1 Analysis Operations Encoded as QBF

Lemma 1. *(Implements) Given an SPL, a set of components C, and a feature f, implements(C, f) iff $form\_implements_f(v_1, \ldots, v_n)$, where $\bar{C} = \langle v_1, \ldots, v_n \rangle$, evaluates to TRUE.*

Proof. $(\Rightarrow)$: Let $\mathcal{T}(f) = \{C_1, \ldots, C_k\}$ and assume $implements(C, f)$. By definition, there is a $C_j \in \mathcal{T}(f)$ such that $C_j \subseteq C$. Let $C_j = \{p_{c_{\ell_1}}, \ldots, p_{c_{\ell_m}}\}$. Then, $v_{\ell_i} = 1$ for all $i = 1, \ldots, m$. We have to show that

$$\forall p_{c_1} \ldots p_{c_n} \{[\bigwedge_{i=1}^{n} (v_i \Rightarrow p_{c_i})] \Rightarrow formula\_\mathcal{T}(f)\}.$$

Let $\langle w_1, \ldots, w_n \rangle$ be an assignment of boolean values (TRUE or FALSE) to the propositional variables $p_{c_1} \ldots p_{c_n}$ such that $\bigwedge_{i=1}^{n}(v_i \Rightarrow p_{c_i}) = \bigwedge_{i=1}^{n}(v_i \Rightarrow w_i)$ evaluates to TRUE. Since $v_{\ell_i} = 1$ for all $i = 1, \ldots, m$, this implies $w_{\ell_j} = 1$ for all $j = 1, \ldots, m$ as well. Therefore, for $C_j \in \mathcal{T}(f)$, $(\bigwedge_{c_i \in C_j} w_i) = w_{\ell_1} \wedge \ldots \wedge w_{\ell_m}$ = TRUE.

Since $formula\_\mathcal{T}(f) = \bigvee_{j=1..k} \bigwedge_{c_i \in C_j} p_{c_i} = \bigvee_{j=1..k} \bigwedge_{c_i \in C_j} w_i$, and one of the disjuncts C_j is TRUE, the entire expression for $formula\_\mathcal{T}(f)$ evaluates to TRUE.

$(\Leftarrow)$: Assume that $implements(C, f)$ does not hold. Then, for every $C_j \in \mathcal{T}(f)$, $C_j \not\subseteq C$. This implies that for all $j \in \{1 \ldots k\}$, there is a $c_j \in C_j \setminus C$. Define an assignment to the propositional variables as follows: $v_i = 1$ for p_{c_i} such that $c_i \in C$ and 0 for the rest. Hence, $v_j = 0$ for the proposition p_{c_j} corresponding to component $c_j \notin C$.

This assignment evaluates the antecedent $\bigwedge_{i=1}^{n}(v_i \Rightarrow p_{c_i})$ to TRUE. But the consequent $formula\_\mathcal{T}(f) = \bigvee_{j=1..k} \bigwedge_{c_i \in C_j} p_{c_i}$ = FALSE for the above assignment because each disjunct is falsified by the presence of an assignment $v_j = 0$ for the proposition $p_{c_j} \notin C$. Therefore, $form\_implements_f(v_1, \ldots, v_n)$ evaluates to FALSE.

Lemma 2. *(Realizes, Covers) Given a set of components C and a set of features F, let $\bar{C} = (c'_1, \ldots, c'_n)$ and $\bar{F} = (f'_1, \ldots, f'_m)$. Then the following statements hold:*

1. *C covers F iff $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$*
2. *C realizes F iff $f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$*

Proof. Suppose $\bar{C} = (c'_1, \ldots, c'_n)$ and $\bar{F} = (f'_1, \ldots, f'_m)$. If C covers F, then $Provided\_by(C) = \{f \mid Implements(C, f)\}$ will contain F. Pick $f \in F$. Since $F \subseteq Provided\_by(C)$, $\exists C_1 \in \mathcal{T}(f), C_1 \subseteq C$. We then have $\widehat{C} \Rightarrow \widehat{C}_1$ and $\widehat{C}_1 \Rightarrow formula\_\mathcal{T}(f)$. Therefore, $f\_implements_f(c'_1, \ldots, c'_n)$ holds. Hence, it will be the case that $p_{f'_i} \Rightarrow f\_implements_{f_i}(c'_1, \ldots, c_n)$ for every $f'_i \in \bar{F}$. Hence, $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$.

Conversely, suppose $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$. Then $\bigwedge_{i=1}^{m}(p_{f'_i} \Rightarrow f\_implements_{f_i}(c'_1, \ldots, c'_n))$ holds. If $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$

holds, then (1) either $p_{f'_j} = 0$, or (2) $p_{f'_j} = 1$ and $f\_implements_{f_j}(c'_1, \ldots, c'_n)$ holds. As seen in Lemma 1, $f\_implements_{f_i}(c'_1, \ldots, c'_n)$ holds good when there exists a set $C_{j_i} \in \mathcal{T}(f_i)$ such that $C_{j_i} \subseteq C$. Since this is true for all f_i, we have $\bigcup_{j_i} C_{j_i} \subseteq C$. Therefore, $Provided\_by(C) \supseteq Provided\_by(\bigcup_{j_i} C_{j_i})$. By the formula $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$, whenever $p_{f'_i} = 1$, there exists $C_{j_i} \subseteq C$ which implements f_i. Therefore, C implements possibly a superset of F, hence covers F.

The proof for realizes is similar. The only difference is that the implication is both ways, which ensures that C implements F exactly.

Lemma 3. *(Completeness, Soundness) Let $\Psi = (\overline{\mathcal{F}}, \overline{\mathcal{C}}, \mathcal{T})$ be an SPL, with $\mathcal{F} = \{f_1, \ldots, f_m\}$ and $\mathcal{C} = \{c_1, \ldots, c_n\}$.*

1. *Ψ is complete iff*
   $\forall f'_1 \ldots f'_m[CON_F(f'_1, \ldots, f'_m) \Rightarrow \exists c'_1 \ldots c'_n[CON_I(c'_1, \ldots, c'_n) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)]]$
2. *Ψ is sound iff*
   $\forall c_1 \ldots c_n[CON_I(c_1, \ldots, c_k) \Rightarrow \exists f_1 \ldots f_j[CON_F(f_1, \ldots, f_j) \wedge f\_covers(c_1, \ldots, c_k, f_1, \ldots, f_j)]]$

Proof. Let $\Psi = (\overline{\mathcal{F}}, \overline{\mathcal{C}}, \mathcal{T})$ be an SPL with n components and m features. Let $\overline{\mathcal{C}} = \{S_1, \ldots, S_k\}$. Given a tuple of component parameters $c'_1, \ldots, c'_n$ where each c'_i is 0 or 1, the predicate $CON_I(c'_1, \ldots, c'_n)$ is defined as

$$\bigvee_j \bigwedge_{c_i \in S_j} c'_i$$

Then $CON_I(c'_1, \ldots, c'_n)$ is satisfied iff $\{c'_k \mid c'_k = 1\} = S_l$ for some $S_l \in \overline{\mathcal{C}}$. $CON_F(f'_1, \ldots, f'_m)$ is defined similarly.

1. Assume that the SPL is complete. Then for every $F \in \overline{\mathcal{F}}$, there exists some $C \in \overline{\mathcal{C}}$ such that $Covers(C, F)$. Pick any $F \in \overline{\mathcal{F}}$ and its corresponding $C \in \overline{\mathcal{C}}$. Let $\bar{F} = (f'_1, \ldots, f'_m)$ and $\bar{C} = (c'_1, \ldots, c'_n)$. Then $CON_F(f'_1, \ldots, f'_m)$ will be 1 iff there exists a set $F_j \in \overline{\mathcal{F}}$ such that $\bar{F}_j(i) = 1$ iff $f'_i = 1$. Since $Covers(C, F)$ holds for all $F \in \overline{\mathcal{F}}$, we have by Lemma 2, $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$ is true for every tuple $(f'_1, \ldots, f'_m)$ such that $f'_i = 1$ iff $\exists F_j \in \overline{\mathcal{F}}$ such that $\bar{F}_j(i) = 1$. That is, for every tuple $(f'_1, \ldots, f'_m)$ that satisfies the feature constraints, there exists some tuple $(c'_1, \ldots, c'_n)$ such that $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$ holds. Thus,
   $\forall f'_1 \ldots f'_m.\{[CON_F(f'_1, \ldots, f'_m) \Rightarrow \exists c'_1 \ldots c'_n[CON_I(c'_1, \ldots, c'_n) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)]\}$
   holds.
   Conversely, assume $\forall f'_1 \ldots f'_m.\{[CON_F(f'_1, \ldots, f'_m) \Rightarrow \exists c'_1 \ldots c'_n[CON_I(c'_1, \ldots, c'_n) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)]\}$ holds. Then for all possible ways of satisfying $CON_F(f'_1, \ldots, f'_n)$ (i.e, over all $F \in \overline{\mathcal{F}}$), there exists some tuple satisfying $CON_I(c'_1, \ldots, c'_n)$ such that $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$. Each tuple satisfying $CON_F(f'_1, \ldots, f'_n)$ corresponds to a set in $\overline{\mathcal{F}}$. Corresponding to each such set, there is a tuple $(c'_1, \ldots, c'_n)$ satisfying $[CON_I(c'_1, \ldots, c'_n) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)]$: that is, there is some $C \in \overline{\mathcal{C}}$ with $\bar{C} = (c'_1, \ldots, c'_n)$ such that $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$. This says that for every $F \in \overline{\mathcal{F}}$, there exists some $C \in \overline{\mathcal{C}}$ such that C covers F. Hence, Ψ is complete.

2. The proof of soundness is similar.

Lemma 4. *(Existentially Explicit Features) Given a set of features F, let $\bar{F} = (f'_1, \ldots, f'_m)$. Then F is existentially explicit iff* $\exists c'_1 \ldots c'_n[CON_I(c'_1, \ldots, c'_n) \wedge f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)]$.

Proof. Suppose F is existentially explicit. Then there exists a $C \in \overline{\mathcal{C}}$ such that C realizes F. By Lemma 2, C realizes F iff $f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$. $CON_I(c'_1, \ldots, c'_n)$ is true iff there exists a $C \in \overline{\mathcal{C}}$ such that $\bar{C} = (c'_1, \ldots, c'_n)$. Hence, if F is existentially explicit, $\exists c'_1, \ldots, c'_n[CON_I(c'_1, \ldots, c'_n) \wedge f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)]$ holds.

Conversely, assume $\exists c'_1, \ldots, c'_n[CON_I(c'_1, \ldots, c'_n) \wedge f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)]$ holds. Then, there exists a $C \in \overline{\mathcal{C}}$ with $\bar{C} = (c'_1, \ldots, c'_n)$ and $f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$. Again, by Lemma 2, $f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$ iff C realizes F, with $\bar{C} = (c'_1, \ldots, c'_n), \bar{F} = (f'_1, \ldots, f'_m)$. That is, for $F \in \overline{\mathcal{F}}$, there exists a $C \in \overline{\mathcal{C}}$ such that C covers F. Hence, F is existentially explicit.

Lemma 5. *(Universally Explicit Features) Given a set of features F, let $\bar{F} = (f'_1, \ldots, f'_m)$. Then F is universally explicit iff φ_F holds, where φ_F is given by* $\exists c'_1 \ldots c'_n[CON_I(c'_1, \ldots, c'_n) \wedge f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)] \wedge$ $\forall c'_1 \ldots c'_n\{[(CON_I(c'_1, \ldots, c'_n) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)] \Rightarrow f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)\}$.

Proof. Assume F is universally explicit. Then by definition, (i) there exists a $C \in \overline{\mathcal{C}}$ such that C realizes F and (ii) for all $C \in \overline{\mathcal{C}}$, C covers $F \Rightarrow C$ realizes F.

The first point (i) can be expressed as $\exists c'_1 \ldots c'_n[CON_I(c'_1, \ldots, c'_n) \wedge f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)]$ (recall that $CON_I(c'_1, \ldots, c'_n)$ holds iff there exists a $C \in \overline{\mathcal{C}}$ with $\bar{C} = (c'_1, \ldots, c'_n)$, and $f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$ holds iff C realizes F by Lemma 2).

To formalize the second point (ii), we have to consider all possible component tuples $(c'_1, \ldots, c'_n)$ satisfying $CON_I(c'_1, \ldots, c'_n)$, such that $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$ holds. For each such tuple, we have to ensure that $f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$ holds. This is true iff $\forall c'_1 \ldots c'_n\{[(CON_I(c'_1, \ldots, c'_n) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)] \Rightarrow f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)\}$ holds. Clearly, if F is universally explicit, then φ_F holds.

Conversely, assume φ_F holds. Now, $\exists c'_1 \ldots c'_n[CON_I(c'_1, \ldots, c'_n) \wedge f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)]$ holds whenever there is a tuple $(c'_1, \ldots, c'_n)$ satisfying $CON_I(c'_1, \ldots, c'_n)$ for which $f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$ is true. This corresponds to a set in $C \in \overline{\mathcal{C}}$ with $\bar{C} = (c'_1, \ldots, c'_n)$ which realizes F. $\forall c'_1 \ldots c'_n\{[(CON_I(c'_1, \ldots, c'_n) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)] \Rightarrow f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)\}$ considers all possible tuples $(c'_1, \ldots, c'_n)$ satisfying $CON_I(c'_1, \ldots, c'_n)$ for which, whenever $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$ is true, so is $f\_realizes(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$. By definition, each tuple satisfying $CON_I(c'_1, \ldots, c'_n)$ corresponds to a set $C \in \overline{\mathcal{C}}$. The formulae holds iff for each such $C \in \overline{\mathcal{C}}$, whenever C covers F, C realizes F. Therefore, F is universally explicit whenever φ_F holds.

Lemma 6. *(Unique Implementation) Given a set of features F, let $\bar{F} = (f'_1, \ldots, f'_m)$. Then F has a unique implementation iff φ_U holds. φ_U is given by* $\exists c'_1 \ldots c'_n[CON_I(c'_1, \ldots, c'_n) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)] \wedge$

$\forall d'_1 \ldots d'_n \{[CON_I(d'_1, \ldots, d'_n) \wedge f\_covers(d'_1, \ldots, d'_n, f'_1, \ldots, f'_m)] \Rightarrow (\wedge_{i=1}^n (d'_i \Leftrightarrow c'_i)\}$

Proof. Let F have a unique implementation. Then there exists a $C \in \overline{\mathcal{C}}$ which covers F and for all $C' \in \overline{\mathcal{C}}$ which covers F, $C = C'$. Two implementations C, C' are same when $\bar{C} = \bar{C}'$. That is, $\bar{C}(i) = \bar{C}'(i)$ for all $1 \leq i \leq n$. As given by the definition of $CON_I(c'_1, \ldots, c'_n)$, $CON_I(c'_1, \ldots, c'_n)$ is satisfiable iff there exists some $C \in \overline{\mathcal{C}}$ with $\bar{C} = (c'_1, \ldots, c'_n)$. Such a C covers F iff $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$ as given by Lemma 2. We have to check that there is a unique $C \in \overline{\mathcal{C}}$ that can cover F - for this, we enumerate over all possible tuples $(d'_1, \ldots, d'_n)$ that satisfy $CON_I(d'_1, \ldots, d'_n)$, and then ensure that $(d'_1, \ldots, d'_n) = (c'_1, \ldots, c'_n)$. This check is given by $\forall d'_1 \ldots d'_n \{[CON_I(d'_1, \ldots, d'_n) \wedge f\_covers(d'_1, \ldots, d'_n, f'_1, \ldots, f'_m)] \Rightarrow (\wedge_{i=1}^n (d'_i \Leftrightarrow c'_i)\}$ Thus, if F has a unique implementation, we have φ_U holds.

The converse is similar.

Lemma 7. *(Common, live and dead elements)*

1. *A component c is common iff* $\forall c'_1, \ldots, c'_n, f'_1, \ldots, f'_m \{[CON_I(c'_1, \ldots, c'_n) \wedge CON_F(f'_1, \ldots, f'_m) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)] \Rightarrow p_c\}$ *holds.*
2. *A component c is live iff* $\exists c'_1, \ldots, c'_n, f'_1, \ldots, f'_m \{[CON_I(c'_1, \ldots, c'_n) \wedge CON_F(f'_1, \ldots, f'_m) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m) \wedge p_c\}$
3. *A component c is dead iff* $\forall c'_1, \ldots, c'_n, f'_1, \ldots, f'_m \{[CON_I(c'_1, \ldots, c'_n) \wedge CON_F(f'_1, \ldots, f'_m) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)] \Rightarrow \neg p_c\}$ *holds.*

Proof. 1. Assume that c is a common component. Then by definition, for all $\langle F, C \rangle \in Prod(\Psi)$, $c \in C$. To enumerate all possible $\langle F, C \rangle$ for $F \in \overline{\mathcal{F}}$ and $C \in \overline{\mathcal{C}}$, we consider all possible tuples $(c'_1, \ldots, c'_n)$ as well as $(f'_1, \ldots, f'_m)$ for which $CON_I(c'_1, \ldots, c'_n) \wedge CON_F(f'_1, \ldots, f'_m)$ holds. Clearly, every pair of tuples satisfying $CON_I(c'_1, \ldots, c'_n) \wedge CON_F(f'_1, \ldots, f'_m)$ corresponds to a pair $\langle F, C \rangle$. For each such pair $\langle F, C \rangle$ of tuples to be in $Prod(\Psi)$, we check if C covers F. By Lemma 2, this holds iff $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$. Clearly, for all pairs of tuples for which this is true, if an element is common, then it will evaluate to 1. This is given by saying $\forall c'_1, \ldots, c'_n, f'_1, \ldots, f'_m \{[CON_I(c'_1, \ldots, c'_n) \wedge CON_F(f'_1, \ldots, f'_m) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)] \Rightarrow p_c\}$. Note that in $f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)$, we are evaluating over all possible values of $p_{c_1}, \ldots, p_{c_n}$. In particular, for $c = c_i$, we are checking that whenever $c'_i = 1$ in a product, then $p_{c_i} = 1$.
   The converse is similar.
2. Assume c is live. Then there is a pair $\langle F, C \rangle \in Prod(\Psi)$ such that $c \in C$. The existence of a pair $\langle F, C \rangle \in Prod(\Psi)$ is expressed by saying $\exists c'_1, \ldots, c'_n, f'_1, \ldots, f'_m [CON_I(c'_1, \ldots, c'_n) \wedge CON_F(f'_1, \ldots, f'_m) \wedge f\_covers(c'_1, \ldots, c'_n, f'_1, \ldots, f'_m)]$. Clearly, if c is in one such tuple, $p_c = 1$. This is written by conjuncting p_c and obtaining $\exists c'_1, \ldots, c'_n, f'_1, \ldots, f'_m \{[CON_I(c'_1, \ldots, c'_n) \wedge$

$CON_F(f'_1,\ldots,f'_m) \wedge f\_covers(c'_1,\ldots,c'_n,f'_1,\ldots,f'_m) \wedge p_c\}$. The converse is similar.

3. This is similar to 1.

Lemma 8. *(Superflous) A component c_i is superflous iff*
$\forall c'_1,\ldots,c'_n,f'_1,\ldots,f'_m\{[c'_i \wedge CON_I(c'_1,\ldots,c'_n) \wedge CON_F(f'_1,\ldots,f'_m) \wedge f\_covers(c'_1,\ldots,c'_i,\ldots,c'_n,f'_1,\ldots,f'_m)] \Rightarrow$
$\exists d'_1,\ldots,d'_n[\neg d'_i \wedge CON_I(d'_1,\ldots,d'_n) \wedge f\_covers(d'_1,\ldots,d'_n,f'_1,\ldots,f'_m)]\}$.

Proof. Assume c_i is superflous. Then by definition, for all $C \in \overline{\mathcal{C}}$ containing c_i and which covers F, there exists $C' \in \overline{\mathcal{C}}$ which does not contain c_i and which covers F. First consider all $C \in \overline{\mathcal{C}}$ containing c_i which covers F. This is given by considering all tuples $(c'_1,\ldots,c'_n)$ and $(f'_1,\ldots,f'_m)$ which satisfy $CON_I(c'_1,\ldots,\ldots,c'_n) \wedge CON_F(f'_1,\ldots,f'_m)$, $c'_i = 1$, for which $f\_covers(c'_1,\ldots,c'_i,\ldots,c'_n,f'_1,\ldots,f'_m)$ holds. The pairs $\langle C,F\rangle$ are enumerated by considering all tuples satisfying $CON_I(c'_1,\ldots,c'_i,\ldots,c'_n) \wedge CON_F(f'_1,\ldots,f'_m)$, and for those in $Prod(\Psi)$, we need to check that C covers F. Now, if such a C contains c_i, then $c'_i = 1$ in the tuple $(c'_1,\ldots,c'_n)$. To check if there exists a pair in $Prod(\Psi)$ which does not contain the ith component c_i, among all tuples $(c'_1,\ldots,c'_n)$, we check if there exists a tuple $(d'_1,\ldots,d'_n)$ such that $CON_I(d'_1,\ldots,d'_n) \wedge f\_covers(d'_1,\ldots,d'_n,f'_1,\ldots,f'_m)$ holds and where $d'_i = 0$. This is expressed by $\neg d'_i$. Thus, if c_i is superflous, we have $\forall c'_1,\ldots,c'_n,f'_1,\ldots,f'_m\{[c'_i \wedge CON_I(c'_1,\ldots,c'_n) \wedge CON_F(f'_1,\ldots,f'_m) \wedge f\_covers(c'_1,\ldots,c'_n,f'_1,\ldots,f'_m)] \Rightarrow \exists d'_1,\ldots,d'_n[\neg d'_i \wedge CON_I(d'_1,\ldots,d'_n) \wedge f\_covers(d'_1,\ldots,d'_n,f'_1,\ldots,f'_m)]\}$ holds.

The converse is similar.

Lemma 9. *(Redundant) A component c_i is redundant iff*
$\forall c'_1,\ldots,c'_n f'_1\ldots,f'_m\{[c'_i \wedge CON_I(c'_1,\ldots,c'_n) \wedge CON_F(f'_1,\ldots,f'_m) \wedge f\_covers(c'_1,\ldots,c'_n,f'_1,\ldots,f'_m)] \Rightarrow$
$f\_covers(c'_1,\ldots,\neg c'_i,\ldots,c'_n,f'_1,\ldots,f'_m)\}$

Proof. Suppose c_i is redundant. Then for every $C \in \overline{\mathcal{C}}$ containing c_i, there exists a $C' \in \overline{\mathcal{C}}$, $C' \subseteq C$, $c_i \notin C'$, and $Provided\_by(C) = Provided\_by(C')$. First, we have to enumerate all tuples $(c'_1,\ldots,c'_n)$ for which we have $CON_I(c'_1,\ldots,c'_n)$ and c'_i (i.e, enumerate all members of $\overline{\mathcal{C}}$ containing c_i). Now, we have to look at the sets of features F that these implementations cover - by definition of covers, this means that the set $Provided\_by(C)$ is in $\overline{\mathcal{F}}$, and $F \subseteq Provided\_by(C)$. This basically means to look at all tuples $(f'_1,\ldots,f'_m)$ such that $CON_F(f'_1,\ldots,f'_m)$ (which correspond to some element of $\overline{\mathcal{F}}$) and $f\_covers(c'_1,\ldots,c'_n,f'_1,\ldots,f'_m)$ (which are covered by C). This is expressed by specifying $\forall c'_1,\ldots,c'_n,f'_1,\ldots,f'_m\{[c'_i \wedge CON_I(c'_1,\ldots,c'_n) \wedge CON_F(f'_1,\ldots,f'_m) \wedge f\_covers(c'_1,\ldots,c'_n,f'_1,\ldots,f'_m)]\}$. For each such C covering F, we want to say that there exists a $C' \subseteq C$ which covers F and which does not contain c'_i. This is expressed by saying that there exists a tuple $(d'_1,\ldots,d'_n)$ for which (i) $CON_I(d'_1,\ldots,d'_n)$ holds, (ii) $f\_covers(d'_1,\ldots,d'_n,f'_1,\ldots,f'_m)$ holds, (iii) $\neg c'_i$ holds, and $(iv)(\bigwedge_{i=1}^n c'_i \Rightarrow \bigwedge_{i=1}^n d'_i)$. This last condition checks that $C' \subseteq C$. Thus, the required formula to hold is $\forall c'_1,\ldots,c'_n f'_1\ldots,f'_m\{[c'_i \wedge CON_I(c'_1,\ldots,c'_n) \wedge CON_F(f'_1,\ldots,f'_m) \wedge f\_covers(c'_1,\ldots,c'_n,f'_1,\ldots,f'_m)] \Rightarrow \exists d'_1\ldots d'_n[\neg d'_i \wedge (\bigwedge_{i=1}^n c'_i \Rightarrow \bigwedge d'_i) \wedge CON_I(d'_1,\ldots,d'_n) \wedge f\_covers(d'_1,\ldots,d'_n,f'_1,\ldots,f'_m)]\}$.

The converse is similar.

Lemma 10. *(Critical) A component c is critical for f_j iff* $\forall p_{c_1}, \ldots, p_{c_n}\{formula\_\mathcal{T}(f_j) \Rightarrow p_c\}$.

Proof. Assume c is critical for f_j. Then, every implementation which does not contain c cannot implement f_j. In other words, every implementation in $\overline{\mathcal{C}}$ which implements f_j must contain c. Lets look at $\mathcal{T}(f_j) = \{C_1, \ldots, C_k\}$. Then, c must belong to all the C_i's. Clearly, if this is the case, then whenever $\bigvee_{i=1}^{k} \bigwedge_{d \in C_i} p_d$ is true, so must be p_c : Assume there exists $C_l \in \mathcal{T}(f_j)$ such that $c \notin C_l$. Then clearly, we have an assignment of $p_{c_1}, \ldots, p_{c_n}$ where $\bigwedge_{d \in C_l} p_d$ is true, but $p_c = 0$ (as $c \notin C_l$). Thus, c is critical for f_j iff $\forall p_{c_1}, \ldots, p_{c_n}\{formula\_\mathcal{T}(f_j) \Rightarrow p_c\}$.

Lemma 11. *(Extends) Let F and F' be subsets of features. Let $\bar{F} = (f_1, \ldots, f_m)$ and $\bar{F}' = (f'_1, \ldots, f'_m)$. Then F' extends F iff $\bigwedge_{i=1}^{m}(f_i \Rightarrow f'_i)$ is true. F' is extendable iff $\exists f'_1, \ldots, f'_m[\bigwedge_{i=1}^{m} f_i \Rightarrow f'_i)]$.*

Proof. If F' extends F, then $\bar{F}(i) = 1 \Rightarrow \bar{F}'(i) = 1$. Then clearly, $\bigwedge_{i=1}^{m}(f_i \Rightarrow f'_i)$ is true. Conversely, if $\bigwedge_{i=1}^{m}(f_i \Rightarrow f'_i)$, then whenever $f_i = 1$, $f'_i = 1$. That is, $\bar{F}(i) = 1 \Rightarrow \bar{F}'(i) = 1$. Clearly, then F' extends F. If F is extendable, then there exists some F' such that F' extends F. This is same as existentially quantifying the variables of F' such that the implication holds.

2 QPRO Syntax

The QPRO input format is divided into two section, *preamble* and the $formula$.

1. $Preamble$: The Preamble contains different types of information about the file, namely,
   (a) $Comments$: Each comment line should start with lower case character 'c'. There can be multiple comment lines in the File.
   Format:
   `c COMMENT_STRING`
   Example:
   `c Testing QBF formulae.`
   `c QPRO file for completeness.`
   (b) QBF : After the comments, the string 'QBF' is followed by positive integer. The integer indicates the number of variables occurring in the formula. First variable name is associated with integer 2 and so on.
   Format:
   `QBF` $< number\ \ of\ \ variables >$
   Example:
   `QBF 10`
2. $formula$: The formula may contain either a conjunction, a disjunction or a quantifier.
   (a) $quantifier\ block$: The quantifier block always start with lower case character 'q' and end with '/q'. The line after 'q' start with letter 'a' or 'e' indicating universal quantifier or existential quantifier respectively.

Format:

```
q
a var1 var2 ...
e var11 var22 ...
...
/q
```

Example:

```
q
a 2 3
e 4 5
...
/q
```

(b) *conjunction block* : The conjunction block always start with lower case character 'c' and end with '/c'. The first line after 'c' contain all the positive literals and the second line contain negative literals.
Format:
`c`
positive literals
negative literals
`...`
`/c`

Example:
The propositional formula $c2 \wedge c3 \wedge \neg c4$ can be written as:

```
c
2 3
4
/c
```

(c) *disjunction block* : The disjunction block always start with lower case character 'd' and end with '/d'. The first line after 'd' contain all the positive literals and the second line contain negative literals.
Format:
`d`
positive literals
negative literals
`...`
`/d`

Example:
The propositional formula $c2 \vee c3 \vee \neg c4$ can be written as:

```
d
2 3
4
```

```
    /d
```

As an example, the QPRO format for the formula $\forall X \exists Y ((X \vee \neg Y) \wedge (\neg X \vee Y))$ is as follows.

```
c Illustration
QBF 3
q
a 2
e 3
c


d
2
3
/d
d
3
2
/d
/c
/q
```