Full Proofs

No Author Given

No Institute Given

$$\delta_{cf}(\mathscr{K},\mathscr{K}') = |\Delta(A^x_{\mathscr{K}}, A^x_{\mathscr{K}'})| \tag{1}$$

$$l_{\min}(P,\mathscr{K}') := \min_{y \in I_n} \delta_{lm}(x,y,\mathscr{K}') \quad \text{and} \quad l_{\text{mean}}(P,\mathscr{K}') := \frac{1}{|I_n|} \sum_{y \in I_n} \delta_{lm}(x,y,\mathscr{K}') \tag{2}$$

Theorem 1. *The Algorithm 1 is correct: Given a KB $\mathscr{K}$ and an update request $P = \langle C(x), rem \rangle$, such that $\mathscr{K} \not\vdash P$, Algorithm 1 returns a collection KBs such that each $\mathscr{K}_i' \in KBs$ is a local counterfactual for P without redundancy in $\mathscr{K}$.*

Proof. To establish the correctness we first prove that $\mathscr{K}' \vdash P$ is true for every KB $\mathscr{K}'$ returned by Algorithm 1. Since $\mathscr{K}' \in KBs$, there is some candidate i (Line 10) such that $\mathscr{K}' = \mathscr{K}_i'$, that is, $\mathscr{K}_i'$ is obtained from $\mathscr{K}$ after applying changes due to some candidate set i. Moreover, each candidate i is returned after the call $\texttt{find\_candidates}(\mathscr{K}, \{C_j\}, \emptyset, \emptyset, x)$ is made for some concept $C_j \in C$. This implies that for each $\mathscr{K}_i'$, there is some $C_j \in C$ such that $\mathscr{K}_i' \not\models C_j(x)$ since all the assertions necessary to infer $\mathscr{K} \models C_j(x)$ have been removed (Lines 10–19). Consequently, $\mathscr{K}_i' \not\models C(x)$. The fact that each $\mathscr{K}_i'$ is non-redundant follows from Line 13 in Algorithm 2, since exactly one candidate/relevant concept is chosen if some $D \in \mathscr{D}$ is a conjunction. This proves that the KBs returned by Algorithm 1 are indeed local counterfactuals, and each of them fulfils the request P. Finally, Lines 22–24 sort the output KBs according to their edit distance (Eq. 1) and likeliness (Eq. 2).

Conversely, let $\mathscr{L}$ be a non-redundant local counterfactual of $\mathscr{K}$. Then $\mathscr{L} \not\models C(x)$ and consequently, $\mathscr{L} \not\models C_j(x)$ for at least one $C_j \in C$. We prove that $\mathscr{L} \in KBs$. Let $R^j_{rem} = \{R_1, \ldots, R_m\}$ denotes the set of all candidates returned by the call $\texttt{find\_candidates}(\mathscr{K}, \{C_j\}, \emptyset, \emptyset, x)$. Clearly, R^j_{rem} includes all possible ways of removing assertions from $\mathscr{K}$ to fail the inference $\mathscr{K} \models C_j(x)$. Then, there must be some $R_i \in R^j_{rem}$ such that $\mathscr{L}$ is obtained from $\mathscr{K}$ by removing assertions corresponding to the candidate R_i since $\mathscr{L} \models C_j$ is true otherwise. We claim that $\mathscr{L}$ is returned in the iteration i when R_i is selected as a candidate in Line 10. Suppose to the contrary that $\mathscr{L}$ is obtained from $\mathscr{K}$ by removing not only the concept assertions corresponding to concepts in R_i but requires some further changes. With out loss of generality, we assume that these changes only include the removal of ABox assertions since adding assertions is redundant. But this implies that either the update $\mathscr{K} \to \mathscr{L}$ is redundant (if

[3] To get $\mathscr{P}$, assume we take sets of all possible combinations with one DI_{j_n} out of each DI_j. However, double entries and those sets that contain redundancy are not considered. Example 1: $j = 2$, DI_1 = {A, B} and DI_2 = {C, D, E}, then $\{P\}$ = {{A, C}, {A, D}, {A, E}, {B, C}, {B, D}, {B, E}}. Example 2: $j = 2$, DI_1 = {A, B} and DI_2 = {A, C}, then $\{P\}$ = {{A}, {B, C}}.

```
Input: KB $\mathscr{K}$ and a counterfactual request $P = \langle C(x), rem \rangle$ such that $\mathscr{K} \not\vdash P$
Output: Local counterfactuals $\mathscr{K}_i'$ in $\mathscr{K}$ sorted by likeliness such that with $\mathscr{K}_i' \vdash P$,
  global variable $candidates$
Function create_candidates_neg($\mathscr{K}$, $C$, $x$):
    $KBs \leftarrow \emptyset$
    $candidates \leftarrow \emptyset$
    $C \leftarrow \{C_1, \ldots, C_n\}$
    for $C_j$ in $C$ do
        find_candidates($\mathscr{K}, \{C_j\}, \emptyset, \emptyset, x$)
    end
    for $i$ in $candidates$ do
        $\mathscr{K}_i' \leftarrow$ deepcopy($\mathscr{K}$)
        // Get copy $\mathscr{K}_i'$ of $\mathscr{K}$
        for $cn$ in $i$ do
            if $cn \equiv \top$ then
                $\mathscr{K} \leftarrow None$
            else if $cn$ is an atomic class then
                Remove $cn(x)$ from $\mathscr{K}_i'$
            else if $cn = \exists r.A$ then
                Remove all assertions $\{r(x,a) \mid \mathscr{K} \models A(a)$ from $\mathscr{K}_i'\}$
        end
        $KBs \leftarrow KBs \cup \{\mathscr{K}_i'\}$
    end
    $cfs \leftarrow \arg\min_{\mathscr{K}_i' \in KBs} \delta_{cf}(\mathscr{K}, \mathscr{K}_i')$
    $cfs\_min \leftarrow$ sort $cfs$ by $l_{\min}$
    $cfs\_mean \leftarrow$ sort $cfs$ by $l_{\text{mean}}$
return $cfs\_min, cfs\_mean$
```

Algorithm 1: Finds all non-redundant updates $\mathscr{K} \to \mathscr{K}_i'$ such that $\mathscr{K}_i' \not\models C(x)$

these additional removal concerns concepts not in R_i) or $\mathscr{L}$ is not a local counterfactual candidate for P (if these concern some individual $y \neq x$). This is due to the reason that the removal of assertions for x corresponding to concepts in R_1 suffices to create a KB that fulfills P. This leads to a contradiction since $\mathscr{L}$ is a non-redundant local counterfactual of $\mathscr{K}$ for P. This completes the correctness proof.

Theorem 2. *The Algorithm 3 is correct: Given a KB $\mathscr{K}$ and an update request $P = \langle C(x), add \rangle$ such that $\mathscr{K} \not\vdash P$, Algorithm 1 returns an updated KB $\mathscr{K}'$ such that $\mathscr{K}'$ is a counterfactual for P without redundancy in $\mathscr{K}$.*

Proof. We first prove that $\mathscr{K}' \vdash P$ is true for $\mathscr{K}'$ returned by Algorithm 3. As before, assume without loss of generality that C can be written as a conjunct $C_1 \sqcap \ldots \sqcap C_n$ and no C_j contains an intersection on the outer level anymore. Clearly, $\mathscr{K}' \models C_j(x)$ is true once Lines 5–7 have been executed for each $C_j \in C$. This implies that $\mathscr{K}' \models C(x)$. Moreover, the applied changes are non-redundant, minimal, and only local. Conversely, suppose there exists $\mathscr{L} = (\mathscr{T}, \mathscr{A}'')$ such that $\mathscr{L} \models C(x)$. Suppose that $\mathscr{L} \neq \mathscr{K}'$. Clearly, Lines 6–11 state the necessary conditions to infer that $\mathscr{L} \models C_j(x)$ for a KB $\mathscr{L}$. This

```
Input: knowledge base $\mathscr{K} = (\mathscr{T}, \mathscr{A})$, set of concepts to look at in this iteration ($c\_set$),
  set of concepts already visited, set of visited concepts relevant for this candidate, target
  individual $x$, global variable $candidates$
Function find_candidates($\mathscr{K}$, $c\_set$, $visited$, $relevant$, $x$):
    $relevant \leftarrow relevant \cup c\_set$
    $\mathscr{E} \leftarrow \{E \mid \exists F : (E \sqsubseteq F) \in \mathscr{T}, \mathscr{K} \models E \sqsubseteq C_i$ for at least one $C_i \in c\_set\}$
    $\mathscr{D} \leftarrow \{D \mid D \in \mathscr{E}, D \notin visited$ and $\mathscr{K} \models D(x)\ \}$
    for each $D_j \in \mathscr{D}$ that is an existential restriction or an atomic concept do
        $relevant \leftarrow relevant \cup \{D_j\}$
    end
    for each $D_j \in \mathscr{D}$ that is an intersection do
        Bring to NNF $D_j = D_{j_1} \sqcap D_{j_2} \sqcap \ldots \sqcap D_{j_n}$
        $DI_j \leftarrow \{D_{j_1}, \ldots, D_{j_n}\}$
    end
    $\mathscr{P} \leftarrow \{P \subseteq \cup_j DI_j \mid \forall_j |P \cap DI_j| \geq 1$ and $\nexists p \in P \forall_j |P \setminus \{p\} \cap DI_j| \geq 1\}$
    // all non-redundant combinations covering each $DI_j$ (example[3])
    for each $E \in \mathscr{E}$ do
        $visited \leftarrow visited \cup \{E\}$
    end
    if $\mathscr{P} = \emptyset$ then
        $candidates \leftarrow candidates \cup \{relevant\}$ // candidates is a set of sets
    else
        for each $P \in \mathscr{P}$ do
            find_candidates($\mathscr{K}$, $P$, $visited$, $relevant$, $x$)
        end
    end
```

Algorithm 2: Recursively follow path of subsumptions, add set(s) of relevant concepts to global variable *candidates*

implies that $\mathscr{L}$ applies not only these changes but adds or removes further assertions to the ABox $\mathscr{A}''$. Then a similar line of reasoning as in the proof of Theorem 1 implies that either the update $\mathscr{K} \rightarrow \mathscr{L}$ is redundant or $\mathscr{L}$ is not a local counterfactual of $\mathscr{K}$ for P.

```
**Input:** KB $\mathscr{K}$ and a counterfactual request $P = \langle C(x), add \rangle$ such that $\mathscr{K} \not\vdash P$
(To get $\mathscr{K}'$ and still keep $\mathscr{K}$ for comparison, give a copy $\mathscr{K}'$ of $\mathscr{K}$ as input.)
**Output:** Local counterfactual $\mathscr{K}'$ in $\mathscr{K}$ such that $\mathscr{K}' \vdash P$
**Function** create_candidates_pos($\mathscr{K}$, $C$, $x$):
    $C \leftarrow \{C_1, \ldots, C_n\}$
    **for** $C_j$ *in* $C$ **do**
        **if** $C_j$ *is an existential restriction* $\exists r.D$ **then**
            add a new individual $y$ and the assertion $r(x, y)$ to $\mathscr{K}$
            $\mathscr{K} \leftarrow$ create_candidates_pos($\mathscr{K}$, $D$, $y$) // recursive call
        **else**
            add $C_j(x)$ to $\mathscr{K}$
    **end**
**return** $\mathscr{K}$
```

Algorithm 3: Updated KB $\mathscr{K}$ such that $\mathscr{K} \models C(x)$