Substitution and Flip BDDs

Rune M. Jensen and Henrik R. Andersen

IT University Technical Report Series

TR-2003-41

ISSN 1600–6100

December 2003

ISSN 1600–6100

ISBN 87-7949-055-7

Copies may be obtained by contacting:

IT University of Copenhagen
Glentevej 67
DK-2400 Copenhagen NV
Denmark

Telephone:	**+45 38 16 88 88**
Telefax:	**+45 38 16 88 99**
Web	**`www.itu.dk`**

Substitution and Flip BDDs

Rune M. Jensen and Henrik R. Andersen

Abstract

This report introduces two novel approaches for representing transition functions of finite transition systems encoded as Binary Decision Diagrams (BDDs). The first approach is *substition BDDs* where each transition is represented by a corresponding substitution on state variables. The second is *flip BDDs* where each transition is defined by the set of state variables with flipped value. We show that substitution BDDs can be used to find and propagate write conflicts in synchronous and asynchronous compositions. Furthermore, our experimental evaluation suggest that the complexity of image computations based on flip BDDs may compare positively to the usual relational product computation.

1 Introduction

Binary Decision Diagrams (BDDs,[2]) have been successfully applied to implicitly search large transition systems in a wide range of fields including formal verification, planning, control, and game theory (e.g., [3, 4, 1, 9, 5]). Two major approaches for representing the transition function have been developed. The first represents the transition function as a vector of Boolean functions where each function defines the value in the next state of a single state variable (e.g.,[8]). The approach, however, is limited to deterministic transition functions. The second approach is probably the most popular. It represents the transition function as the characteristic function of the transition relation [7]. It can represent deterministic as well as nondeterministic transition functions.

In this report, we introduce two novel transition function representations. Both can represent deterministic as well as nondeterministic transition functions. The first is called *substitution BDDs* and is based on representing each transition by a substitution on state variables. Since this representation is redundant, it allows write conflicts on state variables to be detected and propagated. The second approach is called *flip BDDs*. The idea is to represent each transition by the set of state variables it changes. This set is uniquely defined. Thus, flip BDDs are non-redundant. Our preliminary experimental evaluation indicates that image computations based on flip BDDs may have lower complexity than the relational product operation used to compute the image when the transition function is represented by the characteristic function of the transition relation.

The remainder of the report is organized as follows. In Section 2, we introduce a guarded command language and substitution BDDs for representing its semantics. We also define specialized BDD operators for making synchronous and asynchronous compositions of guarded commands and computing the image of a set of states given a transition system represented as a substitution BDD. In Section 3, we introduce flip BDDs. We show how to build a flip BDD from a transition system definition. In addition, we define the image operation for a transition systems represented by a flip BDD and present preliminary experimental results comparing the performance of fixed point computations based on flip BDDs and fixed point computations based on the usual relational product. Finally, we draw conclusions and discuss directions for future work in Section 4.

2 Substitution BDDs

Substitution BDDs represent synchronous and asynchronous compositions of guarded commands. The abstract syntax of the guarded command language is given by

COM

$$\begin{array}{rcl} a & ::= & g \rightarrow \vec{x} := \textbf{any}\ \vec{x}'.\, \varphi, \\ c & ::= & a \ \mid\ c \,||\, c \ \mid\ c * c. \end{array}$$

The operators $*$ and $||$ denote synchronous and asynchronous composition. The variables g and φ are Boolean expressions on a nonempty set of Boolean state variables, $\text{Var} = \{x_1, \cdots, x_n\}$ given by

$$\begin{array}{rcl} g & ::= & x \mid \textbf{true} \mid \textbf{false} \mid \neg g \mid g \wedge g, \\ \varphi & ::= & x \mid x' \mid \textbf{true} \mid \textbf{false} \mid \neg\varphi \mid \varphi \wedge \varphi, \quad \text{where} \quad x \in \text{Var}. \end{array}$$

An *atomic command* a is executable in a state that satisfies the expression g on the current state variables. If it executes, the k state variables in $\vec{x} = (x_{i(1)}, \cdots, x_{i(k)})$ change value to one of the assignments of $\vec{x}' = (x'_{i(1)}, \cdots, x'_{i(k)})$ satisfying φ such that $x_{i(1)} = x'_{i(1)}, \cdots, x_{i(k)} = x'_{i(k)}$ in the resulting state. We regard the set of possible assignments of the k state variables in $\vec{x}$ as substitutions. Example 1 shows four atomic commands.

Example 1 Four atomic commands are shown below

$$\begin{array}{rr} a_1: & x_1 \wedge x_2 \rightarrow x_1 := \textbf{any}\, x'_1.\, \neg x'_1, \\ a_2: & x_2 \wedge x_2 \rightarrow x_2 := \textbf{any}\, x'_2.\, \neg x'_2, \\ a_3: & x_1 \wedge x_2 \rightarrow (x_1, x_2) := \textbf{any}\, (x'_1, x'_2).\, \neg x'_1 \wedge x'_2, \\ a_4: & x_1 \wedge x_2 \rightarrow (x_1, x_2) := \textbf{any}\, (x'_1, x'_2).\, x'_1 \wedge x'_2. \end{array}$$

$\diamond$

In an asynchronous composition $c_1 \,||\, c_2$ of two guarded commands c_1 and c_2 at most one transition (or substitution) takes place in each time step. Thus, the substitutions of each state in the resulting command is equal to the union of substitutions for that state of c_1 and c_2. Synchronous composition $c_1 * c_2$ is more complicated, since a substitution in both c_1 and c_2 happens simultaneously. For a given state this means that any combination of substitutions of c_1 and c_2 can happen. In addition, substitutions may conflict by writing to the same state variable. We assume that a conflict occur even when the two substitutions agree on the new value of the variable. Obviously other models of synchronous composition could be used. In order to define the semantics of a guarded command formally, we first define a substitution as a partial mapping from variables to truth values or the error substitution, $\perp$

$$\text{Subst} = (\text{Var} \rightharpoonup \mathbb{B}) \cup \{\perp\}.$$

A command c denotes a domain $\mathcal{C}[\![c]\!] : \text{D}$, where D is a mapping from states to sets of substitutions

$$\begin{array}{rcl} \text{D} & = & \mathbb{B}^{|\text{Var}|} \rightarrow 2^{\text{Subst}} \\ & = & \mathbb{B}^{n} \rightarrow 2^{\text{Subst}} \end{array}$$

The semantic function $\mathcal{C}[\![c]\!]$ is defined by structural induction on the abstract syntax

$$\begin{array}{rcl} \mathcal{C}[\![g \rightarrow \vec{x} := \textbf{any}\, \vec{x}'.\, \varphi]\!]s & = & \{[x_1/b_1, \cdots, x_k/b_k] \mid\ \mathcal{B}[\![g]\!]s \wedge \mathcal{B}[\![\varphi]\!](s, \vec{b})\} \\ \mathcal{C}[\![c_1 \,||\, c_2]\!]s & = & \mathcal{C}[\![c_1]\!]s \cup \mathcal{C}[\![c_1]\!]s \\ \mathcal{C}[\![c_1 * c_2]\!]s & = & \{\sigma_1 + \sigma_2 \mid\ \sigma_1 \in \mathcal{C}[\![c_1]\!]s,\, \sigma_2 \in \mathcal{C}[\![c_2]\!]s\}, \ \text{where} \end{array}$$

$$\sigma_1 + \sigma_2 = \begin{cases} \perp & : \ dom(\sigma_1) \cap dom(\sigma_2) \neq \emptyset \\ \perp & : \ \sigma_1 = \perp \\ \perp & : \ \sigma_2 = \perp \\ \sigma_1 \cup \sigma_2 & : \ \text{otherwise.} \end{cases}$$

The function $\mathcal{C}[\![\psi]\!]$ defines the usual semantics of a Boolean expression ψ. Example 2 shows various compositions of the four atomic commands introduced in Example 1.

Example 2 The semantics of the four atomic substitutions presented in Example 1 and a few compositions are illustrated below

a_1 $\quad 11 \xrightarrow{[x_1/0]} 01$

a_2 $\quad 11 \xrightarrow{[x_2/0]} 10$

a_3 $\quad 11 \xrightarrow{[x_1/0, x_2/1]} 01$

a_4 $\quad$ $\begin{tikzcd} 11 \arrow[loop right, "{[x_1/1, x_2/1]}"] \end{tikzcd}$

$a_1 \,||\, a_2$ $\quad$ $\begin{tikzcd} 11 \arrow[r, "{[x_1/0]}"] \arrow[d, "{[x_2/0]}"] & 01 \\ 10 & \end{tikzcd}$

$a_1 \,||\, a_3$ $\quad 11 \xrightarrow[{[x_1/0, x_2/1]}]{[x_1/0]} 01$

$(a_1 \,||\, a_2) * a_4$ $\quad 11 \xrightarrow{\perp}$

$a_1 * a_3$ $\quad 11 \xrightarrow{\perp}$

$\diamond$

2.1 Domain Embedding

In order to use BDDs to represent the substitution domain of a guarded command, we embed D in a pair of Boolean functions

$$\begin{aligned} \mathrm{D} &= \mathbb{B}^{|\mathrm{Var}|} \to 2^{\mathrm{Subst}} \\ &= \mathbb{B}^n \to 2^{(\mathrm{Var} \rightharpoonup \mathbb{B}) \cup \{\perp\}} \\ &\cong \big(\mathbb{B}^n \to 2^{\mathrm{Var} \rightharpoonup \mathbb{B}}\big) \times \big(\mathbb{B}^{|\mathrm{Var}|} \to 2^{\{\perp\}}\big) \end{aligned}$$

$$\begin{aligned} \mathrm{D} &\hookrightarrow \big(\mathbb{B}^n \times \mathbb{B}^{|\mathrm{Var}|^2} \to \mathbb{B}\big) \times \big(\mathbb{B}^{|\mathrm{Var}|} \to \mathbb{B}\big) \\ &= \big(\mathbb{B}^{3n} \to \mathbb{B}\big) \times \big(\mathbb{B}^n \to \mathbb{B}\big). \end{aligned}$$

The first function encodes all the non-conflicting substitutions, while second encodes all the error substitutions. To define these functions, we introduce some auxiliary notation

- For a pair $p = (l, r)$, let $fst(p) = l$ and $snd(p) = r$,
- For a Boolean function $\psi(y_1, \cdots, y_m)$, let the variable names and domain of ψ be simultaneously defined by $\psi : \mathbb{B}^{\{y_1, \cdots, y_m\}}$,
- For a Boolean function $\psi : \mathbb{B}^{\{y_1, y_1', \cdots, y_m, y_m'\}}$ and an assignment of the arguments $t = (b_1, b_1', \cdots, b_m, b_m') \in \mathbb{B}^{2m}$, let $t(y_i) = b_i$, $t(y_i') = b_i'$, $t|_{\bar{y}} = (b_1, \cdots, b_m)$, and $t|_{\bar{y}'} = (b_1', \cdots, b_m')$.

A domain embedding is defined by the function ε

$$\varepsilon \;:\; \mathrm{D} \to (\mathbb{B}^{\mathrm{VAR}} \to \mathbb{B}) \times (\mathbb{B}^{\mathrm{Var}} \to \mathbb{B})$$

where $\mathrm{VAR} = \{x, x', x'' \mid x \in \mathrm{Var}\}$. Given a domain $d \in \mathrm{D}$, a substitution transition $t = (b_1, b_1', b_1'', \cdots, b_n, b_n', b_n'')$, and a state $s = (b_1, \cdots, b_n)$, we have

$$\begin{aligned} fst(\varepsilon(d))t &= \exists \sigma \in d(t|_{\bar{x}}) \setminus \{\perp\} \,.\, subval(\sigma, t) \wedge subdom(\sigma, t), \\ snd(\varepsilon(d))s &= \perp \in d(s), \quad \text{where} \end{aligned}$$

$$\begin{aligned} subval(\sigma, t) &= \forall x \in dom(\sigma) \,.\, \big(t(x') = \sigma(x)\big), \\ subdom(\sigma, t) &= \forall x \in \mathrm{Var} \,.\, \big(t(x'') = x \in dom(\sigma)\big). \end{aligned}$$

Thus, $snd(\varepsilon(d))$ and $fst(\varepsilon(d))$ encode a state with the $\vec{x}$ variables. In addition, $fst(\varepsilon(d))$ encodes the value of a substitution with the $\vec{x}'$ variables (see $subval$) and the set of variables in the substitution with the $\vec{x}''$ variables (see $subdom$).

2.2 The BDD Representation of the Embedding

Let the set of BDD variables equal VAR and assume without loss of generality that the variables are ordered ascendingly

$$x_1'' \prec x_1' \prec x_1 \prec \cdots \prec x_n'' \prec x_n' \prec x_n.$$

The BDD representation of a command c is given by the function

$$\tau(c) \;:\; (\mathbb{B}^{\mathrm{VAR}} \to \mathbb{B}) \times (\mathbb{B}^{\mathrm{Var}} \to \mathbb{B})$$

defined by

$$\begin{aligned}
\tau(g \to \vec{x} := \mathbf{any}\,\vec{x}' \,.\, \varphi) &= (\tilde{g}(\vec{x}) \wedge \tilde{\varphi}(\vec{x}, \vec{x}') \wedge x_1'' \wedge \cdots \wedge x_k'' \wedge \neg x_{k+1}'' \wedge \cdots \wedge \neg x_n'', \mathbf{0}) \\
\tau(c_1 \parallel c_2) &= (fst(\tau(c_1)) \vee fst(\tau(c_2)), snd(\tau(c_1)) \vee snd(\tau(c_2))) \\
\tau(c_1 * c_2) &= \tau(c_1) * \tau(c_2),
\end{aligned}$$

where $\tilde{g}(\vec{x})$ and $\tilde{\varphi}(\vec{x}, \vec{x}')$ are BDD representations of g and φ using $\vec{x}$ to encode the state and $\vec{x}'$ to encode the substitution values. The $*$ operation is defined by structural induction on the BDD representation. To simplify the definition, we assume that the BDD is non-reduced such that all nodes are present in the decision tree. Let $L = \tau(c_1)$ and $L = \tau(c_2)$.

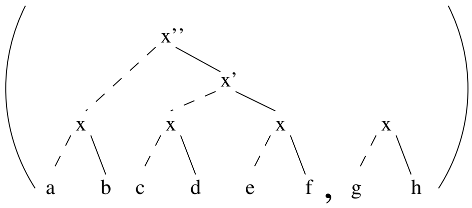
Base Case $(L, R \in \mathbb{B}^2)$

$$L * R = \Big(fst(L) \wedge fst(R),\; fst(L) \wedge snd(R) \vee snd(L) \wedge fst(R) \vee snd(L) \wedge snd(R)\Big)$$

Inductive Case

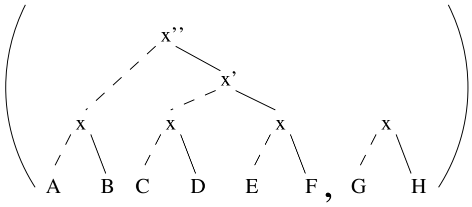
Let

L =



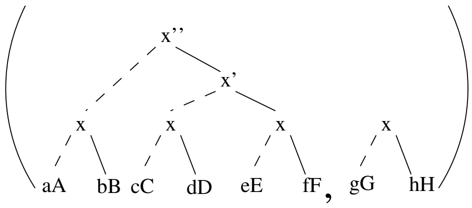
and

R =



We then have

L $*$ R =



where

$$\begin{array}{rcllllllll}
\text{aA} & = & \mathit{fst}\big((a,g) & * & (A,G)\big) \\
\text{bB} & = & \mathit{fst}\big((b,h) & * & (B,H)\big) \\
\text{cC} & = & \mathit{fst}\big((c,g) & * & (A,G)\big) & \vee & \mathit{fst}\big((a,g) & * & (C,G)\big) \\
\text{dD} & = & \mathit{fst}\big((d,h) & * & (B,H)\big) & \vee & \mathit{fst}\big((b,h) & * & (D,H)\big) \\
\text{eE} & = & \mathit{fst}\big((e,g) & * & (A,G)\big) & \vee & \mathit{fst}\big((a,g) & * & (E,G)\big) \\
\text{fF} & = & \mathit{fst}\big((f,h) & * & (B,H)\big) & \vee & \mathit{fst}\big((b,h) & * & (F,H)\big) \\
\text{gG} & = & \mathit{snd}\big((a,g) & * & (A,G)\big) & \vee & \mathit{snd}\big((c,g) & * & (A,G)\big) & \vee\ \mathit{snd}\big((e,g) * (A,G)\big) \ \vee \\
 & & \mathit{snd}\big((a,g) & * & (C,G)\big) & \vee & (c,g) & \otimes & (C,G) & \vee\ (e,g) \otimes (C,G) \ \vee \\
 & & \mathit{snd}\big((a,g) & * & (E,G)\big) & \vee & (c,g) & \otimes & (E,G) & \vee\ (e,g) \otimes (E,G) \\
\text{hH} & = & \mathit{snd}\big((b,h) & * & (B,H)\big) & \vee & \mathit{snd}\big((d,h) & * & (B,H)\big) & \vee\ \mathit{snd}\big((f,h) * (B,H)\big) \ \vee \\
 & & \mathit{snd}\big((b,h) & * & (D,H)\big) & \vee & (d,h) & \otimes & (D,H) & \vee\ (f,h) \otimes (D,H) \ \vee \\
 & & \mathit{snd}\big((b,h) & * & (F,H)\big) & \vee & (d,h) & \otimes & (F,H) & \vee\ (f,h) \otimes (F,H)
\end{array}$$

where

$$L \otimes R = \Big(\big(\exists \vec{x}'', \vec{x}' \,.\, \mathit{fst}(L)\big) \vee \mathit{snd}(L)\Big) \wedge \Big(\big(\exists \vec{x}'', \vec{x}' \,.\, \mathit{fst}(R)\big) \vee \mathit{snd}(R)\Big).$$ [1]

The definition of synchronous composition is complex due to the induction on a pair of BDDs instead of just a single BDD. The first BDD in the pair is easiest to understand. There are two symmetrically distinct cases

$$\begin{array}{rcl}
\text{aA} & = & \mathit{fst}\big((a,g) * (A,G)\big), \\
\text{cC} & = & \mathit{fst}\big((c,g) * (A,G)\big) \ \vee\ \mathit{fst}\big((a,g) * (C,G)\big).
\end{array}$$

The BDD aA represents the continuation of non-conflicting substitutions where no value is substituted for x, and x is false in the current state. This BDD is the first element of the synchronous composition of the subsystems represented by (a,g) and (A,G) of R and L since the substitutions represented by these systems are the only continuations of substitutions not containing x, when x is false. Similarly, cC represents the continuation of non-conflicting substitutions where x is assigned false. This can happen in two ways: L does not substitute on x and R assigns it to false, or L assigns x to false and R does not substitute on it. The second BDD represents the error states. There is one symmetrically distinct case

$$\begin{array}{rcllllllll}
\text{gG} & = & \mathit{snd}\big((a,g) & * & (A,G)\big) & \vee & \mathit{snd}\big((c,g) & * & (A,G)\big) & \vee\ \mathit{snd}\big((e,g) * (A,G)\big) \ \vee \\
 & & \mathit{snd}\big((a,g) & * & (C,G)\big) & \vee & (c,g) & \otimes & (C,G) & \vee\ (e,g) \otimes (C,G) \ \vee \\
 & & \mathit{snd}\big((a,g) & * & (E,G)\big) & \vee & (c,g) & \otimes & (E,G) & \vee\ (e,g) \otimes (E,G)
\end{array}$$

The five expressions on the form $\mathit{snd}(x * y)$ are the conflicts happening in the previous or remaining levels, while the four expressions on the form $x \otimes y$ are conflicts happening at the current level: L assigns x to true and R assigns x to true, L assigns x to true and R assigns x to false, L assigns x to false and R assigns x to true, and L assigns x to false and R assigns x to false. Due to the synchronous composition, $L \oplus R$ is the intersection of all the possible states reachable by L and R (notice that this includes error states). The correctness of the BDD implementation is proven in Appendix A.

Theorem 1 (Correctness) *For any command,* $c \in \mathbf{COM}$, $\varepsilon(\mathcal{C}[\![c]\!]) = \tau(c)$.
Proof: See Appendix A.

2.3 Image Computation

The $D \odot R$ operation computes the image of a domain represented by the embedding D of a set of states R represented in the usual way by a BDD. The image operation is undefined if D has a conflict substitution ($\perp$) for some state in R. The $\odot$ operation is defined by structural induction on D and R.

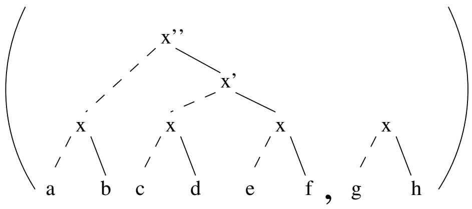
Base Case ($D \in \mathbb{B}^2, R \in \mathbb{B}$)

$$D \odot R = \begin{cases} \mathit{undefined} & D = (1,1) \textbf{ and } R = 1 \\ \mathit{fst}(D) \wedge R & \textbf{otherwise} \end{cases}$$

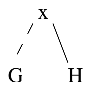
[1] $\otimes$ can also be defined inductively in the same way as $*$.

Inductive Case

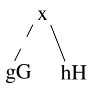
Let

D = 

and

R = 

We then have

$D \odot R =$ 

where

$$\begin{aligned} \text{gG} &= a \odot G \vee c \odot G \vee d \odot H \\ \text{hH} &= b \odot H \vee f \odot H \vee e \odot G \end{aligned}$$

The preimage computation can be defined in a similar way.

3 Flip BDDs

Flip BDDs are related to substitution BDDs except that there is a one-to-one correspondence between a flip BDD and the transition system it represents. The flip BDD representation has been developed to investigate the computational efficiency of the image computation based on this representation compared to the usual image computation based on the characteristic function of the transition relation.

As usual let $\mathrm{Var} = \{x_1, \cdots, x_n\}$ define a nonempty set of Boolean state variables. A possibly nondeterministic transition system on Var is defined by the pair $\langle Q, \delta \rangle$ where $Q = \mathbb{B}^n$ is the finite state space spanned by Var and $\delta : Q \to 2^Q$ is a transition function. A flip BDD f is a BDD on the ordered set of variables

$$x'_1 \prec x_1 \prec \cdots \prec x'_n \prec x_n$$

defined by

$$f(x_1, x'_1, \cdots, x_n, x'_n) = \exists s \in \delta(x_1, \cdots, x_n) \,.\, \forall x \in \mathrm{Var} \,.\, x' = (s(x) \neq s(x')).$$

Thus, an assignment to $x_1, x'_1, \cdots, x_n, x'_n$ uniquely defines a transition where the primed variables encode the set of variables that have their truth value flipped by the transition.

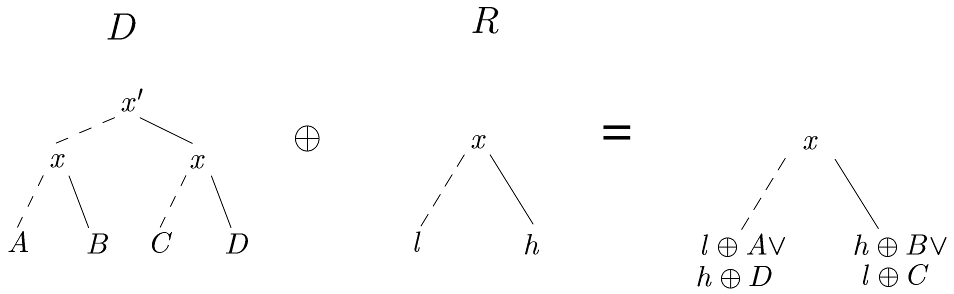
3.1 Image calculation

The $D \oplus R$ operation computes the image of a set of states. The set of states R is represented by a BDD in the usual way, while the transition system D is given as a flip BDD. The ordering of the variables in R is the same as the ordering of the pairs in D.

Base Case ($D, R \in \{1, 0\}$)

$$D \oplus R = R \wedge D$$

Inductive Case



3.2 Experimental Results

The following experiments are based on an implementation of $\oplus$ in the Buddy package [6]. The first experiment compares the CPU time of 10 fixed point calculations using flip BDDs and the relational product operator[2]. The Buddy package was initialized with 10000 BDD nodes and an operator cache varying from 10 to 100 percent of the number of BDD nodes. The transition system and the set of initial states are randomly generated using 10 state variables and 1000 transitions. The results are shown in table 1. As depicted in the table, the flip BDD image computations seem to

Cache size (%)	$T_{\oplus}$ (sec)	$T_{relprod}$ (sec)
10	0.43	0.73
20	0.24	0.30
30	0.21	0.18
40	0.08	0.19
50	0.02	0.05
60	0.02	0.03
70	0.02	0.03
80	0.01	0.02
90	0.01	0.04
100	0.01	0.04

Table 1: CPU time of 10 fixed point calculations as a function of operator cache size for flip BDD image computations ($T_{\oplus}$) and image computations based on relational products ($T_{relprod}$).

be less cache dependent than image computations based on the relational product. This is an encouraging result since cache efficiency is essential for keeping the complexity of BDD operations low. Moreover, these results were obtained with a naive experimental implementation of $\oplus$ that actually unfolds the BDDs on the fly to perform the computation. It is, however, surprising that the relational product operation has highest performance at 80% rather than 100%.

[2] As most other BDD packages, Buddy employs a highly optimized version of the relational product operation.

The second experiment compares the CPU time of 1000 fixed point calculations using flip BDDs and the relational product operator. The Buddy package was initialized to 100000 BDD nodes and the cache to 50000 BDD nodes. The transition system and the set of initial states are randomly generated using 10 state variables and 1000 to 10000 transitions. The results are shown in table 2. These results show some, but not an overwhelming, advantage of the flip

Transitions	$T_{\oplus}$ (sec)	$T_{relprod}$ (sec)
1000	0.02	0.04
2000	0.00	0.00
3000	0.11	0.14
4000	0.10	0.12
5000	0.07	0.09
6000	0.24	0.26
7000	0.09	0.30
8000	0.09	0.11
9000	0.09	0.11
10000	0.09	0.11

Table 2: CPU time of 1000 fixed point calculations as a function of number of transitions for flip BDD image computations ($T_{\oplus}$) and image computations based on relational products ($T_{relprod}$).

BDD approach. For 1000 randomly generated experiments, it is surprising that the problems with 6000 transitions are so much harder than the ones with 5000 transitions. Also, information about the cache miss rates for both approach would be valuable. It seems necessary to conduct further experiments to get a deeper insight in the performance differences between the two approaches. It is also a problem that the experiments are based solely on randomly generated transition systems. It is well known that there often is a significant difference between the performance on random and real-world problems for a particular approach.

4 Conclusion

In this report, we have introduced two novel BDD representations of transition systems called substitution and flip BDDs. The first representation is suitable for tracing variable assignment conflicts in synchronous and asynchronous system compositions. The second provides a promising alternative to the relational product operation for state space exploration. Future work includes developing efficient algorithms for manipulating these representations without unfolding the BDDs on the fly. In addition, more experiments are necessary comparing the efficiency of image computations based on flip BDDs and the relational product. These experiments should include both real-world and random problems.

Acknowledgments

Most of the work presented in this rapport was carried out in the summer 2000 and 2001 while the first author was visiting the IT University of Copenhagen. This work was supported in part by the Danish Research Agency.

References

[1] E. Asarin, O. Maler, and A. Pnueli. Symbolic controller synthesis for discrete and timed systems. In *Hybrid Systems*, pages 1–20, 1994.

[2] R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, 8:677–691, 1986.

[3] J. R. Burch, E. Clarke, D. L. Long, K. L McMillan, and D. L. Dill. Symbolic model checking for sequential circuit verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(4):401–424, 1994.

[4] A. Cimatti, M. Roveri, and P. Traverso. Automatic OBDD-based generation of universal plans in non-deterministic domains. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98)*, pages 875–881. AAAI Press, 1998.

[5] L. De Alfaro, Henzinger T. A., and O. Kupferman. Concurrent reachability games. In *IEEE Symposium on Foundations of Computer Science*, pages 564–575, 1998.

[6] J. Lind-Nielsen. BuDDy - A Binary Decision Diagram Package. Technical Report IT-TR: 1999-028, Institute of Information Technology, Technical University of Denmark, 1999. `http://cs.it.dtu.dk/buddy`.

[7] K. L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publ., 1993.

[8] C. Meinel and T. Theobald. *Algorithms and Data Structures in VLSI Design*. Springer, 1998.

[9] A. Vahidi, B. Lennartson, D. Arkeryd, and M. Fabian. Efficient application of symbolic tools for resource booking problems. In *Proceedings of the American Control Conference*, 2001.

A Proof of Correctness

First we introduce some auxiliary definitions and lemmas.

Lemma 1 ($\otimes$)

$$\big(\varepsilon(\mathcal{C}[\![c_1]\!]) \otimes \varepsilon(\mathcal{C}[\![c_1]\!])\big)s \;=\; \mathcal{C}[\![c_1]\!]s \neq \emptyset \wedge \mathcal{C}[\![c_1]\!]s \neq \emptyset$$

Proof:

$$\begin{array}{ll} & \big(\varepsilon(\mathcal{C}[\![c_1]\!]) \otimes \varepsilon(\mathcal{C}[\![c_1]\!])\big)s \\ = & \Big(\big(\exists \vec{x}'', \vec{x}' \,.\, fst(\varepsilon(\mathcal{C}[\![c_1]\!]))\big)s \vee snd(\varepsilon(\mathcal{C}[\![c_1]\!]))s\Big) \wedge \\ & \Big(\big(\exists \vec{x}'', \vec{x}' \,.\, fst(\varepsilon(\mathcal{C}[\![c_2]\!]))\big)s \vee snd(\varepsilon(\mathcal{C}[\![c_2]\!]))s\Big) \\ & \qquad \text{due to def. of } \otimes \\ = & \mathcal{C}[\![c_1]\!]s \neq \emptyset \wedge \mathcal{C}[\![c_1]\!]s \neq \emptyset \\ & \qquad \text{due to def. of } \varepsilon \end{array}$$

$\square$

Definition 1 ($\tau(c)|$)

$$\begin{array}{ll} & \tau(c)|_{\alpha(y''),\beta(y'),\gamma(y)} \\ = & \Big(\lambda t \in \mathbb{B}^{\mathit{VAR}\setminus\{y'',y',y\}} \,.\, fst(\tau(c))(t \cup \{\alpha(t(y'')), \beta(t(y')), \gamma(t(y))\}), \\ & \lambda s \in \mathbb{B}^{\mathit{Var}\setminus\{y\}} \,.\, snd(\tau(c))(s \cup \{\gamma(s(y))\})\Big) \end{array}$$

Definition 2 ($\varepsilon(\mathcal{C}[\![c]\!])|$)

$$\begin{array}{ll} & \varepsilon(\mathcal{C}[\![c]\!])|_{\alpha(y''),\beta(y'),\gamma(y)} \\ = & \Big(\lambda t \in \mathbb{B}^{\mathit{VAR}\setminus\{y'',y',y\}} \,.\, fst(\varepsilon(\mathcal{C}[\![c]\!]))(t \cup \{\alpha(t(y'')), \beta(t(y')), \gamma(t(y))\}), \\ & \lambda s \in \mathbb{B}^{\mathit{Var}\setminus\{y\}} \,.\, snd(\varepsilon(\mathcal{C}[\![c]\!]))(s \cup \{\gamma(s(y))\})\Big) \end{array}$$

Definition 3 (aA)

$$\mathcal{C}[\![c_{(\neg y'',\neg y)}]\!] = \lambda s \in \mathbb{B}^{\mathit{Var}\setminus\{y\}} \,.\, \{\sigma|\ \sigma \in \mathcal{C}[\![c]\!](s \cup \{\neg s(y)\}) \wedge y \notin dom(\sigma)\}$$

Lemma 2 (aA)

$$\varepsilon(\mathcal{C}[\![c]\!])|_{\neg y'',\neg y} = \varepsilon(\mathcal{C}[\![c_{(\neg y'',\neg y)}]\!])$$

Proof:

$$\begin{array}{ll} & fst(\varepsilon(\mathcal{C}[\![c]\!])|_{\neg y'',\neg y})t,\;\; t \in \mathbb{B}^{\mathrm{VAR}\setminus\{y'',y',y\}} \\ = & fst(\tau(c))(t \cup \{\neg t(y''), \neg t(y)\}) \\ & \qquad \text{due to def. } \varepsilon(\mathcal{C}[\![c]\!])| \\ = & \exists \sigma \in \mathcal{C}[\![c]\!]t_{\bar{x}\cup\{\neg t(y)\}} \setminus \{\bot\} \,. \\ & \big((\forall x \in dom(\sigma) \,.\, t(x') = \sigma(x)) \wedge \\ & \forall x \in \mathrm{Var} \,.\, t(x'') = x \in dom(\sigma)\big) \wedge \\ & \neg t(y'') \end{array}$$

$$
\begin{aligned}
& \quad \text{due to def. of } \varepsilon \\
= \;& \exists \sigma \in \mathcal{C}[\![c]\!] t_{\bar{x} \cup \{\neg t(y)\}} \setminus \{\perp\} \,. \\
& \big((\forall x \in dom(\sigma) \,.\, t(x') = \sigma(x)) \wedge \\
& \forall x \in \text{Var} \setminus \{y\} \,.\, t(x'') = x \in dom(\sigma)\big) \wedge \\
& y \notin dom(\sigma) \\
& \quad \text{logic} \\
= \;& \exists \sigma \in \{\sigma | \mathcal{C}[\![c]\!] t_{\bar{x} \cup \{\neg t(y)\}} \wedge y \notin dom(\sigma)\} \setminus \{\perp\} \,. \\
& \big((\forall x \in dom(\sigma) \,.\, t(x') = \sigma(x)) \wedge \\
& \forall x \in \text{Var} \setminus \{y\} \,.\, t(x'') = x \in dom(\sigma)\big) \\
& \quad \text{logic} \\
= \;& fst(\varepsilon(\mathcal{C}[\![c_{(\neg y'', \neg y)}]\!]))t \\
& \quad \text{due to def. aA}
\end{aligned}
$$

$$
\begin{aligned}
& snd(\varepsilon(\mathcal{C}[\![c]\!])|_{\neg y'', \neg y})e, \;\; s \in \mathbb{B}^{\text{Var} \setminus \{y\}} \\
= \;& snd(\varepsilon(\mathcal{C}[\![c]\!]))(s \cup \{\neg s(y)\}) \\
& \quad \text{due to def. } \varepsilon(\mathcal{C}[\![c]\!])| \\
= \;& \perp \in \{\sigma | \sigma \in \mathcal{C}[\![c]\!](s \cup \{\neg s(y)\})\} \\
& \quad \text{due to def. of } \varepsilon \\
= \;& \perp \in \{\sigma | \sigma \in \mathcal{C}[\![c]\!](s \cup \{\neg s(y)\}) \wedge y \notin dom(\sigma)\} \\
& \quad \text{logic} \\
= \;& snd(\varepsilon(\mathcal{C}[\![c_{(\neg y'', \neg y)}]\!])) \\
& \quad \text{due to def. aA}
\end{aligned}
$$

$\square$

Definition 4 (cC)

$$\mathcal{C}[\![c_(y'', \neg y', \neg y)]\!] = \lambda s \in \mathbb{B}^{Var \setminus \{y\}} \,.\, \{\sigma | \; \sigma \in \mathcal{C}[\![c]\!](s \cup \{\neg s(y)\}) \wedge ([y/0] \in \sigma \vee \sigma = \perp)\}$$

Lemma 3 (cC)

$$\varepsilon(\mathcal{C}[\![c]\!])|_{y'', \neg y', \neg y} = \varepsilon(\mathcal{C}[\![c_{(y'', \neg y', \neg y)}]\!])$$

Proof:

$$
\begin{aligned}
& fst(\varepsilon(\mathcal{C}[\![c]\!])|_{y'', \neg y', \neg y})t, \;\; t \in \mathbb{B}^{\text{VAR} \setminus \{y'', y', y\}} \\
= \;& fst(\varepsilon(\mathcal{C}[\![c]\!]))(t \cup \{t(y''), \neg t(y'), \neg t(y)\}) \\
& \quad \text{due to def. } \varepsilon(\mathcal{C}[\![c]\!])| \\
= \;& \exists \sigma \in \mathcal{C}[\![c]\!] t_{\bar{x} \cup \{\neg t(y)\}} \setminus \{\perp\} \,. \\
& \big((\forall x \in dom(\sigma) \,.\, t(x') = \sigma(x)) \wedge \\
& \forall x \in \text{Var} \,.\, t(x'') = x \in dom(\sigma)\big) \wedge \\
& t(y'') \wedge \neg t(y') \\
& \quad \text{due to def. of } \varepsilon \\
= \;& \exists \sigma \in \mathcal{C}[\![c]\!] t_{\bar{x} \cup \{\neg t(y)\}} \setminus \{\perp\} \,. \\
& \big((\forall x \in dom(\sigma) \setminus \{y\} \,.\, t(x') = \sigma(x)) \wedge
\end{aligned}
$$

$$\forall x \in \mathrm{Var} \setminus \{y\} \,.\, t(x'') = x \in dom(\sigma)\big) \wedge$$
$$y \in dom(\sigma) \wedge \neg\sigma(y)$$
logic
$$= \exists \sigma \in \{\sigma | \mathcal{C}[\![c]\!] t_{\bar{x} \cup \{\neg t(y)\}} \wedge [y/0] \in \sigma\} \setminus \{\perp\} \,.$$
$$\big((\forall x \in dom(\sigma) \setminus \{y\} \,.\, t(x') = \sigma(x)) \wedge$$
$$\forall x \in \mathrm{Var} \setminus \{y\} \,.\, t(x'') = x \in dom(\sigma)\big)$$
logic
$$= \exists \sigma \in \{\sigma | \mathcal{C}[\![c]\!] t_{\bar{x} \cup \{\neg t(y)\}} \wedge ([y/0] \in \sigma \vee \sigma = \perp)\} \setminus \{\perp\} \,.$$
$$\big((\forall x \in dom(\sigma) \setminus \{y\} \,.\, t(x') = \sigma(x)) \wedge$$
$$\forall x \in \mathrm{Var} \setminus \{y\} \,.\, t(x'') = x \in dom(\sigma)\big)$$
logic
$$= fst(\varepsilon(\mathcal{C}[\![c_{(y'', \neg y', \neg y)}]\!]))t$$
due to def. cC

$$snd(\varepsilon(\mathcal{C}[\![c]\!])|_{y'', \neg y', \neg y})e, \;\; s \in \mathbb{B}^{\mathrm{Var} \setminus \{y\}}$$
$$= snd(\varepsilon(\mathcal{C}[\![c]\!]))(s \cup \{\neg s(y)\})$$
due to def. $\varepsilon(\mathcal{C}[\![c]\!])|$
$$= \perp \in \{\sigma | \sigma \in \mathcal{C}[\![c]\!](s \cup \{\neg s(y)\})\}$$
due to def. of ε
$$= \perp \in \{\sigma | \sigma \in \mathcal{C}[\![c]\!](s \cup \{\neg s(y)\}) \wedge ([y/0] \in \sigma \vee \sigma = \perp)\}$$
logic
$$= snd(\varepsilon(\mathcal{C}[\![c_{(y'', \neg y', \neg y)}]\!]))$$
due to def. cC

□

Definition 5 (eE)

$$\mathcal{C}[\![c_(y'', y', \neg y)]\!] = \lambda s \in \mathbb{B}^{Var \setminus \{y\}} \,.\, \{\sigma |\; \sigma \in \mathcal{C}[\![c]\!](s \cup \{\neg s(y)\}) \wedge ([y/1] \in \sigma \vee \sigma = \perp)\}$$

Lemma 4 (eE)

$$\varepsilon(\mathcal{C}[\![c]\!])|_{y'', y', \neg y} = \varepsilon(\mathcal{C}[\![c_{(y'', y', \neg y)}]\!])$$

Proof: Symmetric to the proof of lemma cC □

We are now ready to prove the main theorem

Theorem 1 (Correctness) *For any command,* $c \in \mathbf{COM}$*,* $\varepsilon(\mathcal{C}[\![c]\!]) = \tau(c)$.

Proof: We prove by structural induction on c.

Basis

In the base case, c is an atomic command

$$c = a = g \rightarrow \vec{x} := \mathbf{any}\; \vec{x}' . \, \varphi.$$

We have

$$
\begin{array}{rl}
& fst(\varepsilon(\mathcal{C}[\![a]\!]))t \\
= & \exists \sigma \in \{[x_1/x'_1, \cdots, x_k/x'_k] \mid \ \mathcal{B}[\![g(t|_{\bar{x}})]\!] \wedge \mathcal{B}[\![\varphi(t|_{\bar{x}}, t|_{\bar{x}'})]\!]\} \, . \\
& \big((\forall x \in dom(\sigma) \, . \, t(x') = \sigma(x)) \wedge \\
& \forall x \in \mathrm{Var} \, . \, t(x'') = x \in dom(\sigma)\big) \\
& \quad \text{due to def. of } \varepsilon \text{ and } \mathcal{C} \\
= & (\tilde{g}(\vec{x}) \wedge \tilde{\varphi}(\vec{x}, \vec{x}') \wedge x''_1 \wedge \cdots \wedge x''_k \wedge \neg x''_{k+1} \wedge \cdots \wedge \neg x''_n)t \\
& \quad \text{logic} \\
= & fst(\tau(a))t \\
& \quad \text{due to def. of } \tau
\end{array}
$$

and

$$
\begin{array}{rl}
& snd(\varepsilon(\mathcal{C}[\![a]\!]))s \\
= & \perp \in \mathcal{C}[\![a]\!]s \\
& \quad \text{due to def. of } \varepsilon \\
= & (\mathbf{0})s \\
& \quad \text{logic} \\
= & snd(\tau(a))s \\
& \quad \text{due to def. of } \tau
\end{array}
$$

Thus, $\varepsilon(\mathcal{C}[\![a]\!]) = \tau(a)$.

Inductive Cases ($c = c_1 \mid\mid \ c_2$ and $c = c_1 * c_2$)

The structural induction hypothesis (IH I) is

$$
\begin{array}{rcl}
\tau(c_1) & = & \varepsilon(\mathcal{C}[\![c_1]\!]) \\
\tau(c_2) & = & \varepsilon(\mathcal{C}[\![c_2]\!])
\end{array}
$$

I. Assume $c = c_1 \mid\mid \ c_2$.

We have

$$
\begin{array}{rl}
& fst(\varepsilon(\mathcal{C}[\![c_1 \mid\mid \ c_2]\!]))t \\
= & \exists \sigma \in \mathcal{C}[\![c_1]\!]t|_{\bar{x}} \cup \mathcal{C}[\![c_2]\!]t|_{\bar{x}} \setminus \{\perp\} \, . \\
& \big((\forall x \in dom(\sigma) \, . \, t(x') = \sigma(x)) \wedge \\
& \forall x \in \mathrm{Var} \, . \, t(x'') = x \in dom(\sigma)\big) \\
& \quad \text{due to def. of } \varepsilon \text{ and } \mathcal{C} \\
= & \exists \sigma \in \mathcal{C}[\![c_1]\!]t|_{\bar{x}} \setminus \{\perp\} \, . \\
& \big((\forall x \in dom(\sigma) \, . \, t(x') = \sigma(x)) \wedge
\end{array}
$$

$$
\begin{array}{ll}
& \forall x \in \mathrm{Var} \,.\, t(x'') = x \in dom(\sigma)\big) \\
& \vee \\
& \exists \sigma \in \mathcal{C}[\![c_2]\!]t|_{\bar{x}} \setminus \{\perp\} \,. \\
& \big((\forall x \in dom(\sigma) \,.\, t(x') = \sigma(x)) \wedge \\
& \forall x \in \mathrm{Var} \,.\, t(x'') = x \in dom(\sigma)\big) \\
& \quad \text{logic} \\
= & fst(\tau(c_1))t \vee fst(\tau(c_2))t \\
& \quad \text{due to IH I} \\
= & fst(\tau(c_1 \,||\, c_2))t \\
& \quad \text{due to def. of } \tau
\end{array}
$$

and

$$
\begin{array}{ll}
& snd(\varepsilon(\mathcal{C}[\![c_1 \,||\, c_2]\!]))s \\
= & \perp \in \mathcal{C}[\![c_1]\!]s \cup \mathcal{C}[\![c_2]\!]s \\
& \quad \text{due to def. of } \varepsilon \text{ and } \mathcal{C} \\
= & \perp \in \mathcal{C}[\![c_1]\!]s \vee \perp \in \mathcal{C}[\![c_2]\!]e \\
& \quad \text{logic} \\
= & snd(\tau(c_1))s \vee snd(\tau(c_2))s \\
& \quad \text{due to IH I} \\
= & snd(\tau(c_1 \,||\, c_2))s \\
& \quad \text{due to def. of } \tau
\end{array}
$$

Thus, $\varepsilon(\mathcal{C}[\![c_1 \,||\, c_2]\!]) = \tau(c_1 \,||\, c_2)$.

II. Assume $c = c_1 \,*\, c_2$.

Consider the BDD, $\tilde{c}$, representing the command c,

$$\tilde{c} = \tau(c).$$

Without loss of generality we assume that $\tilde{c}$ is a non-reduced BDD. Let $|\tilde{c}|$ equal the number of (x'', x', x) levels $\tilde{c}$. Since $\tilde{c}$ is assumed to be non-reduced, we have

$$|\tau(c)| = |\mathrm{Var}|.$$

Consider the property $p(n)$ with the definition

$$p(n) \Leftrightarrow_{\mathrm{def}} \tau(c_1 * c_2) = \varepsilon(\mathcal{C}[\![c_1 * c_2]\!]) \text{ for } |\tau(c_1 * c_2)| \; = n.$$

We want to prove that $p(n)$ holds for all $n > 0$. We do this by induction on n.

Basis $(n = 1)$

Assume $|\tau(c_1 * c_2)| = 1$ and $\mathrm{Var} = \{x\}$. First we prove

$$fst(\tau(c_1 * c_2)) = fst(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!])).$$

Case $t = (0,0,0)$

$$
\begin{array}{ll}
& fst(\tau(c_1 * c_2))(0,0,0) \\
= & fst(\tau(c_1) * \tau(c_2))(0,0,0) \\
& \quad \text{due to def. of } \tau \\
= & fst(\varepsilon(\mathcal{C}[\![c_1]\!]) * \varepsilon(\mathcal{C}[\![c_2]\!]))(0,0,0) \\
& \quad \text{due to IH I} \\
= & fst(\varepsilon(\mathcal{C}[\![c_1]\!]))(0,\_,0) \wedge fst(\varepsilon(\mathcal{C}[\![c_2]\!]))(0,\_,0) \\
& \quad \text{due to base case def. of } * \\
= & (\exists\sigma \in \mathcal{C}[\![c_1]\!](0) \setminus \{\bot\} \,.\, x \notin dom(\sigma)) \wedge (\exists\sigma \in \mathcal{C}[\![c_2]\!](0) \setminus \{\bot\} \,.\, x \notin dom(\sigma)) \\
& \quad \text{due to def. of } \varepsilon \\
= & \exists\sigma \in \{\sigma_1 + \sigma_2 |\; \sigma_1 \in \mathcal{C}[\![c_1]\!](0), \sigma_2 \in \mathcal{C}[\![c_2]\!](0)\} \setminus \{\bot\}. x \notin dom(\sigma) \\
& \quad \text{due to def. of } + \\
= & \exists\sigma \in \mathcal{C}[\![c_1 * c_2]\!]) t|_{\bar{x}} \setminus \{\bot\} \,.\, \big((\forall x \in dom(\sigma) \,.\, t(x') = \sigma(x)) \wedge \\
& \forall x \in \mathrm{Var} \,.\, t(x'') = x \in dom(\sigma)\big), \text{ for } t = (0,0,0) \\
& \quad \text{logic} \\
& fst(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!]))(0,0,0) \\
& \quad \text{due to def. of } \varepsilon
\end{array}
$$

Case $t = (0,0,1), t = (0,1,0)$ and $t = (0,1,1)$
Symmetric to case $t = (0,0,0)$

Case $t = (1,0,0)$

$$
\begin{array}{ll}
& fst(\tau(c_1 * c_2))(1,0,0) \\
= & fst(\varepsilon(\mathcal{C}[\![c_1]\!]) * \varepsilon(\mathcal{C}[\![c_2]\!]))(1,0,0) \\
& \qquad \text{using the same steps as in case } t = (0,0,0) \\
= & fst(\varepsilon(\mathcal{C}[\![c_1]\!]))(1,0,0) \wedge fst\varepsilon(\mathcal{C}[\![c_2]\!]))(0,,0) \vee \\
& fst(\varepsilon(\mathcal{C}[\![c_1]\!]))(0,,0) \wedge fst\varepsilon(\mathcal{C}[\![c_2]\!]))(1,0,0) \\
& \qquad \text{due to base case def. of } * \\
= & (\exists\sigma \in \mathcal{C}[\![c_1]\!](0) \setminus \{\bot\} \,.\, [x/0] \in \sigma) \wedge (\exists\sigma \in \mathcal{C}[\![c_2]\!](0) \setminus \{\bot\} \,.\, x \notin dom(\sigma)) \vee \\
& (\exists\sigma \in \mathcal{C}[\![c_1]\!](0) \setminus \{\bot\} \,.\, x \notin dom(\sigma)) \wedge (\exists\sigma \in \mathcal{C}[\![c_2]\!](0) \setminus \{\bot\} \,.\, [x/0] \in \sigma) \\
& \qquad \text{due to def. of } \varepsilon \\
= & \exists\sigma \in \{\sigma_1 + \sigma_2 |\; \sigma_1 \in \mathcal{C}[\![c_1]\!](0), \sigma_2 \in \mathcal{C}[\![c_2]\!](0)\} \setminus \{\bot\} \,.\, [x/0] \in \sigma \\
& \qquad \text{due to def. of } + \\
= & \exists\sigma \in \mathcal{C}[\![c_1 * c_2]\!]) t|_{\bar{x}} \setminus \{\bot\} \,.\, \big((\forall x \in dom(\sigma) \,.\, t(x') = \sigma(x)) \wedge \\
& \forall x \in \mathrm{Var} \,.\, t(x'') = x \in dom(\sigma)\big), \text{ for } t = (1,0,0) \\
& \qquad \text{logic} \\
& fst(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!]))(1,0,0) \\
& \qquad \text{due to def. of } \varepsilon
\end{array}
$$

Case $t = (1,0,1), t = (1,1,0)$ and $t = (1,1,1)$
Symmetric to case $t = (1,0,0)$

We then prove

$$snd(\tau(c_1 * c_2)) = snd(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!]))$$

Case $s = (0)$

$$\begin{array}{rl}
& snd(\tau(c_1 * c_2))(0) \\
= & snd(\varepsilon(\mathcal{C}[\![c_1]\!]) * \varepsilon(\mathcal{C}[\![c_2]\!]))(0) \\
& \quad \text{using the same steps as in case } t = (0,0,0) \\
= & fst(\varepsilon(\mathcal{C}[\![c_1]\!]))(0,_,0) \wedge snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0) \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0) \wedge fst(\varepsilon(\mathcal{C}[\![c_2]\!]))(0,_,0) \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0) \wedge snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0) \vee \\
& fst(\varepsilon(\mathcal{C}[\![c_1]\!]))(1,0,0) \wedge snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0) \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0) \wedge fst(\varepsilon(\mathcal{C}[\![c_2]\!]))(0,_,0) \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0) \wedge snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0) \vee \\
& fst(\varepsilon(\mathcal{C}[\![c_1]\!]))(1,1,0) \wedge snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0) \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0) \wedge fst(\varepsilon(\mathcal{C}[\![c_2]\!]))(0,_,0) \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0) \wedge snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0) \vee \\
& fst(\varepsilon(\mathcal{C}[\![c_1]\!]))(0,_,0) \wedge snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0) \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0) \wedge fst(\varepsilon(\mathcal{C}[\![c_2]\!]))(1,0,0) \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0) \wedge snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0) \vee \\
& fst(\varepsilon(\mathcal{C}[\![c_1]\!]))(0,_,0) \wedge snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0) \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0) \wedge fst(\varepsilon(\mathcal{C}[\![c_2]\!]))(1,1,0) \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0) \wedge snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0) \vee \\
& \big((\exists x'', x' \,.\, fst(\varepsilon(\mathcal{C}[\![c_1]\!]))(1,0,0)) \vee snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0)\big) \wedge \\
& \big((\exists x'', x' \,.\, fst(\varepsilon(\mathcal{C}[\![c_2]\!]))(1,0,0)) \vee snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0)\big) \vee \\
& \big((\exists x'', x' \,.\, fst(\varepsilon(\mathcal{C}[\![c_1]\!]))(1,1,0)) \vee snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0)\big) \wedge \\
& \big((\exists x'', x' \,.\, fst(\varepsilon(\mathcal{C}[\![c_2]\!]))(1,0,0)) \vee snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0)\big) \vee \\
& \big((\exists x'', x' \,.\, fst(\varepsilon(\mathcal{C}[\![c_1]\!]))(1,0,0)) \vee snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0)\big) \wedge \\
& \big((\exists x'', x' \,.\, fst(\varepsilon(\mathcal{C}[\![c_2]\!]))(1,1,0)) \vee snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0)\big) \vee \\
& \big((\exists x'', x' \,.\, fst(\varepsilon(\mathcal{C}[\![c_1]\!]))(1,1,0)) \vee snd(\varepsilon(\mathcal{C}[\![c_1]\!]))(0)\big) \wedge \\
& \big((\exists x'', x' \,.\, fst(\varepsilon(\mathcal{C}[\![c_2]\!]))(1,1,0)) \vee snd(\varepsilon(\mathcal{C}[\![c_2]\!]))(0)\big) \\
& \quad \text{due to base case def. of } * \\
= & (\exists \sigma \in \mathcal{C}[\![c_1]\!](0) \setminus \{\bot\} \,.\, x \notin dom(\sigma)) \wedge \bot \in \mathcal{C}[\![c_2]\!](0) \vee \\
& \bot \in \mathcal{C}[\![c_1]\!](0) \wedge (\exists \sigma \in \mathcal{C}[\![c_2]\!](0) \setminus \{\bot\} \,.\, x \notin dom(\sigma)) \vee \\
& \bot \in \mathcal{C}[\![c_1]\!](0) \wedge \bot \in \mathcal{C}[\![c_2]\!](0) \vee \\
& (\exists \sigma \in \mathcal{C}[\![c_1]\!](0) \setminus \{\bot\} \,.\, [x/0] \in \sigma) \wedge \bot \in \mathcal{C}[\![c_2]\!](0) \vee \\
& \bot \in \mathcal{C}[\![c_1]\!](0) \wedge (\exists \sigma \in \mathcal{C}[\![c_2]\!](0) \setminus \{\bot\} \,.\, x \notin dom(\sigma)) \vee \\
& \bot \in \mathcal{C}[\![c_1]\!](0) \wedge \bot \in \mathcal{C}[\![c_2]\!](0) \vee \\
& (\exists \sigma \in \mathcal{C}[\![c_1]\!](0) \setminus \{\bot\} \,.\, [x/1] \in \sigma) \wedge \bot \in \mathcal{C}[\![c_2]\!](0) \vee \\
& \bot \in \mathcal{C}[\![c_1]\!](0) \wedge (\exists \sigma \in \mathcal{C}[\![c_2]\!](0) \setminus \{\bot\} \,.\, x \notin dom(\sigma)) \vee \\
& \bot \in \mathcal{C}[\![c_1]\!](0) \wedge \bot \in \mathcal{C}[\![c_2]\!](0) \vee \\
& (\exists \sigma \in \mathcal{C}[\![c_1]\!](0) \setminus \{\bot\} \,.\, x \notin dom(\sigma)) \wedge \bot \in \mathcal{C}[\![c_2]\!](0) \vee
\end{array}$$

$$
\begin{aligned}
& \bot \in \mathcal{C}[\![c_1]\!](0) \wedge (\exists \sigma \in \mathcal{C}[\![c_2]\!](0) \setminus \{\bot\} \,.\, [x/0] \in \sigma) \vee \\
& \bot \in \mathcal{C}[\![c_1]\!](0) \wedge \bot \in \mathcal{C}[\![c_2]\!](0) \vee \\
& (\exists \sigma \in \mathcal{C}[\![c_1]\!](0) \setminus \{\bot\} \,.\, x \notin dom(\sigma)) \wedge \bot \in \mathcal{C}[\![c_2]\!](0) \vee \\
& \bot \in \mathcal{C}[\![c_1]\!](0) \wedge (\exists \sigma \in \mathcal{C}[\![c_2]\!](0) \setminus \{\bot\} \,.\, [x/1] \in \sigma) \vee \\
& \bot \in \mathcal{C}[\![c_1]\!](0) \wedge \bot \in \mathcal{C}[\![c_2]\!](0) \vee \\
& \big((\exists \sigma \in \mathcal{C}[\![c_1]\!](0) \,.\, [x/0] \in \sigma) \vee \bot \in \mathcal{C}[\![c_1]\!](0)\big) \wedge \\
& \big((\exists \sigma \in \mathcal{C}[\![c_2]\!](0) \,.\, [x/0] \in \sigma) \vee \bot \in \mathcal{C}[\![c_2]\!](0)\big) \vee \\
& \big((\exists \sigma \in \mathcal{C}[\![c_1]\!](0) \,.\, [x/1] \in \sigma) \vee \bot \in \mathcal{C}[\![c_1]\!](0)\big) \wedge \\
& \big((\exists \sigma \in \mathcal{C}[\![c_2]\!](0) \,.\, [x/0] \in \sigma) \vee \bot \in \mathcal{C}[\![c_2]\!](0)\big) \vee \\
& \big((\exists \sigma \in \mathcal{C}[\![c_1]\!](0) \,.\, [x/0] \in \sigma) \vee \bot \in \mathcal{C}[\![c_1]\!](0)\big) \wedge \\
& \big((\exists \sigma \in \mathcal{C}[\![c_2]\!](0) \,.\, [x/1] \in \sigma) \vee \bot \in \mathcal{C}[\![c_2]\!](0)\big) \vee \\
& \big((\exists \sigma \in \mathcal{C}[\![c_1]\!](0) \,.\, [x/1] \in \sigma) \vee \bot \in \mathcal{C}[\![c_1]\!](0)\big) \wedge \\
& \big((\exists \sigma \in \mathcal{C}[\![c_2]\!](0) \,.\, [x/1] \in \sigma) \vee \bot \in \mathcal{C}[\![c_2]\!](0)\big) \\
& \quad \text{due to def. of } \varepsilon \\
= \; & \exists \sigma_1 \in \mathcal{C}[\![c_1]\!](0), \sigma_2 \in \mathcal{C}[\![c_2]\!](0) \,.\, \sigma_1 = \bot \vee \sigma_2 = \bot \vee (dom)(\sigma_1) \cap (dom)(\sigma_2) \neq \emptyset \\
& \quad \text{logic} \\
= \; & \bot \in \{\sigma_1 + \sigma_2 | \sigma_1 \in \mathcal{C}[\![c_1]\!](0), \sigma_2 \in \mathcal{C}[\![c_2]\!](0)\} \\
& \quad \text{due to def. of } + \\
= \; & snd(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!]))(0) \\
& \quad \text{due to def. of } \varepsilon
\end{aligned}
$$

Case $s = (1)$
Symmetric to case $s = (0)$

Thus $\tau(c_1 * c_2) = \varepsilon(\mathcal{C}[\![c_1 * c_2]\!])$ for $|\tau(c_1 * c_2)| = 1$.

Inductive Step $(n > 0)$

The induction hypothesis (IH II) is that $p(n-1)$ holds. That is

$$\tau(c_1 * c_2) = \varepsilon(\mathcal{C}[\![c_1 * c_2]\!]) \text{ for } |\tau(c_1 * c_2)| = n - 1$$

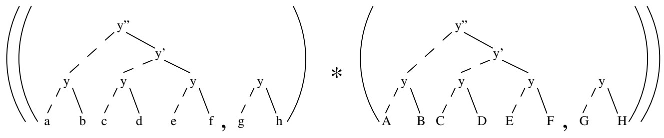
First we prove

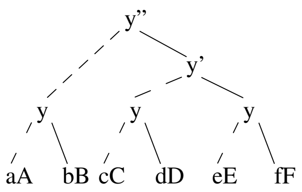
$$fst(\tau(c_1 * c_2)) = fst(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!]))$$

We have

$fst(\tau(c_1 * c_2))$
$= fst(\tau(c_1) * \tau(c_2))$
due to def. of τ

$= fst$



=


where

$$\begin{array}{rcllllllll}
\text{aA} & = & fst\big((a,g) & * & (A,G)\big) & & & & \\
\text{bB} & = & fst\big((b,h) & * & (B,H)\big) & & & & \\
\text{cC} & = & fst\big((c,g) & * & (A,G)\big) & \vee & fst\big((a,g) & * & (C,G)\big) \\
\text{dD} & = & fst\big((d,h) & * & (B,H)\big) & \vee & fst\big((b,h) & * & (D,H)\big) \\
\text{eE} & = & fst\big((e,g) & * & (A,G)\big) & \vee & fst\big((a,g) & * & (E,G)\big) \\
\text{fF} & = & fst\big((f,h) & * & (B,H)\big) & \vee & fst\big((b,h) & * & (F,H)\big)
\end{array}$$

$$= \lambda t \in \mathbb{B}^{\text{VAR}} . \begin{cases} \text{aA}(t \setminus \{t(y''), t(y'), t(y)\}) & : \neg t(y'') \wedge \neg t(y) \\ \text{bB}(t \setminus \{t(y''), t(y'), t(y)\}) & : \neg t(y'') \wedge t(y) \\ \text{cC}(t \setminus \{t(y''), t(y'), t(y)\}) & : t(y'') \wedge \neg t(y') \wedge \neg t(y) \\ \text{dD}(t \setminus \{t(y''), t(y'), t(y)\}) & : t(y'') \wedge \neg t(y') \wedge t(y) \\ \text{eE}(t \setminus \{t(y''), t(y'), t(y)\}) & : t(y'') \wedge t(y') \wedge \neg t(y) \\ \text{fF}(t \setminus \{t(y''), t(y'), t(y)\}) & : t(y'') \wedge t(y') \wedge t(y) \end{cases}$$

due to the semantics of BDDs

Case aA

We want to prove

$$\text{aA}t = fst(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!]))(t \cup \{\neg t(y''), \neg t(y)\})$$

where $t \in \mathbb{B}^{\text{VAR}\setminus\{y'',y',y\}}$. We have

$$\begin{array}{cl}
 & \text{aA}t \\
= & fst(\tau(c_1)|_{\neg y'',\neg y} * \tau(c_1)|_{\neg y'',\neg y})t \\
= & fst(\varepsilon(\mathcal{C}[\![c_1]\!])|_{\neg y'',\neg y} * \varepsilon(\mathcal{C}[\![c_2]\!])|_{\neg y'',\neg y})t \\
 & \quad \text{due to IH I} \\
= & fst(\varepsilon(\mathcal{C}[\![c_{1(\neg y'',\neg y)}]\!]) * \varepsilon(\mathcal{C}[\![c_{2(\neg y'',\neg y)}]\!]))t \\
 & \quad \text{due to lem. aA} \\
= & fst(\tau(c_{1(\neg y'',\neg y)}) * \tau(c_{2(\neg y'',\neg y)}))t \\
 & \quad \text{due t0 IH I} \\
= & fst(\tau(c_{1(\neg y'',\neg y)} * c_{2(\neg y'',\neg y)}))t \\
 & \quad \text{due to def. of } \tau \\
= & fst(\varepsilon(\mathcal{C}[\![c_{1(\neg y'',\neg y)} * c_{2(\neg y'',\neg y)}]\!]))t \\
 & \quad \text{due to IH II} \\
= & \exists \sigma \in \{\sigma_1 + \sigma_2 | \sigma_1 \in \mathcal{C}[\![c_{1(\neg y'',\neg y)}]\!]t|_{\bar{x}}, \sigma_2 \in \mathcal{C}[\![c_{2(\neg y'',\neg y)}]\!]t|_{\bar{x}}\} \setminus \{\perp\} . \\
 & \big((\forall x \in dom(\sigma) . t(x') = \sigma(x)) \wedge \\
 & \forall x \in \text{Var} \setminus \{y\} . t(x'') = x \in dom(\sigma)\big) \\
 & \quad \text{due to def. of } \varepsilon \text{ and } * \\
= & \exists \sigma \in \{\sigma_1 + \sigma_2 | \sigma_1 \in \mathcal{C}[\![c_1]\!])t|_{\bar{x}\cup\{\neg t(y)\}} \wedge y \notin dom(\sigma_1) \wedge \\
 & \sigma_2 \in \mathcal{C}[\![c_2]\!])t|_{\bar{x}\cup\{\neg t(y)\}} \wedge y \notin dom(\sigma_2)\} \setminus \{\perp\} . \\
 & \big((\forall x \in dom(\sigma) . t(x') = \sigma(x)) \wedge
\end{array}$$

$\forall x \in \mathrm{Var} \setminus \{y\} \,.\, t(x'') = x \in \mathit{dom}(\sigma)\big)$

due to lem. aA

$= \exists \sigma \in \{\sigma_1 + \sigma_2 | \sigma_1 \in \mathcal{C}[\![c_1]\!])t|_{\bar{x} \cup \{\neg t(y)\}} \wedge$
$\sigma_2 \in \mathcal{C}[\![c_2]\!])t|_{\bar{x} \cup \{\neg t(y)\}}\} \setminus \{\bot\} \,.$
$\big((\forall x \in \mathit{dom}(\sigma) \,.\, t(x') = \sigma(x)) \wedge$
$\forall x \in \mathrm{Var} \setminus \{y\} \,.\, t(x'') = x \in \mathit{dom}(\sigma)\big) \wedge$
$y \notin \mathit{dom}(\sigma)$

due to def. of $+$

$= \exists \sigma \in \{\sigma_1 + \sigma_2 | \sigma_1 \in \mathcal{C}[\![c_1]\!])t'|_{\bar{x}} \wedge$
$\sigma_2 \in \mathcal{C}[\![c_2]\!])t'|_{\bar{x}}\}\} \setminus \{\bot\} \,.$
$\big((\forall x \in \mathit{dom}(\sigma) \,.\, t'(x') = \sigma(x)) \wedge$
$\forall x \in \mathrm{Var} \,.\, t'(x'') = x \in \mathit{dom}(\sigma)\big)$
where $t' = t \cup \{\neg ty'', \neg ty\}$

logic

$= \exists \sigma \in \mathcal{C}[\![c_1 * c_2]\!])t' \setminus \{\bot\} \,.$
$\big((\forall x \in \mathit{dom}(\sigma) \,.\, t'(x') = \sigma(x)) \wedge$
$\forall x \in \mathrm{Var} \,.\, t'(x'') = x \in \mathit{dom}(\sigma)\big)$

due to def. of $*$

$= \mathit{fst}(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!]))(t \cup \{\neg t(y''), \neg t(y)\})$

due to def. of ε

Case bB

symmetric to case aA

Case cC

We want to prove

$$\mathrm{cC}t = \mathit{fst}(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!]))(t \cup t(y''), \neg t(y'), \neg t(y')\})$$

where $t \in \mathbb{B}^{\mathrm{VAR} \setminus \{y'', y', y\}}$. We have

$\mathrm{cC}t$

$= \mathit{fst}(\varepsilon(\mathcal{C}[\![c_{1(y'',\neg y',\neg y)} * c_{2(\neg y'',\neg y)}]\!]))t \vee$
$\mathit{fst}(\varepsilon(\mathcal{C}[\![c_{1(\neg y'',\neg y)} * c_{2(y'',\neg y',\neg y)}]\!]))t$

using the same steps as in the proof of case aA

$= \sigma \in \{\sigma_1 + \sigma_2 | \sigma_1 \in \mathcal{C}[\![c_{1(y'',\neg y',\neg y)}]\!]t|_{\bar{x}}, \sigma_2 \in \mathcal{C}[\![c_{2(\neg y'',\neg y)}]\!]t|_{\bar{x}}\} \setminus \{\bot\} \,.$
$\big((\forall x \in \mathit{dom}(\sigma) \,.\, t(x') = \sigma(x)) \wedge$
$\forall x \in \mathrm{Var} \setminus \{y\} \,.\, t(x'') = x \in \mathit{dom}(\sigma)\big)$
$\vee$
$\sigma \in \{\sigma_1 + \sigma_2 | \sigma_1 \in \mathcal{C}[\![c_{1(\neg y'',\neg y)}]\!]t|_{\bar{x}}, \sigma_2 \in \mathcal{C}[\![c_{2(y'',\neg y',\neg y)}]\!]t|_{\bar{x}}\} \setminus \{\bot\} \,.$
$\big((\forall x \in \mathit{dom}(\sigma) \,.\, t(x') = \sigma(x)) \wedge$
$\forall x \in \mathrm{Var} \setminus \{y\} \,.\, t(x'') = x \in \mathit{dom}(\sigma)\big)$

due to def. of ε and $*$

$= \sigma \in \{\sigma_1 + \sigma_2 |$

$$\sigma_1 \in \mathcal{C}[\![c_{1(y'',\neg y',\neg y)}]\!]t|_{\bar{x}} \wedge \sigma_2 \in \mathcal{C}[\![c_{2(\neg y'',\neg y)}]\!]t|_{\bar{x}} \vee$$
$$\sigma_1 \in \mathcal{C}[\![c_{1(\neg y'',\neg y)}]\!]t|_{\bar{x}} \wedge \sigma_2 \in \mathcal{C}[\![c_{2(y'',\neg y',\neg y)}]\!]t|_{\bar{x}}$$
$$\} \setminus \{\bot\} .$$
$$\big((\forall x \in dom(\sigma) \,.\, t(x') = \sigma(x)) \wedge$$
$$\forall x \in \mathrm{Var} \setminus \{y\} \,.\, t(x'') = x \in dom(\sigma)\big)$$

logic

$$= \quad \sigma \in \{\sigma_1 + \sigma_2|$$
$$\sigma_1 \in \mathcal{C}[\![c_1]\!]t|_{(\bar{x}\cup\{\neg t(y)\})} \wedge ([y/0] \in \sigma \vee \sigma = \bot) \wedge$$
$$\sigma_2 \in \mathcal{C}[\![c_2]\!]t|_{(\bar{x}\cup\{\neg t(y)\})} \wedge y \notin dom(\sigma_2) \vee$$
$$\sigma_1 \in \mathcal{C}[\![c_1]\!]t|_{(\bar{x}\cup\{\neg t(y)\})} \wedge y \notin dom(\sigma_2) \wedge$$
$$\sigma_2 \in \mathcal{C}[\![c_2]\!]t|_{(\bar{x}\cup\{\neg t(y)\})} \wedge ([y/0] \in \sigma \vee \sigma = \bot)$$
$$\} \setminus \{\bot\} .$$
$$\big((\forall x \in dom(\sigma) \,.\, t(x') = \sigma(x)) \wedge$$
$$\forall x \in \mathrm{Var} \setminus \{y\} \,.\, t(x'') = x \in dom(\sigma)\big)$$

due to lem. cC

$$= \quad \exists \sigma \in \{\sigma_1 + \sigma_2 | \sigma_1 \in \mathcal{C}[\![c_1]\!])t|_{\bar{x}\cup\{\neg t(y)\}} \wedge$$
$$\sigma_2 \in \mathcal{C}[\![c_2]\!])t|_{\bar{x}\cup\{\neg t(y)\}}\} \setminus \{\bot\} .$$
$$\big((\forall x \in dom(\sigma) \,.\, t(x') = \sigma(x)) \wedge$$
$$\forall x \in \mathrm{Var} \setminus \{y\} \,.\, t(x'') = x \in dom(\sigma)\big) \wedge$$
$$y \in dom(\sigma) \wedge \neg\sigma(y)$$

due to def. of $+$

$$= \quad \exists \sigma \in \{\sigma_1 + \sigma_2 | \sigma_1 \in \mathcal{C}[\![c_1]\!])t'|_{\bar{x}} \wedge$$
$$\sigma_2 \in \mathcal{C}[\![c_2]\!])t'|_{\bar{x}}\} \setminus \{\bot\} .$$
$$\big((\forall x \in dom(\sigma) \,.\, t'(x') = \sigma(x)) \wedge$$
$$\forall x \in \mathrm{Var} \,.\, t'(x'') = x \in dom(\sigma)\big)$$

where $t' = t \cup \{t(y''), \neg ty', \neg ty\}$

logic

$$= \quad \exists \sigma \in \mathcal{C}[\![c_1 * c_2]\!])t' \setminus \{\bot\} .$$
$$\big((\forall x \in dom(\sigma) \,.\, t'(x') = \sigma(x)) \wedge$$
$$\forall x \in \mathrm{Var} \,.\, t'(x'') = x \in dom(\sigma)\big)$$

due to def. of $*$

$$= \quad fst(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!]))(t \cup \{t(y), \neg t(y''), \neg t(y)\})$$

due to def. of ε

Case dD,eE and fF

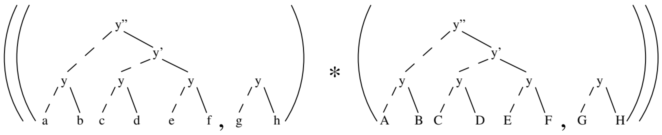
symmetric to case cC

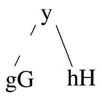
Thus $fst(\tau(c_1 * c_2)) = fst(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!]))$. We then prove

$$snd(\tau(c_1 * c_2)) = snd(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!]))$$

We have

$$
\begin{aligned}
& snd(\tau(c_1 * c_2)) \\
= \; & snd(\tau(c_1) * \tau(c_2)) \\
& \quad \text{due to def. of } \tau
\end{aligned}
$$

$= snd$ 

$=$ 

where

$$
\begin{array}{rcllllllll}
\text{gG} & = & snd\big((a,g) * (A,G)\big) & \vee & snd\big((c,g) * (A,G)\big) & \vee & snd\big((e,g) * (A,G)\big) & \vee \\
& & snd\big((a,g) * (C,G)\big) & \vee & (c,g) \otimes (C,G) & \vee & (e,g) \otimes (C,G) & \vee \\
& & snd\big((a,g) * (E,G)\big) & \vee & (c,g) \otimes (E,G) & \vee & (e,g) \otimes (E,G) & \\
\text{hH} & = & snd\big((b,h) * (B,H)\big) & \vee & snd\big((d,h) * (B,H)\big) & \vee & snd\big((f,h) * (B,H)\big) & \vee \\
& & snd\big((b,h) * (D,H)\big) & \vee & (d,h) \otimes (D,H) & \vee & (f,h) \otimes (D,H) & \vee \\
& & snd\big((b,h) * (F,H)\big) & \vee & (d,h) \otimes (F,H) & \vee & (f,h) \otimes (F,H) &
\end{array}
$$

$$
= \lambda s \in \mathbb{B}^{\text{Var}} . \begin{cases} \text{gG}(s \setminus \{s(y)\}) & : \neg s(y'') \\ \text{hH}(s \setminus \{s(y)\}) & : s(y'') \end{cases}
$$

due to the semantics of BDDs

Case gG

We want to prove

$$\text{gG}e = snd(\varepsilon(\mathcal{C}[\![c_1 * c_2]\!]))(e \cup \{\neg t(y)\})$$

where $e \in \mathbb{B}^{\text{Var} \setminus \{y\}}$.

we get

$$
\begin{aligned}
& \text{gG}t \\
= \; & snd(\varepsilon(\mathcal{C}[\![c_{1(\neg y'',\neg y)} * c_{2(\neg y'',\neg y)}]\!]))s \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_{1(y'',\neg y',\neg y)} * c_{2(\neg y'',\neg y)}]\!]))s \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_{1(y'',y',\neg y)} * c_{2(\neg y'',\neg y)}]\!]))s \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_{1(\neg y'',\neg y)} * c_{2(y'',\neg y',\neg y)}]\!]))s \vee \\
& snd(\varepsilon(\mathcal{C}[\![c_{1(\neg y'',\neg y)} * c_{2(y'',y',\neg y)}]\!]))s \vee \\
& \mathcal{C}[\![c_{1(y'',\neg y',\neg y)}]\!]s \neq \emptyset \wedge \mathcal{C}[\![c_{2(y'',\neg y',\neg y)}]\!]s \neq \emptyset \vee \\
& \mathcal{C}[\![c_{1(y'',y',\neg y)}]\!]s \neq \emptyset \wedge \mathcal{C}[\![c_{2(y'',\neg y',\neg y)}]\!]s \neq \emptyset \vee \\
& \mathcal{C}[\![c_{1(y'',\neg y',\neg y)}]\!]s \neq \emptyset \wedge \mathcal{C}[\![c_{2(y'',y',\neg y)}]\!]s \neq \emptyset \vee \\
& \mathcal{C}[\![c_{1(y'',y',\neg y)}]\!]s \neq \emptyset \wedge \mathcal{C}[\![c_{2(y'',y',\neg y)}]\!]s \neq \emptyset \\
& \quad \text{using the same steps as in the proof of case aA and lemma } \otimes \\
= \; & \bot \in \{\sigma_1 + \sigma_2 | \\
& \sigma_1 \in \mathcal{C}[\![c_1]\!](s \cup \{\neg s(y)\}) \wedge y \notin dom(\sigma_1) \wedge \\
& \sigma_2 \in \mathcal{C}[\![c_2]\!](s \cup \{\neg s(y)\}) \wedge (y \notin dom(\sigma_2) \vee \\
& [y/0] \in \sigma_2 \vee [y/1] \in \sigma_2 \vee \sigma_2 = \bot) \vee \\
& \sigma_2 \in \mathcal{C}[\![c_2]\!](s \cup \{\neg s(y)\}) \wedge y \notin dom(\sigma_2) \wedge
\end{aligned}
$$

$$
\begin{array}{ll}
& \sigma_1 \in \mathcal{C}[\![c_1]\!](s \cup \{\neg s(y)\}) \wedge (y \notin dom(\sigma_1) \vee \\
& [y/0] \in \sigma_1 \vee [y/1] \in \sigma_1 \vee \sigma_1 = \bot)\} \\
& \vee \\
& \{\sigma | \sigma \in \mathcal{C}[\![c_1]\!](s \cup \{\neg s(y)\}) \wedge ([y/0] \in \sigma \vee [y/0] \in \sigma \vee \sigma = \bot)\} \neq \emptyset \wedge \\
& \{\sigma | \sigma \in \mathcal{C}[\![c_2]\!](s \cup \{\neg s(y)\}) \wedge ([y/0] \in \sigma \vee [y/0] \in \sigma \vee \sigma = \bot)\} \neq \emptyset \\
& \quad \text{logic and def. of } \varepsilon \\
= & \bot \in \{\sigma_1 + \sigma_2 | \\
& \sigma_1 \in \mathcal{C}[\![c_1]\!](s \cup \{\neg s(y)\}) \wedge y \notin dom(\sigma_1) \wedge \\
& \sigma_2 \in \mathcal{C}[\![c_2]\!](s \cup \{\neg s(y)\}) \wedge (y \notin dom(\sigma_2) \vee \\
& [y/0] \in \sigma_2 \vee [y/1] \in \sigma_2 \vee \sigma_2 = \bot) \vee \\
& \sigma_2 \in \mathcal{C}[\![c_2]\!](s \cup \{\neg s(y)\}) \wedge y \notin dom(\sigma_2) \wedge \\
& \sigma_1 \in \mathcal{C}[\![c_1]\!](s \cup \{\neg s(y)\}) \wedge (y \notin dom(\sigma_1) \vee \\
& [y/0] \in \sigma_1 \vee [y/1] \in \sigma_1 \vee \sigma_1 = \bot)\} \\
& \vee \\
& \bot \in \{\sigma_1 + \sigma_2 | \\
& \sigma_1 \in \mathcal{C}[\![c_1]\!](s \cup \{\neg s(y)\}) \wedge ([y/0] \in \sigma_1 \vee [y/0] \in \sigma_1) \wedge \\
& \sigma_2 \in \mathcal{C}[\![c_2]\!](s \cup \{\neg s(y)\}) \wedge ([y/0] \in \sigma_2 \vee [y/0] \in \sigma_2)\} \\
& \quad \text{logic} \\
= & \bot \in \{\sigma_1 + \sigma_2 | \sigma_1 \in \mathcal{C}[\![c_1]\!]s', \sigma_2 \in \mathcal{C}[\![c_2]\!]e'\} \wedge \neg(\text{conflict in } y \text{ substitution}) \\
& \vee \\
& \bot \in \{\sigma_1 + \sigma_2 | \sigma_1 \in \mathcal{C}[\![c_1]\!]s', \sigma_2 \in \mathcal{C}[\![c_2]\!]s'\} \wedge (\text{conflict in } y \text{ substitution}), \\
& \text{where } s' = (s \cup \neg s(y) \\
& \quad \text{due to def. of } + \\
= & \bot \in \{\sigma_1 + \sigma_2 | \sigma_1 \in \mathcal{C}[\![c_1]\!]s', \sigma_2 \in \mathcal{C}[\![c_2]\!]s'\} \\
& \quad \text{logic} \\
= & snd(\varepsilon(\mathcal{C}[\![c_1 * c_2))(s \cup \{\neg s(y)\}) \\
& \quad \text{due to def. of } \varepsilon \text{ and } *
\end{array}
$$

Case hH
Symmetric to case gG

Thus $\tau(c_1 * c_2) = \varepsilon(\mathcal{C}[\![c_1 * c_2]\!])$ for $|\tau(c_1 * c_2)| = n$. $\square$