Article

Social Science Computer Review
31(4) 505-521

DOI: 10.1177/0894439313478999
ssc.sagepub.com
SAGE

Boolean Minimization in Social Science Research: A Review of Current Software for Qualitative Comparative Analysis (QCA)

Alrik Thiem[1] and Adrian Dușa[2]

Abstract
Besides an increase in the number of empirical applications, the widening landscape of tailored computer programs attests to the success of qualitative comparative analysis (QCA) as a social research method. Users now have the choice between three graphical user interface (GUI) and three command line interface (CLI) solutions. In addition to different functional foci, each program possesses several technical particularities, some of which the vast majority of end users remain unaware of. Since these particularities may influence results and in turn substantive conclusions, this review is a timely undertaking. More specifically, we compare the two most common GUIs *fs/QCA* and *Tosmana* as well as the CLI *QCA*. By reanalyzing data from a sociological study on rural grassroots associations in Norway, major differences and similarities with respect to truth table construction, minimization algorithms, and prime implicant chart management are illustrated.

Keywords
qualitative comparative analysis, Boolean minimization, fs/QCA, QCA, Tosmana

About two decades ago, three articles in this journal took stock of the extent to which computing technology had made its way into the discipline of sociology (Blank, 1991; Brent, 1993; Heise, 1992). Among other things, they also underlined the contribution qualitative comparative analysis (QCA; Ragin, 1987) and its eponymous computer program (Drass, 1988) had made to formal data processing at the juncture between qualitative and quantitative research. Although not even a handful of substantive applications of QCA had been published in scientific periodicals (Amenta, Carruthers, & Zylan, 1992; Griffin, Botsko, Wahl, & Isaac, 1991; Ragin, Mayer, & Drass, 1984; Wickham-Crowley, 1991), and only one tailored computer program existed at that time, the potential of QCA beyond the discipline of sociology was already being recognized.

[1] Department of Humanities, Social and Political Sciences, Swiss Federal Institute of Technology, Zurich, Switzerland
[2] Department of Sociology, University of Bucharest, Bucharest, Romania

Corresponding Author:
Alrik Thiem, Department of Humanities, Social and Political Sciences, Swiss Federal Institute of Technology Zurich (ETHZ), IFW C29.2, Haldeneggsteig 4, 8092 Zurich, Switzerland.
Email: thiem@sipo.gess.ethz.ch

In accord with these expectations, major methodological, applicative, distributional, and computational developments have marked the past 20 years. Not only has the method branched out into the three variants of crisp-set QCA (csQCA; Rihoux & de Meur, 2009), multi-value QCA (mvQCA; Cronqvist & Berg-Schlosser, 2009), and fuzzy-set QCA (fsQCA; Ragin, 2000, 2008, 2009), but its substantive applications in peer-reviewed journals have also risen in number to about 250 (Thiem & Duşa, 2013). In addition to the diffusion of QCA across education research (e.g., Glaesser & Cooper, 2011), health policy (e.g., Harkreader & Imershein, 1999), management, and organization studies (e.g., Grandori & Furnari, 2008) as well as business and economics (e.g., Seeleib-Kaiser & Fleckenstein, 2009), political scientists have now overtaken sociologists as the main "consumer group" (e.g., Fischer, Kaiser, & Rohlfing, 2006; Redding & Viterna, 1999; Thiem, 2011). To meet this demand, no fewer than six computer programs now compete on the market (Thiem & Duşa, 2012).

In this review, we introduce and compare the current versions of three computer programs. The *fs/QCA* software (Ragin & Davey, 2009)—the trailblazer in this area (Weitzman & Miles, 1995)—remains the most popular solution with more than 80% market share. In contrast, *QCA* (Duşa & Thiem, 2012) is a recent extension package for the *R* environment for statistical computing and graphics (R Development Core Team, 2012). The third program is *Tosmana* (Cronqvist, 2011), which has been the first software for processing multivalent crisp sets. These programs were chosen for three reasons: first, all differ in their capabilities, thereby providing maximum variation in terms of functional scope; second, each offers a different approach to deriving QCA solutions; and third, *fs/QCA* and *Tosmana* form a duopoly on the QCA software market.

The objectives of this review are clearly defined. Neither is it intended as a summarizing snapshot of the contemporary QCA software landscape, nor does it seek to provide orientation for prospective end users who are still unsure about which program would fit their analytical requirements best. Also, it is not a supplementary manual for any of the three programs. Both *fs/QCA* and *Tosmana* come shipped with user-friendly manuals, and Thiem and Duşa (2013) provide extensive documentation for *QCA*. Instead, emphasis is placed on technical particularities, ranging from the construction of truth tables to the derivation of final solutions. As such, the target audience includes intermediate to advanced users of QCA.

The article is structured as follows. The first section provides a concise repetition of the most important concepts in QCA, mainly those relating to truth tables and Boolean minimization. In this connection, the nomenclature to be followed throughout the text will also be introduced. The second section provides an overview of the main differences between the three programs in terms of their basic characteristics. The third section is the main part. It covers the topics of truth table construction, minimization algorithms, prime implicant (PI) chart management, and the derivation of solutions. For the purpose of illustration, we use data from Wollebæk's (2010) sociological study on rural populations of grassroots associations in 22 Norwegian municipalities.

Set Theory and Boolean Minimization

The increasing popularity of QCA in social science research is to a not inconsiderable degree a consequence of the elegant simplicity of the method. In essence, two steps constitute its basic mechanism, the first being the construction of a truth table and the second the minimization of the Boolean function this table represents. In the following, each of these steps is explained in brief, mainly for the purpose of establishing the nomenclature for the remainder of this review.

Truth Tables

A *truth table* is a two-dimensional array of $k + 1$ columns and d rows. A row in such a table, minus its element in the rightmost column, is called a *configuration* and represents a unique combination of

Table 1. Truth Table.

$\mathbf{X}_1$	$\mathbf{X}_2$	OUT	Inclusion	*n*	Cases
0	0	0	0.000	2	a, b
0	1	?	—	—	—
1	0	1	1.000	1	c
1	1	1	1.000	3	d, e, f

k values from all *conditions* $\mathbf{X}_j$, written $\mathbf{X}_j\{v_h\}$, with $j = 1, 2, \ldots, k$ and $h = 1, 2, \ldots, p$, such that only a single value from each condition is included at a time. The total number of configurations d is determined by the product of the total number of values p_j of all conditions $\mathbf{X}_j$. A simple truth table with $k = 2$ and $p_j = 2$ is presented in Table 1.

The "OUT" column represents the *outcome value*, which results from a set-theoretic operation on the configurations and the *outcome set* called *inclusion* (Thiem & Duşa, 2013) or *consistency* (Ragin, 2006). Inclusion provides a summary measure of the degree to which the hypothesis that the configuration is a subset of the outcome set can be preliminarily considered as true. Based on their respective inclusion scores, configurations can then be coded as *negative* (0) if they do not form a subset of the outcome set or *positive* (1) if they do. Configurations for which no such statement can be established for lack of empirical evidence are called *logical remainders* (?). If the evidence is mixed, the configuration is called a *contradiction* (C). Truth tables are often supplemented with further information, such as the number of cases (*n*) and their labels (Cases), but although generally useful, these columns form no necessary parts of a truth table. In addition to the configurations, only the "OUT" column is essential.

Boolean Minimization

Given the information in Table 1, the system of condition–outcome set relations forms a Boolean function that can be expressed as a *canonical union* of *fundamental intersections* (FI), each of which corresponds to a positive configuration. If the outcome set value to be explained is denoted by $\mathbf{O}\{1\}$, then the canonical union can be written as $(\mathbf{X}_1\{1\} \cap \mathbf{X}_2\{0\}) \cup (\mathbf{X}_1\{1\} \cap \mathbf{X}_2\{1\}) \subseteq \mathbf{O}\{1\}$. The goal of Boolean minimization is to eliminate as many conditions as possible from this set relation. A condition that can be eliminated is irrelevant because the outcome is not affected by any of its values.

The elimination of sets requires the application of the laws and theorems of Boolean algebra. One of the two distributive laws says that $\boldsymbol{a}$ AND ($\boldsymbol{b}$ OR $\boldsymbol{c}$) = ($\boldsymbol{a}$ AND $\boldsymbol{b}$) OR ($\boldsymbol{a}$ AND $\boldsymbol{c}$) (Edwards, 1973, p. 19; Hohn, 1966, p. 12). Using the familiar operators for set union and intersection instead, this equation becomes $\boldsymbol{a} \cap (\boldsymbol{b} \cup \boldsymbol{c}) = (\boldsymbol{a} \cap \boldsymbol{b}) \cup (\boldsymbol{a} \cap \boldsymbol{c})$. If $\boldsymbol{a} = \mathbf{X}_1\{1\}$, $\boldsymbol{b} = \mathbf{X}_2\{0\}$, and $\boldsymbol{c} = \mathbf{X}_2\{1\}$, where $\{1\}$ stands for one set value and $\{0\}$ for its negation, then the canonical union formed above can be rewritten as $\mathbf{X}_1\{1\} \cap (\mathbf{X}_2\{0\} \cup \mathbf{X}_2\{1\}) \subseteq \mathbf{O}\{1\}$. By the law of excluded middle, according to which $\mathbf{X}_j\{1\} \cup \mathbf{X}_j\{0\} = 1$, it follows that $\mathbf{X}_1\{1\} \cap (\mathbf{X}_2\{0\} \cup \mathbf{X}_2\{1\}) \subseteq \mathbf{O}\{1\} = \mathbf{X}_1\{1\} \subseteq \mathbf{O}\{1\}$. Only $\mathbf{X}_1\{1\}$ is relevant with respect to $\mathbf{O}\{1\}$. The condition $\mathbf{X}_2$ is redundant because it makes no difference in the system of condition–outcome relations. In contrast, $\mathbf{X}_1\{1\}$ is an *implicant* of $\mathbf{X}_1\{1\} \cap \mathbf{X}_2\{0\}$ and $\mathbf{X}_1\{1\} \cap \mathbf{X}_2\{1\}$ which cannot be absorbed. Moreover, $\mathbf{X}_1\{1\}$ is also a PI because it cannot be simplified further. Irrespective of the fact that it contains only a single term in this example, the final reduced form of the canonical union is called the *minimal union*.

Introducing the Software

Important technical details and the main functional aspects of each program are presented in Table 2. In particular, we compare version 2.5 of *fs/QCA*, 1.0–4 of *QCA*, and 1.3.2.0 of *Tosmana*, each of

Table 2. Software Overview.

		fs/QCA	QCA	Tosmana
Technical details	Current version	2.5	1.0–4	1.3.2.0
	License type	Freeware	Freeware	Freeware
	User interface	Graphical	Command line	Graphical
	Operating systems	Windows	All *R*-compatible systems	Windows, Mac
	Minimization algorithm	QMC	eQMC	GBA
	Input data formats	.csv,.dat,.qdm,.txt	All *R*-importable formats	.csv,.dat,.tosmana,.xml
Functionality	Variants	csQCA, fsQCA	csQCA, mvQCA, fsQCA	csQCA, mvQCA
	Solution types	Complex, intermediate, parsimonious	Complex, intermediate, parsimonious	Complex, parsimonious
	Parameters of fit	Consistency, PRI, raw coverage, unique coverage	Inclusion, PRI, raw coverage, unique coverage	No parameters of fit
	Case identification	Only in PIs	In truth tables and PIs	In truth tables and PIs
	Simplifying assumptions	No	Yes	Yes
	Factorization	No	Yes	Yes
	Graphics	Bar charts, XY plot	No internal functionality	Venn diagrams for csQCA (up to five sets)
	Special features	Free selection of inessential PIs, directional expectations	Automated necessity analysis, directional expectations, threshold setter for calibration	Threshold setter for calibration, free selection of PIs

Note. fsQCA = fuzzy-set qualitative comparative analysis; QMC = Quine–McCluskey; eQMC = enhanced Quine–McCluskey; GBA = graph-based agent; csQCA = crisp-set QCA; mvQCA = multi-value QCA; PI = prime implicant; PRI = proportional reduction in inconsistency.

which is available as freeware. All programs run under the Windows operating systems (OSs), but only *QCA* and *Tosmana* are also compatible with Mac OS. As a package for the *R* environment for statistical computing and graphics (R Development Core Team, 2012), *QCA* runs under all systems for which *R* can be installed.

On the surface, the most obvious difference between the three programs is their interface. Both *fs/QCA* and *Tosmana* employ a graphical user interface (GUI) and *QCA* a command line interface (CLI). Most social scientists surely feel more comfortable with GUIs, but for users with some experience in syntax-based software or programming, *QCA* may be a preferable alternative. We will not enumerate the advantages of either interface approach here but simply note that a decision for one or the other is generally a matter of mathematical affinity and previous experience.

Each solution has a different orientation with regard to functional capabilities. While all are able to process csQCA as the most basic variant, the focus of *fs/QCA* is on fsQCA and that of *Tosmana* on mvQCA. In contrast, the repertoire of the *QCA* package includes all three variants, which makes it the most versatile of the three programs.

In QCA, different types of solutions can be derived (Ragin & Sonnett, 2005). *Solution types* can be categorized according to the stringency with which logical remainders are allowed to become part of the canonical union as FIs prior to minimization. If analysts let the minimization algorithm choose any logical remainder and its respective supersets that make it possible to absorb a condition and so generate a simpler equivalent to the canonical union, then the *parsimonious solution* will result. If the plausibility of logical remainders is assessed insofar as all of them are declared negative and do not become part of the canonical union, then the *complex* or *conservative* (Schneider & Wagemann, 2012, pp. 165–167) *solution* will be derived. If at least one logical remainder is explicitly added to the canonical union because the analyst considers it a plausible FI, which is often referred to as an *easy counterfactual*, the result will be one element in the larger set of possibilities for an *intermediate solution*. Put differently, the three solution types form a continuum of possible minimization outcomes, with the parsimonious and complex solution types at its extremes. All three programs are capable of deriving complex and parsimonious solutions, but only *fs/QCA* and *QCA* allow analysts to generate intermediate solutions.

The graphical capabilities of the three programs differ immensely, mainly as a result of their functional orientation and user interface approach. While *QCA* provides no indigenous graphical facilities of its own because the *R* environment offers plenty of powerful tools in base and extension packages already (Chen & Boutros, 2011), *fs/QCA* and *Tosmana* cannot draw on this resource as stand-alone programs. In line with *fs/QCA*'s focus, two-dimensional scatterplots are its primary tool for visualizing fuzzy set relations. *Tosmana*, in contrast, offers the "visualizer," with which Venn diagrams of up to five sets can be produced, including the option to color-code intersections according to their status as positive, negative, contradictory, and logical remainder configurations. The largest disadvantage of the visualizer, however, given the focus of *Tosmana* on mvQCA, is the incapability of generating Venn diagrams with multivalent sets.

Each solution also comes with a set of individual features. The *fs/QCA* program provides the possibility to formulate directional expectations about the counterfactual relation between each single condition that is part of a logical remainder and the outcome value. This allows analysts to simplify Boolean models by automatically filtering out those counterfactuals from the parsimonious solution that are considered difficult in order to produce intermediate solutions.

The automatic exclusion of difficult logical remainders is also possible in *QCA*, which offers this tool for all variants, including mvQCA. Another important feature is its ability to search through all set-theoretic unions and intersections and find those that fulfill the analyst's requirements in terms of inclusion and coverage but are minimally complex at the same time. This avoids the manual analysis of necessity relations, which can become a tedious task in *fs/QCA* and *Tosmana*.

Besides the visualizer, another of *Tosmana*'s special features is the "threshold setter", which has been welcomed by many researchers as a useful device (Vink & van Vliet, 2009, p. 266). It clusters cases into distinct groups along the values of the underlying base variable in order to determine their set membership and thus aids the analyst in the set calibration process.

Comparing fs/QCA, QCA, and Tosmana

In this section, we review and compare each program with respect to truth table construction, minimization algorithms, and PI chart management for the derivation of solutions. To illustrate the differences in these aspects, we reanalyze the data from the sociological study by Wollebæk (2010) on rural populations of grassroots associations in 22 municipalities in Hordaland County, Norway, between 1980 and 2000. Among other goals, the author seeks to identify the essential combinations of conditions that prove sufficient for rural grassroots associations to thrive. As it employs csQCA, the study is suitable for a comparative replication of results in all three programs.

Wollebæk introduces eight conditions: the percentage growth in the municipality's population (**PG**), the percentage change in the population living in densely populated areas (**RB**), the percentage change in the population with a higher education (**EL**), the proportion of people leaving state church (**SE**), the proportion of Christian associations (**CS**), the organization density rate (**OD**), the proportion of associations covering a smaller area than the entire municipality (**PC**), and urban proximity (**UP**). The outcome set (**GR**) is defined as the overall percentage change in the number of organizations. The calibrated set data are shown in Table 3.

Truth Tables

Four possible outcome values that can appear in truth tables have been mentioned above: "0" for negative configurations, "1" for positive configurations, "?" for logical remainders, and "C" for contradictions. At the very least, the assignment of one of these values to a configuration depends on its inclusion score and its number of cases. More precisely, the number of cases first influences whether a configuration is coded as a remainder or a nonremainder, following which the inclusion score determines whether a nonremainder configuration is negative, positive, or contradictory. Each software program has a different mechanism for assigning outcome values, and not all make use of all four of them.

The *fs/QCA* truth table is shown in Figure 1. It contains the matrix of configurations, the number of cases with membership above 0.5 in that configuration, the outcome value (which is misleadingly labeled with the name of the outcome set), and three additional consistency columns. A major disadvantage for small *n* case-oriented research is the lack of an option for identifying the cases in their respective configuration.

The *fs/QCA* software does not offer the possibility of coding configurations as contradictions. As a result, the analyst is forced to take a clear decision on the truth value of the subset relationship between a configuration and the outcome set. If there are configurations with mixed evidence and the analyst would like to test whether results change with the inclusion or exclusion of this evidence, the entire coding procedure has to be repeated. However, users can set a cutoff for the number of empirical instances below which a configuration is coded as a logical remainder and a consistency cutoff below which nonremainders are coded as negative, and above which they are coded as positive. The possibility to control these two parameters avoids an overly deterministic approach to assessments of subset relations.

The *QCA* package is more flexible insofar as it allows the outcome value to be based on two different inclusion cutoffs as well as the number of a configuration's cases. Besides the possibility of

Table 3. Set Data From Wollebæk (2010).

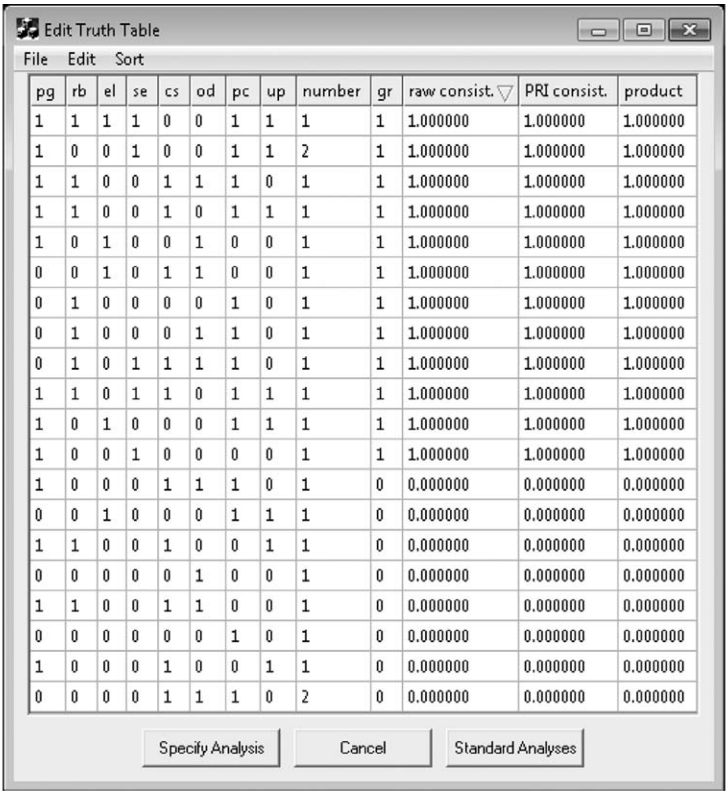
	Conditions								Outcome
Case	PG	RB	EL	SE	CS	OD	PC	UP	GR
Etne	0	1	0	1	1	1	1	0	1
Ølen	1	1	0	0	1	1	1	0	1
Stord	1	0	0	1	0	0	0	0	1
Fitjar	1	1	0	0	1	1	0	0	0
Tysnes	1	0	1	0	0	0	1	1	1
Kvinnherad	1	0	0	0	1	1	1	0	0
Ullensvang	0	0	0	0	0	0	1	0	0
Eidfjord	0	1	0	0	0	1	1	0	1
Ulvik	0	0	0	0	0	1	0	0	0
Granvin	1	0	1	0	0	1	0	0	1
Voss	0	1	0	0	0	0	1	0	1
Kvam	0	0	0	0	1	1	1	0	0
Samnanger	1	1	0	0	1	0	0	1	0
Os	1	0	0	0	1	0	0	1	0
Austevoll	1	0	0	1	0	0	1	1	1
Sund	1	1	0	0	1	0	1	1	1
Fjell	1	1	0	1	1	0	1	1	1
Vaksdal	0	0	1	0	0	0	1	1	0
Meland	1	1	1	1	0	0	1	1	1
Øygarden	1	0	0	1	0	0	1	1	1
Fedje	0	0	1	0	1	1	0	0	1
Masfjorden	0	0	0	0	1	1	1	0	0

Note. PG = percentage growth in the municipality's population; RB = percentage change in the population living in densely populated areas; EL = percentage change in the population with a higher education; SE = proportion of people leaving state church; CS = proportion of Christian associations; OD = organization density rate; PC = proportion of associations covering a smaller area than the entire municipality; UP = urban proximity; GR = outcome set.

using only one inclusion cutoff for coding a configuration as either positive or negative, an interval between a lower and an upper cutoff can be set. Over this interval, a configuration is coded as a contradiction.

The *QCA* truth table is shown in Figure 2. It can be printed on screen either in its full form or in a reduced form that contains only the empirical part. If so specified in the options to the truth table-generating function, the case names can be displayed as well. Each configuration has a unique index value, which is given in the leftmost column. These values serve as identifiers, but they are of no direct relevance to the analyst. In addition to the number of a configuration's cases, *QCA* returns the outcome value, the (sufficiency) inclusion score, and the (sufficiency) PRI (proportional reduction in inconsistency) score.

Tosmana's truth table is shown in Figure 3. In contrast to its two competitors, the program does not offer the option of setting inclusion or case number cutoffs and does not compute any parameters of fit. Instead, it sets two cutoffs implicitly by using $1/n$ as the lower cutoff and $(n - 1)/n$ as the upper cutoff, where n is the number of cases in that configuration. The lack of any information about the degree of set inclusion represents a major disadvantage because it renders a distinction between fragmentary and solid evidence against the hypothesis that the configuration implies the outcome impossible. Irrespective of whether *Tosmana* detects a single case in the data that falsifies the hypothesis in light of more robust empirical evidence in favor of it, or a single case that confirms the hypothesis in light of more robust empirical evidence against it, the software will code either configuration as a contradiction.



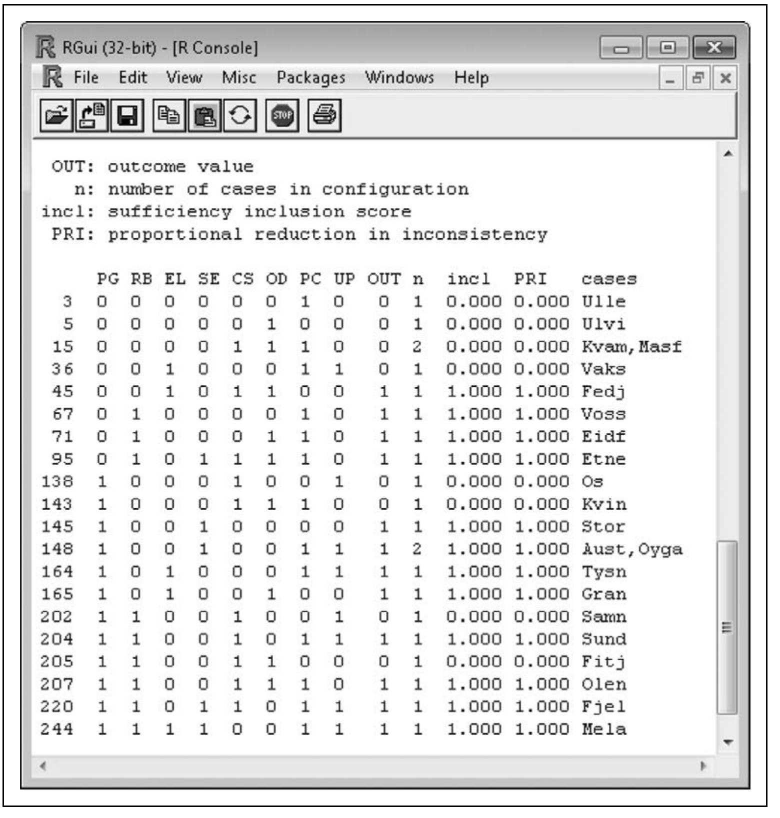
pg	rb	el	se	cs	od	pc	up	number	gr	raw consist.	PRI consist.	product
1	1	1	1	0	0	1	1	1	1	1.000000	1.000000	1.000000
1	0	0	1	0	0	1	1	2	1	1.000000	1.000000	1.000000
1	1	0	0	1	1	1	0	1	1	1.000000	1.000000	1.000000
1	1	0	0	1	0	1	1	1	1	1.000000	1.000000	1.000000
1	0	1	0	0	1	0	0	1	1	1.000000	1.000000	1.000000
0	0	1	0	1	1	0	0	1	1	1.000000	1.000000	1.000000
0	1	0	0	0	0	1	0	1	1	1.000000	1.000000	1.000000
0	1	0	0	0	1	1	0	1	1	1.000000	1.000000	1.000000
0	1	0	1	1	1	1	0	1	1	1.000000	1.000000	1.000000
1	1	0	1	1	0	1	1	1	1	1.000000	1.000000	1.000000
1	0	1	0	0	0	1	1	1	1	1.000000	1.000000	1.000000
1	0	0	1	0	0	0	0	1	1	1.000000	1.000000	1.000000
1	0	0	0	1	1	1	0	1	0	0.000000	0.000000	0.000000
0	0	1	0	0	0	1	1	1	0	0.000000	0.000000	0.000000
1	1	0	0	1	0	0	1	1	0	0.000000	0.000000	0.000000
0	0	0	0	0	1	0	0	1	0	0.000000	0.000000	0.000000
1	1	0	0	1	1	0	0	1	0	0.000000	0.000000	0.000000
0	0	0	0	0	0	1	0	1	0	0.000000	0.000000	0.000000
1	0	0	0	1	0	0	1	1	0	0.000000	0.000000	0.000000
0	0	0	0	1	1	1	0	2	0	0.000000	0.000000	0.000000

Figure 1. Truth table in *fs/QCA*.

Algorithms

The reduction of complex canonical unions to simpler equivalents with identical properties has posed a challenge to electrical engineers and computer scientists for more than half a century. Besides limited graphical tools of solving Boolean minimization problems, including the famous Karnaugh-Veitch map (Karnaugh, 1953; Veitch, 1952), many more sophisticated procedures have been developed. The traditional Quine–McCluskey (QMC) algorithm is implemented in *fs/QCA*, *QCA* employs enhanced QMC (*e*QMC) and *Tosmana* draws on GBA (graph-based agent). A fuller explanation of each algorithm would require a more involved treatment, so we only introduce their basic idea, advantages, and disadvantages in this section.

QMC has been the most well-known procedure for minimizing Boolean functions (McCluskey, 1956; Quine, 1952, 1955). No textbook on Boolean algebra, switching circuit theory or logic design misses a section on it (Edwards, 1973, p. 98ff.; Hohn, 1966, 201ff.; Lewin & Protheroe, 1992, p. 76ff.). It is exact and able to process moderately complex models within a reasonable time frame given modern computing technology. The core idea has already been introduced in the section on Boolean minimization above. In essence, all FIs and their implicants are examined exhaustively in a systematic way to test for the applicability of the law of excluded middle. The output from this process is a complete list of PIs, which is then further reduced to find the minimal union. Notwithstanding QMC's robustness and exactitude, its brute-force approach is demanding in both time and computer memory resources, as a result of which the upper limit of model complexity that can be handled by *fs/QCA* lies between 11 and 12 conditions.




```
 OUT: outcome value
   n: number of cases in configuration
incl: sufficiency inclusion score
 PRI: proportional reduction in inconsistency
```

	PG	RB	EL	SE	CS	OD	PC	UP	OUT	n	incl	PRI	cases
3	0	0	0	0	0	0	1	0	0	1	0.000	0.000	Ulle
5	0	0	0	0	0	1	0	0	0	1	0.000	0.000	Ulvi
15	0	0	0	0	1	1	1	0	0	2	0.000	0.000	Kvam,Masf
36	0	0	1	0	0	0	1	1	0	1	0.000	0.000	Vaks
45	0	0	1	0	1	1	0	0	1	1	1.000	1.000	Fedj
67	0	1	0	0	0	0	1	0	1	1	1.000	1.000	Voss
71	0	1	0	0	0	1	1	0	1	1	1.000	1.000	Eidf
95	0	1	0	1	1	1	1	0	1	1	1.000	1.000	Etne
138	1	0	0	0	1	0	0	1	0	1	0.000	0.000	Os
143	1	0	0	0	1	1	1	0	0	1	0.000	0.000	Kvin
145	1	0	0	1	0	0	0	0	1	1	1.000	1.000	Stor
148	1	0	0	1	0	0	1	1	1	2	1.000	1.000	Aust,Oyga
164	1	0	1	0	0	0	1	1	1	1	1.000	1.000	Tysn
165	1	0	1	0	0	1	0	0	1	1	1.000	1.000	Gran
202	1	1	0	0	1	0	0	1	0	1	0.000	0.000	Samn
204	1	1	0	0	1	0	1	1	1	1	1.000	1.000	Sund
205	1	1	0	0	1	1	0	0	0	1	0.000	0.000	Fitj
207	1	1	0	0	1	1	1	0	1	1	1.000	1.000	Olen
220	1	1	0	1	1	0	1	1	1	1	1.000	1.000	Fjel
244	1	1	1	1	0	0	1	1	1	1	1.000	1.000	Mela

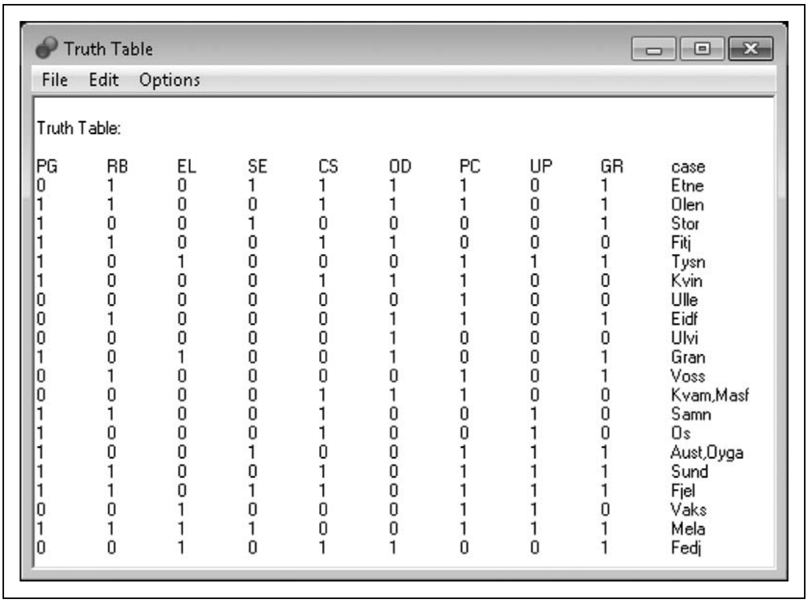
Figure 2. Truth table in *QCA*.

The *e*QMC algorithm of *QCA* compares the set of positive configurations with the set of negative configurations. More precisely, it treats each configuration as a compound set, all of whose supersets form the entire set of possible implicants. By consecutively eliminating those supersets that have a subset in both the set of positive and negative configurations using line index vectors from an auxiliary implicant matrix, *e*QMC ultimately finds all PIs that are supersets of all positive configurations without also being supersets of any negative configuration. In the final step, it eliminates those PIs that have supersets of their own among the surviving ones. As this procedure is very economical, *e*QMC can process much more complex models than QMC.

Similar to *e*QMC, *Tosmana*'s GBA does not need to run through all possible pairs of configurations and implicants. Instead, it compares two sets of cases in a matrix whose two dimensions represent the conditions along one axis and their values along the other. The first set includes all cases that share the value of the outcome set to be explained and the second set includes all cases that do not share this value. Implicants are then found by establishing common paths through the matrix for the first set of cases, which are not completely followed by any path through the matrix for the second set of cases. Since this approach is also highly efficient, *Tosmana* can handle an even larger number of conditions than *e*QMC under some circumstances.

Solution Types and PI Charts

The Boolean minimization process often ends before the derivation of a minimal union. If not only essential PIs make up the solution and alternative minimal unions exist, a PI chart is constructed and



PG	RB	EL	SE	CS	OD	PC	UP	GR	case
0	1	0	1	1	1	1	0	1	Etne
1	1	0	0	1	1	1	0	1	Olen
1	0	0	1	0	0	0	0	1	Stor
1	1	0	0	1	1	0	0	0	Fitj
1	0	1	0	0	0	1	1	1	Tysn
1	0	0	0	1	1	1	0	0	Kvin
0	0	0	0	0	0	1	0	0	Ulle
0	1	0	0	0	1	1	0	1	Eidf
0	0	0	0	0	1	0	0	0	Ulvi
1	0	1	0	0	1	0	0	1	Gran
0	1	0	0	0	0	1	0	1	Voss
0	0	0	0	1	1	1	0	0	Kvam,Masf
1	1	0	0	1	0	0	1	0	Samn
1	0	0	0	1	0	0	1	0	Os
1	0	0	1	0	0	1	1	1	Aust,Oyga
1	1	0	0	1	0	1	1	1	Sund
1	1	0	1	1	0	1	1	1	Fjel
0	0	1	0	0	0	1	1	0	Vaks
1	1	1	1	0	0	1	1	1	Mela
0	0	1	0	1	1	0	0	1	Fedj

Figure 3. Truth table in *Tosmana*.

Table 4. Generic PI Chart With Row Dominance.

	FI					
PI	C_1	C_2	C_3	C_4	C_5	C_6
$\mathbf{X}_1\{1\}\mathbf{X}_2\{1\}\mathbf{X}_3\{1\}$	×	–	×	×	×	×
$\mathbf{X}_1\{1\}\mathbf{X}_2\{0\}\mathbf{X}_3\{1\}$	–	×	–	–	–	×
$\mathbf{X}_1\{1\}\mathbf{X}_2\{1\}\mathbf{X}_3\{0\}$	–	×	–	–	–	–

must be solved according to a specified set of rules. The procedures for solving these charts are independent of the minimization algorithm, but while analysts need not necessarily fully understand how canonical unions are reduced, it is absolutely crucial to comprehend the purpose and structure of PI charts. Based on conversations with colleagues, conference presentations, and our experiences as reviewers of submissions to scientific journals, we can claim with some confidence that the PI chart is one of the least understood devices in QCA.

In the classical version of such a chart, the column headings are the index values of the FIs (usually the positive configurations from the truth table) and the rows contain the PIs. A generic PI chart of six FIs and three PIs is shown in Table 4. Crosses (×) indicate that a PI includes an FI, whereas dashes (–) mean that an FI is not included by a PI. The *complete union* is the set-theoretic union of all PIs, but usually only the minimal union is sought. However, sometimes there exist multiple PIs that could be part of a minimal union. PIs with no alternative are *essential*. Conversely, those which have alternatives are *inessential*. After all essential PIs and those inessential PIs which are minimally required to cover the canonical union have been incorporated into the minimal union, the final solution has been derived.

For example, Table 4 shows a PI chart that contains exactly one essential and two inessential PIs: $\mathbf{X}_1\{1\}\mathbf{X}_2\{1\}\mathbf{X}_3\{1\}$ is essential because there is no other PI covering configurations C_1, C_3, C_4,

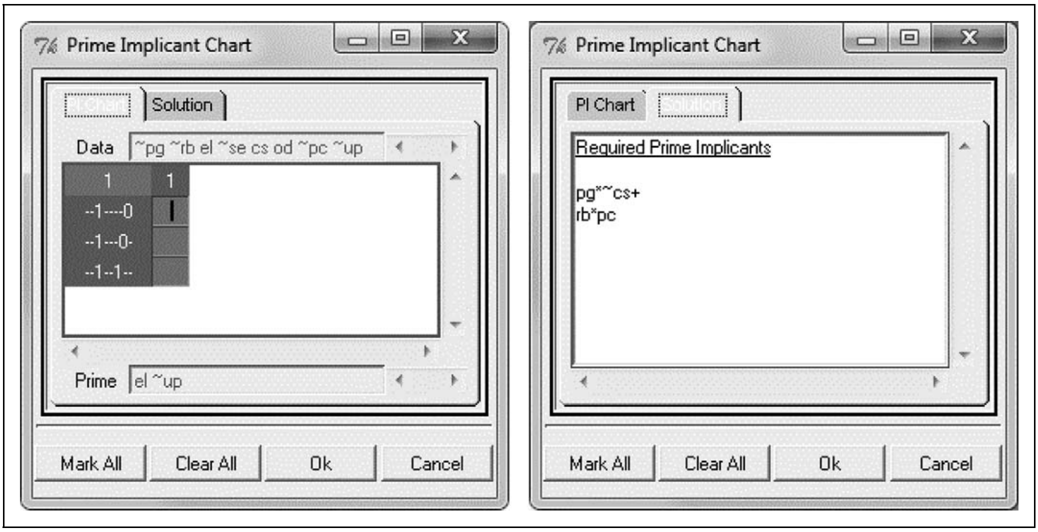


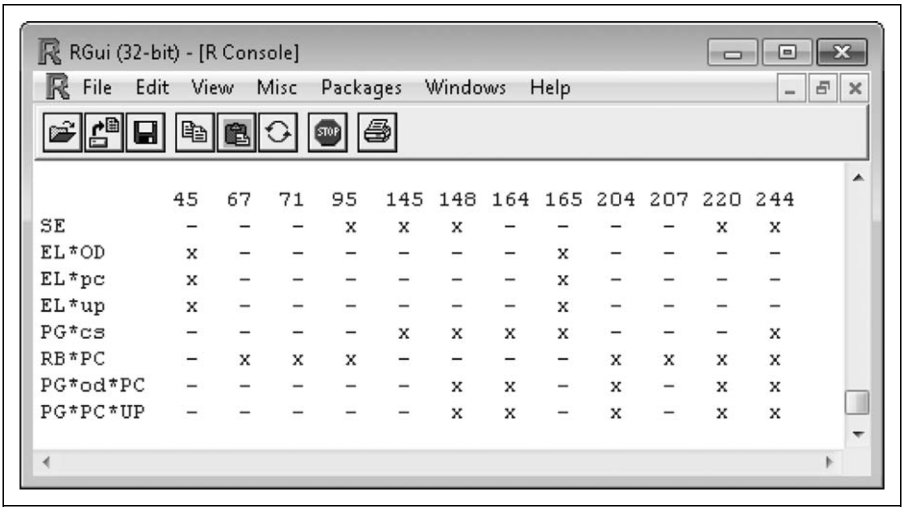
Figure 4. PI chart in *fs/QCA*.

and C_5 (for the sake of brevity, the intersection operator is dropped). Consequentially, this PI has to be a part of any solution. Both $\mathbf{X}_1\{1\}\mathbf{X}_2\{0\}\mathbf{X}_3\{1\}$ and $\mathbf{X}_1\{1\}\mathbf{X}_2\{1\}\mathbf{X}_3\{0\}$, in contrast, are inessential because either one covers C_2. Therefore, either one or both also have to be part of the solution. The three software programs all have their own style of presenting and solving PI charts, so it is important for end users to be aware of similarities and differences as results may be affected. Multiple possibilities for deriving a minimal union may occur under all solution types, but it is most probable in the case of the parsimonious solution because the likelihood of one PI including a growing number of FIs increases. In the demonstrations to follow, we focus on the parsimonious solution.

Figure 4 shows the two tabs of *fs/QCA*'s PI chart. Strictly speaking, the software does not present the full chart but only that submatrix of it which contains the FIs in question and the PIs that cover them. This window will pop up automatically whenever two or more implicants survive the minimization process as inessential PIs.

The tab "PI Chart", shown in the left panel of Figure 4, presents the analyst with the options available for completing the minimal union. The row called "Data" shows the FI that still needs covering in the solution. Right below this row, *fs/QCA* prints the concerned part of the PI chart in binary format and a "Prime" row, which translates this binary information into the corresponding label format. The binary representation of the inessential PI "el ∼up" (**EL**{1}**UP**{0}) is "- -1- - - - 0", where each dash stands for an absorbed condition in the order of the conditions originally provided by the user for the construction of the truth table (**PG**, **RB**, **SE**, **CS**, **OD**, and **PC**), the value "1" denotes the presence of **EL** (**EL**{1}) and the value "0" the absence of **UP** (**UP**{0}). Analogously, "- -1 - - -0-" stands for "el ∼pc" (**EL**{1}**PC**{0}) and "- -1- -1- -" for "el ∼od" (**EL**{1}**OD**{0}). Thus, three inessential PIs exist, each of which on its own could complete the solution. But users may also choose any combination of two PIs or even all three by clicking the "Mark All" button at the bottom of the window or by marking each cell.

The tab "Solution", shown in the right panel of Figure 4, presents the union of essential PIs. Both "pg*∼cs" (**PG**{1}**CS**{0}) and "rb*pc" (**RB**{1}**PC**{1}) constitute nonsubstitutable parts of any minimal union. Once the desired PIs have been chosen, a click on the Ok button will apply the constraints submitted by the user to derive the solution. If the procedure is repeated for each PI, *fs/QCA*



	45	67	71	95	145	148	164	165	204	207	220	244
SE	-	-	-	x	x	x	-	-	-	-	x	x
EL*OD	x	-	-	-	-	-	-	x	-	-	-	-
EL*pc	x	-	-	-	-	-	-	x	-	-	-	-
EL*up	x	-	-	-	-	-	-	x	-	-	-	-
PG*cs	-	-	-	-	x	x	x	x	-	-	-	x
RB*PC	-	x	x	x	-	-	-	-	x	x	x	x
PG*od*PC	-	-	-	-	-	x	x	-	x	-	x	x
PG*PC*UP	-	-	-	-	-	x	x	-	x	-	x	x

Figure 5. PI chart in *QCA*.

will therefore find three minimal unions: **PG**{1}**CS**{0} ∪ **RB**{1}**PC**{1} ∪ **EL**{1}**OD**{1}, **PG**{1}**CS**{0} ∪ **RB**{1}**PC**{1} ∪ **EL**{1}**PC**{0}, and **PG**{1}**CS**{0} ∪ **RB**{1}**PC**{1} ∪ **EL**{1}**UP**{0}. Logically, users would concentrate only on these three possibilities in their substantive interpretations, particularly with regard to the role of the three distinct elements in each minimal union (**OD**{1}, **PC**{0} and **UP**{0}).

The *QCA* package takes a very different approach. It allows no direct user intervention in the solution of the PI chart but only permits analysts to decide beforehand whether or not the principle of *row dominance* should be applied. As the columns always represent the FIs, this principle refers to the PIs. Put precisely, one PI P_1 dominates another P_2 if all FIs covered by P_2 are also covered by P_1 and both are not interchangeable (cf. McCluskey, 1965, p. 150). For example, $\mathbf{X}_1${1}$\mathbf{X}_2${0}$\mathbf{X}_3${1} dominates $\mathbf{X}_1${1}$\mathbf{X}_2${1}$\mathbf{X}_3${0} in Table 3 because not only do both PIs cover configuration C_2, but they are also not interchangeable. Besides C_2, the former PI includes C_6 in addition, which the latter does not. In contrast to *QCA*, *fs/QCA* never applies the principle of row dominance. Since this principle has originated from cost considerations in the design of electrical switching circuits, its absence causes no problems in social–scientific research.

As a result, PI charts in *QCA* can only be inspected by the user after a solution has been found. The PI chart of Wollebæk's Boolean model with the principle of row dominance applied is shown in Figure 5. Instead of using the label format of the FIs, the column header entries are index numbers that equal the index numbers shown in the truth table in Figure 2. With regard to the essential PIs, the PI chart generated by *QCA* shows both "PG*cs" (**PG**{1}**CS**{0}) and "RB*PC" (**RB**{1}**PC**{1}) together to cover all FIs except C_{45}. Most importantly, however, C_{45} corresponds to the truth table row displayed in the "Data" row of *fs/QCA*'s PI chart (Figure 4). Recall that *fs/QCA* showed this configuration to be covered by any one of three PIs **EL**{1}**UP**{0}, **EL**{1}**PC**{0}, and **EL**{1}**OD**{1}. In the *QCA* PI chart, it can be seen that these three PIs are not only functionally equivalent but also fully interchangeable. No PI dominates another because all three cover no configuration apart from C_{45} and C_{165}. Users cannot obtain this information from the PI chart in *fs/QCA*.

While the fact that presented PIs may be fully interchangeable has only informative value, a more important difference between *fs/QCA* and *QCA* is consequential for the substantive interpretation of

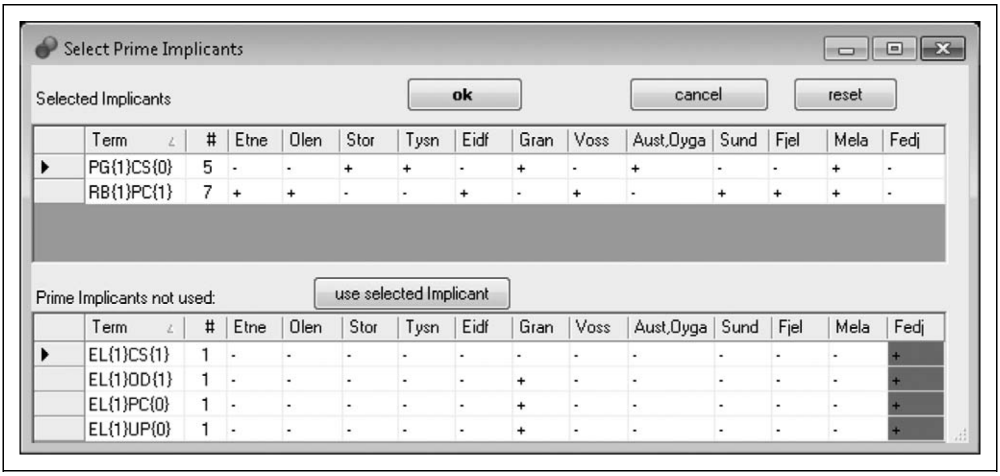


Figure 6. PI chart in *Tosmana*.

results. The *fs/QCA* software will not present dominated inessential PIs if at least two inessential but fully interchangeable PIs exist. This implies that alternative minimal unions will be hidden from the complete union if multiple, but functionally identical inessential implicants survive the Boolean minimization process. If only a single dominating PI exists, all inessential PIs, both dominated and dominating ones, will be listed.

The PI chart of *Tosmana* is shown in Figure 6. It resembles that of *QCA* insofar as a full PI chart is displayed, with case names as column header entries instead of configuration index numbers, but similar to *fs/QCA* the software will not reveal all information under certain conditions. More precisely, if inessential dominated PIs are functionally more complex than the functionally least complex dominating inessential PIs, they will be ignored. In contrast, inessential dominated PIs which are functionally not more complex than the inessential dominating PIs will be included in the PI chart. We elaborate on this idiosyncrasy in the following.

For deriving solutions, *Tosmana* offers three options, called *Selection Modes*. The first mode ‘‘Show All’’ lets the software derive the solution, the second mode ‘‘Select if Possible’’ allows the manual selection of those inessential PIs that would have also been used if the first mode had been chosen, and the third mode ‘‘Free Selection’’ lets the user construct the entire solution manually by adding PIs until all FIs are covered. Since the last option is of little direct analytical importance, we focus on the second selection mode, also because this mode is closest to the procedure of solving PI charts in *fs/QCA*.

Choosing the selection mode ‘‘Select if Possible’’, *Tosmana* presents the following four inessential PIs: **EL**{1}**CS**{1}, **EL**{1}**OD**{1}, **EL**{1}**PC**{0}, and **EL**{1}**UP**{0}. The last three of these PIs are also presented in *fs/QCA*’s and *QCA*’s chart, but the first PI appears in neither. It is clear that **EL**{1}**CS**{1} would complete the solution because it covers ‘‘Fedj’’ (C_{45}), but it is also dominated by the other three alternatives as it does not cover ‘‘Gran’’ (C_{165}). The analyst cannot conclude from this chart that *Tosmana* does not apply the principle of row dominance, but it is obvious that its PI chart differs from that of *fs/QCA*. The latter does not list **EL**{1}**CS**{1} as an inessential PI. Since neither program offers the option of controlling row dominance, we will first inspect the results *QCA* generates, which are exact and fully exhaustive. The parsimonious solution with row dominance disabled returns the seven minimal unions S_1 to S_7 listed in Figure 7. Unless raw coverage acts a selection criterion, none is preferable over the other. Each inessential PI receives the same unique coverage score of 0.077.

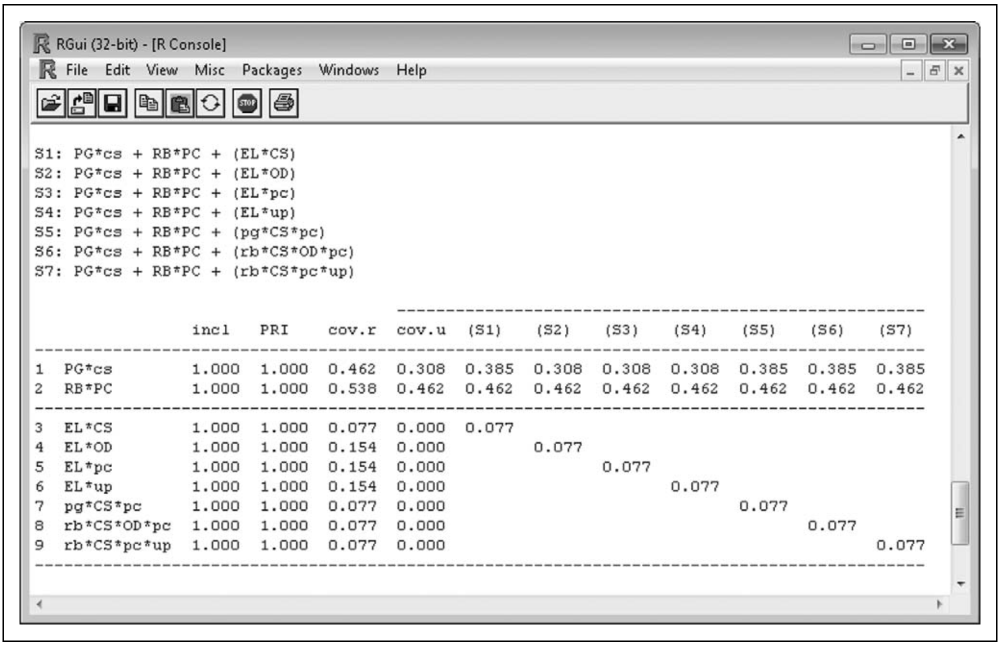


Figure 7. Solution output in *QCA*.

For reasons of notational consistency, these seven alternative minimal unions are reproduced below.

$$S_1 : \mathbf{PG}\{\mathbf{1}\}\mathbf{CS}\{\mathbf{0}\} \cup \mathbf{RB}\{\mathbf{1}\}\mathbf{PC}\{\mathbf{1}\} \cup \mathbf{EL}\{\mathbf{1}\}\mathbf{CS}\{\mathbf{1}\}$$

$$S_2 : \mathbf{PG}\{\mathbf{1}\}\mathbf{CS}\{\mathbf{0}\} \cup \mathbf{RB}\{\mathbf{1}\}\mathbf{PC}\{\mathbf{1}\} \cup \mathbf{EL}\{\mathbf{1}\}\mathbf{OD}\{\mathbf{1}\}$$

$$S_3 : \mathbf{PG}\{\mathbf{1}\}\mathbf{CS}\{\mathbf{0}\} \cup \mathbf{RB}\{\mathbf{1}\}\mathbf{PC}\{\mathbf{1}\} \cup \mathbf{EL}\{\mathbf{1}\}\mathbf{PC}\{\mathbf{0}\}$$

$$S_4 : \mathbf{PG}\{\mathbf{1}\}\mathbf{CS}\{\mathbf{0}\} \cup \mathbf{RB}\{\mathbf{1}\}\mathbf{PC}\{\mathbf{1}\} \cup \mathbf{EL}\{\mathbf{1}\}\mathbf{UP}\{\mathbf{0}\}$$

$$S_5 : \mathbf{PG}\{\mathbf{1}\}\mathbf{CS}\{\mathbf{0}\} \cup \mathbf{RB}\{\mathbf{1}\}\mathbf{PC}\{\mathbf{1}\} \cup \mathbf{PG}\{\mathbf{0}\}\mathbf{CS}\{\mathbf{1}\}\mathbf{PC}\{\mathbf{0}\}$$

$$S_6 : \mathbf{PG}\{\mathbf{1}\}\mathbf{CS}\{\mathbf{0}\} \cup \mathbf{RB}\{\mathbf{1}\}\mathbf{PC}\{\mathbf{1}\} \cup \mathbf{RB}\{\mathbf{0}\}\mathbf{CS}\{\mathbf{1}\}\mathbf{OD}\{\mathbf{1}\}\mathbf{PC}\{\mathbf{0}\}$$

$$S_7 : \mathbf{PG}\{\mathbf{1}\}\mathbf{CS}\{\mathbf{0}\} \cup \mathbf{RB}\{\mathbf{1}\}\mathbf{PC}\{\mathbf{1}\} \cup \mathbf{RB}\{\mathbf{0}\}\mathbf{CS}\{\mathbf{1}\}\mathbf{PC}\{\mathbf{0}\}\mathbf{UP}\{\mathbf{0}\}$$

As only inessential PIs that consist of two condition values are returned by *Tosmana* (Figure 6), it only finds S_1 to S_4 but not S_5 to S_7. This leads to the presumption that the software chooses its PIs with regard not only to row dominance but also to PI complexity. The only inessential dominated PI that is not more complex than the three dominating PIs is $\mathbf{EL}\{1\}\mathbf{CS}\{1\}$. Users would therefore concentrate on these four possibilities in their substantive interpretations, particularly with regard to the role of the four distinct condition values in each minimal union: $\mathbf{CS}\{1\}$, $\mathbf{OD}\{1\}$, $\mathbf{PC}\{0\}$, and $\mathbf{UP}\{0\}$. If they had worked with *fs/QCA*, the conjunction of $\mathbf{EL}\{1\}$ and $\mathbf{CS}\{1\}$ would not have been revealed. However, *Tosmana* also misses three equally fitting minimal unions that could have formed important parts in the substantive interpretation or further case study analysis.

In summary, both *fs/QCA* and *Tosmana* show some idiosyncrasies in solving PI charts for deriving minimal unions. Most importantly, *fs/QCA* has been most selective in its presentation of inessential PIs, returning only three of seven possible ones, all of which were dominating. *Tosmana* presented four PIs, one of which was dominated, but it did not present dominated inessential PIs of a higher complexity. These particularities pose major problems for empirical research because users' attention is directed toward specific subsets of possible solutions, whereas equally fitting models remain hidden from their eyes. In addition, this behavior is not documented. The manual of neither program includes a section on or reference to its exact procedure for solving PI charts. Given that the PI chart seems to be the most confusing device in QCA for many end users, this situation warrants more attention by software developers, course instructors, and journal reviewers.

Conclusion

The QCA software market has broadened considerably over the last couple of years, testifying to the diffusion of QCA as a method across many subfields of the social sciences. In this review, we surveyed three computer programs with regard to some important yet often not well-understood technical particularities. Although more programs for performing QCA exist by now, concentrating on the three most developed of them allowed us to review each solution in more detail. After having provided a brief repetition of the most important basic concepts in QCA and a concise introduction to the functional capabilities of each program, the focus has been put on the topics of truth table construction, minimization algorithms, PI chart management, and the derivation of solutions because each program shows some particularities in these respects, most importantly with regard to PI chart management.

Although the first QCA software has been around for almost 25 years now, computational advances in terms of both functional breadth and depth appear on a regular basis. While this is to be highly welcomed, it also increases the demands put on end users with respect to methodological literacy and computational proficiency. If software developers, course instructors, and journal reviewers succeed in helping applied end users meet these demands, the positive prospects for QCA at the juncture of qualitative and quantitative data analysis outlined by Blank (1991), Brent (1993), and Heise (1992) some 20 years ago may surely be renewed at this point.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

Amenta, E., Carruthers, B. G., & Zylan, Y. (1992). A hero for the aged: The townsend movement, the political mediation model, and United-States old-age policy, 1934-1950. *American Journal of Sociology*, *98*, 308–339.

Blank, G. (1991). Why sociological computing gets no respect. *Social Science Computer Review*, *9*, 593–611.

Brent, E. E. (1993). Computational sociology: Reinventing sociology for the next millennium. *Social Science Computer Review*, *11*, 487–499.

Chen, H., & Boutros, P. (2011). VennDiagram: A package for the generation of highly-customizable Venn and Euler diagrams in R. *BMC Bioinformatics*, *12*, 35–41.

Cronqvist, L. (2011). *Tosmana: Tool for small-n analysis, Version 1.3.2.0 [Computer Program]*. Trier, Germany: University of Trier.

Cronqvist, L., & Berg-Schlosser, D. (2009). Multi-value QCA (mvQCA). In B. Rihoux & C. C. Ragin (Eds.), *Configurational comparative methods: Qualitative comparative analysis (QCA) and related techniques* (pp. 69–86). London, England: Sage.

Drass, K. A. (1988). *QCA: Qualitative Comparative Analysis [Computer Program]*. Dallas, TX: Southern Methodist University, Department of Sociology.

Duşa, A., & Thiem, A. (2012). *QCA: Qualitative Comparative Analysis, R Package Version 1.0-4* [Computer Program]. Bucharest: University of Bucharest.

Edwards, F. H. (1973). *The principles of switching circuits*. Cambridge, MA: MIT Press.

Fischer, J., Kaiser, A., & Rohlfing, I. (2006). The push and pull of ministerial resignations in Germany, 1969-2005. *West European Politics*, *29*, 709–735.

Glaesser, J., & Cooper, B. (2011). Selectivity and flexibility in the German secondary school system: A configurational analysis of recent data from the German socio-economic panel. *European Sociological Review*, *27*, 570–585.

Grandori, A., & Furnari, S. (2008). A chemistry of organization: Combinatory analysis and design. *Organization Studies*, *29*, 459–485.

Griffin, L. J., Botsko, C., Wahl, A.-M., & Isaac, L. W. (1991). Theoretical generality, case particularity: Qualitative comparative analysis of trade-union growth and decline. *International Journal of Comparative Sociology*, *32*, 110–136.

Harkreader, S., & Imershein, A. W. (1999). The conditions for state action in Florida's health-care market. *Journal of Health and Social Behavior*, *40*, 159–174.

Heise, D. R. (1992). Computer assistance in qualitative sociology. *Social Science Computer Review*, *10*, 531–543.

Hohn, F. E. (1966). *Applied Boolean algebra: An elementary introduction* (2nd ed.). New York, NY: Macmillan.

Karnaugh, M. (1953). The map method for synthesis of combinational logic circuits. *Transactions of the AIEE*, *72*, 593–599.

Lewin, D., & Protheroe, D. (1992). *Design of logic systems* (2nd ed.). London, England: Chapman & Hall.

McCluskey, E. J. (1956). Minimization of Boolean functions. *Bell Systems Technical Journal*, *35*, 1417–1444.

McCluskey, E. J. (1965). *Introduction to the Theory of Switching Circuits*. Princeton: Princeton University Press.

Quine, W. V. (1952). The problem of simplifying truth functions. *American Mathematical Monthly*, *59*, 521–531.

Quine, W. V. (1955). A way to simplify truth functions. *American Mathematical Monthly*, *62*, 627–631.

Ragin, C. C. (1987). *The comparative method: Moving beyond qualitative and quantitative strategies*. Berkeley: University of California Press.

Ragin, C. C. (2000). *Fuzzy-set social science*. Chicago, IL: University of Chicago Press.

Ragin, C. C. (2006). Set relations in social research: Evaluating their consistency and coverage. *Political Analysis*, *14*, 291–310.

Ragin, C. C. (2008). *Redesigning social inquiry: Fuzzy sets and beyond*. Chicago, IL: University of Chicago Press.

Ragin, C. C. (2009). Qualitative comparative analysis using fuzzy sets (fsQCA). In B. Rihoux & C. C. Ragin (Eds.), *Configurational comparative methods: Qualitative comparative analysis (QCA) and related techniques* (pp. 87–121). London, England: Sage.

Ragin, C. C., & Davey, S. (2009). *fs/QCA: Fuzzy-Set/Qualitative Comparative Analysis Version 2.5 [Computer Program]*. Tucson: Department of Sociology, University of Arizona.

Ragin, C. C., Mayer, S. E., & Drass, K. A. (1984). Assessing discrimination: A Boolean approach. *American Sociological Review*, *49*, 221–234.

Ragin, C. C., & Sonnett, J. (2005). Between complexity and parsimony: Limited diversity, counterfactual cases and comparative analysis. In S. Kropp & M. Minkenberg (Eds.), *Vergleichen in der Politikwissenschaft* (pp. 180–197). Wiesbaden, Germany: VS Verlag für Sozialwissenschaften.

R Development Core Team. (2012). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Redding, K., & Viterna, J. S. (1999). Political demands, political opportunities: Explaining the differential success of left-libertarian parties. *Social Forces*, *78*, 491–510.

Rihoux, B., & de Meur, G. (2009). Crisp-set qualitative comparative analysis (csQCA). In B. Rihoux & C. C. Ragin (Eds.), *Configurational comparative methods: Qualitative comparative analysis (QCA) and related techniques* (pp. 33–68). London, England: Sage.

Schneider, C. Q., & Wagemann, C. (2012). *Set-theoretic methods for the social sciences: A guide to qualitative comparative analysis (QCA)*. Cambridge, MA: Cambridge University Press.

Seeleib-Kaiser, M., & Fleckenstein, T. (2009). The political economy of occupational family policies: Comparing workplaces in Britain and Germany. *British Journal of Industrial Relations*, *47*, 741–764.

Thiem, A. (2011). Conditions of intergovernmental armaments cooperation in Western Europe, 1996-2006. *European Political Science Review*, *3*, 1–33.

Thiem, A., & Duşa, A. (2012). Introducing the QCA package: a market analysis and software review. *Qualitative & Multi-Method Research*, *10*, 45–49.

Thiem, A., & Duşa, A. (2013). *Qualitative comparative analysis with R: A user's guide*. New York, NY: Springer.

Veitch, E. W. (1952). A chart method for simplifying truth functions. In *Proceedings of the 1952 ACM National Meeting* (pp. 127–133). Pittsburgh, PA: ACM.

Vink, M. P., & van Vliet, O. (2009). Not quite crisp, not yet fuzzy? Assessing the potentials and pitfalls of multi-value QCA. *Field Methods*, *21*, 265–289.

Weitzman, E. A., & Miles, M. B. (1995). *Computer programs for qualitative data analysis: A software sourcebook*. Thousand Oaks, CA: Sage.

Wickham-Crowley, T. P. (1991). A qualitative comparative approach to Latin American revolutions. *International Journal of Comparative Sociology*, *32*, 82–109.

Wollebæk, D. (2010). Volatility and growth in populations of rural associations. *Rural Sociology*, *75*, 144–166.

Author Biographies

Alrik Thiem is a postdoctoral researcher in the Department of Humanities, Social and Political Sciences of the Swiss Federal Institute of Technology Zurich (ETHZ), Switzerland. His research interests are in social science methods and foreign policy analysis.

Adrian Duşa is a lecturer in the Department of Sociology of the University of Bucharest, Romania, and director of the Romanian Social Data Archive. His research interests are in social science research methods, social inequalities, and stratification.