

# The **qpro** Input Format: A Textual Syntax for QBFs in Negation Normal Form Version 1.0

Martina Seidl

August 26, 2009

## 1 Background

This document specifies an input format for solvers capable to evaluate quantified Boolean formulas (QBF) in negation normal form (NNF). As this input format was originally designed for the QBF solver **qpro**, we call it the **qpro input format** which may be considered as an extension of the current standard input format QDIMACS.

The QDIMACS<sup>1</sup> format allows the description of quantified Boolean formulas in prenex conjunctive normal form only, hence it provides no syntactical means for nesting conjunctions, disjunctions, and quantifications arbitrarily. Nevertheless, this format is extremely easy to parse for the solvers. Consequently, it is the basis for the **qpro** input format. In QDIMACS, a QBF is divided into three sections: the *preamble*, the *quantifier prefix*, and the *matrix*. The preamble may contain arbitrary many, human readable comments which are ignored by the solvers followed by the so-called problem line which contains information about the number of variables and number of clauses occurring in the QBF. Information on the quantifier blocks is encoded in the quantifier prefix. Each line starts either with an **e** or an **a** representing an existentially or, respectively, a universally quantified set of variables. Variable names are positive integers smaller than the number of variables defined in the problem line. Finally, the matrix consists of constraints (one constraint per line). Every line of the matrix describes a disjunction of literals. Each line contains a sequence of integers, now positive as well as negative ones. Positive integers stand for positive literals, whereas negative integers represent negative literals.

## 2 The Input Format

The syntax of the **qpro** input format is shown in Figure 1. It allows the representation of QBFs in negation normal form. In contrast to QDIMACS, a formula

---

<sup>1</sup><http://www.qbflib.org/qdimacs.html>

is only divided into two sections instead of three, namely the *preamble* and the *formula*. There is no special area for the quantifier blocks as they are included directly in the formula.

A QBF file may start with human readable comments (lines marked with *c*) which are ignored by the solvers. After the comments, the string **QBF** followed by a positive integer in the same or in the next line states the number of variables occurring in the formula. If the formula respects the **qpro** input format then it may be assumed that the names of the variables are numbers between 1 and the given integer.

The main connective of the actual formula may be either a conjunction, a disjunction, or a quantified formula. We consider conjunction and disjunction as *n*-ary connectives and we do not allow that a direct subformula of a conjunction is a conjunction and that the direct subformula of a disjunction is a disjunction. The same holds for a QBF with a quantifier as the main connective: it may contain an arbitrary number of alternating quantifier blocks, but it is not allowed to have a direct subformula which starts with a quantifier. We denote a formula with a quantifier as main connective a *qformula*, accordingly disjunction and conjunction are called *dformula* and *cformula*. A *qformula* starts with a line containing the single letter **q** and ending with a line **/q**. For the disjunction (resp. for the conjunction) the **q** is exchanged by a **d** (resp. a **c**).

The lines after the **q** contain the quantifier blocks. They start either with an **a** or an **e** indicating whether quantifier is universal (**a**) or existential (**e**). A line starting with **e** may not be followed by a line starting with an **a** and vice versa. The letters at the beginning of the line are followed by an ordered list of positive integers—the variable names which are separated by blanks. So the **q** and the closing **/q** define the scope of a variable. After the alternating quantifier blocks a *cformula* or a *dformula* follows.

The conjunctions and disjunctions, i.e., *cformulas* and *dformulas*, are constructed similarly: the first two lines contain ordered lists of positive integers. In the first line all positive literals are collected whereas we find all the negative literals in the second one. If a formula contains no positive or negative integers, the respective line remains blank. Then arbitrary many subformulas (which are not literals) follow before the formula is closed either by **/c** or by **/d**.

### 3 Example

The QBF  $\forall a_2 \exists e_3 e_4 (e_4 \wedge \forall a_5 a_6 ((a_5 \wedge \neg a_6) \vee \exists e_7 e_8 e_9 e_{10} (e_7 \wedge e_8 \wedge e_9 \wedge e_{10})))$  in **qpro**'s input format is shown in Figure 2.

$\langle qbf \rangle$	$\rightarrow$	$\langle preamble \rangle \langle formula \rangle \langle footer \rangle$
$\langle preamble \rangle$	$\rightarrow$	$\langle comment \rangle^* \text{QBF} (\backslash n)? \langle posInteger \rangle \backslash n$
$\langle comment \rangle$	$\rightarrow$	$c \langle string \rangle \backslash n$
$\langle formula \rangle$	$\rightarrow$	$\langle qformula \rangle \backslash n \mid$ $\langle cformula \rangle \backslash n \mid$ $\langle dformula \rangle \backslash n$
$\langle qformula \rangle$	$\rightarrow$	$q \backslash n$ $(\langle aQBlock \rangle \mid \langle eQBlock \rangle)$ $(\langle dformula \rangle \mid \langle cformula \rangle)$ $/q \backslash n$
$\langle aQBlock \rangle$	$\rightarrow$	$\langle aQuant \rangle (\langle eQuant \rangle \langle aQuant \rangle)^* (\langle eQuant \rangle)?$
$\langle eQBlock \rangle$	$\rightarrow$	$\langle eQuant \rangle (\langle aQuant \rangle \langle eQuant \rangle)^* (\langle aQuant \rangle)?$
$\langle aQuant \rangle$	$\rightarrow$	$a \langle posInteger \rangle \langle posIntSeq \rangle$
$\langle eQuant \rangle$	$\rightarrow$	$e \langle posInteger \rangle \langle posIntSeq \rangle$
$\langle dformula \rangle$	$\rightarrow$	$d \backslash n$ $\langle posIntSeq \rangle$ $\langle posIntSeq \rangle$ $(\langle qformula \rangle \mid \langle cformula \rangle)^*$ $/d \backslash n$
$\langle cformula \rangle$	$\rightarrow$	$c \backslash n$ $\langle posIntSeq \rangle$ $\langle posIntSeq \rangle$ $(\langle qformula \rangle \mid \langle dformula \rangle)^*$ $/c \backslash n$
$\langle posIntSeq \rangle$	$\rightarrow$	$\langle posInteger \rangle \langle posIntSeq \rangle \mid \backslash n$
$\langle footer \rangle$	$\rightarrow$	$\text{QBF}$

Figure 1: The syntax of the qpro input format.

```

QBF
10
q
a 2
e 3 4
c
4

q
a 5 6
d

c
5
6
/c
q
e 7 8 9 10
c
7 8 9 10

/c
/q
/d
/q
/c
/q
QBF

```

Figure 2: The QBF  $\forall a_2 \exists e_3 e_4 (e_4 \wedge \forall a_5 a_6 ((a_5 \wedge \neg a_6) \vee \exists e_7 e_8 e_9 e_{10} (e_7 \wedge e_8 \wedge e_9 \wedge e_{10})))$  in qpro's input format.