# 1 Preliminaries

Let $\mathbf{C}$ be a set of concept names and $\mathbf{R}$ a set of role names such that they are disjoint.

**Definition 1** ($QFBAPA$)
*Let $T$ be a set of symbols*

- *set terms over $T$ are:*
  - *empty set $\emptyset$ and universal set $\mathcal{U}$*
  - *every set symbol in $T$*
  - *if $s, t$ are set terms then also $s \cap t$, $s \cup t$ and $s^{\neg}$*
- *set constraints over $T$ are*
  - *$s \subseteq t$ and $s \not\subseteq t$*
  - *$s = t$ and $s \neq t$*

  *where $s, t$ are set terms*
- *cardinality terms over $T$ are:*
  - *every number $n \in \mathbb{N}$*
  - *$|s|$ if $s$ is a set term*
  - *if $k, l$ are cardinality terms then also $k + l$ and $n \cdot k$, $n \in \mathbb{N}$*
- *cardinality constraints over $T$ are:*
  - *$k = l$ and $k \neq l$*
  - *$k < l$ and $k \not< l$*
  - *$k \leq l$ and $k \not\leq l$*
  - *$n$ dvd $k$ and $n$ $\neg$dvd $k$*

  *where $k, l$ are cardinality terms and $n \in \mathbb{N}$*

**Definition 2** ($\mathcal{ALCSCC}$)
*Concepts are:*

- *all concept names*
- *if $C, D$ are concepts then:*
  - *$\neg C$*
  - *$C \sqcup D$*
  - *$C \sqcap D$*
- *$succ(c)$ if $c$ is a set or cardinality constraint over $\mathcal{ALCSCC}$ concepts and role names*

The set $S$ is a set of constraints of the form $x : C$ and $(x, y) : s$, where $C$ is a concept, $s$ a set term and $x, y$ variables. The constraint $(x, y) : s$ denotes that $y$ is a successor of $x$, while $y$ must satisfies $s$ which means that for every role $r$ occurring in $s$ it must hold that $x.r.y$ and for every concept $C$ occurring in $s$ it must hold that $y : C$).
For the next definition we denote with $s \in k$ that the set term $s$ occurs in the cardinality constraint $k$.

**Definition 3** (Number of successors)
*Let $S$ be a set of constraints, $x$ be a variable and $s$ be a set term. The number of successors of $x$ which satisfy a cardinality term $k$ is define as*

$$n_k(x_S) = |\{y \mid (x, y) : s \in S \text{ and } |s| \text{ occurs in } k\}|$$

*A constraint regarding a variable $x$ is violated if*

- $x : succ(k \leq n)$ *and* $n < n_k(x_S)$
- $x : succ(k \geq n)$ *and* $n > n_k(x_S)$
- $x : succ(k \leq l)$ *and* $n_l(x_S) < n_k(x_S)$
- $x : succ(k \geq l)$ *and* $n_l(x_S) > n_k(x_S)$
- $x : succ(n\, dvd\, k)$ *and* $mod(n_k(x_S), n) \neq 0$

*where $n \in \mathbb{N}$.*

Note that we must consider only the set terms $s$ which occur as $|s|$ in $S$ because otherwise we count in variables which are successors of $x$ but might not satisfy the whole constraint. As example we look at $S := \{x : succ(c = 1), x.s.y\}$ with $c = |r \cap B \cap C|$. The variable $y$ should be counted in $n_c(x)$ because $y : B$ and $y : C$ are not in $S$.

## 2 Tableau

**Definition 4** (Merge)
*Merging $y_1$ and $y_2$ results in one variable $y$: replace all occurrence of $y_1$ and $y_2$ with $y$.*

Note that by merging two successors other constraints might become violated:

$$\begin{aligned} S = \{&x : succ(|r \cap A| = 1) \sqcap succ(|r \cap B| = 1) \sqcap succ(|r| > 1), \\ &y_1 : A, y_2 : B, x.r.y_1, x.r.y_2\} \end{aligned} \tag{1}$$

If we merge $y_1$ and $y_2$ then the constraint $x : succ(|r| > 1)$ which was satisfied becomes violated.
But not only constraints regarding $x$ might become violated after merging two successors of $x$:

$$\begin{aligned} S = \{&x : succ(|r| \leq 1), x.r.y_1, x.r.y_2, \\ &y_1 : succ(|s| \leq 1), y_2 : succ(|s| \leq 1), y_1.s.z_1, y_2.s.z_2\} \end{aligned} \tag{2}$$

We see that the first constraint is violated and therefore merging $y_1$ and $y_2$ to $y$ would solve the problem but on the other hand the constraints regarding $y$ become violated:

$$S = \{x : succ(|r| \leq 1), x.r.y, y : succ(|s| \leq 1), y.s.z_1, y.s.z_2\}$$

To solve this problem we have to merge $z_1$ and $z_2$.

**Definition 5** (Clash)
*A constraint set $S$ contains a clash if*

- $\{x :\perp\} \subseteq S$ *or*
- $\{x : A,\ x : \neg A\} \subseteq S$ *or*
- $\{x : succ(c)\} \subseteq S$ *and $c$ is violated regarding $x$*

For the next definition we define first properties of the following notations:

- Conjunction binds stronger than disjunction: $s \cup t \cap u = s \cup (t \cap u)$
- $k \leq l$ and $k \geq l$ iff $k = l$, where $k, l$ are cardinality terms
- $s \subseteq t$ and $s \supset t$ iff $s = t$, where $s, t$ are set terms

**Definition 6** (Tableau)
*Let $S$ be a set of constraints.*

1. *$\sqcap$-rule: In $S$ is $x : C_1 \sqcap C_2$ but not both $x : C_1$ and $x : C_2$*
   *$\rightarrow S := S \cup \{x : C_1, x : C_2\}$*
2. *$\sqcup$-rule: In $S$ is $x : C_1 \sqcup C_2$ but neither $x : C_1$ or $x : C_2$*
   *$\rightarrow S := S \cup \{x : C_1\}$ or $S := S \cup \{x : C_2\}$*
3. *choose-rule: In $S$ are $x : succ(k \leq l)$, $y : C$ or $x.r.y$ and*
   *$C$ or $r$ occur in $k$ but $(x, y) : s \notin S$ for every $|s|$ occurring in $k$*
   *$\rightarrow$ choose one $|s|$, in which $C$ or $r$ occurs, and either $S := S \cup \{(x, y) : s\}$ or $S := S \cup \{(x, y) : \neg s\}$*
4. *cardinality-rule: In $S$ is $x : succ(c)$, with $c \in \{k \leq l,\ k < l,\ n\, dvd\, l\}$, which is violated regarding $x$*
   a) *if there is a set term $s$ in $l$*
      *$\rightarrow$ introduce new variable $y$ and $S := S \cup \{(x, y) : s\}$*
   b) *if $l \in \mathbb{N}$ does not contain a set term then merge two successor $y_1 \neq y_2$ of $x$ for which $(x, y_1) : k \in S$ and $(x, y_2) : k \in S$ if no other constraints regarding $x$ become violated*
5. *set-rule: In $S$ are $x : succ(c_1 \subseteq c_2)$ and $(x, y) : c_1$ but not $(x, y) : c_2$*
   *$\rightarrow S := S \cup \{(x, y) : c_2\}$*

*6. set.term-rule (Repeat until inapplicable): In $S$ is $(x, y) : s$ and*

*a)* $s = s_1 \cap s_2$ *but* $\{(x, y) : s_1,\ (x, y) : s_2\} \not\subseteq S$ *then*
$\rightarrow S := S \cup \{(x, y) : s_1,\ (x, y) : s_2\}$

*b)* $s = s_1 \cup s_2$ *and neither* $\{(x, y) : s_1\} \subseteq S$ *nor* $S\{(x, y) : s_2\} \subset S$ *then*
$\rightarrow$ *either* $S := S \cup \{(x, y) : s_1\}$ *or* $S := S \cup \{(x, y) : s_2\}$

*c)* $s = r$ *and* $x.r.y \notin S$ *then*
$\rightarrow S := S \cup \{x.r.y\}$

*d)* $s = C$ *and* $y : C \notin S$, *where* $C$ *is a* $\mathcal{ALCSCC}$ *concepts then*
$\rightarrow S := S \cup \{y : C\}$

*Any constraint set $S'$ which is obtained from applying a finite number of rules on $S$ is called derived set.*

Note that:

- $4b$ is never applicable for $n\,dvd\,l$
- $n_1\,dvd\,n_2 \cdot l$ and $n_1\,\neg dvd\,n_2$ then $n_1\,dvd\,l$ eventually

Example for $4b$:

$$S = \{x : succ(|r| = 1) \sqcap succ(|r \cap s| = 1) \sqcap succ = (|r \cap C| = 1|\} \tag{3}$$

After rule 1 (two times):

$$\begin{aligned} S = \{x &: succ(|r| = 1) \sqcap succ(|r \cap s| = 1) \sqcap succ = (|r \cap C| = 1| \\ x &: succ(|r| = 1), x : succ(|r \cap s| = 1), x : succ = (|r \cap C| = 1|\} \end{aligned}$$

If we try to satisfy at least two of the new constraints by the Tableau-algorithm above we end up with at least one constraint being violated. Let say we use the rules on the three new constraints sequentially. Then we have

$$\begin{tikzcd} & x \arrow[dl, "r"'] \arrow[d, "r \cap s"'] \arrow[dr, "r"] & \\ y_1 & y_2 & y_3 \\ & & C \end{tikzcd}$$

After using rule 4b two times we have the variable $x$ and its only $r \cap s$-successor $y$ which is of the concept $C$. We could use this rule because we do not violate any other constraints. This condition helps to prevent an infinite chain of rule application:

$$S = \{x : succ(|r| < 2) \sqcap succ(|r| \geq 2)\} \tag{4}$$

First we apply the rules 4a and 6c two times to add two $r$-successors for $x$ hence $x$ : $succ(|r| < 2)$ is not satisfied any more. If we ignore the condition in rule 4b and apply it then we merge the two successors leading to $x$ : $succ(|r| < 2)$ being satisfied but $x$ : $succ(|r| \geq 2)$ being violated. Then we apply the rules 4a and 6c again leading to $x$ : $succ(|r| < 2)$ being violated again and so on. By the condition in 4b we can not use rule 4b from the beginning which means the algorithm terminates with a clash stating that the constraint set is unsatisfiable.
First we prove that the tableau algorithm terminates.

**Proposition 1**
*Let $C$ be a concept in negation normal form. Then there is no infinite chain of applications of any tableau rules issuing from $\{x : C\}$.*

To prove this we mapped any derived sets $S$ to an element $\Psi(S)$ from a set $Q$. The elements from $Q$ can be ordered by a strict partial order $\prec$. A strict partial order is well-founded which means that there exists no infinite decreasing chains. If we can show that by obtaining $S'$ by applying a rule on $S$ results in $\Psi(S') \prec \Psi(S)$ then by the property of $\prec$ the tableau algorithm terminates.
The elements in $Q$ are quintuples, whose elements are multisets of integers ($\mathbb{Z}$). A quintuple $q = (q_1, \ldots, q_5)$ is larger than another $q' = (q'_1, \ldots, q'_5)$ if for the first $i$, $1 \leq i \leq 5$ for which $q_i$ and $q'_i$ differs $q_i$ is larger than $q'_i$ (also called lexicographical ordering). A mutliset $q_i$ is larger than $q'_i$ if $q'_i$ can be obtained from $q_i$ by replacing an integer $c$ in $q_i$ by a finite number of integers which are all smaller than $c$. Note that the finite number can also be 0.
Two more definition are needed: Two non-negative integer $n, m$ have the asymmetrical difference $n - m$ if $n \geq m$ and 0 if $n < m$. It is denoted as $n \dot{-} m$. For a concept $C$ its size $|C|$ is inductively defined as

- 1, if $C$ is a primitive concept of $\mathbf{C}$
- $|\neg C| = |C|$
- $|succ(c)| = 1 + \sum_{C \in \mathbf{C} \text{ occurs in c}} |C|$
- $|C \sqcap D| = |C \sqcup D| = |C| + |D|$

The quintuples in $Q$ are defined as follows

**Definition 7**
*Let $S$ be a constraint set. Then $\Psi(S)$ is a multiset containing for each variable $x$ occurring in $S$ a quintuple $\psi_S(x)$ where*

- *the first component is a non-negative integer $max\{|C| \mid x : C \in S\}$*
- *the second component is a multiset of integers containing for each $x : C \sqcap D$ on which the $\sqcap$-rule is applicable the non-negative integer $|C \sqcap D|$ (respectively for $C \sqcup D$)*

- *the third component is a multiset of integers containing for each* $x : succ(k \geq n) \in S$ *the non-negative integer* $n \dot{-} n_k(x_S)$ *and for each* $x : succ(k \geq l) \in S$ *the non-negative integer* $n_l(x_S) \dot{-} n_k(x_S)$
- *the fourth component denotes the number of all successors of* $x$ *in* $S$
- *the fifth component is a multiset of integers containing for each* $x : succ(k \leq n) \in S$ *the number of all successors* $y$ *of* $x$ *such that for a role* $r$ *we have* $x.r.y$, $r$ *occurs in* $k$ *but for a* $|s|$ *in* $k$ *we have neither* $(x, y) : s \in S$ *or* $(x, y) : \neg s \in S$

**Lemma 1**
*The following properties hold*

1. *For any concept* $C$ *we have* $|C| \geq |NNF(\neg C)|$
2. *Any variable* $y$ *in a derived set* $S$ *has at most one predecessor* $x$ *in* $S$
3. *If* $x.r.y \in S$ *then*

$$max\{|C| \mid x : C \in S\} > max\{|D| \mid y : D \in S\}$$

*Proof.*

1. By induction over the number of applications to compute the negation normal form we have $|C| = |NNF(\neg C)|$. Because $\neg succ(k > 0)$ can be replace by $\perp$ which is *smaller* than $\neg succ(k > 0)$, we have $|C| \geq |NNF(\neg C)|$
2. If $y$ is a newly introduced variable then it can only be introduced by one exactly one variable $x$ which it is $y$'s only predecessor. If two variable are merged together by rule 4b then both variable must have the same predecessor $x$ by the condition of the rule
3. By the second fact we know that $x$ is the only predecessor of $y$. When $y$ is introduced by 4a then we have $y : C$ for every concept $C$ occurring in $l$. But by the definition of the concept size $max\{|C| \mid x : C \in S\}$ is greater by at least 1. A new constraint $y : D$ can occur either because rule 1 or 2 are applicable on $y : C$, which neither raise $max\{|D| \mid y : D \in S\}$, or because rule 3 is applicable on $x : C$. If $y$ get merged together with another variable $z$ then $y$ and $z$ must have the same predecessor which means that all concept size regarding $z$ are also smaller then $max\{|C| \mid x : C \in S\}$.

□