# A Tableau algorithm for $\mathcal{ALCSCC}$

Ryny Khy

January 2, 2021

1. Introduction
2. Difficulties with $\mathcal{ALCSCC}$
3. Dealing with successor constraints
4. Tableau for $\mathcal{ALCSCC}$
5. Example

# Motivation

- Reasoning in data base

# Motivation

- Reasoning in data base
- Satiasfiability check for reasoning

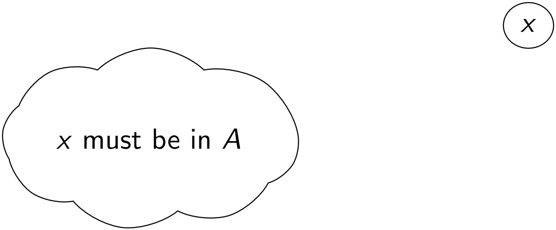
# Motivation

- Reasoning in data base
- Satiasfiability check for reasoning
- Tableau algorithm for satisfiabilty check

# Tableau Algorithm

Main Idea:

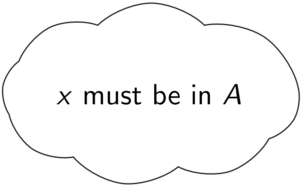
$$x : A \sqcap B \sqcap \exists r.C$$



# Tableau Algorithm

Main Idea:

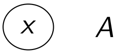
$$x : A \sqcap B \sqcap \exists r.C$$

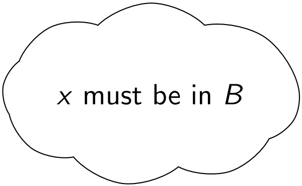



# Tableau Algorithm

Main Idea:

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



# Tableau Algorithm

Main Idea:

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

$x$ $\quad A, B$

$x$ must be in $B$

# Tableau Algorithm

Main Idea:

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

$x$ $A, B$

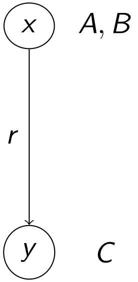
$x$ must have
an $r$-successor
in $C$

# Tableau Algorithm

Main Idea:

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

$x$ must have
an $r$-successor
in $C$



## $\mathcal{ALCSCC}$: 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:**

- $r \subseteq s$
- $C \cap r \subseteq D$
- $succ(C \cap r) \subseteq succ(D)$

**cardinality constraint**

- $2\, dvd\, |r|$
- $|C \cap r| \leq |D|$
- $|succ(C \cap r)| \leq |succ(D)|$

# Problem with successors constraints

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

# 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

$$x : succ(2 \cdot |r| \leq 5 \cdot |s|) \sqcap succ(5 \cdot |s| \leq 2 \cdot |r|) \sqcap succ(|r| > 1)$$

$s$-successors

0

$r$-successors

0

# Problem with blocking

$$x : succ(2 \cdot |r| \leq 5 \cdot |s|) \sqcap succ(5 \cdot |s| \leq 2 \cdot |r|) \sqcap \underline{succ(|r| > 1)}$$

| $s$-successors | $r$-successors |
|---|---|
| 0 | 1 |

# Problem with blocking

$$x : \underline{succ(2 \cdot |r| \leq 5 \cdot |s|)} \sqcap succ(5 \cdot |s| \leq 2 \cdot |r|) \sqcap succ(|r| > 1)$$

$s$-successors

1

$r$-successors

1

# QFBAPA formula and solver

$$x : succ(2 \cdot |r| \leq 5 \cdot |s|) \sqcap succ(5 \cdot |s| \leq 2 \cdot |r|) \sqcap succ(|r| > 1)$$

# QFBAPA formula and solver

$$x : succ(2 \cdot |r| \leq 5 \cdot |s|) \sqcap succ(5 \cdot |s| \leq 2 \cdot |r|) \sqcap succ(|r| > 1)$$

$$\downarrow$$

$$2 \cdot |X_r| \leq 5 \cdot |X_s| \wedge 5 \cdot |X_s| \leq 2 \cdot |X_r| \wedge |X_r| > 1$$

# QFBAPA formula and solver

$$x : succ(2 \cdot |r| \leq 5 \cdot |s|) \sqcap succ(5 \cdot |s| \leq 2 \cdot |r|) \sqcap succ(|r| > 1)$$

$$\downarrow$$

$$2 \cdot |X_r| \leq 5 \cdot |X_s| \wedge 5 \cdot |X_s| \leq 2 \cdot |X_r| \wedge |X_r| > 1$$

$$\downarrow$$

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

# QFBAPA formula and solver

What about

$$2 \cdot |X_r| \leq 5 \cdot |X_s| \wedge 5 \cdot |X_s| \leq 2 \cdot |X_r| \wedge |X_r| > 1$$

$$\downarrow$$

$$|X_r| = 10 \text{ and } |X_s| = 4$$

# QFBAPA formula and solver

What about

$$2 \cdot |X_r| \leq 5 \cdot |X_s| \wedge 5 \cdot |X_s| \leq 2 \cdot |X_r| \wedge |X_r| > 1$$

$$\downarrow$$

$$|X_r| = 100 \text{ and } |X_s| = 40$$

# QFBAPA formula and solver

We have infinite possible solutions

# QFBAPA formula and solver

We have infinite possible solutions

Do we need all of them?

# QFBAPA formula and solver

We have infinite possible solutions

Do we need all of them?

No: Consider formula as ILP and calculate an upper bound

# ILP and upper bound

$$2 \cdot |X_r| \leq 5 \cdot |X_s| \wedge 5 \cdot |X_s| \leq 2 \cdot |X_r| \wedge |X_r| > 1$$

$$\downarrow$$

$$\begin{aligned} -5 \cdot |X_s| + 2 \cdot |X_r| &\leq 0 \\ 5 \cdot |X_s| - 2 \cdot |X_r| &\leq 0 \\ -|X_r| &< -1 \end{aligned}$$

$$\downarrow$$

$$\begin{aligned} -5 \cdot |X_s| + 2 \cdot |X_r| + l_1 &= 0 \\ 5 \cdot |X_s| - 2 \cdot |X_r| + l_2 &= 0 \\ -|X_r| + l_3 &= -1 \end{aligned}$$

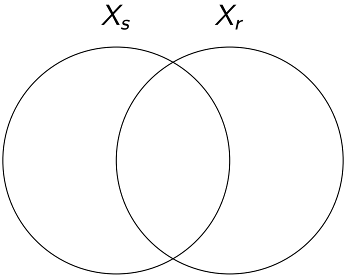
# ILP and upper bound

**Problem**: Are the variables disjoint or not?

# ILP and upper bound

**Problem**: Are the variables disjoint or not?

**Solution**: Venn region



# Venn regions

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

$$\begin{aligned}
v_1 &= X_s \cap X_r \\
v_2 &= X_s^\neg \cap X_r \\
v_3 &= X_s \cap X_r^\neg \\
v_4 &= X_s^\neg \cap X_r^\neg
\end{aligned}$$

# 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 $X_r = v_1 \cup v_2$

# ILP

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

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

# ILP and upper bound

Christos H. Papadimitriou[1]:

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

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

# Tableau algorithm

- $\sqcap$-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

# Tableau algorithm

- $\sqcap$-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)

# Tableau algorithm

- *successor*-rule:

$$x \longrightarrow \begin{matrix} x : succ(c), \\ x : succ(d), \\ \dots \end{matrix} \longrightarrow \begin{matrix} |X_r| \leq |X_s| \\ \wedge \dots \end{matrix}$$

rule not applied yet

set of successor constraints

QFBAPA formula

# Tableau algorithm

- *successor*-rule:

$$|X_r| \leq |X_s| \\ \wedge \ldots$$

$\longrightarrow$ calculate upper bound

# Tableau algorithm

- *successor*-rule:

$$|X_r| \leq |X_s| \wedge \ldots \longrightarrow \text{calculate upper bound} \longrightarrow \text{use QFBAPA solver}$$

# Tableau algorithm

- *successor*-rule:

use QFBAPA solver

$\downarrow$

If it returns unsatisfiable

$\downarrow$

add $x : \perp$

# Tableau algorithm

- *successor*-rule:

use QFBAPA solver

If it returns unsatisfiable

add $x : \perp$

If it returns a solution

add variables and its constraints

# Example

$$\mathit{fruitcake} : \mathit{succ}(|\mathit{need} \cap \mathit{Cake}| \geq 1) \sqcap$$

$$\mathit{succ}(|\mathit{need} \cap \mathit{succ}(|\mathit{is} \cap (\mathit{Strawberry} \cup \mathit{Tangerine} \cup \mathit{Apple})| \geq 1)| \geq 1)$$

## Example

Apply $\sqcap$-rule first:

# Example

Apply $\sqcap$-rule first:

$$fruitcake : succ(|need \cap Cake| \geq 1) \sqcap$$

$$succ(|need \cap succ(|is \cap (Strawberry \cup Tangerine \cup Apple)| \geq 1)| \geq 1),$$

$$fruitcake : succ(|need \cap Cake| \geq 1), fruitcake : succ(|need \cap$$

$$succ(|is \cap (Strawberry \cup Tangerine \cup Apple)| \geq 1)| \geq 1)$$

# Example

- impossible to decompose further
- *successor*-rule has not been applied yet

# Example

- impossible to decompose further
- *successor*-rule has not been applied yet

$\rightarrow$ Apply *successor*-rule next!

# Example

Let $c = |is \cap (Strawberry \cup Tangerine \cup Apple)| \geq 1$

$$fruitcake : succ(|need \cap Cake| \geq 1),$$
$$fruitcake : succ(|need \cap succ(c)| \geq 1)$$

# Example

Let $c = |is \cap (Strawberry \cup Tangerine \cup Apple)| \geq 1$

$$fruitcake : succ(|need \cap Cake| \geq 1),$$
$$fruitcake : succ(|need \cap succ(c)| \geq 1)$$

$$\downarrow$$

$$|X_{need} \cap X_{Cake}| \geq 1 \wedge$$
$$|X_{need} \cap X_{succ(c)}| \geq 1$$

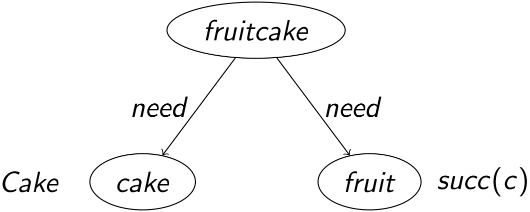
# Example

Assume solver returns:

$X_{need} \cap X_{Cake} = \{cake\}$ and $X_{need} \cap X_{succ(c)} = \{fruit\}$

# Example

Assume solver returns:
$X_{need} \cap X_{Cake} = \{cake\}$ and $X_{need} \cap X_{succ(c)} = \{fruit\}$



# Example

Same procedure for $fruit : succ(c)$