Simple Laws about
Nonprominent Properties of Binary Relations

Jochen Burghardt

jochen.burghardt@alumni.tu-berlin.de

Jun 2018

Abstract

We checked each binary relation on a 5-element set for a given set of properties, including usual ones like asymmetry and less known ones like Euclideanness. Using a poor man’s Quine-McCluskey algorithm, we computed prime implicants of non-occurring property combinations, like “not irreflexive, but asymmetric”. We considered the non-trivial laws obtained this way, and manually proved them true for binary relations on arbitrary sets, thus contributing to the encyclopedic knowledge about less known properties.

Key words: Binary relation; Quine-McCluskey algorithm; Hypotheses generation

Contents

1 Introduction 2
2 Formal proofs of property laws 5
3 Computed relation properties 13
References 14

List of Figures

1 Timing vs. universe cardinality 4
2 Encoding scheme for 3 × 3 relations for a [Bur02] approach 4
3 Tree grammar sketch for [Bur02] approach 4
4 Right Euclidean relation 12
5 Left Euclidean relation 12
6 Search graph for the Quine-McCluskey algorithm on 3 variables 14
7 Reported laws for level 3 15
8 Reported laws for level 2 16
9 Reported laws for level 4 16
10 Reported laws for level 5 16
1 Introduction

In order to flesh out encyclopedic articles\footnote{at https://en.wikipedia.org} about less common properties (like e.g. anti-transitivity) of binary relations, we implemented a simple C program to iterate over all relations on a small finite set and to check each of them for given properties. We implemented checks for the 15 properties given in Def. 1 below.

This way, we could, in a first stage, (attempt to) falsify intuitively found hypotheses about laws involving such properties, and search for illustrative counter-examples to known, or intuitively guessed, non-laws (such as a right Euclidean relation that isn’t transitive).

Relations on a set of up to 6 elements could be dealt with in reasonable time on a 2.3 GHz CPU. Figure 1 gives an overview, where all times are wall clock times in seconds, and “tr⇒qtr” indicates the task of validating that each transitive binary relation is also quasi-transitive. Note the considerable amount of compile time, caused by excessive use of inlining, deeply nested loops, and abuse of array elements as loop variables.

In a second stage, we aimed at supporting the generation of law hypotheses, rather than their validation.

We used a 5-element universe set, and checked each binary relation for each of the 15 properties. The latter were encoded in the order from Def. 1 by bits of a 16-bit word, starting from the least significant one. After that, we applied a poor-man’s Quine-McCluskey algorithm\footnote{See [McC56a,McC56b] for the original algorithm.} (denoted “QMc” in Fig. 1) to obtain a short description of property combinations that didn’t occur at all. For example, an output line “\texttt{~Irrefl ASym}” indicated that the program didn’t find any relation that was asymmetric but not irreflexive, i.e. that each asymmetric relation on a 5-element set is irreflexive. Section 3 shows the complete output on a 5-element universe.

We took each printed law as a suggestion to be proven for all binary relations (on arbitrary sets), except that we didn’t yet consider laws involving semi-order properties 1 or 2. Many of the considered laws were trivial, in particular those involving co-reflexivity, as this property applies only to a relatively small number of relations (32 on a 5-element set).

A couple of laws appeared to be interesting, and we could prove them fairly easily by hand for the general case\footnote{We needed to require a minimum cardinality of the universe set in Lemma 8.2}. For those laws involving less usual prop-
erties (like anti-transitivity, quasi-transitivity, Euclideanness) there is good chance that they haven’t been stated in the literature before. However, while they may contribute to the completeness of an encyclopedia, it is not clear whether they may serve any other purpose.

One of the laws, viz. Lemma 9.5, appeared surprising, but turned out during the proof to be vacuously true. The proof attempt to some laws gave rise to the assertion of other lemmas that weren’t directly obtained from the computed output: Lemma 4.1 was needed for the proof of Lemma 5.5, and Lemma 8.3 was needed for Lemma 9.7.

Our Quine-McCluskey approach restricts law suggestions to formulas of the form $\forall R. \, \text{prop}_1(R) \lor \ldots \lor \text{prop}_n(R)$, where the quantification is over all binary relations, and $\text{prop}_i$ is one of the 15 considered properties or a negation thereof.

For an approach to compute more general forms of law suggestions, see [Bur02]; however, due to its run-time complexity this approach is feasible only for even smaller universe sets. In order to handle all relations on a 3-element set, a regular tree grammar of 512 nonterminals, one for each relation, plus 2 nonterminals, one for each truth value, would be needed. Using the encoding scheme from Fig. 2, the original grammar would consist of rules as sketched in Fig. 3. However, this grammar grows very large, and its $n$-fold product would be needed if all laws in $n$ variables were to be computed.

The rest of this paper is organized as follows. In Sect. 2, we formally define each considered property, and give the proofs of nontrivial laws about them. In addition, we state the proofs of some laws that weren’t of the form admitted by our approach; some of them were, however, obtained using the assistance of the counter-example search in our C program. In Sect. 3, we comment on some program details, and show the annotated output for a run on a 5-element set, also indicating which law suggestions gave rise to which lemmas of Sect. 2.

4 For sake of simplicity, only one unary and one binary operation on relations is considered, viz. symmetric closure $\text{symCls}$ and union $\cup$. Only two properties of relations are considered, viz. reflexivity $\text{isRefl}$ and symmetry $\text{isSym}$. It should be obvious how to incorporate more operators and predicates on relations. By additionally providing a sort for sets, operations like dom, ran, restriction, etc. could be considered also.
| Universe cardinality | 3   | 4 | 5   | 6   |
|----------------------|-----|---|-----|-----|
| Relation count       | 512 | 65 536 | 33 554 432 | 68 719 476 736 |
| Compile time         | 14.730 | 6.502 | 6.152 | 8.967 |
| Run time tr⇒qtr      | 0.005 | 0.022 | 3.171 | 5717.611 |
| Run time QMc         | 2.485 | 2.335 | 19.732 | (not tried) |

Figure 1. Timing vs. universe cardinality

```
x\y  0  1  2
   0  1  2  4
   1  8 16 32
   2 64 128 256
```

Figure 2. Encoding scheme for $3 \times 3$ relations for a [Bur02] approach

```
R_0 ::= symCls(R_0) | R_0 \cup R_0
\vdots
R_{10} ::= symCls(R_2) | symCls(R_8) | symCls(R_{10})
| R_0 \cup R_{10} | R_2 \cup R_8 | R_2 \cup R_{10} | R_8 \cup R_2 | R_8 \cup R_{10}
| R_{10} \cup R_0 | R_{10} \cup R_2 | R_{10} \cup R_8 | R_{10} \cup R_{10}
\vdots
R_{511} ::= symCls(R_{311}) | \ldots | symCls(R_{511})
| R_0 \cup R_{511} | \ldots | R_{511} \cup R_{511}
true ::= isRefl(R_{273}) | \ldots | isRefl(R_{511})
| isSym(R_0) | \ldots | isSym(R_{511})
| false \lor true | true \lor false | true \lor true | true \land true | \neg false
false ::= isRefl(R_0) | \ldots | isSym(R_2) | \ldots | \neg true
```

Figure 3. Tree grammar sketch for [Bur02] approach
2 Formal proofs of property laws

**Definition 1 (Binary relation properties)** Let $X$ be a set. A (homogeneous) binary relation $R$ on $X$ is a subset of $X \times X$. The relation $R$ is called

1. reflexive if $\forall x \in X. \ xRx$;
2. irreflexive if $\forall x \in X. \ \neg xRx$;
3. co-reflexive if $\forall x, y \in X. \ xRy \rightarrow x = y$;
4. symmetric if $\forall x, y \in X. \ xRy \rightarrow yRx$;
5. asymmetric if $\forall x, y \in X. \ xRy \rightarrow \neg yRx$;
6. anti-symmetric if $\forall x, y \in X. \ xRy \land x \neq y \rightarrow \neg yRx$;
7. semi-connex if $\forall x, y \in X. \ xRy \lor yRx \lor x = y$;
8. connex if $\forall x, y \in X. \ xRy \lor yRx$;
9. transitive if $\forall x, y, z \in X. \ xRy \land yRz \rightarrow xRz$;
10. anti-transitive if $\forall x, y, z \in X. \ xRy \land yRz \rightarrow \neg xRz$;
11. quasi-transitive if $\forall x, y, z \in X. \ xRy \land \neg yRx \land yRz \land \neg zRy \rightarrow xRz \land \neg zRx$;
12. right Euclidean if $\forall x, y, z \in X. \ xRy \land xRz \Rightarrow yRz$;
13. left Euclidean if $\forall x, y, z \in X. \ yRx \land zRx \Rightarrow yRz$;
14. semi-order property 1 if $\forall w, x, y, z \in X. \ wRx \land \neg xRy \land \neg yRx \land yRz \Rightarrow wRz$;
15. semi-order property 2 if $\forall w, x, y, z \in X. \ xRy \land yRz \Rightarrow wRx \lor xRw \lor wRy \lor yRw \lor wRz \lor zRw$.

**Definition 2 (Kinds of binary relation)** A binary relation $R$ on a set $X$ is called

1. an equivalence if it is reflexive, symmetric, and transitive;
2. a non-strict partial order if it is reflexive, anti-symmetric, and transitive;
3. a strict partial order if it is irreflexive, asymmetric, and transitive;
4. a non-strict partial order if it is reflexive, anti-symmetric, and transitive;
5. a strict partial order if it is irreflexive, asymmetric, and transitive;
6. a semi-order if it is asymmetric and satisfies semi-order properties 1 and 2.

**Definition 3 (Operations on relations)**

1. For a relation $R$ on a set $X$ and a subset $Y \subseteq X$, we write $R|_Y$ for the restriction of $R$ to $Y$. Formally, $R|_Y$ is the relation on $Y \times Y$ defined by $x(R|_Y)y :\Leftrightarrow xRy$ for each $x, y \in Y$.
2. For an equivalence relation $R$ on a set $X$, we write $[x]_R$ for the equivalence class of $x \in X$ w.r.t. $R$. Formally, $[x]_R := \{y \in X | xRy\}$.
3. For a relation $R$ on a set $X$ and $x, y \in X$, we write $xR$ for the set of
elements \( x \) is related to, and \( R_y \) for the set of elements that are related to \( y \). Formally, \( xR := \{ y \in X \mid xRy \} \) and \( R_y := \{ x \in X \mid xRy \} \).

 Lemma 4 (Co-reflexive relations)

(1) The union of a co-reflexive relation and a transitive relation is always transitive.

**PROOF.**

(1) Let \( C \) be co-reflexive and \( T \) be transitive. Let \( R = C \cup T \). Assume \( xRy \land yRz \). We distinguish four cases:

- If \( xTy \land yTz \), then \( xTz \) by transitivity of \( T \), and hence \( xRz \).
- If \( xTy \land yCz \), then \( y = z \) by co-reflexivity of \( C \), hence \( xTz \) by substitutivity, hence \( xRz \).
- Similarly, \( xCy \land yTz \Rightarrow x = yTz \Rightarrow xRz \).
- If \( xCy \land yCz \), then \( x = yCz \) implies \( xRz \). \( \square \)

 Lemma 5 (Quasi-transitive relations)

(1) \( R \) is a quasi-transitive relation iff \( R = I \cup P \) for some symmetric relation \( I \) and some transitive relation \( P \).

(2) \( I \) and \( P \) are not uniquely determined by a given \( R \).

(3) The definitions \( xIy :\Leftrightarrow xRy \land yRx \) and \( xPy :\Leftrightarrow xRy \land \neg yRx \) lead to the minimal \( P \).

(4) Each symmetric relation is quasi-transitive; each transitive relation is quasi-transitive.

(5) An anti-symmetric and quasi-transitive relation is always transitive.

**PROOF.**

(1) “\( \Rightarrow \)”: Let \( R \) be quasi-transitive. Following [Sen71, ?], Define \( xIy :\Leftrightarrow xRy \land yRx \) and \( xPy :\Leftrightarrow xRy \land \neg yRx \). Then

- \( I \) and \( P \) are disjoint:
  \( xIy \land xPy \)
  \( \Rightarrow yRx \land \neg yRx \) using the definitions of \( I \) and \( P \)
  \( \Rightarrow false \)
- Their union is \( R \):
  \( xIy \lor xPy \)
  \( \Leftrightarrow (xRy \land yRx) \lor (xRy \land \neg yRx) \) by definition of \( I \) and \( P \)
  \( \Leftrightarrow xRy \land (yRx \lor \neg yRx) \) by distributivity
  \( \Leftrightarrow xRy \)
• \( I \) is symmetric:
  \[
  xIy \\
  \Rightarrow xRy \land yRx \\
  \Rightarrow yIx
  \]

• \( P \) is transitive:
  \[
  xPy \land yPz \\
  \Rightarrow xRy \land \neg yRx \land yRz \land \neg zRy \quad \text{by definition of } P \\
  \Rightarrow xRz \land \neg zRx \quad \text{by quasi-transitivity of } R \\
  \Rightarrow xPz \quad \text{by definition of } P
  \]

“\( \Leftarrow \)”: Let \( R = I \cup P \) for some symmetric relation \( I \) and some transitive relation \( R \). Assume \( xRy \) and \( yRz \) hold, but neither \( yRx \) nor \( zRy \) does. We observe the following facts:
(a) \( xIy \) is false, since else \( xIy \Rightarrow yIx \Rightarrow yRx \), contradicting our assumptions.
(b) \( xPy \) holds, since \( xRy \Rightarrow xIy \lor xPy \Rightarrow xPy \) by 1a.
(c) \( yPz \) follows by an argument similar to 1a and 1b.
(d) Hence \( xPz \) holds, by transitivity of \( P \).
(e) Hence \( xRz \).
(f) Since \( I \) and \( P \) are disjoint, we obtain \( \neg xIz \) from 1d; hence \( \neg zIx \) by symmetry of \( I \).
(g) Finally, we have \( \neg zRx \), since else \( zPx \) by 1f, which in turn would imply \( zPy \) by 1b and the transitivity of \( P \), which would imply \( zRy \), contradicting our assumptions.

From 1e and 1g, we conclude the quasi-transitivity of \( R \).
(2) For example, if \( R \) is an equivalence relation, \( I \) may be chosen as the empty relation, or as \( R \) itself, and \( P \) as its complement.
(3) Given \( R \), whenever \( xRy \land \neg yRx \) holds, the pair \( \langle x, y \rangle \) can’t belong to the symmetric part, but must belong to the transitive part.
(4) Follows from 1 and the transitivity and symmetry of the empty relation.
(5) Let \( R \) be anti-symmetric and quasi-transitive. We use the definitions of \( I \) and \( P \) from 3. We have \( xIy \Rightarrow xRy \land yRx \Rightarrow x = y \) by anti-symmetry, hence \( I \) is co-reflexive. By Lemma 4.1, \( R = I \cup P \) is transitive.  \( \square \)

For sake of completeness, we repeat here the well-known laws about asymmetry.

**Lemma 6** (Asymmetric relations)

(1) A relation is asymmetric iff it is anti-symmetric and irreflexive.
(2) A transitive relation is asymmetric iff it is irreflexive.

**PROOF.**

(1) “\( \Rightarrow \)”: Let \( R \) be asymmetric. Then \( xRx \) implies the contradicting \( \neg xRx \);

---

7
hence $R$ is irreflexive. Moreover, $xRy \vee x \neq y$ implies $\neg yRx$, since its left disjunct does; hence $R$ is anti-symmetric.

---

(2) $\Rightarrow$: Let $R$ be transitive and asymmetric. By 1, irreflexivity follows from asymmetry alone.

---

**Lemma 7** (Anti-transitive relations)

(1) An anti-transitive relation is always irreflexive.

(2) For an anti-transitive relation, $xRy \land xRz \Rightarrow \neg yRz$ and $yRz \land xRz \Rightarrow \neg xRy$ holds for all $x, y, z$.

**PROOF.**

(1) Assume $xRx$ holds. Then $xRx \land xRx$ implies $\neg xRx$ by anti-transitivity, which is a contradiction.

(2) Both implications follow by contraposition from the anti-transitivity implication. $\Box$

**Lemma 8** (Connex and semi-connex relations)

(1) A relation is connex iff it is semi-connex and reflexive.

(2) A semi-connex relation on a set $X$ cannot be anti-transitive, provided $X$ has at least 4 elements.

(3) If $R$ is a semi-connex relation on $X$, then the set $X \setminus \operatorname{ran}(R)$ has at most one element; the same applies to $X \setminus \operatorname{dom}(R)$.

**PROOF.**

(1) $\Rightarrow$: Both properties follow trivially.

$\Leftarrow$: For $x \neq y$, the semi-connex property implies $xRy \vee yRx$. For $x = y$, reflexivity implies $xRy$.

(2) Assume $R$ is both semi-connex and anti-transitive. Consider the directed graph corresponding to $R$, with its vertices being the elements of $X$, and its edges being the pairs related by $R$.

Consider three arbitrary distinct vertices. By the semi-connex property, each pair of them must be connected by an edge. By anti-transitivity, none of them may be the source of more than one edge. Hence, the three edges must be oriented in such a way that they for a directed cycle.

Let $w, x, y, z$ be four distinct elements. W.l.o.g. assume the subgraph for $x, y, z$ is oriented a directed cycle corresponding to $xRy \land yRz \land zRx$. The
subgraph for \( w, x, y \) must be oriented as a directed cycle, too; therefore \( wRx \land xRy \land yRw \) must hold. But then, the subgraph for \( w, x, z \) is not oriented as a cycle, since \( wRx \land zRx \). This contradicts the cycle-property shown above.

(3) Let \( x, y \in X \setminus \text{ran}(R) \). Since \( R \) is semi-connex, \( xRy \) or \( yRx \) or \( x = y \) must hold. The first two possibilities are ruled out by assumption, so the third one must apply, i.e. \( x \) and \( y \) can’t be distinct. A similar argument applies to \( \text{dom}(R) \). \( \square \)

**Lemma 9 (Euclidean relations)**

(1) For symmetric relations, transitivity, right Euclideanness, and left Euclideanness all coincide.

(2) A relation which is both right Euclidean and reflexive is an equivalence relation. Similarly, each left Euclidean and reflexive relation is an equivalence.

(3) The range of a right Euclidean relation is always a subset of its domain. The restriction of a right Euclidean relation to its range is always an equivalence. Similarly, the domain of a left Euclidean relation is a subset of its range, and the restriction of a left Euclidean relation to its domain is an equivalence.

(4) A relation \( R \) is both left and right Euclidean, iff the domain and the range set of \( R \) agree, and \( R \) is an equivalence relation on that set.

(5) A right Euclidean relation is always quasi-transitive, and so is a left Euclidean relation.

(6) A semi-connex right Euclidean relation is always transitive, and so is a semi-connex left Euclidean relation.

(7) If \( X \) has at least 3 elements, a semi-connex right Euclidean relation on \( X \) is never anti-symmetric, and neither is a semi-connex left Euclidean relation on \( X \).

(8) A relation \( R \) on a set \( X \) is right Euclidean iff \( R' := R|_{\text{ran}(R)} \) is an equivalence and \( \forall x \in X \setminus \text{ran}(R) \exists y \in \text{ran}(R). xR \subseteq [y]_{R'} \), cf. Fig. 4. Similarly, \( R \) on \( X \) is left Euclidean iff \( R' := R|_{\text{dom}(R)} \) is an equivalence and \( \forall y \in X \setminus \text{dom}(R) \exists x \in \text{dom}(R). yR \subseteq [x]_{R'} \), cf. Fig. 5.

**PROOF.**

(1) Trivial.

(2) \( R \) is symmetric, since \( xRy \) and \( xRx \) implies \( yRx \). Hence, by 1, \( R \) is also transitive.

(3) If \( y \) is in the range of \( R \), then \( xRy \land xRx \) implies \( yRx \), for some suitable \( x \). This also proves that \( y \) is in the domain of \( R \). By 2, \( R \) is therefore an equivalence.
(4) “⇒”: follows by 3.
“⇐”: Assume $aRb$ and $aRc$, then $a, b, c$ are members of the domain and range of $R$, hence $bRc$ by symmetry and transitivity. Left Euclideanness of $R$ follows similarly.

(5) Let $R$ be right Euclidean. Let $xRy \land \neg yRx \land yRz \land \neg zRy$ hold. Observe that $y, z \in \text{ran}(R)$. By 3, $R$ is symmetric on $\text{ran}(R)$, hence $yRz$, which is a contradiction. Hence, $R$ is vacuously quasi-transitive, since the assumptions about $x, y, z$ can never be met.

A similar argument applies to left Euclidean relations, exploiting that $x, y \in \text{dom}(R)$.

(6) Let $R$ be semi-connex and right Euclidean. Let $xRy \land yRz$ hold. Observe again that $y, z \in \text{ran}(R)$. Since $R$ is semi-connex, the following case distinction is exhaustive:

- $xRz$ holds.
  Then we are done immediately.
- $zRx$ holds.
  Then also $x \in \text{ran}(R)$; hence $xRz$, since $R$ is symmetric on its range by 3.
- $x = z$.
  Then also $x \in \text{ran}(R)$; hence $xRz$, since $R$ is reflexive on its range by 3.

Again, a similar argument applies to semi-connex and Euclidean relations, using $x, y \in \text{dom}(R)$.

(7) Let $R$ be semi-connex and right Euclidean. By Lemma 8.3, at most one element of $X$ is not in the range of $R$. Hence, by assumption, two distinct elements $x, y \in \text{ran}(R)$ exist. Since $R$ is semi-connex and $x \neq y$, we have $xRy$ or $yRx$. By Lemma 9.3, we obtain both $xRy$ and $yRx$. This contradicts the anti-symmetry requirement.

(8) “⇒”: By 3, $R|_{\text{ran}(R)}$ is an equivalence. Let $x \in X \setminus \text{ran}(R)$. If $xRy_1$ and $xRy_2$, then $y_1, y_2 \in \text{ran}(R)$, and $y_1 R y_2$ by right Euclideanness of $R$, that is, $y_1, y_2$ belong to the same equivalence class w.r.t. $R'$.

“⇐”: Let $x, y, z \in X$ such that $xRy \land xRz$, we show $yRz$. Observe $y, z \in \text{ran}(R)$. We distinguish two cases:

- If $x \in \text{ran}(R)$,
  then $xR' y \land xR' z$, hence $yR' z$ by symmetry and transitivity of $R'$, hence $yRz$.
- If $x \not\in \text{ran}(R)$,
  then let $w \in \text{ran}(R)$ with $xR \subseteq [w]_{R'}$. We have $y, z \in [w]_{R'}$ by assumption, hence $yR' z$, hence $yRz$. □

Based on Lem. 9.8, Fig. 4 shows a schematized Right Euclidean relation. Deeply-colored squares indicate equivalence classes of $R|_{\text{ran}(R)}$, assuming $X$’s elements are arranged in such a way that equivalent ones are adjacent. Pale-colored rectangles indicate possible relationships of elements in $X \setminus \text{ran}(R)$,
again assuming them to be arranged in convenient order. In these rectangles, relationships may, or may not, hold. A light grey color indicates that the element corresponding to the line is unrelated to that corresponding to the column; in particular, the lighter grey right rectangle indicates that no element at all can be related to some in the set $\text{rest} := X \setminus \text{ran}(R)$. The diagonal line indicates that $xRx$ holds iff $x \in \text{ran}(R)$.

Figure 5 shows a similar schema for a Left Euclidean relation,
Figure 4. Right Euclidean relation

Figure 5. Left Euclidean relation
3 Computed relation properties

In this section, we comment on some program details, and show the annotated output of a program run on a 5-element set.

The source code is available as an ancillary file at [arxiv.org](http://arxiv.org). Its procedure `computeLaws` iterates over all possible relations, determining for each the set (bit vector) of its properties, and counting the number of occurrences of each such vector. After that, it calls the Quine-McCluskey implementation `qmc` to compute all prime implicants of the non-occurring vectors. The latter procedure performs a top-down breadth-first search on the search graph.

An example graph, showing all possible prime implicants for a Boolean function of 3 variables is given in Fig. 6. At each node of the search graph, the corresponding conjunction is checked by the procedure `qmcRect`: if no combination in its covered rectangle is “off” and at least one is “on” then it is actually a prime implicant. In that case, we output its description using `qmcPrint`, and set all vectors in its covered rectangle to "don’t care".

Note that we can’t perform a depth-first search: for example, if $a$ isn’t a prime implicant, we can’t check its child $ab$ next, since it could satisfy the above primness criterion, but nevertheless be covered by a simpler prime implicant, such as $b$.

In the Fig. 7 to 10, we list the computed prime implicants for missing relation property combinations on a 5-element universe set. Suggestions are grouped by the number of conjuncts; tables have been reordered to utilize page space to full advantage.

In the leftmost column, the binary representation is given, “0”, “1”, and “-” indicating that the column’s property doesn’t hold, does hold, and is irrelevant, respectively. In the middle column, the same information is given in textual representation, “¬” denoting negation, and juxtaposition used for conjunction. In the rightmost column, we annotated a reference to the lemma where the law has been formally proven, or “c” if the law is trivial due to the involvement of co-reflexivity, “s” if a semi-order property is involved, “t” if the law is trivial, or “w” if the law can be obtained as a weakening or a

---

5 This terminology is inspired by the Karnaugh diagram method; in Fig. 6, the rectangle covered by a node corresponds to the set of all leaves below it.
6 Since we are interested in non-occurring vectors, “on” corresponds to an occurrence count of zero, and “off” to a count $> 0$.
7 Columns are ordered as in Def. 1, starting from the rightmost, i.e. least significant, bit.
8 We didn’t yet attempt to prove such laws.
Figure 6. Search graph for the Quine-McCluskey algorithm on 3 variables

For example, the 4th line in level 2 reports that no relation was non-irreflexive (negation of property 1.2) and asymmetric (property 1.5); we formally proved that every irreflexive relation is asymmetric in Lem. 6.1.

No laws were reported for level 1 and 6–16.

References

[Bur02] Jochen Burghardt. Axiomatization of finite algebras. In Proc. KI 2002, number 2479 in LNAI, pages 222–234. Springer, 2002.

[McC56a] E. McCluskey. Minimization of boolean functions. Bell System Technical Journal, 35:437–457, 1956.

[McC56b] E. J. McCluskey. Minimization of boolean functions. Bell System Technical Journal, 35(5):1417–1444, 1956.

[Sen71] Amartya K. Sen. Choice functions and revealed preference. The Review of Economic Studies, 38(3):307–317, Jul 1971.
Figure 7. Reported laws for level 3
| --- | --- | --- | --- | --- | --- |
| 11 | 01 | 11 | 01 | 11 | 01 |
| Refl | Irrefl | CoRef1 | ~Sym | Refl | ASym |
| t | c | t | 6.1 |
| --- | --- | --- | --- | --- | --- |
| 01 | 01 | 11 | 11 | 01 | 01 |
| CoRef1 | ~AntiSym | ASym | ~AntiSym | CoRef1 | SemiConnex |
| c | c | c | 6.1 |
| --- | --- | --- | --- | --- | --- |
| 11 | 11 | 11 | 11 | 11 | 11 |
| CoRef1 | Connex | SemiConnex | Connex | CoRef1 | Connex |
| c | c | c | c |
| 01 | 01 | 01 | 01 | 01 | 01 |
| CoRef1 | Connex | SemiConnex | Connex | CoRef1 | Connex |
| c | c | c | c |
| 01 | 01 | 01 | 01 | 01 | 01 |
| CoRef1 | Connex | SemiConnex | Connex | CoRef1 | Connex |
| c | c | c | c |

Figure 8. Reported laws for level 2

| --- | --- | --- | --- | --- | --- |
| 10 | 10 | 10 | 10 | 10 | 10 |
| ~Refl | AntiTrans | AntiTrans | AntiTrans | AntiTrans | AntiTrans |
| c | c | c | c |
| 01 | 01 | 01 | 01 | 01 | 01 |
| CoRef1 | ~QuasiTrans | ~QuasiTrans | ~QuasiTrans | ~QuasiTrans | ~QuasiTrans |
| c | c |
| 01 | 01 | 01 | 01 | 01 | 01 |
| CoRef1 | ~QuasiTrans | ~QuasiTrans | ~QuasiTrans | ~QuasiTrans | ~QuasiTrans |
| c | c |
| 01 | 01 | 01 | 01 | 01 | 01 |
| Connex | ~SemiOrd1 | ~SemiOrd1 | ~SemiOrd1 | ~SemiOrd1 | ~SemiOrd1 |
| s |

Figure 9. Reported laws for level 4

| 10 | 10 | 10 | 10 | 10 | 10 |
| ~Refl | ~Irrefl | Sym | SemiConnex | Trans | Trans |
| t | s |

Figure 10. Reported laws for level 5