A Tableau algorithm for $\mathcal{ALCSCC}$

Ryny Khy

January 12, 2021

Motivation

- Reasoning in data base

Motivation

- Reasoning in data base
- Satisfiability check for reasoning

Motivation

- Reasoning in data base
- Satisfiability check for reasoning
- Tableau algorithm for satisfiability check

Tableau Algorithm

Main Idea (for $\mathcal{ALCQ}$):

$$x : A \sqcap B \sqcap \exists r.C$$

x $\quad$ A

x must be in A

Tableau Algorithm

Main Idea (for $\mathcal{ALCQ}$):

$x : A \sqcap B \sqcap \exists r.C$

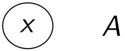
x A

x must be in A

Tableau Algorithm

Main Idea (for $\mathcal{ALCQ}$):

$$x : A \sqcap B \sqcap \exists r.C$$



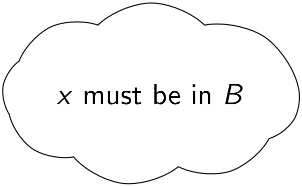


Tableau Algorithm

Main Idea (for $\mathcal{ALCQ}$):

$x : A \sqcap B \sqcap \exists r.C$

x $\quad A, B$

x must be in B

Tableau Algorithm

Main Idea (for $\mathcal{ALCQ}$):

$x : A \sqcap B \sqcap \exists r.C$

x A, B

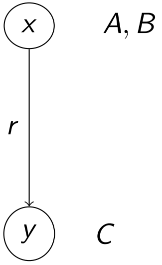
x must have
an r-successor
in C

Tableau Algorithm

Main Idea (for $\mathcal{ALCQ}$):

$x : A \sqcap B \sqcap \exists r.C$

x must have
an r-successor
in C



Goal

Tableau algorithm for $\mathcal{ALCSCC}$ concepts

$\mathcal{ALCSCC}$: successors

$\cancel{\exists r.C}$ $\cancel{\forall r.C}$ $\cancel{\leq m\, r.C}$ $\cancel{\geq m\, r.C}$

$$\boxed{succ(c)}$$

c: **set constraint** or **cardinality constraint**

$\mathcal{ALCSCC}$: constraints

set constraint:

- $hasChild \cap Female \subseteq Mother$

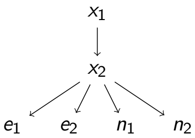
cardinality constraint

- $2\ dvd\ |hasLegs|$
- $|Edges| \leq |Nodes|$

$\mathcal{ALCSCC}$: constraints

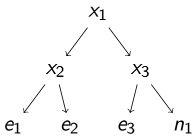
set constraint:

- $succ(2\ dvd\ |Edges|) \subseteq succ(|Edges| = |Nodes|)$



cardinality constraint

- $succ(2\ dvd\ |Edges|) \leq succ(|Edges| = |Nodes|)$



Problem with successors constraints

$$x : \mathit{succ}(|s| > 1) \sqcap \mathit{succ}(|r| \leq |s|) \sqcap \mathit{succ}(|r| > |s|)$$

Problem with successors constraints

$$x: \; succ(|s| > 1) \sqcap succ(|r| \leq |s|) \sqcap succ(|r| > |s|)$$

- endless loop of adding r- and s-successors

Problem with successors constraints

$$x : \ succ(|s| > 1) \sqcap succ(|r| \leq |s|) \sqcap succ(|r| > |s|)$$

- endless loop of adding r- and s-successors
- blocking?

Problem with blocking

$$x : succ(3 \cdot |r| \leq 5 \cdot |s|) \sqcap succ(5 \cdot |s| \leq 3 \cdot |r|) \sqcap succ(|r| > 1)$$

s-successors	r-successors
0	0

Problem with blocking

$$x : succ(3 \cdot |r| \leq 5 \cdot |s|) \sqcap succ(5 \cdot |s| \leq 3 \cdot |r|) \sqcap \underline{succ(|r| > 1)}$$

s-successors	r-successors
0	1

Problem with blocking

$$x : \underline{succ(3 \cdot |r| \leq 5 \cdot |s|)} \sqcap succ(5 \cdot |s| \leq 3 \cdot |r|) \sqcap succ(|r| > 1)$$

s-successors	r-successors
0	1

Problem with blocking

$$x : \underline{succ(3 \cdot |r| \leq 5 \cdot |s|)} \sqcap succ(5 \cdot |s| \leq 3 \cdot |r|) \sqcap succ(|r| > 1)$$

s-successors

1

r-successors

1

Problem with blocking

$$x : succ(3 \cdot |r| \leq 5 \cdot |s|) \sqcap \underline{succ(5 \cdot |s| \leq 3 \cdot |r|)} \sqcap succ(|r| > 1)$$

s-successors	r-successors
1	1

Problem with blocking

$$x : \underline{succ(3 \cdot |r| \leq 5 \cdot |s|)} \sqcap succ(5 \cdot |s| \leq 3 \cdot |r|) \sqcap succ(|r| > 1)$$

s-successors	r-successors
1	1

QFBAPA formula and solver

$$x : succ(3 \cdot |r| \leq 5 \cdot |s|) \sqcap succ(5 \cdot |s| \leq 3 \cdot |r|) \sqcap succ(|r| > 1)$$

QFBAPA formula and solver

$$x : succ(3 \cdot |r| \leq 5 \cdot |s|) \sqcap succ(5 \cdot |s| \leq 3 \cdot |r|) \sqcap succ(|r| > 1)$$

$$\downarrow$$

$$3 \cdot |X_r| \leq 5 \cdot |X_s| \wedge 5 \cdot |X_s| \leq 3 \cdot |X_r| \wedge |X_r| > 1$$

QFBAPA formula and solver

$$x : succ(3 \cdot |r| \leq 5 \cdot |s|) \sqcap succ(5 \cdot |s| \leq 3 \cdot |r|) \sqcap succ(|r| > 1)$$

$$\downarrow$$

$$3 \cdot |X_r| \leq 5 \cdot |X_s| \wedge 5 \cdot |X_s| \leq 3 \cdot |X_r| \wedge |X_r| > 1$$

$$\downarrow$$

$X_r = \{r_1, r_2, \ldots, r_5\}$ and $X_s = \{s_1, s_2, s_3\}$

QFBAPA formula and solver

What about

$$3 \cdot |X_r| \leq 5 \cdot |X_s| \wedge 5 \cdot |X_s| \leq 3 \cdot |X_r| \wedge |X_r| > 1$$

$$\downarrow$$

$$|X_r| = 10 \text{ and } |X_s| = 6$$

QFBAPA formula and solver

What about

$$3 \cdot |X_r| \leq 5 \cdot |X_s| \wedge 5 \cdot |X_s| \leq 3 \cdot |X_r| \wedge |X_r| > 1$$

$$\Big\downarrow$$

$$|X_r| = 100 \text{ and } |X_s| = 60$$

QFBAPA formula and solver

$\rightarrow$ Upper bound

$\rightarrow$ ILP

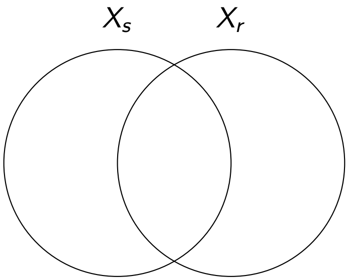
Venn regions

Problem: Are the variables disjoint or not?

Venn regions

Problem: Are the variables disjoint or not?

Solution: Venn region



Venn regions

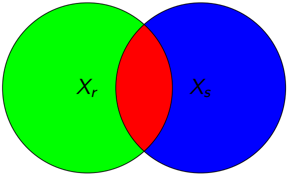
Two variables X_s and X_r $\rightarrow$ four Venn regions:

$v_1 = X_s \cap X_r$

$v_2 = X_s^{\neg} \cap X_r$

$v_3 = X_s \cap X_r^{\neg}$

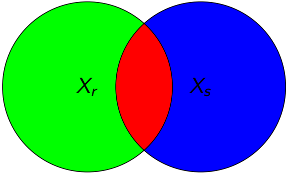
$v_4 = X_s^{\neg} \cap X_r^{\neg}$



Venn regions

Two variables X_s and X_r $\rightarrow$ four Venn regions:

$v_1 = X_s \cap X_r$
$v_2 = X_s^\neg \cap X_r$
$v_3 = X_s \cap X_r^\neg$
$v_4 = X_s^\neg \cap X_r^\neg$



$X_s = v_1 \cup v_3$ and $X_r = v_1 \cup v_2$

Venn regions

Franz Baader [1]:

For every QFBAPA formula ϕ there is a number N, which is polynomial in the size of ϕ and can be computed in polynomial time such, that for every solution σ of ϕ there exists a solution σ' of ϕ such that:

- $|\{v|v \text{ is a Venn region and } \sigma'(v) \neq \emptyset\}| \leq N$
- $\{v|v \text{ is a Venn region and } \sigma'(v) \neq \emptyset\} \subseteq \{v|v \text{ is a Venn region and } \sigma(v) \neq \emptyset\}$

[1] A New Description Logic with Set Constraints and Cardinality Constraints on Role Successors. Springer International Publishing: 43-59, 2017

Venn regions

In short:

2^n Venn regions,

n : amount of set variables

N Venn regions,

N : polynomial in the size of formula

Transforming formula into ILP

$$3 \cdot |X_r| \leq 5 \cdot |X_s| \wedge 5 \cdot |X_s| \leq 3 \cdot |X_r| \wedge |X_r| > 1$$

$$\downarrow$$

$$\begin{aligned} -5 \cdot |X_s| + 3 \cdot |X_r| &\leq 0 \\ 5 \cdot |X_s| - 3 \cdot |X_r| &\leq 0 \\ |X_r| &> 1 \end{aligned}$$

Transforming formula into ILP

$$\downarrow$$

$$\begin{aligned} -5 \cdot |X_s| + 3 \cdot |X_r| &\leq 0 \\ 5 \cdot |X_s| - 3 \cdot |X_r| &\leq 0 \\ |X_r| &> 1 \end{aligned}$$

$$\downarrow$$

$$\begin{aligned} -5 \cdot |X_s| + 3 \cdot |X_r| + l_1 &= 0 \\ 5 \cdot |X_s| - 3 \cdot |X_r| + l_2 &= 0 \\ |X_r| - l_3 &= 2 \end{aligned}$$

Transforming formula into ILP

$$\downarrow$$

$$\begin{aligned} -5 \cdot |X_s| + 3 \cdot |X_r| + l_1 &= 0 \\ 5 \cdot |X_s| - 3 \cdot |X_r| + l_2 &= 0 \\ |X_r| - l_3 &= 2 \end{aligned}$$

$$\downarrow$$

$$\begin{aligned} -5 \cdot |v_1 \cup v_3| + 3 \cdot |v_1 \cup v_2| + l_1 &= 0 \\ 5 \cdot |v_1 \cup v_3| - 3 \cdot |v_1 \cup v_2| + l_2 &= 0 \\ |v_1 \cup v_2| - l_3 &= 2 \end{aligned}$$

ILP

$$
\begin{array}{c}
\\
-5\cdot|X_s|+2\cdot|X_r|+l_1 \\
5\cdot|X_s|-2\cdot|X_r|+l_2 \\
|X_r|-l_3
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & l_1 & l_2 & l_3 \end{array} \\
\begin{pmatrix}
-5+3 & 3 & -5 & 0 & 1 & 0 & 0 \\
5-3 & -3 & 5 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & -1
\end{pmatrix}
\end{array}
\cdot
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}
$$

$$
= \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}
$$

Upper bound

Christos H. Papadimitriou[2]:

For each ILP $Ax = b$, A $m \times n$ matrix, $x \in \mathbb{N}^n$ and $b \in \mathbb{R}^m$ there exists an upper bound M and a solution $x' \in \{0, \ldots, M\}^n$ such that

$$Ax' = b$$

[2] On the Complexity of Integer Programming. J. ACM,28(4):765-768, Oct. 1981.

Tableau for $\mathcal{ALCSCC}$

Now we consider the rules for the tableau algorithm.

Decomposing rules

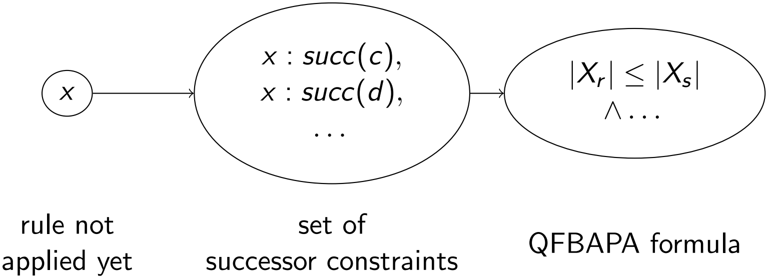
- $\sqcap$-rule:
  If $x : A \sqcap B$ is in ABox then $x : A$ and $x : B$ must be in ABox
- $\sqcup$-rule:
  If $x : A \sqcup B$ is in ABox then $x : A$ or $x : B$ must be in ABox

Decomposing rules

- $\sqcap$-rule:
  If $x : A \sqcap B$ is in ABox then $x : A$ and $x : B$ must be in ABox
- $\sqcup$-rule:
  If $x : A \sqcup B$ is in ABox then $x : A$ or $x : B$ must be in ABox

Always apply when possible (higher priority)

successor-rule



successor-rule

$$\boxed{|X_r| \leq |X_s| \wedge \dots} \longrightarrow \text{calculate upper bound}$$

successor-rule

$$\begin{array}{c} |X_r| \leq |X_s| \\ \land \ldots \end{array} \longrightarrow \begin{array}{c} \text{calculate} \\ \text{upper bound} \end{array} \longrightarrow \begin{array}{c} \text{use} \\ \text{QFBAPA} \\ \text{solver} \end{array}$$

successor-rule

use QFBAPA solver

If it returns unsatisfiable

add $x : \perp$

successor-rule

use QFBAPA solver

If it returns unsatisfiable

add $x : \perp$

If it returns a solution

add variables and its constraints

Example: Concept

$$pancake : succ(|contains \cap Milk| \geq |contains \cap Flavour|) \sqcap$$

$$succ(|contains \cap Milk| \leq 2 \cdot |contains \cap Flavour|) \sqcap$$

$$succ(|contains \cap Egg| \geq 1)$$

Example: Concept

$$\mathit{pancake} : \mathit{succ}(|\mathit{contains} \cap \mathit{Milk}| \geq |\mathit{contains} \cap \mathit{Flavour}|) \sqcap$$

$$\mathit{succ}(|\mathit{contains} \cap \mathit{Milk}| \leq 2 \cdot |\mathit{contains} \cap \mathit{Flavour}|) \sqcap$$

$$\mathit{succ}(|\mathit{contains} \cap \mathit{Egg}| \geq 1)$$

Example: Decomposing

We use the $\sqcap$-rule twice and add

Example: Decomposing

We use the $\sqcap$-rule twice and add

$$\mathit{pancake} : \mathit{succ}(|\mathit{contains} \cap \mathit{Milk}| \geq |\mathit{contains} \cap \mathit{Flavour}|),$$

$$\mathit{pancake} : \mathit{succ}(|\mathit{contains} \cap \mathit{Milk}| \leq 2 \cdot |\mathit{contains} \cap \mathit{Flavour}|),$$

$$\mathit{pancake} : \mathit{succ}(|\mathit{contains} \cap \mathit{Egg}| \geq 1)$$

Example: *successor*-rule

$$succ(|contains \cap Milk| \geq |contains \cap Flavour|)$$
$$succ(|contains \cap Milk| \leq 2 \cdot |contains \cap Flavour|)$$
$$succ(|contains \cap Egg| \geq 1)$$

Example: *successor*-rule

$$succ(|contains \cap Milk| \geq |contains \cap Flavour|)$$
$$succ(|contains \cap Milk| \leq 2 \cdot |contains \cap Flavour|)$$
$$succ(|contains \cap Egg| \geq 1)$$

$$\downarrow$$

$$|X_{contains} \cap X_{Milk}| \geq |X_{contains} \cap X_{Flavour}| \wedge$$
$$|X_{contains} \cap X_{Milk}| \leq 2 \cdot |X_{contains} \cap X_{Flavour}| \wedge$$
$$|X_{contains} \cap X_{Egg}| \geq 1$$

Example: ILP

- 4 set variables $\rightarrow 2^4 = 16$ Venn regions

Example: ILP

- 4 set variables $\rightarrow 2^4 = 16$ Venn regions
- BUT: every successor is a *contains*-successor

Example: ILP

- 4 set variables $\rightarrow 2^4 = 16$ Venn regions
- BUT: every successor is a *contains*-successor
- Assume *Milk*, *Flavour* and *Egg* are disjoint

Example: ILP

- 4 set variables $\rightarrow 2^4 = 16$ Venn regions
- BUT: every successor is a *contains*-successor
- Assume *Milk*, *Flavour* and *Egg* are disjoint

$$
\begin{aligned}
|X_{contains} \cap X_{Milk}| - |X_{contains} \cap X_{Flavour}| - l_1 &= 0 \\
|X_{contains} \cap X_{Milk}| - 2 \cdot |X_{contains} \cap X_{Flavour}| + l_2 &= 0 \\
|X_{contains} \cap X_{Egg}| - l_3 &= 1
\end{aligned}
$$

Example: ILP

$$\begin{array}{r c}
 & \begin{array}{ccccccc} X_M & X_F & X_E & X_c & l_1 & l_2 & l_3 \end{array} \\
\begin{array}{r} \text{Milk}<\text{Flavour} \\ \text{not to much milk} \\ \text{Eggs} \end{array} & \begin{pmatrix} 1 & -1 & 0 & 1 & -1 & 0 & 0 \\ 1 & -2 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 \end{pmatrix}
\end{array} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Example: *successor*-rule

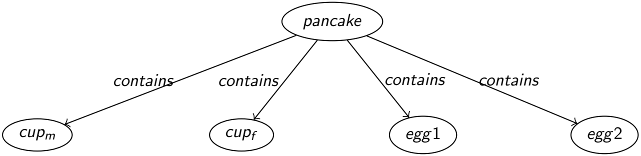
Assume the solver returns

$$X_{Milk} = \{cup_m\},$$
$$X_{Flavour} = \{cup_f\}$$
$$X_{egg} = \{egg1, egg2\} \text{ and}$$
$$X_{contains} = X_{Milk} \cup X_{Flavour} \cup X_{Egg}$$

Example: *successor*-rule

Assume the solver returns

$$X_{Milk} = \{cup_m\},$$
$$X_{Flavour} = \{cup_f\}$$
$$X_{egg} = \{egg1, egg2\} \text{ and}$$
$$X_{contains} = X_{Milk} \cup X_{Flavour} \cup X_{Egg}$$



- Tableau for $\mathcal{ALCSCC}$
- 2ExpSpace because of upper bound

- Tableau for $\mathcal{ALCSCC}$
- 2ExpSpace because of upper bound

For the future:

- use/find smaller upper bound
- extend tableau for whole knowledge base
- tableau without QFBAPA solver