This document describes the calculus **RES** as used in `proof.dtd`:

1 The Calculus

A *clause* is a pair of sequences of atoms, inferences on the left (right) side of the sequent occur only at the outermost left (right) formula(s).

The Rules:

1. structural

$$\frac{A_1, \ldots, A_n \vdash \Delta}{A_{\tau(1)}, \ldots, A_{\tau(n)} \vdash \Delta}\ \pi(\tau) : l \qquad \frac{\Gamma \vdash A_1, \ldots, A_n}{\Gamma \vdash A_{\tau(1)}, \ldots, A_{\tau(n)}}\ \pi(\tau) : r$$

   where τ is a permutation given as a list of cycles.

2. factoring

$$\frac{\Gamma \vdash \Delta, A_1, \ldots, A_n}{(\Gamma \vdash \Delta, A)\sigma}\ f(\sigma) : r \qquad \frac{A_1, \ldots, A_n, \Gamma \vdash \Delta}{(A_1, \Gamma \vdash \Delta)\sigma}\ f(\sigma) : l$$

   where σ denotes an m.g.u. of the set $\{A_i\}_{1 \leq i \leq n}$.

3. resolvent

$$\frac{\Gamma \vdash \Delta, A \quad A', \Pi \vdash \Lambda}{(\Gamma, \Pi \vdash \Delta, \Lambda)\sigma}\ r(\sigma)$$

   where σ is an m.g.u. of $\{A, A'\}$ and the premises are variable disjoint clauses.

4. paramodulation

$$\frac{\Gamma \vdash \Delta, s = t \quad \Pi \vdash \Lambda, A[s']_\Xi}{(\Gamma, \Pi \vdash \Delta, \Lambda, A[t]_\Xi)\sigma}\ p(\sigma, \Xi) : r1$$

$$\frac{\Gamma \vdash \Delta, s = t \quad A[s']_\Xi, \Pi \vdash \Lambda}{(A[t]_\Xi, \Gamma, \Pi \vdash \Delta, \Lambda)\sigma}\ p(\sigma, \Xi) : l1$$

   where Ξ is a set of positions in A, σ is an m.g.u. of $\{s, s'\}$ and the premises are variable disjoint clauses.

$$\frac{\Gamma \vdash \Delta, t = s \quad \Pi \vdash \Lambda, A[s']_\Xi}{(\Gamma, \Pi \vdash \Delta, \Lambda, A[t]_\Xi)\sigma}\ p(\sigma, \Xi) : r2$$

$$\frac{\Gamma \vdash \Delta, t = s \quad A[s']_\Xi, \Pi \vdash \Lambda}{(A[t]_\Xi, \Gamma, \Pi \vdash \Delta, \Lambda)\sigma}\ p(\sigma, \Xi) : l2$$

   where Ξ is a set of positions in A, σ is an m.g.u. of $\{s, s'\}$ and the premises are variable disjoint clauses.

2 The XML Code

For the specification of the calculus-independent part, see `proof.dtd`.

The attribute `type` of the tag `rule` specifies the type of a rule and is defined as follows:

Rule	attribute `type`
$\pi : l$	`perml`
$\pi : r$	`permr`
$f : l$	`factl`
$f : r$	`factr`
r	`res`
$p : l1$	`paral1`
$p : r1$	`parar1`
$p : l2$	`paral2`
$p : r2$	`parar2`

If the type of a rule is `perml` or `permr` then the attribute `param` must contain a string specifying a permutation. The permutation is given in the notation that is common in mathematics: a list of cycles with (and) as delimiters, e.g.:

```
(1 2)(4 5 6)
```

denotes the permutation

$$\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 5 & 6 & 4 \end{matrix}$$

assuming that the length of the permuted part of the sequent is 6. If the length of the permuted part of the sequent is longer than the largest position occurring in the permutation then the rest is padded with the identity permutation.