell) = \tau(\ell')$;
  - $\tau(\ell_0) = \tau(\ell'_0)$.
- Example: RBox $= \{s \sqsubseteq r, r \in \text{Trans}\}$, concept $\neg p \sqcap \exists s.D \sqcap \forall r.\exists s.D$ where $D \stackrel{\text{def}}{=} p \sqcap (\leq 1\ s^{-1}) \sqcap \exists s^{-1}.\neg p$.



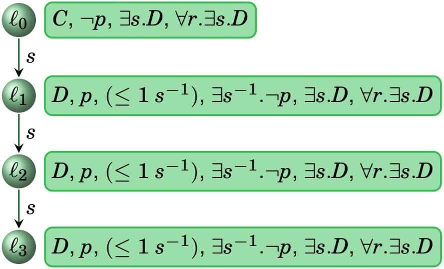
Pairwise Blocking

- A specific blocking mechanism required in extensions of $\mathcal{SHIF}$.
- Block ℓ by ℓ' only if for their predecessors ℓ_0 and ℓ'_0, respectively, the following conditions hold:
  - $\tau(\ell,\ell_0)=\tau(\ell',\ell'_0)$;
  - $\tau(\ell)=\tau(\ell')$;
  - $\tau(\ell_0)=\tau(\ell'_0)$.
- Example: RBox $=\{s\sqsubseteq r, r\in \text{Trans}\}$, concept $\neg p\sqcap\exists s.D\sqcap\forall r.\exists s.D$ where $D\stackrel{\text{def}}{=}p\sqcap(\leq 1\ s^{-1})\sqcap\exists s^{-1}.\neg p$.



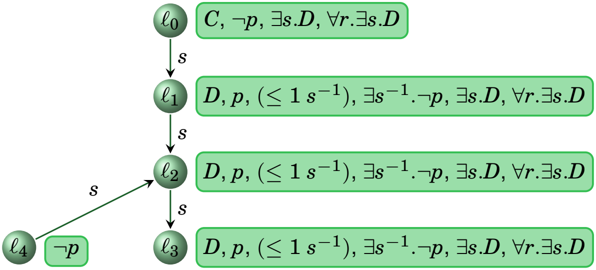
Pairwise Blocking

- A specific blocking mechanism required in extensions of $\mathcal{SHIF}$.
- Block ℓ by ℓ' only if for their predecessors ℓ_0 and ℓ'_0, respectively, the following conditions hold:
  - $\tau(\ell, \ell_0) = \tau(\ell', \ell'_0)$;
  - $\tau(\ell) = \tau(\ell')$;
  - $\tau(\ell_0) = \tau(\ell'_0)$.
- Example: RBox $= \{s \sqsubseteq r, r \in \text{Trans}\}$, concept $\neg p \sqcap \exists s.D \sqcap \forall r.\exists s.D$ where $D \stackrel{\text{def}}{=} p \sqcap (\leq 1\ s^{-1}) \sqcap \exists s^{-1}.\neg p$.



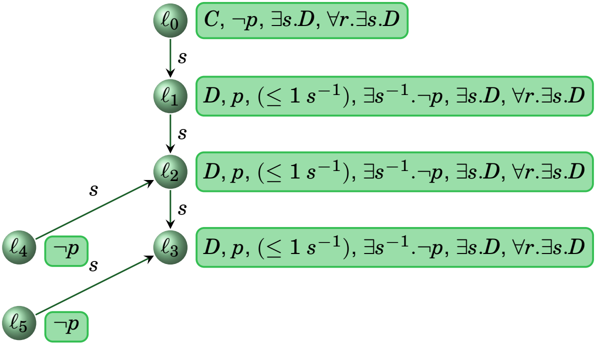
Pairwise Blocking

- A specific blocking mechanism required in extensions of $\mathcal{SHIF}$.
- Block ℓ by ℓ' only if for their predecessors ℓ_0 and ℓ'_0, respectively, the following conditions hold:
  - $\tau(\ell, \ell_0) = \tau(\ell', \ell'_0)$;
  - $\tau(\ell) = \tau(\ell')$;
  - $\tau(\ell_0) = \tau(\ell'_0)$.
- Example: RBox $= \{s \sqsubseteq r, r \in \text{Trans}\}$, concept $\neg p \sqcap \exists s.D \sqcap \forall r.\exists s.D$ where $D \overset{\text{def}}{=} p \sqcap (\leq 1\ s^{-1}) \sqcap \exists s^{-1}.\neg p$.



Pairwise Blocking

- A specific blocking mechanism required in extensions of $\mathcal{SHIF}$.
- Block ℓ by ℓ' only if for their predecessors ℓ_0 and ℓ'_0, respectively, the following conditions hold:
  - $\tau(\ell, \ell_0) = \tau(\ell', \ell'_0)$;
  - $\tau(\ell) = \tau(\ell')$;
  - $\tau(\ell_0) = \tau(\ell'_0)$.
- Example: RBox $= \{s \sqsubseteq r, r \in \text{Trans}\}$, concept $\neg p \sqcap \exists s.D \sqcap \forall r.\exists s.D$ where $D \stackrel{\text{def}}{=} p \sqcap (\leq 1\ s^{-1}) \sqcap \exists s^{-1}.\neg p$.



Successor and Anywhere Blocking

Successor blocking:

- Block ℓ by ℓ' only if ℓ is a successor of ℓ' along some path of role links in the branch.
- It is sufficient for logics which have the tree-model property.

Anywhere blocking:

- Blocks are allowed for any pair of individuals in given branch.

MANCHESTER 1824 The University of Manchester

Outline

1 Crash course on DLs
- ALC
- Universal modality
- TBox
- ABox and individuals
- RBox
- Problem statement
- Tableau procedure

2 Blocking mechanisms
- Subset blocking
- Equality blocking
- Pairwise blocking
- Successor and anywhere blocking

3 General blocking mechanism
- Unrestricted blocking rule
- Simulating existing blocking mechanisms

MANCHESTER 1824 The University of Manchester

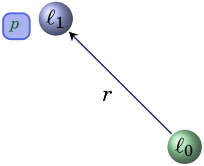
Blocking Problem for $\mathcal{ALBO}$

ℓ_0

$\exists r.p$
$\exists s.p$
$\exists t.\neg\exists t.\exists s^{-1}.\top$
$\neg\exists t.\exists\neg t.\neg\exists s^{-1}.\top$

Blocking Problem for $\mathcal{ALBO}$

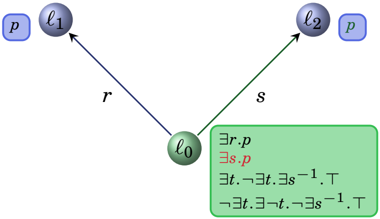
$$\frac{\ell : \exists R.C}{\ell : \exists R.\{\ell'\}, \quad \ell' : C}$$



$\exists r.p$
$\exists s.p$
$\exists t.\neg\exists t.\exists s^{-1}.\top$
$\neg\exists t.\exists\neg t.\neg\exists s^{-1}.\top$

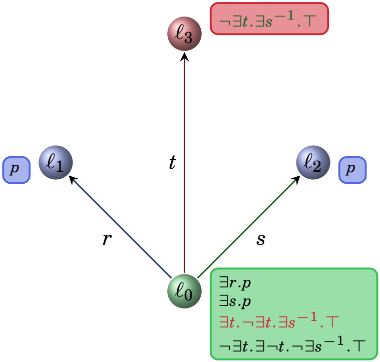
Blocking Problem for $\mathcal{ALBO}$

$$\frac{\ell : \exists R.C}{\ell : \exists R.\{\ell'\}, \quad \ell' : C}$$



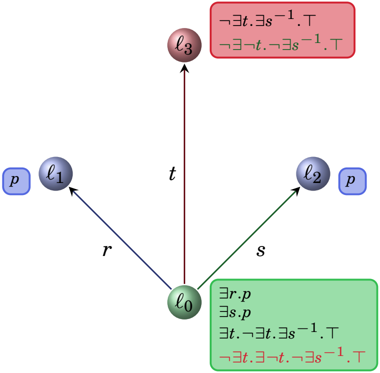
$\ell_1 = \ell_2$?

Blocking Problem for $\mathcal{ALBO}$



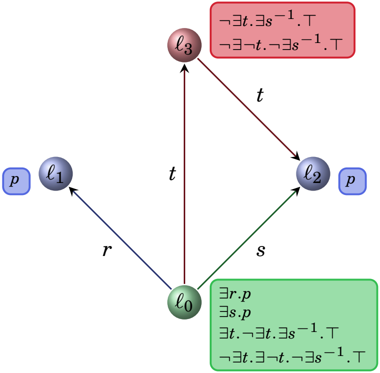
$$\frac{\ell : \exists R.C}{\ell : \exists R.\{\ell'\}, \quad \ell' : C}$$

Blocking Problem for $\mathcal{ALBO}$



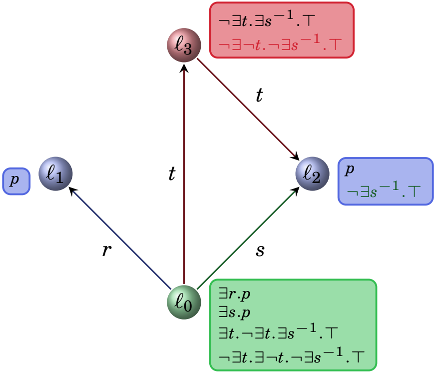
$$\frac{\ell : \neg\exists R.C, \quad \ell : \exists R.\{\ell'\}}{\ell' : \neg C}$$

Blocking Problem for $\mathcal{ALBO}$



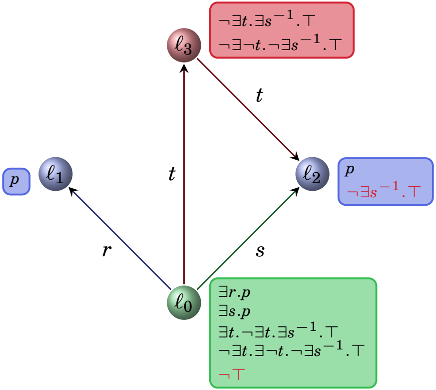
$$\frac{\ell : \neg\exists\neg R.C, \quad \ell' : \{\ell'\}}{\ell : \exists R.\{\ell'\} \mid \ell' : \neg C}$$

Blocking Problem for $\mathcal{ALBO}$



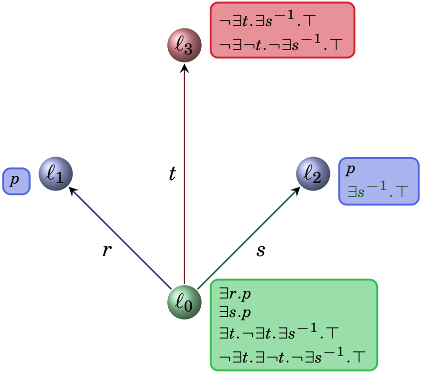
$$\frac{\ell : \neg\exists R.C, \quad \ell : \exists R.\{\ell'\}}{\ell' : \neg C}$$

Blocking Problem for $\mathcal{ALBO}$



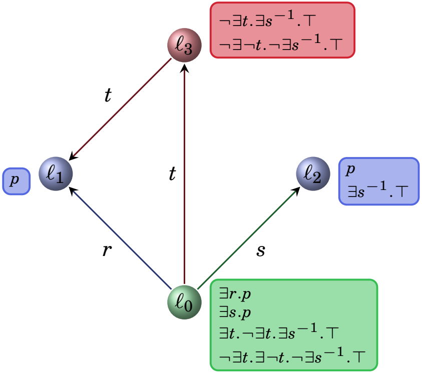
$$\frac{\ell : \neg\exists R^{-1}.C, \quad \ell' : \exists R.\{\ell\}}{\ell' : \neg C}$$

Blocking Problem for $\mathcal{ALBO}$



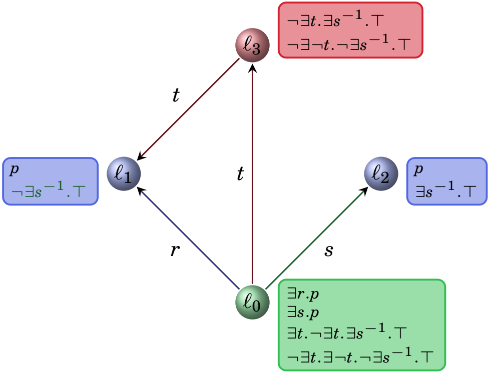
$$\frac{\ell : \neg\exists\neg R.C, \quad \ell' : \{\ell'\}}{\ell : \exists R.\{\ell'\} \mid \ell' : \neg C}$$

Blocking Problem for $\mathcal{ALBO}$



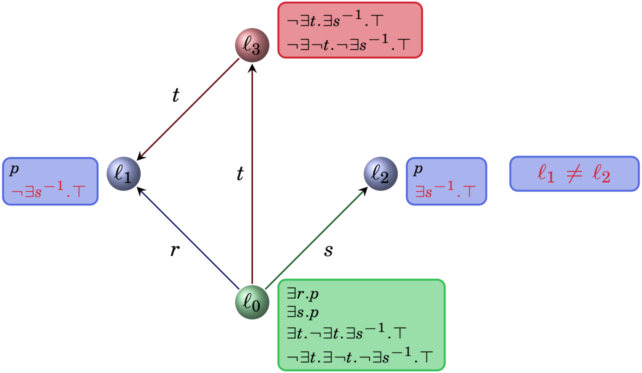
$$\frac{\ell : \neg\exists\neg R.C, \quad \ell' : \{\ell'\}}{\ell : \exists R.\{\ell'\} \mid \ell' : \neg C}$$

Blocking Problem for $\mathcal{ALBO}$



$$\frac{\ell : \neg\exists R.C, \quad \ell : \exists R.\{\ell'\}}{\ell' : \neg C}$$

Blocking Problem for $\mathcal{ALBO}$



Unrestricted Blocking Rule

$$\text{(ub)}\ \frac{\ell : \{\ell\},\quad \ell' : \{\ell'\}}{\ell : \{\ell'\} \mid \ell : \neg\{\ell'\}}$$

Strategy conditions:

1. any rule is applied at most once to the same set of premises.
2. the (∃) rule is not applied to role assertion expressions.
3. if $\ell : \{\ell'\}$ in current branch and $\ell < \ell'$ then no applications of the (∃) rule to expressions $\ell' : \exists R.C$ are performed[1]
4. in every open branch there is some node from which point onwards, all possible applications of the (ub) rule have been performed before any application of the (∃) rule

[1] $<$ reflects the order in which the individuals are introduced

MANCHESTER 1824 The University of Manchester

Unrestricted Blocking Rule

$$\text{(ub)}\ \frac{\ell : \{\ell\},\quad \ell' : \{\ell'\}}{\ell : \{\ell'\} \mid \ell : \neg\{\ell'\}}$$

Strategy conditions:

1. any rule is applied at most once to the same set of premises.
2. the ($\exists$) rule is not applied to role assertion expressions.
3. if $\ell : \{\ell'\}$ in current branch and $\ell < \ell'$ then no applications of the ($\exists$) rule to expressions $\ell' : \exists R.C$ are performed[1]
4. in every open branch there is some node from which point onwards, all possible applications of the (ub) rule have been performed before any application of the ($\exists$) rule

[1] $<$ reflects the order in which the individuals are introduced

Unrestricted Blocking Rule

$$\text{(ub)}\ \frac{\ell : \{\ell\},\quad \ell' : \{\ell'\}}{\ell : \{\ell'\} \mid \ell : \neg\{\ell'\}}$$

Strategy conditions:

1. any rule is applied at most once to the same set of premises.
2. the ($\exists$) rule is not applied to role assertion expressions.
3. if $\ell : \{\ell'\}$ in current branch and $\ell < \ell'$ then no applications of the ($\exists$) rule to expressions $\ell' : \exists R.C$ are performed[1]
4. in every open branch there is some node from which point onwards, all possible applications of the (ub) rule have been performed before any application of the ($\exists$) rule

[1] $<$ reflects the order in which the individuals are introduced

Unrestricted Blocking Rule

$$\text{(ub)}\ \frac{\ell : \{\ell\},\ \ \ell' : \{\ell'\}}{\ell : \{\ell'\} \mid \ell : \neg\{\ell'\}}$$

Strategy conditions:

1. any rule is applied at most once to the same set of premises.
2. the ($\exists$) rule is not applied to role assertion expressions.
3. if $\ell : \{\ell'\}$ in current branch and $\ell < \ell'$ then no applications of the ($\exists$) rule to expressions $\ell' : \exists R.C$ are performed[1]
4. in every open branch there is some node from which point onwards, all possible applications of the (ub) rule have been performed before any application of the ($\exists$) rule

MANCHESTER 1824 The University of Manchester

[1] $<$ reflects the order in which the individuals are introduced

Unrestricted Blocking Rule

$$\text{(ub)}\ \frac{\ell : \{\ell\},\quad \ell' : \{\ell'\}}{\ell : \{\ell'\} \mid \ell : \neg\{\ell'\}}$$

Strategy conditions:

1. any rule is applied at most once to the same set of premises.
2. the ($\exists$) rule is not applied to role assertion expressions.
3. if $\ell : \{\ell'\}$ in current branch and $\ell < \ell'$ then no applications of the ($\exists$) rule to expressions $\ell' : \exists R.C$ are performed[1]
4. in every open branch there is some node from which point onwards, all possible applications of the (ub) rule have been performed before any application of the ($\exists$) rule

MANCHESTER 1824 The University of Manchester

[1] $<$ reflects the order in which the individuals are introduced

Simulating Existing Blocking Mechanisms

- Add conditions for blocking as constraints on application of the unrestricted blocking rule.
- Ensure that tableau uses a fair strategy.
- Ensure the condition 4.
- Tableau algorithm is guaranteed to terminate for logics with the effective finite model property.

Simulating Existing Blocking Mechanisms

- Add conditions for blocking as constraints on application of the unrestricted blocking rule.
- Ensure that tableau uses a fair strategy.
- Ensure the condition 4.
- Tableau algorithm is guaranteed to terminate for logics with the effective finite model property.

MANCHESTER 1824 The University of Manchester

Simulating Existing Blocking Mechanisms

- Add conditions for blocking as constraints on application of the unrestricted blocking rule.
- Ensure that tableau uses a fair strategy.
- Ensure the condition 4.
- Tableau algorithm is guaranteed to terminate for logics with the effective finite model property.

Simulating Existing Blocking Mechanisms

- Add conditions for blocking as constraints on application of the unrestricted blocking rule.
- Ensure that tableau uses a fair strategy.
- Ensure the condition 4.
- Tableau algorithm is guaranteed to terminate for logics with the effective finite model property.

k you! Questions?

nk you! Questions?

ank you! Questions?

MANCHESTER 1824
The University of Manchester

ank you! Questions?

Thank you! Questions?

Thank you! Questions?

Thank you! Questions?

Thank you! Questions?

Thank you! Questions?

Thank you! Questions?

Thank you! Questions?

Thank you! Questions?

Thank you! Questions?

Thank you! Questions?