Canonical Mixed-Polarity Multi-Target Toffoli Circuits: 
Shift and Removal

Hans-Jörg Kreowski and Aaron Lye

University of Bremen, Department of Computer Science
P.O.Box 33 04 40, 28334 Bremen, Germany
{kreo,lye}@informatik.uni-bremen.de

Abstract
In this paper, we study reversible circuits as cascades of multi-target Toffoli gates. This type of gates allows to shift parts of a gate to the preceding gate within a circuit provided that a certain independence condition holds. It turns out that shifts decrease the so-called waiting degree such that shifting as long as possible always terminates and yields shift-reduced circuits. As the main result, we show that shift-reduced circuits are unique canonical representatives of their shift equivalence classes. Canonical circuits are optimal with respect to maximal and as-early-as-possible parallelism of targets within gates. Further, we discuss how successive equal subgates can be removed in order to reduce the waiting degree even more. Interestingly after applying shifts and removals as long as possible again a unique normal form is obtained.

Keywords: canonical form, multi-target Toffoli circuits, reversible computation, shift equivalence, shift-and-remove equivalence

1. Introduction

Reversible computation is an alternative to conventional computing motivated by the fact that the integration density of circuits reaches physical limits in scale and power dissipation. Due to the fact that energy dissipation is significantly reduced or even eliminated in reversible circuits [1], reversible computing is a very promising research area. The fundamental reason for power dissipation arises from [2]. Landauer proved that using irreversible logic gates always lead to energy dissipation regardless of the underlying technology such that exactly $k \cdot T \cdot \log(2)$ Joule of energy is dissipated for each irreversible bit, where $k$ is the Boltzmann constant and $T$ is the temperature.

Reversible circuits are cascades of reversible gates that compute invertible functions on Boolean vectors. This bijective mapping is equivalently described by a permutation matrix mapping input to output vectors. The permutation matrix of the circuit is given by the product of all permutation matrices of the respective gates. Permutation matrices are unitary. This is interesting as unitary matrices are required for expressing quantum-mechanical evolution [3].
To specify reversible circuits, the gate model introduced by Toffoli [4] is frequently used. In the past this model has been generalized in different ways. In this paper (an extended version of [5] which can be found in the proceedings of LATA 2016) we want to study the model of Toffoli circuits with multi-target Toffoli gates.

A (single-target multi-controlled) Toffoli gate consists of a target line and a set of control lines each of which is different from the target line. The lines represent Boolean variables. The target line gets negated if and only if all control lines are carrying the value 1. All other values are kept invariant by the evaluation of the gate. In particular, a Toffoli gate is reversed by itself. Consider now a set of Toffoli gates such that the target lines are pairwise different and all control lines are disjoint from all target lines. Such gates may be called independent because their evaluation in every sequential order yields the same Boolean function. Moreover, their evaluation can be done in parallel because the various negations cannot interfere with each other. This motivates us to consider such sets of independent Toffoli gates as a multi-target Toffoli gate.

As the parallel as well as each sequential evaluation of independent gates yield the same result, a multi-target Toffoli gate can be sequentialized with respect to every partition of the set of target lines. Conversely, two gates can be parallelized into one gate if their sets of target lines are disjoint and no target line is control line of the other gate. There is a weaker form of combining two multi-target Toffoli gates that can be applied much more frequently than the full parallelization: One of the gates is sequentialized first and then only one of the parts is parallelized with the other gate if possible. In this case, a part of a gate is shifted to another gate. The shifts from right to left (together with the parallelization) define a relation on multi-target Toffoli circuits with quite significant properties. First of all, the shift relation has the local Church-Rosser property meaning that the circuits resulting from two shifts on a given circuit can be further shifted into a common result. Secondly, shifts decrease the so-called waiting degree. For each target line of some gate, there is a number of preceding gates. If evaluation is done gate by gate, this is the number of steps a negation must wait before it is executed. The waiting degree sums up all these numbers. As the waiting degree decreases with each shift, the lengths of shift sequences are bounded by the maximum waiting degree (which is $\frac{m(m-1)}{2}$ for the number $m$ of target lines of a circuit). In particular, the iteration of shifts as long as possible terminates always with a circuit reduced with respect to shifting. Combining both results, the shift-reduced circuits turn out to be unique normal forms within the classes of shift-equivalent circuits. Therefore, it is justified to call shift-reduced multi-target Toffoli gates canonical. Canonical circuits are optimal with respect to maximal and as-early-as-possible parallelism of targets within gates.

Shift-reduced canonical forms may be the starting point of further optimizations. For example, the sequential composition of a Toffoli gate with itself computes the identity. Therefore, two identical parts in successive multi-target Toffoli gates can be removed without changing the semantics of a circuit. After-
wards, another round of shift optimization can be started. Interestingly, it turns out that the combination of shifts and removals leads to similar normal-form results than shifts alone.

Shifts, shift equivalence and shift-reduced normal forms as unique canonical representatives of their shift equivalence classes were studied by the first author quite some time ago for parallel derivations in graph grammars (see [6, 7, 8]). Although multi-target Toffoli circuits as considered in this paper provide a setting quite different from parallel graph grammar derivations, the same ideas work.

The paper is organized as follows. Section 2 introduces the characteristics of reversible functions and circuits. In Section 3, multi-target Toffoli circuits are defined, followed by considering sequentialization, parallelization and shift in Section 4. Section 5 introduces the waiting degree. Section 6 covers our theorem on canonical circuits. In Section 7, we investigate the combination of shift with other transformations. In particular, we analyze properties when removing equal successive subgates. Finally, Section 8 contains a conclusion.

In the shorter version of this paper [5] multi-target Toffoli circuits are considered without mixed control and only the normal-form results for the shift equivalence are discussed and proved. The combination of shifts with removals is new.

2. Reversible Circuits

In this section we introduce the background on reversible functions and their relation to reversible circuits.

2.1. Reversible Functions

Reversible logic can be used for realizing reversible functions. Reversible functions are special multi-output functions and defined as follows.

**Definition 1.** Let \( \mathbb{B} = \{0, 1\} \) be the set of truth values with the negation \( \overline{0} = 1 \) and \( \overline{1} = 0 \) and \( ID \) be a set of identifiers serving as a reservoir of Boolean variables. Let \( \mathbb{B}^X \) be the set of all mappings \( a: X \rightarrow \mathbb{B} \) for some \( X \subseteq ID \) where the elements of \( \mathbb{B}^X \) are called assignments. If the set of variables is ordered, each assignment corresponds to a Boolean vector. A bijective Boolean (multi-output) function \( f: \mathbb{B}^X \rightarrow \mathbb{B}^X \) is called reversible.

2.2. Reversible Circuits

Reversible circuits are used for representing reversible functions because a reversible function can be realized by a cascade of reversible gates. Reversible circuits differ from conventional circuits: While conventional circuits rely on the basic binary operations and also fanouts are applied in order to use signal values on several gate inputs, in reversible logic fanouts and feedback are not directly allowed because they would destroy the reversibility of the computation. Also the logic operators AND and OR cannot be used since they are irreversible.
Instead a reversible gate library is applied. Since the Boolean operator NOT is inverse to itself, the NOT-gate is part of this reversible library. To increase the expressiveness the universal Toffoli gate has been introduced, which is a multi-controlled NOT-gate. Since the Toffoli gate is universal (cf. [4]), all reversible functions can be realized by cascades of this gate type alone.

A (multiple-control) Toffoli gate consists of a target line \( t \in ID \) and a set \( C \subseteq ID - \{ t \} \) of control lines and is denoted by \((t, C)\). The gate defines the function \( f(t, C) : B^X \rightarrow B^X \) for each \( X \subseteq ID \) with \( \{ t \} \cup C \subseteq X \) which maps an assignment \( a : X \rightarrow B \) to \( f(t, C)(a) : X \rightarrow B \) given by \( f(t, C)(a)(t) = \overline{a(t)} \) if \( a(c) = 1 \) for all \( c \in C \). In all other cases, \( f(t, C)(a) \) is equal to \( a \). Hence, \( f(t, C) \) inverts the value of the target line if and only if all control lines are set to 1. The values of all other lines always pass through a gate unchanged. Consequently, \( f(t, C) \) is a mapping on \( B^X \) which is inverse to itself and, therefore, reversible in particular. A multiple-control Toffoli gate can be realized by a sequence of Toffoli gates with two control lines.

Example 1. The four simplest multi-controlled Toffoli gates are \( \text{NOT} = (x_1, \emptyset) \), \( \text{CNOT} = (x_1, \{x_2\}) \), \( \text{CCNOT} = (x_1, \{x_2, x_3\}) \) and \( \text{C}^3\text{NOT} = (x_1, \{x_2, x_3, x_4\}) \).

In the graphical representation, the target line is indicated by \( \oplus \) and the control lines by \( \bullet \) vertically connected with their target line (see Fig 1(a)).

In addition to positive control lines, also negative- and mixed-control Toffoli gates have been considered [4]. This gains smaller circuits in general. Nevertheless, the expressiveness remains the same, since each negative control can be replaced by a positive one with a negation before and after the control.

In [9] an \( m\text{EXOR} \) gate has been introduced with multiple control lines and multiple (\( m \)) target lines, denoted by the sets \( C \) and \( T \), respectively, holding \( C \cup T \subseteq X \), \( T \neq \emptyset \) and \( C \cap T = \emptyset \). It realizes the function \( f(a)(x) = \overline{a(x)} \) if \( x \in T \) and \( a(y) = 1 \) for all \( y \in C \), and \( a(x) \) otherwise. This means that the values of all target lines are negated if the value of each control line is 1 and all target lines depend on the same control lines. Sometimes these gates are also called \( \text{extended-target Toffoli gates} \), see e.g. [10]. We discuss a further generalization in the following section.

3. Multi-Target Toffoli Circuits

In this section, we consider the notion of multi-target Toffoli circuits as cascades of multi-target Toffoli gates with mixed control. Such a gate has a set of target lines where each target line is controlled by a set of positive control lines and a set of negative control lines which are disjoint from the set of target lines and from each other.

Definition 2. 1. A (mixed-polarity) multi-target Toffoli gate over a set \( X \) of lines is a triple \((T, cp : T \rightarrow 2^X, cn : T \rightarrow 2^X)\) with \( T \subseteq X \), \( T \neq \emptyset \), \( T \cap \bigcup_{t \in T} (cp(t) \cup cn(t)) = \emptyset \) and \( cp(t) \cap cn(t) = \emptyset \) for all \( t \in T \). \( T \) is the
set of target lines, \( cp(t) \) and \( cn(t) \) are respectively the set of positive and negative control lines of \( t \) for \( t \in T \).

2. A multi-target Toffoli gate \( mtg = (T, cp, cn) \) models the following semantic function \( f_{mtg} \) on \( \mathbb{B}^X \):

\[
f_{mtg}(a)(x) = \begin{cases} 
  a(x) & \text{if } x \in T \text{ and } a(y) = 1 \text{ for all } y \in cp(x) \\
  a(x) & \text{and } a(y) = 0 \text{ for all } y \in cn(x), \\
  \text{otherwise} & \text{otherwise.}
\end{cases}
\]

3. A multi-target Toffoli circuit \( mtc = mtg_1 \cdots mtg_n \) is a sequence of multi-target Toffoli gates. Its length \( n \) is denoted by \( |mtc| \).

4. Let \( mtc \) be a multi-target Toffoli circuit. It models the semantic function \( f_{mtc} \) defined as the sequential composition of the semantic functions of the gates, i.e.

\[
f_{mtc} = f_{mtg_n} \circ \cdots \circ f_{mtg_1}.
\]

Example 2. Consider the multi-target Toffoli circuit \( mtg_1 \circ mtg_2 \) over 4 lines \( x_1 \) to \( x_4 \) where \( mtg_1 = (T_1, cp_1, np_1) \), for \( i = 1, 2 \) are given by \( T_1 = \{x_2, x_3\}, T_2 = \{x_1, x_3\}, cp_1(x_2) = \{x_1, x_4\}, cn_1(x_2) = cp_1(x_3) = cn_1(x_3) = \emptyset, cp_2(x_1) = \{x_4\}, cp_2(x_3) = \{x_2, x_4\}, cn_2(x_1) = \{x_2\}, cn_2(x_3) = \emptyset \). The circuit is depicted in Fig. 1(b). Negative control lines are depicted by \( \circ \). The semantic functions of \( mtg_1 \) and \( mtg_2 \) are as follows: \( f_{mtg_1}(a)(x_2) = a(x_2) \) if \( a(x_1) = a(x_4) = 1 \), \( f_{mtg_1}(a)(x_3) = a(x_3) \), \( f_{mtg_1}(a)(x) = a(x) \) otherwise, and \( f_{mtg_2}(a)(x_3) = a(x_3) \) if \( a(x_2) = a(x_4) = 1 \), \( f_{mtg_2}(a)(x_1) = a(x_1) \) if \( a(x_2) = 0 \) and \( a(x_4) = 1 \), \( f_{mtg_2}(a)(x) = a(x) \) otherwise.

We do not introduce the graphical representation of multi-target Toffoli circuits formally as they are only used to illustrate sample circuits, which are always formally defined by their set-theoretical representation. But while a target line of a gate can be connected with its control lines by a vertical line without unambiguity if there is only a single target (cf. Fig. 1(a)), it should be noted that one may need to bend lines in the case of multiple targets (cf. Fig. 1(b)).

If a multi-target Toffoli gate \( mtg \) has the set \( T \) of target lines and \( T' \) is a subset of \( T \), then \( mtg \) can be restricted to \( T' \) and its complement \( T'' = T - T' \).
yielding the multi-target Toffoli gates $mtg'$ and $mtg''$. It is easy to prove that the sequential compositions $mtg'\cdot mtg''$ and $mtg''\cdot mtg'$ are semantically equivalent to $mtg$.

**Proposition 1.** Let $mtg = (T, cp, cn)$ be a multi-target Toffoli gate, let $T' \subseteq T$ with $\emptyset \neq T' \neq T$.

1. Then $mtg' = (T', cp', cn')$ with $cp'(t') = cp(t')$ and $cn'(t') = cn(t')$ for $t' \in T'$ is a multi-target Toffoli gate. This gate may be denoted by $mtg|_{T'}$, called the restriction of $mtg$ to $T'$.

2. Accordingly, $mtg'' = (T'', cp'', cn'')$ with $T'' = T - T'$ and $cp''(t'') = cp(t'')$ as well as $cn''(t'') = cn(t'')$ for $t'' \in T''$ is also a multi-target Toffoli gate.

3. The sequential compositions $mtg' \cdot mtg''$ and $mtg'' \cdot mtg'$ are semantically equivalent to $mtg$, i.e. $f_{mtg} = f_{mtg' \cdot mtg''} = f_{mtg'' \cdot mtg'}$.

**Proof**

1. $T' \cap \bigcup_{t' \in T'} (cp'(t') \cup cn'(t')) = T' \cap \bigcup_{t' \in T'} (cp(t') \cup cn(t')) \subseteq T \cap \bigcup_{t' \in T'} (cp(t') \cup cn(t')) = \emptyset$ and $cp'(t') \cap cn'(t') = cp(t') \cap cn(t') = \emptyset$.

2. $T' \subseteq T$ and $\emptyset \neq T' \neq T$ imply $T - T' \subseteq T$ and $\emptyset \neq T - T' \neq T$ such that Point 1 applies to $T'' = T - T'$.

3. By definition, we get the following equations for all $a \in \mathbb{B}^X$ and $x \in X$:

$$f_{mtg' \cdot mtg''}(a)(x) = (f_{mtg''} \circ f_{mtg'})(a)(x) = f_{mtg''}(f_{mtg'}(a))(x)$$

$$= \begin{cases} f_{mtg'}(a)(x) & \text{if } x \in T'' \text{ and } f_{mtg'}(a)(y) = 1 \text{ for all } y \in cp''(x) \\ f_{mtg''}(a)(x) & \text{otherwise,} \end{cases}$$

as well as

$$f_{mtg''}(a)(x) = \begin{cases} a(x) & \text{if } x \in T' \text{ and } a(y) = 1 \text{ for all } y \in cp'(x) \text{ and } a(y) = 0 \text{ for all } y \in cn'(x), \\ a(x) & \text{otherwise.} \end{cases}$$

Combining these results and using $T' \cap T'' = \emptyset$, $T' \cup T'' = T$ and the disjointedness of control and target lines, we get:

$$f_{mtg'' \cdot mtg'}(a)(x) = \begin{cases} a(x) & \text{if } x \in T'' \text{ and } a(y) = 1 \text{ for all } y \in cp''(x) \\ a(x) & \text{if } x \in T' \text{ and } a(y) = 1 \text{ for all } y \in cp'(x) \\ a(x) & \text{otherwise,} \end{cases}$$

$$= \begin{cases} a(x) & \text{if } x \in T \text{ and } a(y) = 1 \text{ for all } y \in cp(x) \\ a(x) & \text{otherwise,} \end{cases}$$

$$= f_{mtg}(a)(x).$$
The case \( f_{\text{mtg}' \cdot \text{mtg}''}(a)(x) = f_{\text{mtg}}(a)(x) \) is analog (all ' are replaced by '' and vice versa).

This proves the statement.

Proposition 1 shows that the subgates are commuting.

Given the situation of Proposition 1, the circuit \( \text{mtg}' \cdot \text{mtg}'' \) may be seen as a sequentialization of \( \text{mtg} \) and \( \text{mtg}'' \) as a parallelization of \( \text{mtg}' \cdot \text{mtg}'' \). In the next section, both operations are considered within arbitrary circuits.

4. Sequentialization, Parallelization and Shift

Sequentialization and parallelization can be done within large circuits inducing an equivalence relation on multi-target Toffoli circuits. As parallelization, a particular composition of a sequentialization and a parallelization shifts some target lines of a gate to the preceding gate.

Shifts are defined formally as a generalization of parallelization. The shift operation is quite nondeterministic as there may be many gates within a circuit that allow shifting. But it turns out that shifting has the local Church-Rosser property meaning that two circuits obtained by two shifts on a circuit can always be shifted into a common circuit.

Definition 3. Let \( \text{mtg} = (T, cp, cn) \) be a multi-target Toffoli gate and \( T' \subseteq T \) with \( \emptyset \neq T' \neq T \). Let \( \text{mtg}' \) be the restriction of \( \text{mtg} \) to \( T' \) and \( \text{mtg}'' \) the restriction of \( \text{mtg} \) to \( T'' = T - T' \). Then

1. \( \text{mtg}' \cdot \text{mtg}'' \) is called sequentialization of \( \text{mtg} \) wrt \( T' \) and \( \text{mtg}'' \) parallelization of \( \text{mtg}' \cdot \text{mtg}'' \). The parallelization is also denoted by \( \text{mtg}' + \text{mtg}'' \).

2. Let \( \text{mtc} = \text{mtc}' \cdot \text{mtg} \cdot \text{mtg}'' \) be a multi-target Toffoli circuit and \( \text{mtg}' \cdot \text{mtg}'' \) be the sequentialization of \( \text{mtg} \) wrt \( T' \). Then the two circuits \( \text{mtc} \) and \( \text{mtc}' \cdot \text{mtg}'' \cdot \text{mtg}'' \cdot \text{mtc}'' \) are in seq-relation wrt \( T' \) in gate \( i = |\text{mtc}'| + 1 \), denoted by

\[
\text{mtc} \xrightarrow{\text{seq}(i,T')} \text{mtc}'
\]

as well as in par-relation after gate \( i - 1 = |\text{mtc}'| \), denoted by

\[
\text{mtc} \xrightarrow{\text{par}(i-1)} \text{mtc}.
\]

Remark 1. Let \( \sim_{\text{seq}} \) be the equivalence relation induced by \( \text{seq} \), i.e. the reflexive, symmetric, and transitive closure of \( \text{seq} \) and \( \sim_{\text{par}} \) the corresponding equivalence relation induced by \( \text{par} \). As the relations \( \text{seq} \) and \( \text{par} \) are inverse to each other, the symmetric closures make their induced equivalence relations equal.

Definition 4. Let \( \text{mtc} \) and \( \text{mtc}' \) be two multi-target Toffoli circuits. Then \( \text{mtc} \) is a shift of \( \text{mtc} \) if \( \text{mtc} \xrightarrow{\text{par}(i-1)} \text{mtc} \) or \( \text{mtc} \xrightarrow{\text{seq}(i+1,T')} \text{mtc}' \xrightarrow{\text{par}(i-1)} \text{mtc} \) for some \( i \geq 1 \) and \( T' \subseteq X \), denoted by

\[
\text{mtc} \xrightarrow{\text{sh}(i,T')} \text{mtc}'.
\]
where $T'$ is the set of target lines of the gate $i + 1$ in case that the shift is just a parallelization. If $i$ and $T'$ are clear from the context, then we may write $\text{mtc} \xrightarrow{sh} \text{mtc}$.

**Example 3.** Consider the multi-target Toffoli circuit $\text{mtc} = \text{mtg}_1 \ldots \text{mtg}_5$ over four lines $x_1$ to $x_4$ where the gates $\text{mtg}_i = (T_i, cp_i, cn_i)$ for $i = 1, \ldots, 5$ are given by $T_1 = \{x_3\}$, $T_2 = T_3 = T_4 = \{x_4\}$, $cp_1(x_3) = cp_2(x_2) = cp_4(x_4) = \{x_1\}$, $cn_3(x_4) = cn_4(x_4) = cn_5(x_2) = \{x_3\}$ and $cn_1(x_3) = cn_2(x_3) = cp_3(x_4) = cp_5(x_2) = \emptyset$ as depicted in Fig. 2(a).

Obviously, $\text{mtg}_1$ and $\text{mtg}_2$ can be parallelized because the target line of the one gate is no target or control line of the other gate. The same holds for gates $\text{mtg}_4$ and $\text{mtg}_5$. Hence, we get $\text{mtc} \xrightarrow{\text{par}(0)} \text{mtc}' \xrightarrow{\text{par}(2)} \text{mtc}''$ with $\text{mtc}' = \text{mtg}'_1 \text{mtg}'_2 \text{mtg}'_3 \text{mtg}'_4 \text{mtg}'_5$ and $\text{mtc}'' = \text{mtg}''_1 \text{mtg}''_2 \text{mtg}''_3 \text{mtg}''_4$ where $\text{mtg}'_i = (T'_i, cp'_i, cn'_i)$ as follows: $T'_1 = \{x_2, x_3\}$, $cp'_1(x_2) = cp'_i(x_3) = \{x_1\}$, $cn_1(x_2) = \emptyset$, $T'_4 = \{x_2, x_4\}$, $cp'_4(x_2) = \emptyset, cp'_4(x_4) = \{x_1\}$ and $cn'_3(x_2) = cn'_4(x_4) = \{x_3\}$. Afterwards, we can apply the shift $sh(2, \{x_2\})$ to $\text{mtc}''$ by sequentializing $\text{mtg}''_4$ wrt $\{x_2\}$ and parallelizing the resulting circuit after $\text{mtg}''_1$. This yields the circuit depicted in Fig. 2(b) where $\text{mtg}''_2 = \{x_2, x_4\}$, $cp''_2(x_2) = \emptyset$ and $cn''_2(x_2) = cn''_4(x_4) = \{x_3\}$.

**Proposition 2.** The shift relation has the local Church-Rosser property meaning that two shifts on a circuit $\text{mtc}$

\[ \text{mtc} \xrightarrow{sh} \text{mtc}_1 \xrightarrow{sh} \text{mtc}_2 \]

imply

\[ \text{mtc} \xrightarrow{s} \text{mtc} \xrightarrow{sh} \text{mtc}_1 \xrightarrow{sh} \text{mtc}_2 \]

for some circuit $\overline{\text{mtc}}$ where $\xrightarrow{s}$ is the reflexive and transitive closure of the shift relation $sh$.

**Proof** A shift changes two successive gates of a circuit and keeps the rest invariant. Hence two shifts that change four different gates cannot interfere with each other so that they can be applied in any order yielding the same result. The situation becomes more complicated if the two shifts change two or three successive gates. Then various cases can occur.

Let us start with shifts on the same two gates. Then the given circuit has the form $\text{mtc} = \text{mtc}' \text{gg}' \text{mtc}''$ for some circuits $\text{mtc}'$ and $\text{mtc}''$ and gates
\[ g = (T, cp, cn) \text{ and } g' = (T', cp', cn'). \] Both shifts may be proper shifts of different parts of the second gate or one of the shifts is the parallelization of the two gates. If both shifts are proper, the parts shifted may be incomparable or one may be a subpart of the other. Hence there are three cases to be considered. In the following we are abbreviating \(|mtc'|\) by \(i\). Moreover, we omit \(mtc'\) and \(mtc''\) in diagrams because they are kept invariant by all shifts.

Case 1: Consider the case in the following diagram:

\[
\begin{align*}
\text{sh}(i + 1, T) & \quad \text{sh}(i + 1, T) \\
(g + g'|_T)g'|_{T' - T} & \quad (g + g'|_{T - T})g'|_{T' - (T \cup T)}
\end{align*}
\]

The given shifts of \(T\) and \(\hat{T}\) in gate \(i + 2\) with \(\hat{T} - T \neq \emptyset\) are defined if \(T \cap T = \emptyset = \hat{T} \cap T\) and \(T \cap \bigcup_{t \in \hat{T}} (cp'(t) \cup cn'(t)) = \emptyset = T \cap \bigcup_{t \in \hat{T}} (cp'(t) \cup cn'(t))\).

After the shifts the changed gates are:

\[
\begin{align*}
g + g'|_T &= (T \cup \hat{T}, \hat{cp}, \hat{cn}), & g'|_{T - \hat{T}} &= (T - \hat{T}, cp'|_{T - T}, cn'|_{T - \hat{T}}), & (1) \\
g + g'|_{\hat{T}} &= (T \cup \hat{T}, \hat{cp}, \hat{cn}), & g'|_{\hat{T} - T} &= (T - \hat{T}, cp'|_{\hat{T} - T}, cn'|_{\hat{T} - T}) & (2)
\end{align*}
\]

with \(\hat{cp}(x) = \hat{cp}(x) = cp(x)\) and \(\hat{cn}(x) = \hat{cn}(x) = cn(x)\) for \(x \in T\), \(\hat{cp}(x) = cp'(x)\) and \(\hat{cn}(x) = cn'(x)\) for \(x \in \hat{T}\), as well as \(\hat{cp}(x) = cp'(x)\) and \(\hat{cn}(x) = cn'(x)\) for \(x \in \hat{T}\).

Moreover, the following holds:

\[
\begin{align*}
(T \cup \hat{T}) \cap (T \cup \hat{T}) &= ((\hat{T} - T) \cap T) \cup ((\hat{T} - T) \cap \hat{T}) \subseteq \hat{T} \cap T = \emptyset, & (3) \\
(T \cup \hat{T}) \cap c'(\hat{T} - T) &= (T \cap c'(\hat{T} - T)) \cup (\hat{T} \cap c'(\hat{T} - T)) \subseteq T \cap c'(\hat{T}) = \emptyset & (4)
\end{align*}
\]

where \(c'(X) = \bigcup_{t \in X} (cp'(t) \cup cn'(t))\).

Therefore the shift of \(\hat{T} - T\) in gate \(g'|_{T - \hat{T}}\) to the preceding gate \(g + g'|_T\) is defined because \(\hat{T} - T \neq \emptyset\). Analogously the shift of \(\hat{T} - \hat{T}\) in gate \(g'|_{\hat{T} - \hat{T}}\) to the preceding gate \(g + g'|_{\hat{T}}\) is defined because \(\hat{T} - \hat{T} \neq \emptyset\). The changed gates are:

\[
\begin{align*}
(g + g'|_T) + (g'|_{T - \hat{T}})|_{\hat{T} \cup T} &= g + g'|_{T \cup T}, & (g'|_{T - \hat{T}})|_{(T \cup \hat{T} - T)} &= g'|_{T - (T \cup \hat{T})}, \\
(g + g'|_{\hat{T}}) + (g'|_{\hat{T} - T})|_{\hat{T}} &= g + g'|_{\hat{T}}, & (g'|_{\hat{T} - T})|_{(\hat{T} \cup T - T)} &= g'|_{\hat{T} - (\hat{T} \cup T)}
\end{align*}
\]

This proves that the two further shifts yield the same circuit.

Case 2: Using the same denotation and convention, consider the following
The situation is similar to Case 1 with the exception that $\hat{T} \subseteq \hat{T}$ which implies $\hat{T} - \hat{T} = \emptyset$. Using the same argument as in Case 1 the two shifts of $\hat{T}$ and $\hat{T}$ in gate $i + 2$ are defined. Hence, the changed gates after the shifts are the same as in [1] and [2] and, moreover, also [3] and [4] hold. Therefore, as in Case 1 the shift of $\hat{T} - \hat{T}$ in gate $g'|_{T - \hat{T}}$ of the preceding gate $g + g'|_{\hat{T}}$ is defined because the sequentialization of gate $i + 1$ wrt $\hat{T}$ in $mtc'(g + g'|_{\hat{T}})g'|_{T - \hat{T}} mtc''$ results in

$$mtc'(g + g'|_{\hat{T}})g'|_{\hat{T} - \hat{T}} mtc'' = mtc'(g + g'|_{\hat{T}})g'|_{T - \hat{T}} mtc''$$

because $\hat{T} \subseteq \hat{T}$. A parallelization after gate $i + 1$ in this circuit results in

$$mtc'(g + g'|_{\hat{T}})g'|_{\hat{T} - \hat{T}} mtc'' = mtc'(g + g'|_{\hat{T}})g'|_{T - \hat{T}} mtc''$$

because $g'|_{\hat{T}} + g'|_{\hat{T} - \hat{T}} = g'|_{\hat{T}}$ and $\hat{T} \cap T = \emptyset = (\hat{T} - \hat{T}) \cap \hat{T}$. Therefore, the shift of $\hat{T} - \hat{T}$ after the shift of $\hat{T}$ yields the same result as the shift of $\hat{T}$ in the first place.

Case 3: Given a shift and a parallelization as in the diagram:

The parallelization after the shift is defined and yields the same result as the parallelization directly using arguments similar to Case 1. Specifically, consider a shift after gate $i + 1$ wrt $\hat{T} \subseteq T'$. The changed gates after the shift are given in [1] and [2]. But because of $T \cap T' = \emptyset, \hat{T} \subseteq T'$ and $T \cap c(T') = \emptyset = T' \cap c(T)$ we have $(T \cup T) \cap (T' - \hat{T}) = \emptyset$ and $(T \cup T) \cap c(T' - \hat{T}) = \emptyset = (T' - \hat{T}) \cap c(T - \hat{T})$ where $c(X) = \bigcup \{cp(t) \cup cn(t)\}$. Hence, after the shift $sh(i + 1, \hat{T})$ parallelizing after gate $i + 1$ yields in the same result as the parallelization directly.

Now we consider two shifts changing three successive gates. Then the given circuit has the form $mtc = mtc'gg\ g''$ for some circuits $mtc'$ and $mtc''$ and three gates $g = (T, cp, cn), g' = (T', cp', cn')$ and $g'' = (T'', cp'', cn'')$. Let
\[ i = |\text{mtc}| \text{ again. Both shifts may be parallelizations or one is a parallelization and the other one a proper shift or both are proper shifts. The argumentation that the given shifts can be continued by further shifts into the same result is in all four cases similar to the argumentation in Case 1.} \]

**Case 4:** Consider two parallelizations as in the diagram:

\[ gg'g'' \xrightarrow{\text{par}(i)} (g + g')g'' \]

\[ g(g' + g'') \xrightarrow{\text{sh}(i + 1, T')} (g' + g'') \]

If a parallelization of \( g \) and \( g' \) as well as \( g' \) and \( g'' \) is possible, then \( T \cap T' = \emptyset = T' \cap T'' \), \( T \cap c(T') = \emptyset = T' \cap c(T) \), as well as \( T' \cap c(T'') = \emptyset = T'' \cap c(T') \) where again \( c(X) = \bigcup_{t \in X} (cp(t) \cup cn(t)) \). The resulting gates are \( (g + g') \) and \( (g' + g'') \). Now a sequentialization in gate \( i + 1 \) wrt \( T'' \) in \( \text{mtc}'g(g' + g'')\text{mtc}'' \) results in \( \text{mtc}'g'g''\text{mtc}'' \), i.e. the original circuit because \( (T' \cup T'') \cap T' = T' \) and \( (T' \cup T'') \cap (T'' - T') = T'' \). Hence, as in first place a parallelization after gate \( i \) is possible. This is a parallelization after gate \( i + 1 \) followed by a sequentialization in gate \( i \) wrt \( T' \) and a parallelization after gate \( i + 1 \), i.e. a shift after gate \( i + 1 \) wrt \( T' \), yields in the same circuit as a parallelization at once.

**Case 5:** Now consider a parallelization and a proper shift. In this very case, two further shifts are applied after the given proper shift to keep up with the given parallelization. This can be seen in the following diagram:

\[ gg'g'' \xrightarrow{\text{par}(i)} (g + g')g'' \]

\[ g(g' + g'') \xrightarrow{\text{sh}(i + 2, \hat{T})} (g' + g''|_{\hat{T}})g''|_{T'' - \hat{T}} \]

\[ (g + g')g''|_{\hat{T}}g''|_{T'' - \hat{T}} \xrightarrow{\text{sh}(i + 1, T')} (g + g')g''|_{\hat{T}}g''|_{T'' - \hat{T}} \]

The circuit after the first proper shift wrt \( \hat{T} \subseteq T'' \) has the form

\[ \text{mtc}'g(g' + g'')|_{\hat{T}}g''|_{T'' - \hat{T}}\text{mtc}'' \]

By assumption \( g \) and \( g' \) can be parallelized, i.e., the shift of \( T' \) in gate \( i + 1 \) is defined. Both together yield

\[ \text{mtc}'(g + g')g''|_{\hat{T}}g''|_{T'' - \hat{T}}\text{mtc}'' \]

where \( g''|_{\hat{T}}g''|_{T'' - \hat{T}} \) is a sequentialization of \( g'' \) wrt \( \hat{T} \) so that the parallelization is defined yielding \( \text{mtc}'(g + g')g''\text{mtc}'' \) as stated.

**Case 6:** Considering the converse case of a proper shift and a parallelization
as in the following diagram:

\[
\begin{array}{c}
\text{gg}'g'' \xrightarrow{sh(i+1,T)} (g + g'|_T)g'|_{T' \cap \hat{T}}g'' \\
\xrightarrow{par(i+1)} (g + g'|_T)(g' + g'')|_{(T') \cup \hat{T}' \cup \hat{T}'' - \hat{T}}
\end{array}
\]

Let \( T \cap \hat{T} = \emptyset, \hat{T} \subseteq T, T' \cap T'' = \emptyset. \) Then \( sh(i+1,T) \) and \( par(i+1) \) are defined in \( mtc'gg'g''mtc''. \) The circuit after the proper shift has the form \( mtc'(g + g'|_T)g''|_{T' \cap \hat{T}}\) whereas the circuit after the parallelization has the form \( mtc'g(g' + g'')mtc''. \) Due to the fact that \( T' \cap T'' = \emptyset \) we have \( (T' \cap \hat{T}) \cap T'' = \emptyset. \) Hence, \( par(i+1,T) \) is defined on \( mtc'g(g' + g'')mtc'' \) resulting in

\[
mtc'(g + g'|_T)(g' + g'')|_{(T' \cup T'') \cup \hat{T}}. \tag{5}
\]

On the other hand a sequentialization in \( i + 1 \) wrt \( \hat{T} \) in \( mtc'gg'g''mtc'' \) results in \( mtc'g(g' + g'')|_{T'' \cup \hat{T}'' \cup \hat{T}}. \) Because of \( T \cap \hat{T} = \emptyset \) the circuit can be parallelized after gate \( i \) also yielding in \( \text{[5]} \).

Case 7: Finally, consider the following case with two proper shifts:

\[
\begin{array}{c}
\text{gg}'g'' \xrightarrow{sh(i+2,\hat{T})} (g + g'|_T)g'|_{T' \cap \hat{T}}g'' \\
\xrightarrow{sh(i+1,\hat{T})} (g + g'|_T)(g' + g'')|_{(T' \cup T'') \cup \hat{T}}
\end{array}
\]

Let \( T \cap \hat{T} = \emptyset, \hat{T} \subseteq T', T' \cap \hat{T} = \emptyset. \) Then \( mtc'gg'g''mtc'' \) can be shifted after gate \( i + 1 \) wrt \( \hat{T} \) into

\[
mtc'(g + g'|_T)g'|_{T' \cap \hat{T}}g''\]  \( mtc'' \)

but it can be also shifted after gate \( i + 2 \) wrt \( \hat{T} \) into

\[
mtc'g(g' + g'')|_{T'' \cup \hat{T}'' \cup \hat{T}}. \]

Because of \( T' \cap \hat{T} \) we can shift after gate \( i + 2 \) wrt \( \hat{T} \) in \( mtc'(g + g'|_T)g'|_{T' \cap \hat{T}}g''\]  \( mtc'' \)

resulting in

\[
mtc'(g + g'|_T)(g' + g'')|_{(T' \cup T'') \cup \hat{T}}g''\]  \( mtc'' \) \( \tag{6} \)

and because of \( T \cap \hat{T} \) shifting after gate \( i + 1 \) wrt \( \hat{T} \) in \( mtc'g(g' + g'')|_{T'' \cup \hat{T}'' \cup \hat{T}} \]

also results in \( \text{[6]} \).

As there are no cases left, the local Church-Rosser property of shifts is proved.
5. Waiting Degree

Besides the local Church-Rosser property, the shift operation has a second significant property: It does not allow infinite shift sequences. In other words, the lengths of shift sequences starting in some circuit are bounded. Consequently, shifting as long as possible always terminates in a circuit that is reduced with respect to shifting. To prove this, we introduce the waiting degree and show that it decreases with each shift. The waiting degree of a circuit sums up, for each target line, the number of gates that precede the gate of the target line.

Definition 5 (Waiting degree). Let $m_{tc} = (T_1, cp_1, cn_1) \cdots (T_n, cp_n, cn_n)$ be a multi-target Toffoli circuit. Then the waiting degree of $m_{tc}$ is

$$\text{wait}(m_{tc}) = \sum_{j=1}^{n} (j - 1) \cdot \#T_j$$

where $\#T_j$ denotes the number of elements of $T_j$.

Example 4. The waiting degree of the circuit in Fig. 2a is 10 and the waiting degree of the circuit in Fig. 2b is 4.

Proposition 3.

1. If $m_{tc} \xrightarrow{\text{par}(i-1)} \tilde{m}_{tc}$, then $\text{wait}(\tilde{m}_{tc}) = \text{wait}(m_{tc}) - \sum_{j=i+1}^{n} \#T_j$.

2. If $m_{tc} \xrightarrow{\text{seq}(i+1,T') \text{par}(i-1)} \tilde{m}_{tc}$, then $\text{wait}(\tilde{m}_{tc}) = \text{wait}(m_{tc}) - \#T'$.

Proof

1. In this case,

$$\tilde{m}_{tc} = \text{mtg}_1 \cdots \text{mtg}_{i-1} \text{mtg}_i \text{mtg}_{i+2} \cdots \text{mtg}_n$$

with $\text{mtg}_i = (T_i + T_{i+1}, cp, cn), cp(x) = cp_i(x), cn(x) = cn_i(x)$ for $x \in T_i$ and $cp(x) = cp_{i+1}(x), cn(x) = cn_{i+1}(x)$ for $x \in T_{i+1}$. Therefore,

$$\text{wait}(\tilde{m}_{tc}) = \sum_{j=1}^{n-1} (j - 1) \#\tilde{T}_j$$

$$= \sum_{j=1}^{i-1} (j - 1) \#\tilde{T}_j + (i - 1) \#\tilde{T}_i + \sum_{j=i+1}^{n-1} (j - 1) \#\tilde{T}_j$$

$$= \sum_{j=1}^{i-1} (j - 1) \#T_j + (i - 1) \#(T_i + T_{i+1}) + \sum_{j=i+1}^{n-1} (j - 1) \#T_{j+1}$$

$$= \sum_{j=1}^{i-1} (j - 1) \#T_j + (i - 1) \#T_i + i \#T_{i+1} - \#T_{i+1} + \sum_{j=i+2}^{n} (j - 2) \#T_j$$
\[
\begin{align*}
&= \left( \sum_{j=1}^{i+1} (j-1) \#T_j \right) - \#T_{i+1} + \sum_{j=i+2}^{n} ((j-1) \#T_j - \#T_j) \\
&= \left( \sum_{j=1}^{n} (j-1) \#T_j \right) - \sum_{j=i+1}^{n} \#T_j = \text{wait}(mtc) - \sum_{j=i+1}^{n} \#T_j
\end{align*}
\]

2. The proof in this case is analogously.

**Remark 2.** Let \( mtc = (T_1, cp_1, cn_1) \cdots (T_n, cp_n, cn_n) \) be a multi-target Toffoli circuit. Sequentialize \( mtc \) as long as possible. Then the result has length \( m \) and waiting degree \( \frac{m(m-1)}{2} \). But \( \text{wait}(mtc) \) is not greater because sequentialization increases the waiting degree. Hence, \( \text{wait}(mtc) \leq \frac{m(m-1)}{2} \) for \( m = \sum_{j=1}^{n} \#T_j \).

The properties in Proposition 3 and Remark 2 imply the following corollary.

**Corollary 1.**
1. Let \( mtc \xrightarrow{n \text{ sh}} \overline{mtc} \) be a shift sequence of \( n \) shifts. Then \( n \leq \text{wait}(mtc) \).
2. Let \( mtc = (T_1, cp_1, cn_1) \cdots (T_n, cp_n, cn_n) \) be a multi-target Toffoli circuit. Let \( m = \sum_{i=1}^{n} \#T_i \). Then shifting as long as possible terminates with a circuit that is reduced wrt shifts after at most \( \frac{m(m-1)}{2} \) shifts.

6. **Canonical Circuits**

Circuits that are reduced wrt shifts are called canonical. They are local optima wrt the waiting degree. But this result can be tremendously improved by combining the termination with the local Church-Rosser property. The shifting defines an equivalence relation on circuits. Each equivalence class contains only circuits that are semantically equivalent. Moreover, it turns out that each canonical circuit is a unique representative of its shift equivalence class so that it is a global optimum within its class. To show this, we prove first that shift equivalence is confluent meaning that each two equivalent circuits can be shifted into a common circuit.

**Remark 3.** Let \( \sim \) be the equivalence relation generated by the shift relation, called shift equivalence. By definition \( \text{par} \subseteq \text{shift} \) and \( \text{shift} \subseteq \text{par} \cup \text{par} \circ \text{seq} \subseteq \text{par} \cup (\text{par} \circ \text{par}^{-1}) \subseteq (\text{par} \cup \text{par}^{-1})^* = \sim_{\text{par}} \). Hence, \( \sim \) is equal to \( \sim_{\text{seq}} = \sim_{\text{par}} \).

**Theorem 1.** Shift-equivalent canonical circuits are equal.

**Proof** Let \( mtc \) and \( \overline{mtc} \) be two shift-equivalent canonical circuits. Due to the following lemma, there is a circuit \( \overline{mtc} \) and there are shift sequences from \( mtc \) and \( \overline{mtc} \) into \( mtc \). Because \( mtc \) and \( \overline{mtc} \) are canonical and hence shift-reduced, both shift sequences have length 0 yielding \( mtc = mtc = \overline{mtc} \) as stated.
Lemma 1. mtc \sim \overline{mtc} implies \( mtc \xrightarrow{sh,mtc} mtc \) for some multi-target Toffoli circuit \( mtc \).

Proof: \( mtc \sim \overline{mtc} \) iff there is a sequence \( zz = mtc_0 \cdots mtc_n \) such that \( mtc_0 = mtc, mtc_n = \overline{mtc}, mtc_i \xrightarrow{sh} mtc_{i+1} \) or \( mtc_{i+1} \xrightarrow{sh} mtc_i \) for all \( i = 0, \ldots, n - 1 \), i.e. a zigzag of shifts.

Let \( MTC(zz) = \{ mtc_i \mid i = 0, \ldots, n \} \) and let \( X \) be a finite set of multi-target Toffoli circuits. Let \( \text{reach}(X) = \{ mtc \mid mtc \xrightarrow{sh} mtc, mtc \in X \} \). Note that \( \text{reach}(X) \) is finite. Then, \( mtc \xrightarrow{sh,mtc} mtc_{i-1} \) is a critical pair of \( zz \) if \( mtc, \notin \text{reach}(MTC(zz) - \{ mtc_i \}) \), i.e. \( mtc_i \) is a critical element of \( zz \).

Induction on \#reach(CE(zz)), where \( CE(zz) \) denotes the set of critical elements of \( zz \).

Base: \#reach(CE(zz)) = 0. Then there is no critical element because each critical element is reachable by itself by 0 shifts and belongs to \( reach(CE(zz)) \). Therefore, \( zz \) must contain a multi-target Toffoli circuit \( mtc_i \) with \( mtc_i \xrightarrow{sh} mtc_{i+1} \) for all \( i < i_0 \) and \( mtc_{i+1} \xrightarrow{sh} mtc_i \) for all \( i \geq i_0 \).

Step: Let \#reach(CE(zz)) = \( k \) with \( k > 0 \). Let \( mtc_i \) be a critical element of \( zz \) i.e. \( mtc_i \in CE(zz) \). Then one can replace \( mtc_{i-1} \leftarrow mtc_i \xrightarrow{sh} mtc_{i+1} \) in \( zz \) by the shifts that make the shift relation locally Church-Rosser due to Proposition 2. Defining a new \( zz' \). The new elements of \( zz' \) are not critical as none of them has branching shifts. Hence, \( CE(zz') \subseteq MTC(zz) \subseteq \text{reach}(CE(zz)) \). This implies \( \text{reach}(CE(zz')) \subseteq \text{reach}(\text{reach}(CE(zz)) = \text{reach}(CE(zz)) \). The inclusion is proper as \( mtc_i \notin \text{reach}(CE(zz')) \) because of the following reason. Assuming \( mtc_j \in \text{reach}(CE(zz')) \) then \( mtc_j \xrightarrow{sh,mtc} mtc_i \) for some \( mtc_j \in CE(zz') \).

As \( mtc_j \in MTC(zz') - \{ mtc_i \} \) we get \( mtc_i \in \text{reach}(MTC(zz') - \{ mtc_i \}) \) in contradiction to the choice of \( mtc_i \).

Therefore, \#reach(CE(zz')) < \( k \) so that by induction hypothesis, the lemma holds for \( zz' \) and for \( zz \) too as they connect the same circuits.

Remark 4. How much can one gain by shifting a circuit into its canonical form? In a shift equivalence class, the canonical circuit is a unique optimum with respect to the waiting degree. Moreover, it has the smallest gate count among the equivalent circuits, but it is not unique in this respect. So the quantitative gain depends on the waiting degree of the other equivalent members. One finds always members with \( n \) gates where \( n \) is the number of negations. Their waiting degree is \( \frac{n(n-1)}{2} \). This is also the maximum length of a shift sequence such that the canonical circuits is reached after a quadratic number of shifts at most. There are two extreme cases. First, each two successive gates are dependent of each other. Then no shift is possible, and one wins nothing. Second, each two gates are independent. Then the canonical circuit has waiting degree 0, and the gain lies between 0 and the quadratic upper bound. All other cases are placed in between so that the gain lies between the lower bound given by the canonical
7. Combining Shifts with Other Transformations

Besides shifts, there are further transformations on Toffoli circuits that preserve the functional semantics while decreasing the waiting degree. For example, each Toffoli gate is obviously inverse to itself so that two successive equal gates in a circuit compute the identity and can be removed without change of the semantics, but with a significant drop of the waiting degree. After such a removal, further shifts may be possible such that the optimization with respect to the waiting degree can be continued. Concerning multi-target Toffoli circuits, the same works for a target line in two successive gates if it has the same control in both gates. Clearly, the combination of shifts and removals of successive identical target lines with equal control as long as possible terminates always because both transformations decrease the waiting degree. Like in the case of shifting as long as possible, it turns out that also the combination of shifts and removals yields unique reduced forms.

Definition 6. Let \( mtc = mtc' \cap mtc'' \cap mtc'' \) be a multi-target Toffoli circuit, \( mtc' = (T', cp', cn') \), \( mtc'' = (T'', cp'', cn'') \) and let \( t_0 \in T' \cap T'' \) with \( cp'(t_0) = cp''(t_0) \) and \( cn'(t_0) = cn''(t_0) \). Then the removal of \( t_0 \) in \( mtc' \) and \( mtc'' \) yields the circuit \( \overset{\text{rem}_{(i,t_0)}}{mtc} = mtc' \cap \overset{\text{rem}_{(i,t_0)}}{mtc''} \cap mtc'' \) denoted by \( mtc \overset{\text{rem}_{(i,t_0)}}{\rightarrow} \overset{\text{rem}_{(i,t_0)}}{mtc} \) for \( i = |mtc'| \) where \( \text{rem}_{(i,t_0)}(mtc' \cap mtc'') \) is defined as follows:

\[
\text{rem}_{(i,t_0)}(mtc' \cap mtc'') = \begin{cases} 
\text{mtc'}(T' - \{t_0\}) & \text{if } T' - \{t_0\} \neq \emptyset \neq T' - \{t_0\} \\
\text{mtc''}(T'' - \{t_0\}) & \text{if } T'' - \{t_0\} \neq \emptyset \neq T'' - \{t_0\} \\
\text{mtc''}(T'' - \{t_0\}) & \text{if } T'' - \{t_0\} \neq \emptyset \\
\lambda & \text{if } T'' - \{t_0\} = T'' - \{t_0\} 
\end{cases}
\]

where \( \lambda \) is the empty string.

Example 5. Consider the circuit \( mtc_1 = mtc_1 \cdots mtc_6 \) where the gates \( mtc_i = (T_i, cp_i, cn_i) \) for \( i = 1, \ldots, 6 \) are given by \( T_1 = \{x_2, x_5\}, T_2 = \{x_5\}, T_3 = T_5 = \{x_4\}, T_4 = T_6 = \{x_3\}, cp_1(x_2) = \{x_1, x_4\}, cp_1(x_5) = cp_2(x_5) = cp_6(x_3) = cp_3(x_4) = cp_5(x_4) = \emptyset \) as depicted in Fig. 3(a). The waiting degree of \( mtc_1 \) is 15. The gates \( mtc_3 \) and \( mtc_4 \) can be parallelized yielding
Erasing can decrease the waiting degree. Figure 3: Erasing can decreases the waiting degree

\( \text{mtg}_{34} = (\{x_3, x_4\}, \text{cp}_{34}, \text{cn}_{34}) \) with \( \text{cp}_{34}(x_3) = \emptyset \) and \( \text{cn}_{34}(x_3) = \{x_3\} \). Afterwards, no further shift is possible. The resulting circuit \( \text{mtc}_2 = \text{mtg}_1 \text{mtg}_2 \text{mtg}_{34} \text{mtg}_5 \text{mtg}_6 \) (Fig. 3(b)) has the waiting degree 12. But one can remove the target line \( x_5 \) from \( \text{mtg}_1 \) and \( \text{mtg}_2 \) yielding \( \text{rem}(0, x_5) = (\{x_2\}, \text{cp}', \text{cn}') \) with \( \text{cp}'(x_2) = x_1, x_4 \) and \( \text{cn}'(x_2) = \emptyset \). The resulting circuit \( \text{mtc}_3 = \text{rem}(0, x_5) \text{mtg}_{34} \text{mtg}_5 \text{mtg}_6 \) (Fig. 3(c)) has waiting degree 7. Now another shift is possible: \( \text{sh}(1, \{x_3\}) \); yielding \( \text{mtc}_4 = \text{mtg}_1' \text{mtg}_5 \text{mtg}_5 \text{mtg}_6 \) (Fig 3(d)) with \( \text{mtg}_1' = (\{x_2, x_3\}, \text{cp}', \text{cn}') \) where \( \text{cp}(x_2) = \{x_1, x_4\}, \text{cn}(x_3) = \{x_3\} \) and \( \text{cp}(x_5) = \text{cn}(x_5) = \emptyset \). Its waiting degree is 6. Finally, the two equal gates can be removed yielding \( \text{mtc}_5 = \text{mtg}_1' \text{mtg}_6 \) (Fig 3(e)) with waiting degree 1. No further shift or removal is possible so that \( \text{mtc}_5 \) is a reduced form.

As the example shows, the combination of shifting and removal allows in some cases to decrease the waiting degree beyond the effect of shifting alone. Moreover, the iteration of shifts and removals always terminates as removal decreases the waiting degree like shifting and the reduced forms are also unique. The result follows again from the local Church-Rosser property of removals and of removals vs. shifts.

**Proposition 4.** Let \( \text{mtc} \xrightarrow{\text{rem}(i, t_0)} \) \( \text{mtc} \) be a removal of \( t_0 \) after step \( i \). Then \( \text{wait}(\text{mtc}) \leq \text{wait}(\text{mtc}) - 2i - 1 \).

**Proof** The target line \( t_0 \) in gate \( i + 1 \) waits \( i \) steps and in gate \( i + 2 \) respective \( i + 1 \) steps. Both do not contribute to \( \text{wait}(\text{mtc}) \) after the removal. All other target lines remain in their gates. But the removal may remove the gates \( i + 1 \)
and \(i + 2\) completely so we obtain the following cases for the value of \(\text{wait}(\overline{\text{mtc}})\):

\[
\text{wait}(\overline{\text{mtc}}) - i - (i + 1) \quad \text{if } g_i' \neq \lambda \neq g_{i+2}'.
\]

\[
\sum_{j=1}^{i} (j - 1)\#T_j + i \cdot (\#T_{i+2} - 1) + \sum_{j=i+3}^{n} (j - 2)\#T_j \quad \text{if } g_i' = \lambda \neq g_{i+2}'.
\]

\[
\sum_{j=1}^{i} (j - 1)\#T_j + i \cdot (\#T_{i+1} - 1) + \sum_{j=i+3}^{n} (j - 2)\#T_j \quad \text{if } g_i' \neq \lambda = g_{i+2}'.
\]

\[
\sum_{j=1}^{i} (j - 1)\#T_j + \sum_{j=i+3}^{n} (j - 3)\#T_j \quad \text{if } g_i' = \lambda = g_{i+2}'.
\]

where \(g_i' = g_i|T - \{t_0\}\). Clearly, \(7 > 8 > 10\) and \(7 > 9 > 10\). Hence, \(\text{wait}(\overline{\text{mtc}}) \leq \text{wait}(\text{mtc}) - 2i - 1\).

**Definition 7.**

1. All removals on circuits define the \(\text{rem}\)-relation.
2. The union of the \(\text{rem}\)-relation and the \(\text{shift}\)-relation is called \(\text{sh}&\text{rem}\)-relation.
3. If a circuit \(\overline{\text{mtc}}\) is obtained by a shift or a removal from the circuit \(\text{mtc}\), then this is denoted by \(\text{mtc} \rightarrow \overline{\text{mtc}}\).
4. The equivalence relation, denoted by \(\approx\), induced by the \(\text{sh}&\text{rem}\)-relation is called \(\text{sh}&\text{rem}\)-equivalence.

**Corollary 2.**

1. Let \(\text{mtc} \overset{n}{\rightarrow} \overline{\text{mtc}}\) be a \(\text{sh}&\text{rem}\)-sequence of length \(n\). Then \(n \leq \text{wait}(\text{mtc})\).
2. Let \(\text{mtc} = (T_1,c_{p_1},c_{n_1}) \cdots (T_n,c_{p_n},c_{n_n})\) be a multi-target Toffoli circuit.
   Let \(m = \sum_{i=1}^{n} \#T_i\). Then shifting and removing as long as possible terminates with a \(\text{sh}&\text{rem}\)-reduced circuit after at most \(m(m-1)/2\) steps.

**Proof**

1. Shifting and removal decrease the waiting degree, and
2. \(m(m-1)/2\) is the maximal possible waiting degree.

**Proposition 5.**

The \(\text{sh}&\text{rem}\)-relation has the local Church-Rosser property meaning that two shifts, two removals or a shift and a removal on \(\text{mtc}\)

\[
\text{mtc} \overset{\text{sh}}{\rightarrow} \text{mtc}_1 \quad \text{imply} \quad \text{mtc}_1 \overset{\text{sh}}{\rightarrow} \text{mtc}
\]

for some circuit \(\overline{\text{mtc}}\) where \(\overset{\text{sh}}{\rightarrow}\) is the reflexive and transitive closure of the \(\text{sh}&\text{rem}\)-relation.

**Proof** For two shifts, Proposition 2 applies. It remains to consider \(\text{mtc} \overset{\text{op}_1}{\rightarrow} \text{mtc}_1\) and \(\text{mtc} \overset{\text{op}_2}{\rightarrow} \text{mtc}_2\) where \(\text{op}_1\) is a removal and \(\text{op}_2\) is a removal or a shift.

In several cases, it is obvious the \(\text{op}_2\) can be applied to \(\text{mtc}_1\) and \(\text{op}_1\) to \(\text{mtc}_2\).
both yielding the same circuit \( mtc \) because the operations cannot interfere with each other. Because shift and removal respectively change only two gates within a circuit, this is true in particular if \( op_1 \) and \( op_2 \) change four different gates. This is also true if \( op_2 \) removes a target line different from the one removed by \( op_1 \). This leaves the following cases to be considered where \( op_2 \) is a shift with the exception of Case 2 where only two gates are changed.

1. \( op_1 \) and \( op_2 \) remove the same target line.
2. \( op_1 \) and \( op_2 \) change the same two successive gates.
3. \( op_1 \) removes a target line from the first two successive gates and \( op_2 \) shifts some target lines from the third gate to the second one.
4. \( op_1 \) removes a target line from the second two successive gates and \( op_2 \) shifts target lines from the second gates to the first one.

In all cases, \( mtc \) has the form \( mtc = mtc' gg' mtc'' \) or \( mtc = mtc' gg' gg'' mtc'' \) for some circuits \( mtc' \) and \( mtc'' \) and gates \( g = (T, cp, cn), g' = (T', cp', cn'), \) and \( g'' = (T'', cp'', cn'') \). The operations \( op_1 \) and \( op_2 \) change only \( gg' \) or \( gg' gg'' \). None of the further operations to get the local Church-Rosser property is changing \( mtc' \) and \( mtc'' \) so that one can restrict the consideration to the gates between \( mtc' \) and \( mtc'' \). The operation \( op_1 \) is always a removal, say of \( t_0 \). Then it changes \( gg' \) into \( g|T-\{t_0\}g'|T'-\{t_0\}g'' \) or \( g' g'' \) into \( g'|T-\{t_0\}g''|T''-\{t_0\} \) where \( T \) for some gate \( T \) is \( \lambda \). If \( op_2 \) is also a removal, then it removes after \( t_0 \) with the same effect as \( op_1 \). If \( op_2 \) is a shift, say of the set of target lines \( T \), then it changes either \( gg' \) into \( g+g'|T-\{t_0\}g''|T''-\{t_0\} \) or \( g' g'' \) into \( g'|T-\{t_0\}g''|T''-\{t_0\} \). Let us now look into the four cases separately. In all cases \( i = |mtc'| \).

Case 1: The two removal yield the upper left span in the following diagram:

\[
gg' g'' \xrightarrow{rem(i, t_0)} g|T-\{t_0\}g'|T'-\{t_0\}g'' \xrightarrow{sh(i + 2, \{t_0\})} g|T-\{t_0\}g'|T'-\{t_0\}g''|T''-\{t_0\} =
\]

Due to the given removals, \( t_0 \) has the same control lines in \( g, g' \) and \( g'' \). Therefore, \( t_0 \) in \( g'' \) is independent of \( T' - \{t_0\} \) in \( g' \) provided that \( T' - \{t_0\} \neq \emptyset \). Hence, \( sh(i + 2, \{t_0\}) \) is defined and the middle gate of the result is \( g' \). The same reasoning applies to \( t_0 \) in \( g' \) and \( T - \{t_0\} \) in \( g \) so that \( sh(i + 1, \{t_0\}) \) is defined as given in the diagram.

This works for (a) \( T \neq \{t_0\} \neq T' \) and arbitrary \( T'' \). If \( T'' = \{t_0\} \), then \( g''|T''-\{t_0\} \) is \( \lambda \) without changing the arguments.
(b) $T = \{t_0\} \neq T'$. Then one gets

$$\begin{align*}
&gg'g'' \xrightarrow{rem(i, t_0)} g'|_{T' - \{t_0\}} g'' \\
&\xrightarrow{rem(i + 1, t_0)} g'|_{T' - \{t_0\}} + g''|_{\{t_0\}} g''|_{T'' - \{t_0\}} \\
&= sh(i + 1, \{t_0\}) \xrightarrow{rem(i + 1, t_0)} g' g''|_{T'' - \{t_0\}}
\end{align*}$$

The arguments for $sh(i + 1, \{t_0\})$ are the same as for $sh(i + 2, \{t_0\})$ in Subcase 1a. Because $t_0$ has the same control lines in $g, g'$ and $g''$ and $g = g'|_{\{t_0\}}$, the parallelization of $g$ and $g'|_{T' - \{t_0\}}$ yields $g'$.

(c) $T \neq \{t_0\} = T'$. Then one gets

$$\begin{align*}
&gg'g'' \xrightarrow{rem(i, t_0)} g|_{T - \{t_0\}} g'' \\
&\xrightarrow{rem(i + 1, t_0)} g|_{T - \{t_0\}} + g''|_{\{t_0\}} g''|_{T'' - \{t_0\}} \\
&= sh(i + 1, T' - \{t_0\}) \xrightarrow{rem(i + 1, t_0)} g' g''|_{T'' - \{t_0\}}
\end{align*}$$

The arguments for $sh(i + 1, \{t_0\})$ are the same as before only that it is applied to the independence of $t_0$ in $g''$ and $T - \{t_0\}$.

(d) $T = T' = \{t_0\} \neq T''$. Then one yields

$$\begin{align*}
&gg'g'' \xrightarrow{rem(i, t_0)} g'' \\
&\xrightarrow{rem(i + 1, t_0)} g''|_{T'' - \{t_0\}} + g''|_{\{t_0\}} g''|_{T'' - \{t_0\}} \\
&= sh(i + 1, T'' - \{t_0\}) \xrightarrow{rem(i + 1, t_0)} g''|_{T'' - \{t_0\}}
\end{align*}$$

The arguments for $sh(i + 1, T'' - \{t_0\})$ are the same as for $sh(i + 1, T' - \{t_0\})$ in Subcase 1b.

(e) $T = T' = T'' = \{t_0\}$. Then $g = g' = g''$ so that both removals yield the same result $g$.

Case 2: (a) $T \neq \{t_0\} \neq T'$. The operations $op_1$ and $op_2$ yield the upper left
The $t_0$ in $g'$ is dependent on the $t_0$ in $g$ so that $t_0$ is not shifted and $\text{rem}(i, t_0)$ is defined on the result of $\text{sh}(i + 1, \hat{T})$ yielding

$$g|_{T - \{t_0\}} + g'|_{\hat{T}}(g'|_{T' - \{t_0\}})\big|_{(T' - \{t_0\}) - \hat{T}} = (g + g'|_{\hat{T}})(g'|_{T' - \{t_0\}})\big|_{(T' - \hat{T}) - \{t_0\}}. \quad (11)$$

Moreover, $\hat{T} \subseteq T' - \{t_0\}$ is independent of $T - \{t_0\}$ as it is already independent of $T$. Hence, $\text{sh}(i + 1, T)$ is defined after $\text{rem}(i, t_0)$ yielding

$$(g + g'|_{T'}\big|_{(T \cup \hat{T}) - \{t_0\}}(g'|_{T' - \hat{T}})\big|_{(T' - \hat{T}) - \{t_0\}}. \quad (12)$$

The two gates [11] and [12] are equal because

$$g|_{T - \{t_0\}} + g'|_{\hat{T}} = (g + g'|_{\hat{T}})\big|_{(T \cup \hat{T}) - \{t_0\}}$$

since $T \cap \hat{T} = \emptyset$ and $t_0 \notin \hat{T}$, and

$$(g'|_{T' - \{t_0\}})\big|_{(T' - \{t_0\}) - \hat{T}} = (g'|_{T' - \hat{T}})\big|_{(T' - \hat{T}) - \{t_0\}}$$

because $\overline{g|_{T_2}}|_{T_1} = \overline{g|_{T_1}}$ for any gate $\overline{g} = (\overline{T}, \overline{c}, \overline{n})$ with $T_1 \subseteq T_2 \subseteq T$ and $T - \{t_0\} - T' = (T' - \hat{T}) - \{t_0\}$.

(b) $T = \{t_0\} \neq T'$. Then one gets

$$g|_{T - \{t_0\}} + g'|_{\hat{T}} = (g + g'|_{\hat{T}})\big|_{(T \cup \hat{T}) - \{t_0\}}$$

$\text{rem}(i, t_0)$ is applicable after $\text{sh}(i + 1, \hat{T})$ using the same arguments as in Subcase 2a. In the result, $g$ disappears as $t_0$ is removed leaving $g'|_{\hat{T}}$ as first gate. The second gate is obtained like the second gate in Subcase 2a. Therefore, the result is a sequentialization of $g'|_{T' - \{t_0\}}$ wrt $\hat{T}$ so that the parallelization yields just this gate $g'|_{T' - \{t_0\}}$.
(c) There is no further subcase because \( t_0 \notin \hat{T} \subseteq T' \) so that \( T' = \{ t_0 \} \) cannot occur.

Case 3: The local Church-Rosser property for \( \text{op}_1 = \text{rem}(i, t_0) \) and \( \text{op}_2 = \text{sh}(i + 2, \hat{T}) \) is displayed in the following diagrams. The arguments for the applicability of further operations that lead to confluence are repetition of the arguments in Cases 1 and 2 and are not given in detail.

(a) \( \{ t_0 \} \neq T' \). Then one yields

\[
\begin{align*}
g'g'' & \xrightarrow{\text{rem}(i, t_0)} \frac{g|_{T - \{ t_0 \}}}{g|_{T' - \{ t_0 \}} g''} \\
sh(i + 2, \hat{T}) & \\
g' + g''|_{T'' - \hat{T}} & \xrightarrow{\text{rem}(i, \{ t_0 \})} \frac{g|_{T - \{ t_0 \}}}{g|_{T' - \{ t_0 \}} g''|_{T'' - \hat{T}}}
\end{align*}
\]

Again \( (g'|_{T' - \{ t_0 \}} + g''|_{\hat{T}}) = (g' + g''|_{\hat{T}})|_{(T' \cup \hat{T}) - \{ t_0 \}} \) because \( T' \cap \hat{T} = \emptyset \) and \( t_0 \notin \hat{T} \). If \( T = \{ t_0 \} \), then \( g|_{T - \{ t_0 \}} = \lambda \). But this does not change the reasoning with the assumption that the shift on the right side becomes \( \text{sh}(i + 1, \hat{T}) \). If \( T'' = \hat{T} \), then the same holds for \( g''|_{T'' - \hat{T}} = \lambda \).

(b) \( \{ t_0 \} = T' \) and \( T'' \neq \hat{T} \). Then one gets

\[
\begin{align*}
g'g'' & \xrightarrow{\text{rem}(i, t_0)} \frac{g|_{T' - \{ t_0 \}}}{g|_{T'' - \{ t_0 \}} g''} \\
sh(i + 2, \hat{T}) & \\
g' + g''|_{T'' - \hat{T}} & \xrightarrow{\text{rem}(i, \{ t_0 \})} \frac{g|_{T - \{ t_0 \}}}{g|_{T' - \{ t_0 \}} g''|_{T'' - \hat{T}}}
\end{align*}
\]

If \( T = \{ t_0 \} \), then the same construction works only that the right shift becomes \( \text{sh}(i + 1, T'' - \hat{T}) \).

(c) \( \{ t_0 \} = T' \) and \( T'' = \hat{T} \). Then one gets

\[
\begin{align*}
g'g'' & \xrightarrow{\text{rem}(i, t_0)} \frac{g|_{T' - \{ t_0 \}}}{g|_{T'' - \{ t_0 \}} g''} \\
sh(i + 2, \hat{T}) & \\
g' + g'' & \xrightarrow{\text{rem}(i, t_0)} \frac{g|_{T' - \{ t_0 \}}}{g|_{T'' - \{ t_0 \}} g''}
\end{align*}
\]

Case 4: As in Case 3, only the diagrams are presented that establish the local Church-Rosser property for \( \text{op}_1 = \text{rem}(i + 1, t_0) \) and \( \text{op}_2 = \text{sh}(i + 1, \hat{T}) \).
(a) $T' \neq \{t_0\} \neq T'', \hat{T} \neq T', t_0 \notin \hat{T}$.

$$gg'g'' \xrightarrow{\text{rem}(i+1, t_0)} gg'|_{T' - \{t_0\}}g''|_{T'' - \{t_0\}}$$

\[ sh(i + 1, \hat{T}) \]

\[ g + (g'|_{T' - \{t_0\}})|_{\hat{T} - \{t_0\}}(g'|_{T' - \{t_0\}})|_{(T' - \{t_0\}) - \hat{T}}g''|_{T'' - \{t_0\}} = \]

\[ g + g'|_{\hat{T}}g''|_{T' - \hat{T}}g'' \xrightarrow{\text{rem}(i, t_0)} g + g'|\hat{T} - T' + g''|_{\{t_0\}}g''|_{T'' - \{t_0\}} \]

In this case $(g'|_{T' - \{t_0\}})|_{\hat{T}} = g'|_{\hat{T}}$ because $\hat{T} \subseteq T' - \{t_0\}$ and by the same argument given in Subcase 2a, we can conclude that

$$(g'|_{T' - \{t_0\}})|_{(T' - \{t_0\}) - \hat{T}} = (g'|_{T' - \hat{T}})|_{(T' - \hat{T}) - \{t_0\}}.$$

(b) $T' \neq \{t_0\} \neq T'', \hat{T} \subset T'$, $t_0 \in \hat{T}$.

$$gg'g'' \xrightarrow{\text{rem}(i+1, t_0)} gg'|_{T' - \{t_0\}}g''|_{T'' - \{t_0\}}$$

\[ sh(i + 1, \hat{T} - \{t_0\}) \]

\[ g + (g'|_{T' - \{t_0\}})|_{\hat{T} - \{t_0\}}(g'|_{T' - \{t_0\}})|_{(T' - \{t_0\}) - \hat{T} - \{t_0\}}g''|_{T'' - \{t_0\}} = \]

\[ g + g'|_{\hat{T}}(T \cup \hat{T} - \{t_0\})(g'|_{T' - \hat{T}})|_{(T' - \hat{T}) - \{t_0\}}g''|_{T'' - \{t_0\}} \]

\[ \xrightarrow{\text{rem}(i, t_0)} g + g'|_{\hat{T}}g''|_{T' - \hat{T}} + g''|_{\{t_0\}}g''|_{T'' - \{t_0\}} \]

Here,

$$g + (g'|_{T' - \{t_0\}})|_{\hat{T} - \{t_0\}} = (g + g'|_{\hat{T}})|(T \cup \hat{T}) - \{t_0\}$$

because $\hat{T} \subseteq T'$ and

$$g + (g'|_{T' - \{t_0\}})|_{\hat{T} - \{t_0\}} = (g + (g'|_{T' - \{t_0\}}))|(T \cup \hat{T}) - \{t_0\}.$$

Further,

$$(g'|_{T' - \{t_0\}})|(T' - \{t_0\}) - (\hat{T} - \{t_0\}) = (g'|_{(T' - \hat{T}) - \{t_0\}})$$

because

$$(T' - \{t_0\}) \cap ((T' - \{t_0\}) - (\hat{T} - \{t_0\})) = (T' - \{t_0\}) \cap ((T' - \hat{T}) - \{t_0\})$$

$$= (T' - \hat{T}) \cap ((T' - \hat{T}) - \{t_0\}).$$
(c) $T' \neq \{t_0\} \neq T''$ and $T' = \hat{T}$.

$$
gg'g'' \xrightarrow{\text{rem}(i + 1, t_0)} gg'|_{T' - \{t_0\}}g''|_{T'' - \{t_0\}} \xrightarrow{\text{sh}(i + 1, \hat{T} - \{t_0\})} g + g'|_{T' - \{t_0\}}g''|_{T'' - \{t_0\}} = g + g'|_{(T \cup \hat{T}) - \{t_0\}}g''|_{T'' - \{t_0\}}
$$

$g + g'|_{T' - \{t_0\}} = (g + g')|_{(T \cup \hat{T}) - \{t_0\}}$ because $T' = \hat{T}$ and $\{t_0\} \notin T$ because $\text{sh}(i + 1, \hat{T})$ is defined by assumption.

(d) $T' = \{t_0\} \neq T''$, in particular $\hat{T} = \{t_0\}$.

$$
gg'g'' \xrightarrow{\text{rem}(i + 1, t_0)} gg'|_{T' - \{t_0\}} \xrightarrow{\text{rem}(i, t_0)} g + g'|_{T'' - \{t_0\}}
$$

(e) $T' \neq \{t_0\} = T''$ and $t_0 \notin \hat{T}$.

$$
gg'g'' \xrightarrow{\text{rem}(i + 1, t_0)} gg'|_{T' - \{t_0\}} \xrightarrow{\text{sh}(i + 1, \hat{T})} g + (g'|_{T' - \{t_0\}})|_{\hat{T}}(g'|_{T' - \{t_0\}})|_{(T' - \{t_0\}) - \hat{T}} = g + g'|_{T'(T' - \hat{T}) - \{t_0\}}g''|_{(T' - \hat{T}) - \{t_0\}}
$$

by the same arguments as in Case 4a.

(f) $T' \neq \{t_0\} = T''$ and $t_0 \in \hat{T} \neq T'$.

$$
gg'g'' \xrightarrow{\text{rem}(i + 1, t_0)} gg'|_{T' - \{t_0\}} \xrightarrow{\text{sh}(i + 1, \hat{T} - \{t_0\})} g + (g'|_{T' - \{t_0\}})|_{\hat{T} - \{t_0\}}(g'|_{T' - \{t_0\}})|_{(T' - \{t_0\}) - (\hat{T} - \{t_0\})} = (g + g'|_{\hat{T}})|_{(T \cup \hat{T}) - \{t_0\}}(g'|_{(T' - \hat{T}) \cup \{t_0\}})|_{((T' - \hat{T}) \cup \{t_0\}) - \{t_0\}}
$$

$g + g'|_{\hat{T}}g''|_{T' - \hat{T}} \xrightarrow{\text{sh}(i + 2, \{t_0\})} g + g'|_{\hat{T}}g''|_{(T' - \hat{T}) \cup \{t_0\}}$
Here, both
\[ g + (g'|T'-\{t_0\})|T-\{t_0\} = (g + g'|\hat{T})|T_0,T - \{t_0\}, \]
\[ (g'|T'-\{t_0\})|(T'-\{t_0\})-(T-\{t_0\}) = (g'|\hat{T} - T_0)|((T'-\hat{T})|T - \{t_0\}) - \{t_0\} \]

because \( \hat{T} \subseteq T' \).

(g) \( T' \neq \{t_0\} = T'' \) and \( t_0 \in \hat{T} = T' \).

\[
\begin{array}{c}
gg'g'' \xrightarrow{\text{rem}(i+1, t_0)} gg'|T'-\{t_0\} \\
\xrightarrow{sh(i+1, \hat{T})} g + (g'|T'-\{t_0\})|\hat{T}-\{t_0\} (g'|T'-\{t_0\})|(T'-\{t_0\})-(T-\{t_0\}) = \\
g + g'g'' \xrightarrow{\text{rem}(i+1, t_0)} (g + g')|(T \cup T') - \{t_0\} \\
\end{array}
\]

where both
\[ (g'|T'-\{t_0\})|(T'-\{t_0\})-(T-\{t_0\}) = g'|\emptyset = \lambda, \]
\[ g + (g'|T'-\{t_0\})|\hat{T}-\{t_0\} = g'|T'-\{t_0\} \]

because \( \hat{T} = T' \).

(h) \( T' = \{t_0\} = T'' \), in particular, \( \hat{T} = \{t_0\} \).

\[
\begin{array}{c}
gg'g'' \xrightarrow{\text{rem}(i+1, t_0)} g = (g + g')|(T \cup \{t_0\}) - \{t_0\} \\
\xrightarrow{sh(i+1, \hat{T})} \xrightarrow{\text{rem}(i, t_0)} g + g'g'' \\
\end{array}
\]

**Theorem 2.** \( sh \& \text{rem} \)-equivalent \( sh \& \text{rem} \)-reduced circuits are equal.

**Proof** The proof follows exactly the arguments in the proof of Theorem 1 using the following Lemma 2 instead of Lemma 1. Given two \( sh \& \text{rem} \)-equivalent circuits, then there are \( sh \& \text{rem} \)-sequences into a further circuit. But the reducedness implies that the sequences are empty so that all three involved circuits must be equal.

**Lemma 2.** \( mtc \approx \overline{mtc} \) implies \( \overline{mtc} \) for some multi-target Toffoli circuit \( \overline{mtc} \).

**Proof** The proof repeats the arguments in the proof of Lemma 1 replacing \( \rightarrow_{sh} \) by \( \rightarrow \) and using the local Church-Rosser property of \( \rightarrow \).
The fact that shifting as well as the combination of shifting and removal of two identical successive subsets yield unique reduced circuits is remarkable because the operations on circuits that decrease the waiting degree (or some other measures) may fail in this respect. For example, removal alone does not lead to unique reduced forms. Consider the circuit $mtc = mtg_1 \cdot mtg_2 \cdot mtg_3$ with $mtg_i = (T_i, cp_i, cn_i), T_i = \{t, t'\}$ for $i = 1, 2, 3$ and $cp_1(t) = cp_2(t) = cp_3(t), cn_1(t) = cn_2(t) = cn_3(t)$, but $cp_1(t') \neq cp_2(t') \neq cp_3(t')$. Then $t$ can be removed in two ways yielding $mtg_1|_i \cdot mtg_2|_i \cdot mtg_3$ as well as $mtg_1 \cdot mtg_2|_i \cdot mtg_3|_i$ which are obviously removal-reduced, but also different.

Instead of just erasing two identical successive subgates one can erase any sequence of subgates which yields the identity. These identities can be constructed in many different ways. Several of these identities have been discussed in literature (cf. e.g. [9, 11, 12, 13, 14, 15]). In the following some of them are stated.

**Example 6.** The following circuits yield the identity.

- $mtg_1 \cdot mtg_2 \cdot mtg_3$ with $T_1 = T_2 = T_3 = \{t_0\}, cp_1(t_0) = cn_2(t_0)$ and $cn_1(t_0) = cp_2(t_0) = cp_3(t_0) = cn_3(t_0) = \emptyset$.
- $mtg_1 \cdot \ldots \cdot mtg_5$ with $T_1 = T_3 = T_5 = \{t_0\}, T_2 = T_4 = \{t_1\}, t_0 \neq t_1,$
  $cn_1(t_0) = cp_5(t_0) = \{t_1\}, cn_2(t_1) = cp_4(t_1) = \{t_0\}$ and $cp_1(t_0) = cp_2(t_1) = cp_3(t_0) = cn_3(t_0) = cn_4(t_1) = cn_5(t_0)$.
- $mtg_1 \cdot \ldots \cdot mtg_5$ with $T_1 = T_3 = T_4 = \{t_0\}, T_2 = T_5 = \{t_1\}, t_0 \neq t_1,$
  $cp_1(t_0) = cp_4(t_0) = \{t_1\}$ and $cn_1(t_0) = cp_2(t_1) = cn_2(t_1) = cp_3(t_0) =
  cn_3(t_0) = cn_4(t_0) = cp_5(t_1) = cn_5(t_1) = \emptyset$. The same circuit except
  $cn_1(t_0) = cn_4(t_0) = \{t_1\}$ and $cp_1(t_0) = cp_4(t_0) = \emptyset$ also yields the identity.
- $mtg_1 \cdot \ldots \cdot mtg_6$ with $mtg_1 = mtg_3 = mtg_5, mtg_2 = mtg_4 = mtg_6, T_1 = \{t_0\}, T_2 = \{t_1\}, cp_1(t_0) = \{t_1\}, cp_2(t_1) = \{t_0\}$ and $cn_1(t_0) = cn_2(t_1) = \emptyset$.

The examples above consider only simple gates (single controlled gates with one target line and two target lines in total). However, obviously, the sets of identities with more than just two lines become very large.

These more general removals can be applied in the same way as the removal of the two equal successive subgates. Obviously these more general removals also decrease the waiting degree and terminate after finitely many steps. But is there an open problem whether the local Church-Rosser property holds in the case of more complex removals.

By a combinatorial argument one can show that, if the identity subsequence becomes more complex, it is unlikely to be in the circuit and, in general, not many of such sequences exist in a circuit.

Further, any sequence of subgates $mtc$ may be replaced by a different but shorter sequence of subgates $mtc'$ which is semantically equivalent to $mtc$, i.e. fulfills the properties $f_{mtc} \circ f_{mtc'} = id$ and $|mtc'| < |mtc|$, where $mtc''$ is the reverse circuit to $mtc'$. But as the test for semantic equivalence is NP-complete, it is not easy to find such replacements.
Concerning the quantitative gain of the combination of shift and removal, the remark at the end of Section 6 still holds in principle. Clearly, the gain becomes larger if one finds two negations with the same target line and the same control lines in gate \(i\) and gate \(i + 1\) because the removal already reduces the waiting degree by \(2i - 1\), and the resulting circuit is in a different shift equivalence class so that shifting can decrease the waiting degree further, but the limits 0 and \(\frac{m(m-1)}{2}\) do not change.

8. Conclusion

In this paper, we have studied the class of Toffoli circuits that are sequentially composed of multi-target Toffoli gates with mixed control. Under certain independence conditions parts of a gate can be shifted to the preceding gate within a circuit. It has turned out that shift-reduced circuits are unique canonical representatives of their shift equivalence classes. Canonical circuits have minimal waiting degree within their shift equivalence classes. Moreover, shifts have been combined with removals of identical successive subgates within a multi-target Toffoli circuit. Also in this case, the shift&removal-reduced circuits have been shown to be unique canonical representatives of their shift&removal-equivalence classes. To shed more light on the significance of these considerations, further research on the following topics may be helpful.

1. So far, our considerations are purely theoretical based on our knowledge of parallelism in rule-based systems. It will not be very difficult to implement shifting on circuits as considered so that simulation would be possible. Consequently, one gets a quantitative test of shift equivalence that implies equivalence in the positive case. It is an open problem whether our normal-form results can be of practical use.
2. In the literature on reversible functions and circuits several other gates occur such as the Fredkin gate or the Peres gate. Notions such as shift equivalence should also make sense for other types of gates.
3. As mentioned in the introduction, shifts on parallel graph grammar derivations behave like the shifts on multi-target Toffoli circuits (see, e.g., [6, 16]). Therefore, we wonder whether there is a way to represent Toffoli circuits as parallel derivations so that the canonical-form results for graph grammar derivations apply directly to reversible circuits without the need to mimic the proofs in the context of reversible circuits.

References


