On Theoretical Properties of Sum-Product Networks: Supplementary Material

Robert Peharz
BEE-PRI BioTechMed-Graz
iDN, Inst. of Physiology
Medical University of Graz

Sebastian Tschiatschek
Dept. of Computer Science
ETH Zurich

Franz Pernkopf
SPSC Lab
Graz University
of Technology

Pedro Domingos
Dept. of Computer Science
and Engineering
University of Washington

1 Examples

1.1 Example for partial evidence (Section 1)

Let $\mathbf{X} = \{X_1, X_2, X_3\}$ with $\mathbf{val}(X_1) = [0, 1]$, $\mathbf{val}(X_2) = \{1, 2, 3\}$ and $\mathbf{val}(X_3) = \mathbb{R}$. The partial evidence $\boldsymbol{\mathcal{X}} = [0, 0.25] \cup [0.75, 1] \times \{1, 2, 3\} \times \{\pi\} \in \mathcal{H}_{\mathbf{X}}$ represents evidence that X_1 assumes a value smaller 0.25 or larger 0.75, X_2 assumes any of its possible states, and X_3 is set to the irrational number π.

1.2 Example for Algorithm 2

An example showing the mechanism of Algorithm 2 is shown in Figure 1. See the caption text for details.

1.3 Example for Algorithm 2 applied to a Sum-Product Tree

An example of applying Algorithm 2 to an SPT is shown in Figure 2, illustrating Proposition 3. See the caption text for details.

2 Proofs

Proposition 1. *Let* P *be a product node and* $\mathbf{Y}$ *be its shared RVs.* P *is consistent iff for each* $Y \in \mathbf{Y}$ *there exists a* unique $y^* \in \mathbf{val}(Y)$ *with* $\lambda_{Y=y^*} \in \mathbf{desc}(\mathsf{P})$.

Proof. Direction "$\Leftarrow$", i.e. suppose for each $Y \in \mathbf{Y}$ there exists a unique $y^* \in \mathbf{val}(Y)$ with $\lambda_{Y=y^*} \in \mathbf{desc}(\mathsf{P})$. Consider arbitrary two distinct children $\mathsf{C}', \mathsf{C}'' \in \mathbf{ch}(\mathsf{P})$ and let $\mathbf{Y}' = \mathbf{sc}(\mathsf{C}') \cap \mathbf{sc}(\mathsf{C}'')$, where from Definition 8 (shared RVs) it follows that $\mathbf{Y}' \subseteq \mathbf{Y}$. First, let $X \in \mathbf{sc}(\mathsf{C}') \setminus \mathbf{Y}'$ arbitrary. Since $X \notin \mathbf{sc}(\mathsf{C}'')$, we have

$$\lambda_{X=x} \in \mathbf{desc}(\mathsf{C}') \Rightarrow \forall x' \colon \lambda_{X=x'} \notin \mathbf{desc}(\mathsf{C}''), \quad (1)$$

and in particular

$$\lambda_{X=x} \in \mathbf{desc}(\mathsf{C}') \Rightarrow \forall x' \neq x \colon \lambda_{X=x'} \notin \mathbf{desc}(\mathsf{C}''). \quad (2)$$

Now let $X \in \mathbf{Y}'$ arbitrary. Since $X \in \mathbf{Y}$, there is a unique $y^* \in \mathbf{val}(Y)$ with $\lambda_{Y=y^*} \in \mathbf{desc}(\mathsf{C}')$ and $\lambda_{Y=y^*} \in \mathbf{desc}(\mathsf{C}'')$, and again (2) holds. Since C', C'' and X are arbitrary, P is consistent.

Direction "$\Rightarrow$", i.e. suppose P is consistent. If $\mathbf{Y}$ is empty, i.e. P is decomposable, then the claimed property holds trivially for all $\mathbf{Y}$. Otherwise let $Y \in \mathbf{Y}$ arbitrary and let C' and C'' be arbitrary two children having Y in their scope. There are at least two such children by Definition 8 (shared RVs). There must be a unique y^* with $\lambda_{Y=y^*} \in \mathbf{desc}(\mathsf{C}')$ and $\lambda_{Y=y^*} \in \mathbf{desc}(\mathsf{C}'')$. To see this, note that there must be at least one $y' \in \mathbf{val}(Y)$ with $\lambda_{Y=y'} \in \mathbf{desc}(\mathsf{C}')$ and at least one $y'' \in \mathbf{val}(Y)$ with $\lambda_{Y=y''} \in \mathbf{desc}(\mathsf{C}'')$, since Y is in the scope of both C' and C''. However, there can be at most one such y' and y''. Assume there were y'_1 and y'_2, $y'_1 \neq y'_2$. Then either for $y' = y'_1$ or for $y' = y'_2$ we had a contradiction to

$$\lambda_{Y=y'} \in \mathbf{desc}(\mathsf{C}') \Rightarrow \forall y'' \neq y' \colon \lambda_{Y=y'} \notin \mathbf{desc}(\mathsf{C}''). \quad (3)$$

i.e. P would not be consistent. By symmetry, there is also at most one y'', and (3) can only hold when $y' = y'' =: y^*$. Therefore, since Y, C' and C'' are arbitrary, there must be a unique y^* with $y^* \in \mathbf{desc}(\mathsf{C})$, $\forall \mathsf{C} \in \mathbf{ch}(\mathsf{P}) \colon Y \in \mathbf{sc}(\mathsf{C})$. Therefore, this y^* is also the unique value with $y^* \in \mathbf{desc}(\mathsf{P})$. ☐

Lemma 1. *Let* N *be a node in some complete and consistent SPN over* $\mathbf{X}$, $X \in \mathbf{sc}(\mathsf{N})$ *and* $x \in \mathbf{val}(X)$. *When* $\lambda_{X=x} \notin \mathbf{desc}(\mathsf{N})$, *then* $\forall \mathbf{x} \in \mathbf{val}(\mathbf{X})$ *with* $\mathbf{x}[X] = x$ *we have* $\mathsf{N}(\mathbf{x}) = 0$.

Proof. When $\mathsf{N} = \lambda_{X=x'}$, for some $x' \in \mathbf{val}(X)$, the lemma clearly holds, since $x' \neq x$.

When N is a product or a sum, let $\mathbf{N} = \{\mathsf{N}' \in \mathbf{desc}(\mathsf{N}) \colon X \in \mathbf{sc}(\mathsf{N}')\}$. Let $K = |\mathbf{N}|$ and let $\mathsf{N}_1, \dots, \mathsf{N}_K$ be a topologically ordered list of $\mathbf{N}$, i.e. $\mathsf{N}_k \notin \mathbf{desc}(\mathsf{N}_l)$ when $k > l$. We can assume that for some I, $\mathsf{N}_1, \dots, \mathsf{N}_I$ are IVs of X, $\mathsf{N}_{I+1}, \dots, \mathsf{N}_K$ are sum and product nodes, where

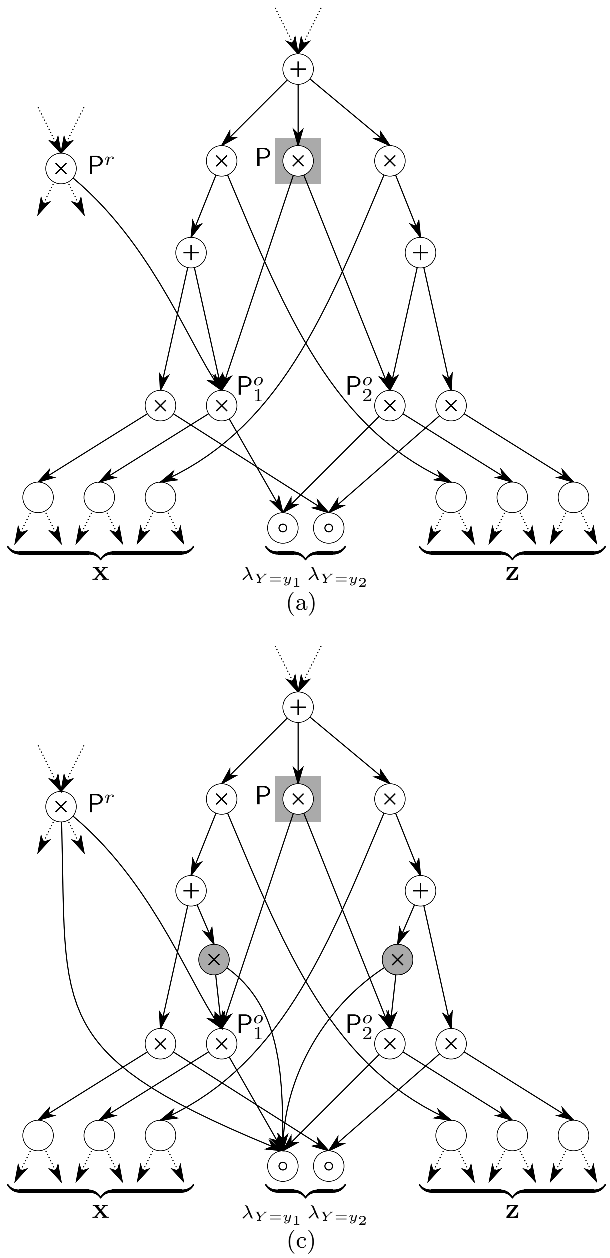

Figure 1: Illustration of Algorithm 2, transforming a complete and consistent SPN into a complete and decomposable one. (a): Excerpt of an SPN containing a single consistent, non-decomposable product P with shared RVs $\mathbf{Y} = \{Y\}$, since $\lambda_{Y=y_1}$ is reached via both P_1^o and P_2^o. Dotted edges denote a continuation of the SPN outside this excerpt. The blank circles at the bottom symbolize sub-SPNs over RV sets $\mathbf{X}$ and $\mathbf{Z}$, where we assume that $\mathbf{X} \cap \mathbf{Z} = \emptyset$. P's children P_1^o and P_2^o have both a sum node as co-parent. P_1^o has additionally a product P^r as co-parent in some remote part of the SPN. (b): Introducing links, depicted as grey product nodes, according to steps 2–7 of Algorithm 2. (c): Steps 15–20. Here, the set $\mathbf{N}^o$ is given as $\mathbf{N}^o = \{\mathsf{P}, \mathsf{P}_1^o, \mathsf{P}_2^o\}$. All parents of $\mathbf{N}^o \setminus \{\mathsf{P}\}$ which are not descendants of P, i.e. the two links and P^r, are connected with the IVs of the respective part of the shared RVs $\mathbf{Y}$, here always $\lambda_{Y=y_1}$. (d): Rendering P decomposable, steps 21–24. IV $\lambda_{Y=y_1}$ is disconnected from $\mathsf{P}_1^o, \mathsf{P}_2^o$ in steps 21–23, cutting $\mathbf{Y}$ from P_1^o, P_2^o and P. Step 24: IV $\lambda_{Y=y_1}$ is re-connected to P, which computes the same distribution as before, but is decomposable now. P_1^o and P_2^o could be short-wired and removed, since they are left with only one child. However, this does not necessarily happen and does not improve the theoretical bound of additionally required multiplications.

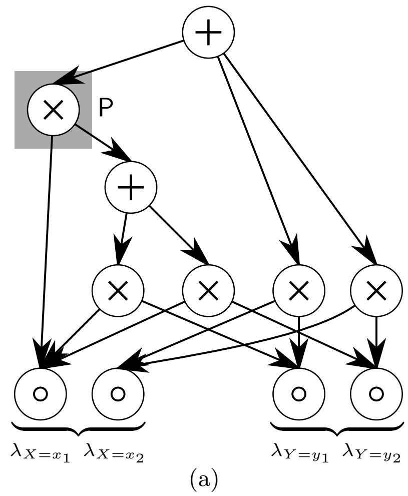

Figure 2: Illustration of Algorithm 2, transforming a complete and consistent, but not decomposable SPT into a complete and decomposable one. This demonstrates that using non-decomposable products is wasteful in SPTs (cf. Proposition 3), since after performing Algorithm 2, the SPT computes the same distribution, but requires fewer multiplications. (a): Complete and consistent SPT over two binary RVs X, Y, containing a non-decomposable product P. (b): Effect of steps 21–23 of Algorithm 2, saving 3 multiplications. (c): Effect of steps 24 of Algorithm 2, reconnecting $\lambda_{X=x_1}$ to P, re-introducing 1 multiplication.

$\mathsf{N}_K = \mathsf{N}$. For any $\mathbf{x} \in \mathbf{val}(\mathbf{X})$ with $\mathbf{x}[X] = x$, we have $\mathsf{N}_1(\mathbf{x}) = \mathsf{N}_2(\mathbf{x}) = \cdots = \mathsf{N}_I(\mathbf{x}) = 0$, since $\lambda_{X=x} \notin \mathbf{desc}(\mathsf{N})$.

When all $\mathsf{N}_1(\mathbf{x}) = \mathsf{N}_2(\mathbf{x}) = \cdots = \mathsf{N}_k(\mathbf{x}) = 0$ for some $k \geq I$, then also $\mathsf{N}_{k+1}(\mathbf{x}) = 0$. When N_{k+1} is a sum node, due to completeness, all children of N_{k+1} must have X in their scope. This means that all children are in $\{\mathsf{N}_1, \dots, \mathsf{N}_k\}$. Therefore $\mathsf{N}_{k+1}(\mathbf{x}) = 0$. When N_{k+1} is a product, at least on child has to be in $\{\mathsf{N}_1, \dots, \mathsf{N}_k\}$. Therefore $\mathsf{N}_{k+1}(\mathbf{x}) = 0$. Thus, by induction, for all $\mathsf{N}' \in \mathbf{N}$ we have $\mathsf{N}'(\mathbf{x}) = 0$. In particular this is true for $\mathsf{N}_K = \mathsf{N}$. □

Lemma 2. *Let P be a probability mass function (PMF) over $\mathbf{X}$ and $\mathbf{Y} \subseteq \mathbf{X}$, $\mathbf{Z} = \mathbf{X} \setminus \mathbf{Y}$ such that there exists a $\mathbf{y}^* \in \mathbf{val}(\mathbf{Y})$ with $P(\mathbf{z}, \mathbf{y}) = 0$ when $\mathbf{y} \neq \mathbf{y}^*$. Then we have $P(\mathbf{z}, \mathbf{y}) = \mathbb{1}(\mathbf{y} = \mathbf{y}^*)\, P(\mathbf{z})$.*

Proof. Since $P(\mathbf{z}, \mathbf{y}) = 0$ for $\mathbf{y} \neq \mathbf{y}^*$, $P(\mathbf{z}) = \sum_{\mathbf{y} \in \mathbf{val}(\mathbf{Y})} P(\mathbf{z}, \mathbf{y}) = P(\mathbf{z}, \mathbf{y}^*)$. Thus

$$P(\mathbf{z}, \mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{y} \neq \mathbf{y}^* \\ P(\mathbf{z}) & \text{if } \mathbf{y} = \mathbf{y}^* \end{cases} = \mathbb{1}(\mathbf{y} = \mathbf{y}^*)\, P(\mathbf{z}). \quad (4)$$

□

Theorem 1. *Let $\mathcal{S}$ be a complete and consistent SPN and P be a non-decomposable product in $\mathcal{S}$, $\mathbf{Y}$ be the shared RVs of P and $\mathbf{y}^*$ the consistent state of $\mathbf{Y}$. For $\mathsf{N} \in \mathbf{desc}(\mathsf{P})$ define $\mathbf{Y}^{\mathsf{N}} := \mathbf{Y} \cap \mathbf{sc}(\mathsf{N})$ and $\mathbf{X}^{\mathsf{N}} := \mathbf{sc}(\mathsf{N}) \setminus \mathbf{Y}^{\mathsf{N}}$. Then for all $\mathsf{N} \in \mathbf{desc}(\mathsf{P})$, and all $\mathbf{x} \in \mathbf{val}(\mathbf{sc}(\mathsf{N}))$:*

$$P_{\mathsf{N}}(\mathbf{x}) = \mathbb{1}\big(\mathbf{x}[\mathbf{Y}^{\mathsf{N}}] = \mathbf{y}^*[\mathbf{Y}^{\mathsf{N}}]\big)\; P_{\mathsf{N}}\left(\mathbf{x}[\mathbf{X}^{\mathsf{N}}]\right), \quad (5)$$

where $\mathbb{1}$ is the indicator function.

Proof. From Proposition 1 we know that $\forall Y \in \mathbf{Y}\colon \lambda_{Y=\mathbf{y}^*[Y]} \in \mathbf{desc}(\mathsf{P})$ and $\forall y \neq \mathbf{y}^*[Y]\colon \lambda_{Y=y} \notin \mathbf{desc}(\mathsf{P})$. Consequently, for any $\mathsf{N} \in \mathbf{desc}(\mathsf{P})$ we have for all $Y \in \mathbf{Y}^{\mathsf{N}}$ that $\lambda_{Y=\mathbf{y}^*[Y]} \in \mathbf{desc}(\mathsf{N})$ and $\forall y \neq \mathbf{y}^*[Y]\colon \lambda_{Y=y} \notin \mathbf{desc}(\mathsf{N})$. With Lemma 1 it follows that for all $\mathbf{x} \in \mathbf{val}(\mathbf{sc}(\mathsf{N}))$ with $\mathbf{x}[\mathbf{Y}^{\mathsf{N}}] \neq \mathbf{y}^*[\mathbf{Y}^{\mathsf{N}}]$, we have $P_{\mathsf{N}}(\mathbf{x}) = 0$. Theorem 1 follows with Lemma 2. □

Corollary 1. *Let P, $\mathbf{Y}$, $\mathbf{y}^*$ be as in Theorem 1. For $\mathsf{C} \in \mathbf{ch}(\mathsf{P})$, let $\mathbf{X}^{\mathsf{C}} := \mathbf{sc}(\mathsf{C}) \setminus \mathbf{Y}$, i.e. the part of $\mathbf{sc}(\mathsf{C})$ which belongs exclusively to C. Then*

$$P_{\mathsf{P}}(\mathbf{x}) = \mathbb{1}(\mathbf{x}[\mathbf{Y}] = \mathbf{y}^*) \prod_{\mathsf{C} \in \mathbf{ch}(\mathsf{P})} \mathsf{C}\left(\mathbf{x}[\mathbf{X}^{\mathsf{C}}]\right). \quad (6)$$

Theorem 2. *For each complete and consistent SPN $\mathcal{S}' = (\mathcal{G}', \boldsymbol{w}')$, there exists a complete, consistent and* locally normalized *SPN $\mathcal{S} = (\mathcal{G}, \boldsymbol{w})$ with $\mathcal{G}' = \mathcal{G}$, such that $\forall \mathsf{N} \in \mathcal{G}\colon \mathcal{S}_{\mathsf{N}} \equiv P_{\mathcal{S}'_{\mathsf{N}}}$.*

Proof. Algorithm 1 finds locally normalized weights without changing the distribution of any node. For deriving the algorithm, we introduce a *correction factor* α_{P} for each product node P, initialized to $\alpha_{\mathsf{P}} = 1$. We redefine the product node P as $\mathsf{P}(\boldsymbol{\lambda}) := \alpha_{\mathsf{P}} \mathsf{P}(\boldsymbol{\lambda})$. At the end of the algorithm, all α_{P} will be 1 again.

Algorithm 1 Locally Normalize SPN

1: Let $\mathsf{N}_1, \ldots, \mathsf{N}_K$ be a topologically ordered list of all sum and product nodes
2: For all product nodes P initialize $\alpha_{\mathsf{P}} \leftarrow 1$
3: **for** $k = 1 : K$ **do**
4:     **if** N_k is a sum node **then**
5:         $\alpha \leftarrow \sum_{\mathsf{C} \in \mathbf{ch}(\mathsf{N}_k)} w_{\mathsf{N}_k,\mathsf{C}}$
6:         $\forall \mathsf{C} \in \mathbf{ch}(\mathsf{N}_k) : w_{\mathsf{N}_k,\mathsf{C}} \leftarrow \frac{w_{\mathsf{N}_k,\mathsf{C}}}{\alpha}$
7:     **end if**
8:     **if** N_k is a product node **then**
9:         $\alpha \leftarrow \alpha_{\mathsf{N}_k}$
10:         $\alpha_{\mathsf{N}_k} \leftarrow 1$
11:     **end if**
12:     **for** $\mathsf{F} \in \mathbf{pa}(\mathsf{N}_k)$ **do**
13:         **if** F is a sum node **then**
14:             $w_{\mathsf{F},\mathsf{N}_k} \leftarrow \alpha\, w_{\mathsf{F},\mathsf{N}_k}$
15:         **end if**
16:         **if** F is a product node **then**
17:             $\alpha_{\mathsf{F}} \leftarrow \alpha\, \alpha_{\mathsf{F}}$
18:         **end if**
19:     **end for**
20: **end for**

Let $\mathsf{N}'_1, \ldots, \mathsf{N}'_K$ be a topologically ordered list of all nodes in the unnormalized $\mathcal{S}'$, i.e. $\mathsf{N}'_k \notin \mathbf{desc}(\mathsf{N}'_l)$ if $k > l$. Let $\mathsf{N}_1, \ldots, \mathsf{N}_K$ be the corresponding list of $\mathcal{S}$, which will be the normalized version after the algorithm has terminated. We have the following loop invariant for the main loop. Given that at the k^{th} entrance of the main loop

1. $P_{\mathsf{N}'_l} = \mathcal{S}_{\mathsf{N}_l}$, for $1 \leq l < k$
2. $\mathcal{S}'_{\mathsf{N}'_m} = \mathcal{S}_{\mathsf{N}_m}$, for $k \leq m \leq K$

the same will hold for $k + 1$ at the end of the loop.

The first point holds since we normalize N_k during the main loop: All nodes prior in the topological order, and therefore all children of N_k are already normalized. If N_k is a sum, then it represents a mixture distribution after step 6. If N_k is a product, then it will be normalized after step 10 since we set $\alpha_{\mathsf{N}_k} = 1$.

The second point holds since the modification of N_k can change any N_m, $m > k$, only via $\mathbf{pa}(\mathsf{N}_k)$. The change of N_k is compensated for all its parents either in step 14 or step 17, depending on whether the parent is a sum or a product node.

From this loop invariance it follows by induction that all $\mathsf{N}_1, \ldots, \mathsf{N}_K$ compute the normalized distributions of $\mathsf{N}'_1, \ldots, \mathsf{N}'_K$ after the K^{th} iteration. □

Proposition 2. *A complete and decomposable SPN computes the NP of some unnormalized distribution.*

Proof. Assume some topological ordering of the SPN nodes. We show by induction over this order, from bottom to top, that each sub-SPN $\mathcal{S}_{\mathsf{N}}$ computes an NP over $\mathbf{sc}(\mathsf{N})$. The induction basis are the IVs $\lambda_{X=x}$ which compute

$$\lambda_{X=x} = \sum_{x' \in \mathbf{val}(X)} \mathbb{1}(x' = x)\, \lambda_{X=x'}, \tag{7}$$

which is an NP over X.

A complete sum node S computes

$$\mathsf{S}(\boldsymbol{\lambda}) = \sum_{\mathsf{C} \in \mathbf{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{C}}\, \mathsf{C}(\boldsymbol{\lambda}) \tag{8}$$

$$= \sum_{\mathsf{C} \in \mathbf{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{C}} \sum_{\mathbf{x} \in \mathbf{val}(\mathbf{sc}(\mathsf{S}))} \Phi_{\mathsf{C}}(\mathbf{x}) \prod_{X \in \mathbf{sc}(\mathsf{S})} \lambda_{X=\mathbf{x}[X]} \tag{9}$$

$$= \sum_{\mathbf{x} \in \mathbf{val}(\mathbf{sc}(\mathsf{S}))} \Phi_{\mathsf{S}}(\mathbf{x}) \prod_{X \in \mathbf{sc}(\mathsf{S})} \lambda_{X=\mathbf{x}[X]}, \tag{10}$$

where Φ_{C} is the unnormalized distributions of the NP of C, and $\Phi_{\mathsf{S}}(\mathbf{x}) := \sum_{\mathsf{C} \in \mathbf{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{C}}\, \Phi_{\mathsf{C}}(\mathbf{x})$. In (9) we apply the induction hypothesis and the fact that for complete sum nodes the scopes of the node and all its children are the same. We see that (10) has the form of an NP, i.e. the induction step holds for complete sum nodes.

Now consider some decomposable product node P with children $\mathbf{ch}(\mathsf{P}) = \{\mathsf{C}_1, \ldots, \mathsf{C}_K\}$. The product node P computes

$$\mathsf{P}(\boldsymbol{\lambda}) = \prod_{k=1}^{K} \mathsf{C}_k(\boldsymbol{\lambda}) \tag{11}$$

$$= \prod_{k=1}^{K} \sum_{\mathbf{x}^k \in \mathbf{val}(\mathbf{sc}(\mathsf{C}_k))} \Phi_{\mathsf{C}_k}(\mathbf{x}^k) \prod_{X \in \mathbf{sc}(\mathsf{C}_k)} \lambda_{X=\mathbf{x}^k[X]} \tag{12}$$

$$= \sum_{\mathbf{x}^1} \cdots \sum_{\mathbf{x}^K} \prod_{k=1}^{K} \left[\Phi_{\mathsf{C}_k}(\mathbf{x}^k) \prod_{X \in \mathbf{sc}(\mathsf{C}_k)} \lambda_{X=\mathbf{x}^k[X]} \right] \tag{13}$$

$$= \sum_{\mathbf{x} \in \mathbf{val}(\mathbf{sc}(\mathsf{P}))} \Phi_{\mathsf{P}}(\mathbf{x}) \prod_{X \in \mathbf{sc}(\mathsf{P})} \lambda_{X=\mathbf{x}[X]}, \tag{14}$$

where Φ_{C_k} is the distributions of the NP of C_k, and $\Phi_{\mathsf{P}}(\mathbf{x}) := \prod_{k=1}^{K} \Phi_{\mathsf{C}_k}(\mathbf{x}[\mathbf{sc}(\mathsf{C}_k)])$. In (12) we use the induction hypothesis, in (13) we apply the distributive law, and in (14) we use the fact that the scope of a decomposable product node is partitioned by the scopes of its children. We see that (14) has the form of an NP, i.e. the induction step holds for complete product nodes.

Consequently, each sub-SPN and therefore also the overall SPN computes an NP of some unnormalized distribution. □

Theorem 3. *Every complete and consistent SPN $\mathcal{S} = ((V, E), \boldsymbol{w})$ over $\mathbf{X}$ can be transformed into a complete and decomposable SPN $\mathcal{S}' = ((V', E'), \boldsymbol{w}')$ over $\mathbf{X}$ such that $P_{\mathcal{S}} \equiv P_{\mathcal{S}'}$, and where $|V'| \in \mathcal{O}(|V|^2)$, $A_{\mathcal{S}'} = A_{\mathcal{S}}$ and $M_{\mathcal{S}'} \in \mathcal{O}(M_{\mathcal{S}}\,|\mathbf{X}|)$.*

Algorithm 2 Transform to decomposable SPN

1: Let $\mathsf{N} = \mathsf{N}_1, \ldots, \mathsf{N}_K$ be a topologically ordered list of all sums and products
2: **for** all sum nodes S and all $\mathsf{C} \in \mathbf{ch}(\mathsf{S})$ **do**
3:     **if** $\mathbf{pa}(\mathsf{C}) > 1$ **then**
4:         Generate a new product node $\mathsf{P}^{\mathsf{S},\mathsf{C}}$
5:         Interconnect $\mathsf{P}^{\mathsf{S},\mathsf{C}}$ between S and C
6:     **end if**
7: **end for**
8: **while** exist non-decomposable products in N **do**
9:     $\mathsf{P} \leftarrow \mathsf{N}_{\min\{k' \,|\, \mathsf{N}_{k'} \text{ is a non-decomposable product}\}}$
10:     $\mathbf{Y} \leftarrow$ shared RVs of P
11:     $\mathbf{y}^* \leftarrow$ consistent state of $\mathbf{Y}$
12:     **if** $\mathbf{sc}(\mathsf{P}) = \mathbf{Y}$ **then**
13:         Replace P by $\prod_{Y \in \mathbf{Y}} \lambda_{Y = \mathbf{y}^*[Y]}$
14:     **else**
15:         $\mathsf{N}^d \leftarrow$ sums and products in $\mathbf{desc}(\mathsf{P})$
16:         $\mathsf{N}^o \leftarrow \{\mathsf{N} \in \mathsf{N}^d \colon \mathbf{sc}(\mathsf{N}) \not\subseteq \mathbf{Y}, \mathbf{sc}(\mathsf{N}) \cap \mathbf{Y} \neq \emptyset\}$
17:         **for** $\mathsf{N} \in \mathsf{N}^o \setminus \{\mathsf{P}\}$ **do**
18:             $\mathsf{F} \leftarrow \mathbf{pa}(\mathsf{N}) \setminus \mathsf{N}^d$
19:             $\forall Y \in \mathbf{Y} \cap \mathbf{sc}(\mathsf{N})$: connect $\lambda_{Y=\mathbf{y}^*[Y]}$ as child of all F
20:         **end for**
21:         **for** $\mathsf{P}^o \in \mathsf{N}^o$ **do**
22:             Disconnect $\mathsf{C} \in \mathbf{ch}(\mathsf{P}^o)$ if $\mathbf{sc}(\mathsf{C}) \subseteq \mathbf{Y}$
23:         **end for**
24:         $\forall Y \in \mathbf{Y}$: connect $\lambda_{Y=\mathbf{y}^*[Y]}$ as child of P
25:     **end if**
26: **end while**
27: Delete all unreachable sums and products

Proof. Due to Theorem 2 we assume w.l.o.g. that $\mathcal{S}'$ is locally normalized, and thus $P_{\mathcal{S}} \equiv \mathcal{S}$. Algorithm 2 transforms $\mathcal{S}$ into a complete and decomposable SPN, representing the same distribution. First it finds a topologically ordered list $\mathsf{N}_1, \ldots, \mathsf{N}_K$ of all sum and product nodes, i.e. $k > l \Rightarrow \mathsf{N}_k \notin \mathbf{desc}(\mathsf{N}_l)$. Then, in steps 2–7, it considers all sum nodes S and all children $\mathsf{C} \in \mathbf{ch}(\mathsf{S})$; if the child C has further parents except S, a newly generated product node $\mathsf{P}^{\mathsf{S},\mathsf{C}}$ is interconnected between S and C, i.e. $\mathsf{P}^{\mathsf{S},\mathsf{C}}$ is connected as child of S with weight $w_{\mathsf{S},\mathsf{C}}$, C is disconnected from S and connected as child of $\mathsf{P}^{\mathsf{S},\mathsf{C}}$. To $\mathsf{P}^{\mathsf{S},\mathsf{C}}$ we refer as *link* between S and C. Note that the link has *only* S as parent, i.e. the link represents a *private* copy of child C for sum node S. Clearly, after step 7, the SPN still computes the same function.

In each iteration of the main loop 8–26, the algorithm finds the lowest non-decomposable product node $\mathsf{N}_k = \mathsf{P}$ w.r.t. the topological ordering. We distinguish two cases: $\mathbf{sc}(\mathsf{P}) = \mathbf{Y}$ and $\mathbf{sc}(\mathsf{P}) \neq \mathbf{Y} \Leftrightarrow \mathbf{Y} \subset \mathbf{sc}(\mathsf{P})$.

In the first case, we know from Corollary 1 that $\mathsf{P}(\mathbf{y}) = \mathbb{1}(\mathbf{y} = \mathbf{y}^*)$, which is equivalent to the decomposable product $\prod_{Y \in \mathbf{Y}} \lambda_{Y=\mathbf{y}^*[Y]}$ replacing P, i.e. this new product is connected as child of all parents of P, and P itself is deleted. Deletion of P might render some nodes unreachable; however, these unreachable nodes do not "influence" the root node and will be safely deleted in step 27.

In the second case, when $\mathbf{Y} \subset \mathbf{sc}(\mathsf{P})$, the algorithm first finds the set N^d of all sum and product descendants of P. It also finds the subset N^o of N^d, containing all nodes whose scope overlaps with $\mathbf{Y}$, but is no subset of $\mathbf{Y}$. Clearly, P is contained in N^o. The basic strategy is to "cut" $\mathbf{Y}$ from the scope of P, i.e. that $\mathbf{Y}$ is marginalized, rendering P decomposable. Then, by re-connecting all indicators $\lambda_{Y=\mathbf{y}^*[Y]}$ to P in step 24, P computes the same distribution as before due to Corollary 1, but is rendered *decomposable* now. Steps 21–23 cut $\mathbf{Y}$ from *all* nodes in N^o, in particular from P, but leave all sub-SPNs rooted at any node in $\mathsf{N}^d \setminus \mathsf{N}^o$ unchanged. To see this, note that $\mathsf{N}^d \setminus \mathsf{N}^o$ contains two types of nodes:

- nodes N^s whose scope is a subset of $\mathbf{Y}$, i.e. $\mathsf{N}^s = \{\mathsf{N} \in \mathsf{N}^d \,|\, \mathbf{sc}(\mathsf{N}) \subseteq \mathbf{Y}\}$, and
- nodes N^n whose scope does not contain any $\mathbf{Y}$, i.e. $\mathsf{N}^n = \{\mathsf{N} \in \mathsf{N}^d \mid \mathbf{sc}(\mathsf{N}) \cap \mathbf{Y} = \emptyset\}$.

Clearly, sub-SPNs rooted at any node in N^s or N^n do not contain any N^o. Steps 21–23 only delete some outgoing edges of some $\mathsf{P}^o \in \mathsf{N}^o$; thus all sub-SPNs rooted at nodes in N^s or N^n remain unchanged, and do not change the output of the overall SPN via nodes outside of N^d. Now consider a topologically ordered list $\mathsf{N}^o_1, \ldots, \mathsf{N}^o_L$ of the nodes in N^o, i.e. $k > l \Rightarrow \mathsf{N}^o_k \notin \mathbf{desc}(\mathsf{N}^o_l)$. N^o_1 must be a product. If it was a sum, all its children would have the same scope as N^o_1, due to completeness. Therefore, all its children would be contained in N^o, contradicting that N^o_1 is the first node in the topological order. Thus N^o_1 is a product. N^o_1 can have three types of children: nodes in N^s, nodes in N^n and IVs. Nodes in N^s and IVs corresponding to some $Y \in \mathbf{Y}$ are disconnected from product N^o_1 in steps 21–23. These children make up the deterministic term $\mathbb{1}(\cdot)$ in Theorem 1. By disconnecting them, $\mathbf{sc}(\mathsf{N}^o_1) \cap \mathbf{Y}$ is "cut" from N^o_1, which now computes the marginal distribution over $\mathbf{sc}(\mathsf{N}^o_1) \setminus \mathbf{Y}$. This is the induction basis. Now assume that after steps 21–23 all $\mathsf{N}^o_1, \ldots, \mathsf{N}^o_l$, $l < L$, compute the marginal, $\mathbf{Y}$ being marginalized. Then also N^o_{l+1} computes such a marginal. If N^o_{l+1} is a sum and thus a mixture distribution, this clearly holds, since mixture sums and marginalization sums can be swapped, i.e. “the mixture of marginals is the marginal of the mixture”. If N^o_{l+1} is a product, it can have four types of children: nodes in $\mathbf{N}^o$, nodes in $\mathbf{N}^s$, nodes in $\mathbf{N}^n$ and IVs. By induction hypothesis $\mathbf{Y}$ is already cut from all children in $\mathbf{N}^o$. Nodes in $\mathbf{N}^s$ and IVs corresponding to some $Y \in \mathbf{Y}$ are disconnected from product N^o_{l+1}. These nodes make up the deterministic term $\mathbb{1}(\cdot)$ in Theorem 1; a subset of $\mathbf{Y}$ might already have been removed, since $\mathbf{Y}$ has been removed from all children in $\mathbf{N}^o$. By disconnecting children in $\mathbf{N}^s$ and IVs corresponding to $\mathbf{Y}$, this deterministic term is fully removed. Thus, by induction, after steps 21–23 all nodes in $\mathbf{N}^o$ compute the marginal distribution, $\mathbf{Y}$ being marginalized.

Although we achieve our primary goal to render P decomposable, steps 21–23 also cut $\mathbf{Y}$ from any other node in $\mathsf{N} \in \mathbf{N}^o$, which would modify the SPN output via N's parents *outside* of $\mathbf{N}^d$, i.e. via $\mathbf{F} = \mathbf{pa}(\mathsf{N}) \setminus \mathbf{N}^d$. Note that all nodes in $\mathbf{F}$ must be products. To see this, assume that $\mathbf{F}$ contains a sum S. This would imply that N is a link, which can reached from P only via its single parent S. This implies $\mathsf{S} \in \mathbf{N}^d$, a contradiction. By Theorem 1, the distribution of N is deterministic w.r.t. $\mathbf{sc}(\mathsf{N}) \cap \mathbf{Y}$. Steps 21–23 cut $\mathbf{Y}$ from N, which would change the distribution of the nodes in $\mathbf{F}$. Thus, in step 19 the IVs corresponding to $\mathbf{Y} \cap \mathbf{sc}(\mathsf{N})$ are connected to all $\mathbf{F}$, such that they still “see” the same distribution after steps 21–23. It is easy to see that if some $\mathsf{F} \in \mathbf{F}$ was decomposable beforehand, it will also be decomposable after step 23, i.e. steps 15–24 do not render other products non-decomposable.

Since the only new introduced nodes are the links between sum nodes and their children, and the number of sum-edges is in $\mathcal{O}(|V|^2)$, we have $|V'| \in \mathcal{O}(|V|^2)$. The number of summations is the same in $\mathcal{S}$ and $\mathcal{S}'$, i.e. $A_{\mathcal{S}'} = A_{\mathcal{S}}$. Furthermore, introducing the links can not introduce more than double the number of multiplications, since we already require one multiplication per sum-edge. Thus, after step 7 we have $M_{\mathcal{S}'} \in \mathcal{O}(M_{\mathcal{S}})$. Since the while-loop in Algorithm 2 can not connect more than one IV per $X \in \mathbf{X}$ to each product node, we have $M_{\mathcal{S}'} \in \mathcal{O}(M_{\mathcal{S}}\,|\mathbf{X}|)$. □

Proposition 3. *Every complete and consistent, but non-decomposable SPT $\mathcal{S} = ((V,E), \boldsymbol{w})$ over $\mathbf{X}$ can be transformed into a complete and decomposable SPT $\mathcal{S}' = ((V',E'), \boldsymbol{w}')$ over $\mathbf{X}$ such that $P_{\mathcal{S}} \equiv P_{\mathcal{S}'}$, and where $|V'| \le |V|$, $A_{\mathcal{S}'} \le A_{\mathcal{S}}$ and $M_{\mathcal{S}'} < M_{\mathcal{S}}$.*

Proof. For SPTs, no links are introduced in steps 2–7 of Algorithm 2. Consider a non-decomposable product P in the SPT, and let $\mathbf{Y}$ be the shared RVs. If $\mathbf{sc}(\mathsf{P}) = \mathbf{Y}$, the whole sub-SPN rooted at P will be replaced by $\prod_{Y \in \mathbf{Y}} \lambda_{Y=\mathbf{y}^*[Y]}$ in step 13. The deleted sub-SPN computes $\prod_{Y \in \mathbf{Y}} (\lambda_{Y=\mathbf{y}^*[Y]})^{k_Y}$, where k_Y are integers with $k_Y \ge 1$, and at least one $k_Y > 1$, since otherwise P would be decomposable. Thus, replacing this sub-SPN by the decomposable product in step 13 saves at least one multiplication.

Similarly, when $\mathbf{sc}(\mathsf{P}) \ne \mathbf{Y}$, steps 21–23 prune nodes from the sub-SPN rooted at P, corresponding to the computation of $\prod_{Y \in \mathbf{Y}} (\lambda_{Y=\mathbf{y}^*[Y]})^{k_Y}$ with $k_Y \ge 1$, and at least one $k_Y > 1$. This requires at least one multiplication more than directly connecting $\lambda_{Y=\mathbf{y}^*[Y]}$ to P in step 24. Clearly, steps 17–20 are never performed in SPTs, since no $\mathsf{N} \in \mathbf{desc}(\mathsf{P}) \setminus \{\mathsf{P}\}$ can have a parent outside the descendants of P.

Thus, for every non-decomposable product, Algorithm 2 saves at least one multiplication. □

Proposition 4. *Let $\mathcal{S}^g$ be a gated SPN. For any $X \in \mathbf{X}$ and $k \in \{1, \dots, |\mathbf{D}^X|\}$, let $\mathbf{X}^D = \mathbf{sc}(\mathsf{D}^{X,k})$ and $\mathbf{X}^R = \mathbf{X} \setminus \mathbf{X}^D$. It holds that $\mathcal{S}^g(\mathbf{X}, Z_X = k, \mathbf{Z} \setminus Z_X) = \mathsf{D}^{X,k}(\mathbf{X}^D)\, \mathcal{S}^g(\mathbf{X}^R, Z_X = k, \mathbf{Z} \setminus Z_X)$.*

Proof. Consider all sum and product nodes $\mathbf{N}$ in $\mathcal{S}^g$ which have X in their scope. We can split $\mathbf{N}$ into two sets $\mathbf{N}', \mathbf{N}''$, where

$$\begin{aligned} \mathbf{N}' &= \{\mathsf{N} \in \mathbf{N} \mid \mathsf{D}^{X,k} \in \mathbf{desc}(\mathsf{N})\} \\ &= \{\mathsf{N} \in \mathbf{N} \mid \lambda_{Z=X,k} \in \mathbf{desc}(\mathsf{N})\} \end{aligned} \tag{15}$$

and $\mathbf{N}'' = \mathbf{N} \setminus \mathbf{N}'$. When we set $Z_X = k$, we get by Lemma 1 that all $\mathsf{N} \in \mathbf{N}''$ output 0.

We now show by induction that all $\mathsf{N} \in \mathbf{N}'$ have the form

$$\mathsf{D}^{X,k}(\mathbf{X}^D)\, \mathsf{N}(\mathbf{X}^{R'}, Z_X = k, \mathbf{Z}' \setminus \{Z_X\}), \tag{16}$$

where $\mathbf{X}^{R'} = \mathbf{X}^R \cap \mathbf{sc}(\mathsf{N})$ and $\mathbf{Z}' = \mathbf{Z} \cap \mathbf{sc}(\mathsf{N})$. Let $\mathsf{N}_1, \dots, \mathsf{N}_K$ be a topologically ordered list of $\mathbf{N}'$, i.e. $\mathsf{N}_k \notin \mathbf{desc}(\mathsf{N}_l)$ when $k > l$, where N_1 is the introduced product when constructing the gated SPN, having $\mathsf{D}^{X,k}$ and $\lambda_{Z_X=k}$ as children, and N_K is the root node. Eq. (16) clearly holds for N_1, since $\mathsf{N}_1(\mathbf{X}^D, Z_X = k) = \mathsf{D}^{X,k}(\mathbf{X}^D)\, \lambda_{Z_X=k}(Z_X = k)$. This is the induction basis. For the induction step assume that (16) holds for $\mathsf{N}_1, \dots, \mathsf{N}_k$. When N_{k+1} is a sum

S, it computes

$$\sum_{\mathsf{C} \in \mathbf{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{C}} \, \mathsf{C}(\mathbf{X}, Z_X = k, \mathbf{Z}' \setminus \{Z_X\}) \tag{17}$$

$$= \sum_{\mathsf{C} \in \mathbf{ch}(\mathsf{S}) \cap \mathbf{N}'} w_{\mathsf{S},\mathsf{C}} \, \mathsf{C}(\mathbf{X}, Z_X = k, \mathbf{Z}' \setminus \{Z_X\}) + \underbrace{\sum_{\mathsf{C} \in \mathbf{ch}(\mathsf{S}) \cap \mathbf{N}''} w_{\mathsf{S},\mathsf{C}} \, \mathsf{C}(\mathbf{X}, Z_X = k, \mathbf{Z}' \setminus \{Z_X\})}_{=0} \tag{18}$$

$$= \mathsf{D}^{X,k}(\mathbf{X}^{\mathsf{D}}) \sum_{\mathsf{C} \in \mathbf{ch}(\mathsf{S}) \cap \mathbf{N}'} w_{\mathsf{S},\mathsf{C}} \, \mathsf{C}(\mathbf{X}^{R'}, Z_X = k, \mathbf{Z}' \setminus \{Z_X\}), \tag{19}$$

i.e. (16) holds for sum nodes. When N_{k+1} is a product P, it must have a single child C in $\mathbf{N}'$, since P itself is contained in $\mathbf{N}'$. All other children are neither in $\mathbf{N}'$ nor $\mathbf{N}''$, due to decomposability. It is immediate, that when (16) holds for C, it also holds for P. Thus, (16) holds for all $\mathbf{N}'$, in particular for the root N_K, and the proposition follows. $\square$