1 Algorithms

Algorithm 1 Function to create counterfactual candidates $\mathcal{K}'$ from $\mathcal{K}$ regarding an individual x such that $\mathcal{K}' \not\models CC(x')$

Input: KB $\mathcal{K}$, Concept CC in $\mathcal{ALCH}$ with $\mathcal{K} \models CC$, individual x, (possibly empty) set "KBs", protected feature set P
Output: Candidates $\mathcal{K}'$ with $\mathcal{K}' \not\models CC(x)$, counterfactual(s) cfs sorted by likeliness
Function `nothold(`$\mathcal{K}$, CC, x`)`**:**

Apply concept rewriting (see section **??**)
Bring concept to top-level conjunctive normal form
for $clause$ in CC **do**
  $\mathcal{K}' = \{\mathcal{T}', \mathcal{A}'\} \leftarrow$ `copy`$(\mathcal{K})$
  for C in $clause$ **do**
    if $\mathcal{A}' \cap (\mathcal{T}'' = \emptyset) \models C(x)$ **then**
    `negative` $(\mathcal{K}', C, x, P)$
  end for
  if $K' \not\models CC$ **then** $KBs \leftarrow KBs \cup \{K'\}$
end for
$cfs \leftarrow \arg\min_{\mathcal{K}' \in KBs} \delta(Axioms(x), Axioms(x'))$
$cfs\_min \leftarrow$ sort cfs by $l_{\min}$
$cfs\_mean \leftarrow$ sort cfs by l_{mean}
return $candidates$

Algorithm 2 Function to create counterfactual candidates $\mathcal{K}'$ from $\mathcal{K}$ regarding an individual x such that $\mathcal{K}' \models CC(x')$

Input: KB $\mathcal{K}$, Concept CC in $\mathcal{ALCH}$ with $\mathcal{K} \not\models CC$, individual x, (possibly empty) set "KBs", protected feature set P
Output: Candidates $\mathcal{K}'$ with $\mathcal{K}' \models CC(x)$, counterfactual(s) cfs sorted by likeliness
Function `hold(`$\mathcal{K}$, CC, x`)`**:**

Apply concept rewriting (see section **??**)
Bring concept to top-level disjunctive normal form
for $term$ in CC **do**
  $\mathcal{K}' = \{\mathcal{T}', \mathcal{A}'\} \leftarrow$ `copy`$(\mathcal{K})$
  for C in $term$ **do**
    if $\mathcal{A}' \cap (\mathcal{T}'' = \emptyset) \not\models C(x)$ **then**
      `positive` $(\mathcal{K}', C, x, P$
    end if
  end for
  if $K' \models CC$ **then** $KBs \leftarrow KBs \cup \{K'\}$
end for
$cfs \leftarrow \arg\min_{\mathcal{K}' \in KBs} \delta(Axioms(x), Axioms(x'))$
$cfs\_min \leftarrow$ sort cfs by $l_{\min}$
$cfs\_mean \leftarrow$ sort cfs by l_{mean}
return $candidates$

Algorithm 3 Subroutine of Alg 1/Alg 4 to change KB K

Input: KB $\mathcal{K}$; concept C in $\mathcal{ALCH}$; individual x, protected feature set P
Output: Updated KB $\mathcal{K}$
`negative(`$\mathcal{K}$, C, x`)`

if $C \equiv \top$ or ($C \in R$ and x is the factual) **then**
  return Error: Counterfactual not possible
else if C is a negated concept **then**
  `positive`$(K, \neg C, x, P)$
else if C is an unnegated concept **then**
  if $C(x) \in \mathcal{A}$ **then** remove $C(x)$.
  if $C = \exists r.D$ **then** remove all $r(x, y_i)$ with $K \models D(y_i)(i \geq 0)$.
  if $C = \forall r.D$ **then** add y, add $r(x, y)$, `hold` $(K, y, \neg D, KBs, P)$.
end if

Algorithm 4 Subroutine of Alg 2/Alg 3 to change KB K

Input: KB $\mathcal{K}$; class, existential restriction or universal restriction C in $\mathcal{ALCH}$; individual x, protected feature set P
Output: Updated KB $\mathcal{K}$
`positive(`$\mathcal{K}$, C, x`)`

if $C \equiv \bot$ or ($C \in R$ and x is the factual) **then**
  return Error: Counterfactual not possible
else if C is a negated concept **then**
  `negative`$(K, \neg C, x, P)$
end if

if $C = \exists r.D$ **then**
  add y, add $r(x, y)$, `hold` (K, y, D, KBs, P).
else if $C = \forall r.D$ **then**
  remove all $r(x, y_i)$ with $K \models \neg D(y_i)(i \geq 0)$.
else
  add $C(x)$
end if

if $\neg C(x) \in \mathcal{A}$ **then**
  remove $\neg C(x)$
end if