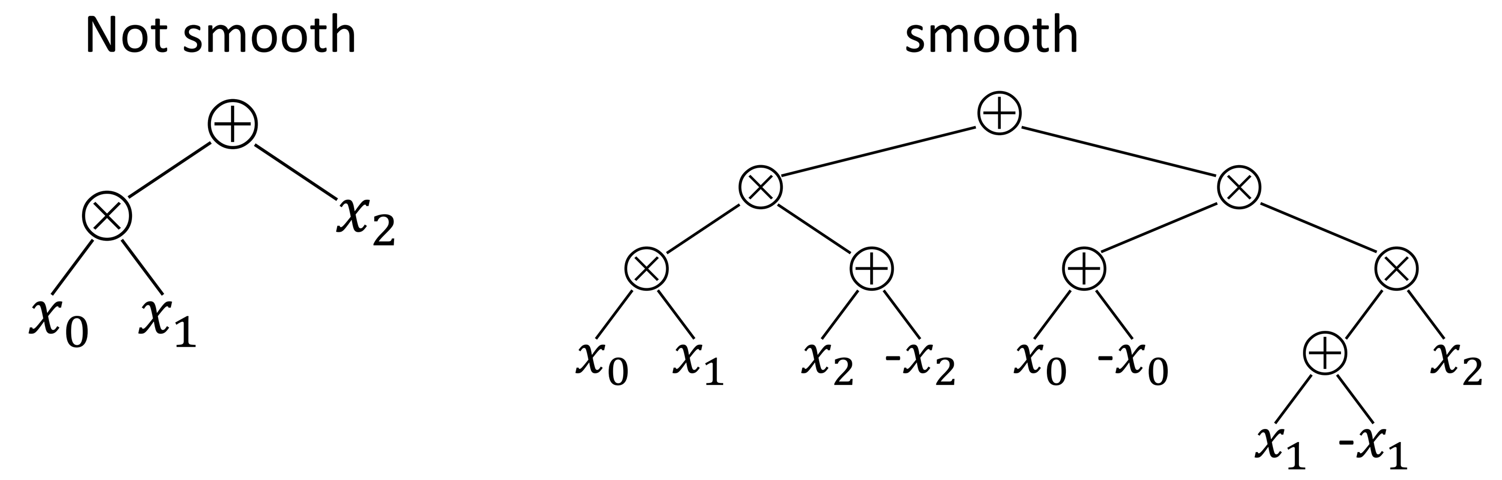
Smoothing Structured Decomposable Circuits

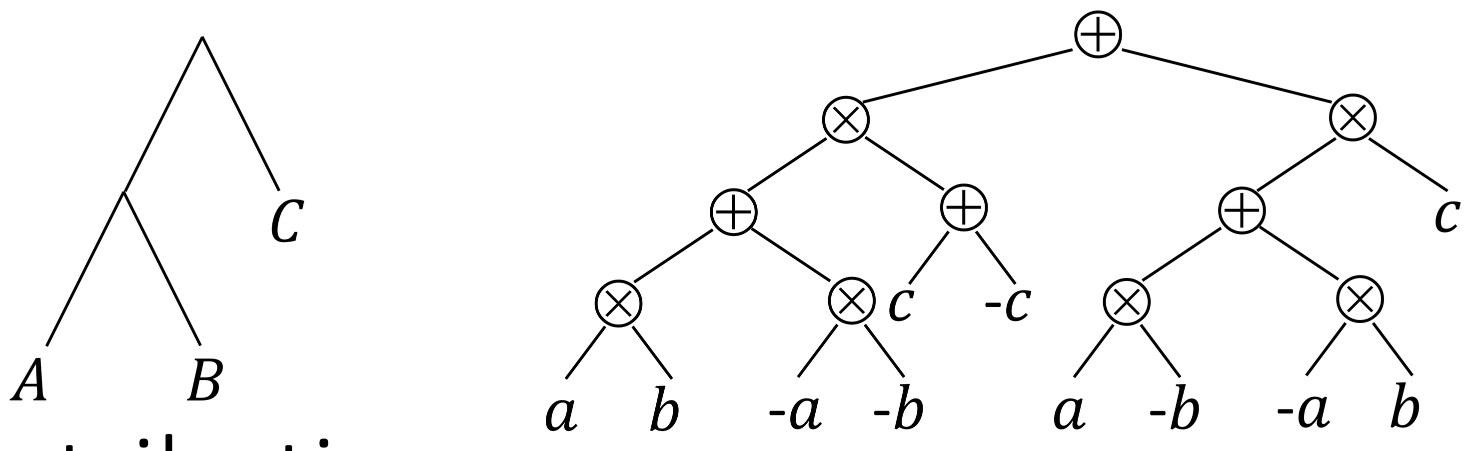
Andy Shih[1], Guy Van den Broeck[1], Paul Beame[2], Antoine Amarilli[3]
[1]University of California, Los Angeles, [2]University of Washington, [3]LTCI Télécom Paris, IP Paris

Smoothness is necessary for linear-time Weighted Model Counting and All-Marginals on circuits. But, the current smoothing algorithm is quadratic.



Structured decomposability

Example from Shen et al. "Tractable operations for arithmetic circuits of probabilistic models."



Contributions

Task	Operations	Complexity
Smoothing	$\oplus, \otimes$	$O(m \cdot \alpha(m, n))$
Smoothing*	$\oplus, \otimes$	$\Omega(m \cdot \alpha(m, n))^*$
All-Marginal	$\oplus, \ominus, \otimes, \oslash$	$\Theta(m)$

* For *smoothing-gate algorithms* on decomposable circuits.

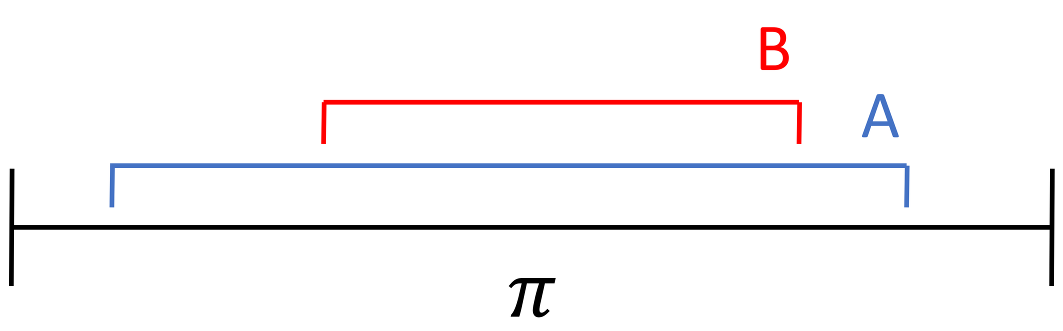
Smoothing gate algorithms

Think "Filling in missing variables":
The output circuit has a subcircuit that is isomorphic to input circuit (after edge contraction).

Missing variables form 2 intervals

Traverse vtree in-order to get ordering of vars π.
The vars of a subtree is a continuous interval in π.

A\B forms 2 intervals



Smoothing in one pass

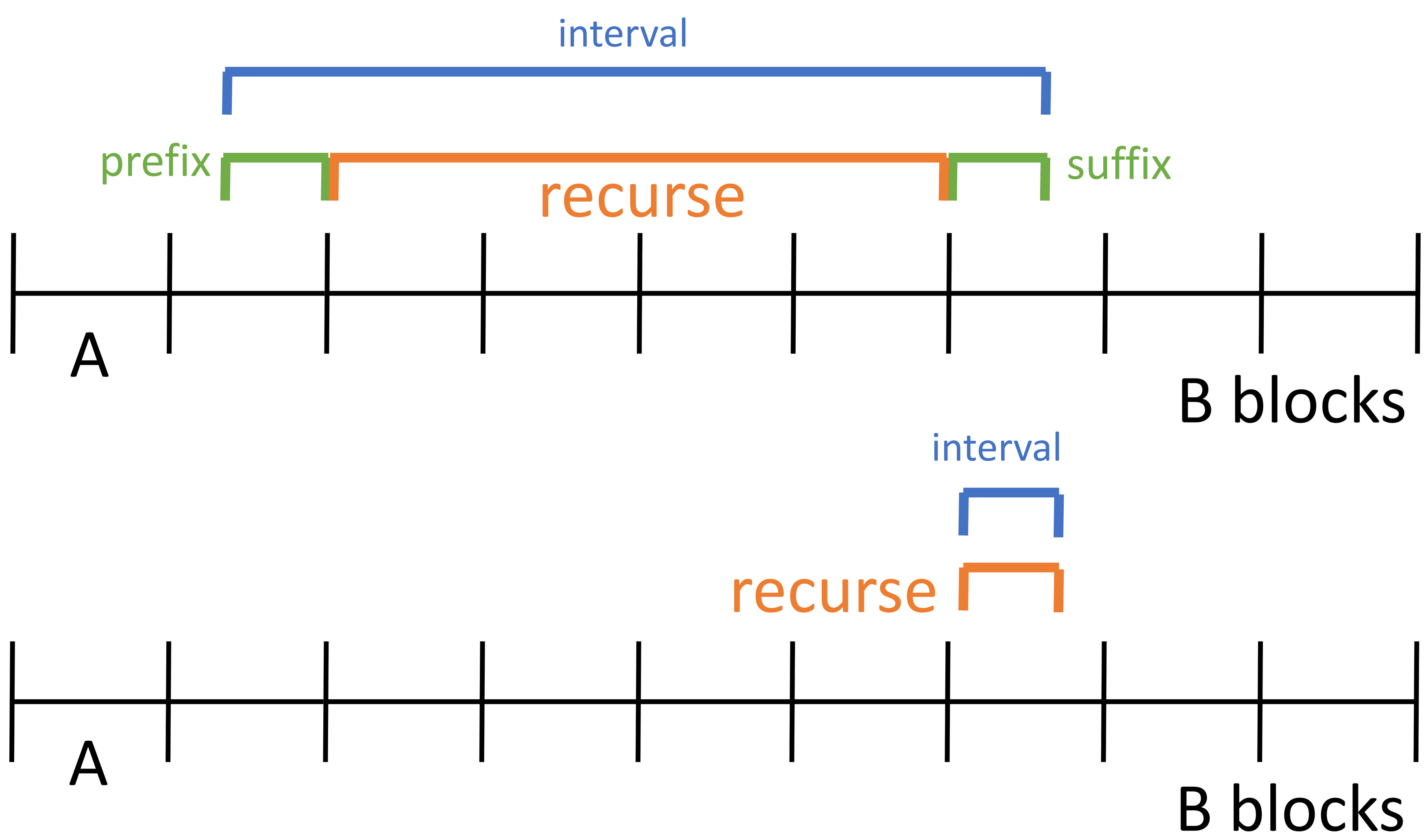
Gate p with interval A. Gate c with interval B.
$\oplus$-gate: replace child c with c $\otimes$ SG(A\B)

Gate p_l with interval A_l. Gate p_r with interval A_r.
Gate c_l with interval B_l. Gate c_r with interval B_r.
$\otimes$-gate: replace child c_l with $c_l \otimes$ SG($A_l \backslash B_l$)
replace child c_r with $c_r \otimes$ SG($A_r \backslash B_r$)

Semigroup Range-Sum

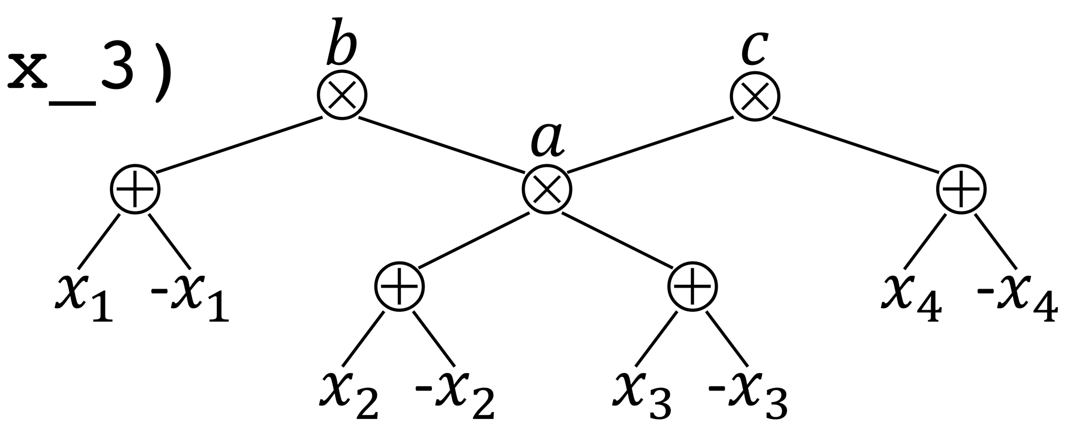
Given n variables defined over a semigroup and m intervals, the sum of each interval can be computed in time O(m · α(m, n)) [Chazelle and Rosenberg, 1989].

Split N into B blocks of size A



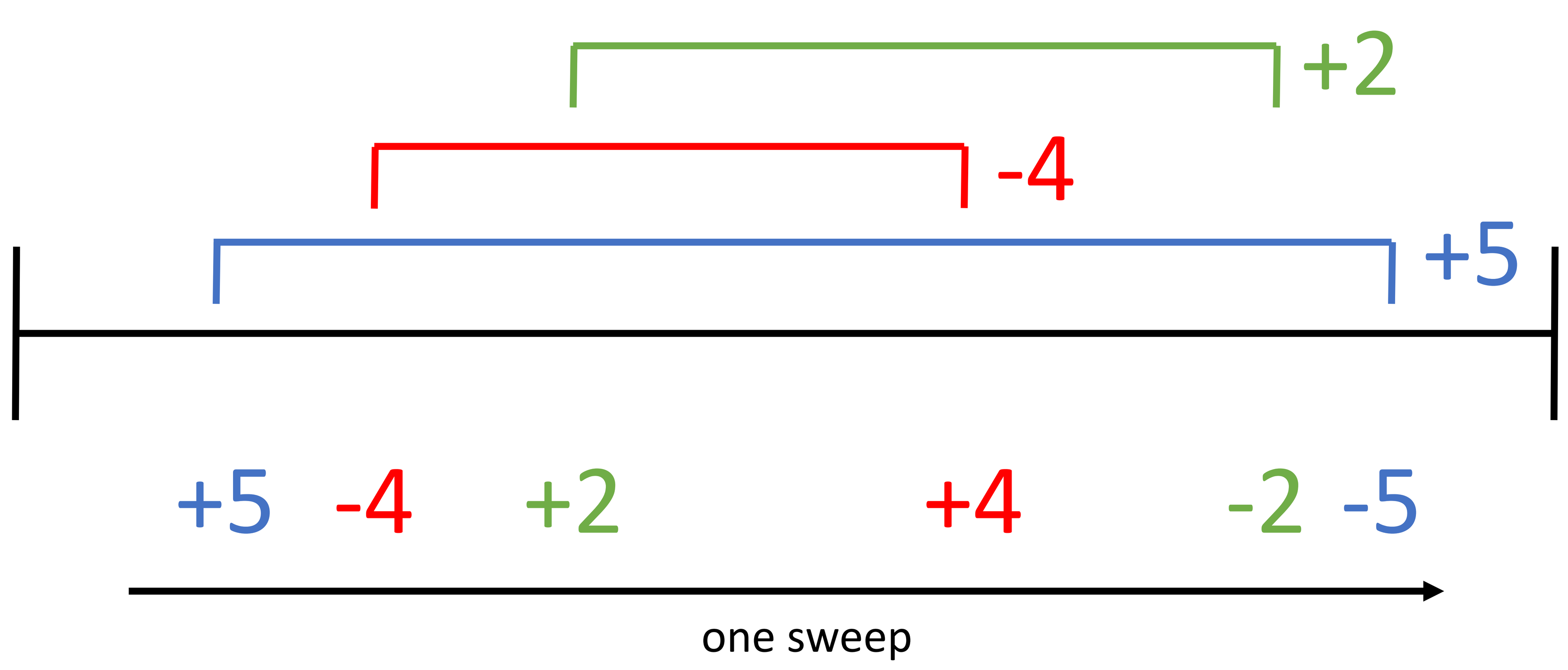
Trace additions using circuits

```
a = z(x_2) + z(x_3)
b = a + z(x_1)
c = a + z(x_4)
output b, c
```



All-Marginals by range increments

1. Backpropagate to get marginals for node
2. Update a range of missing variables



Efficient smoothing / all-marginals

Table 2: Experiments on smoothing hand-crafted circuits and experiments on computing All-Marginals as part of the collapsed sampling algorithm. Sizes are reported in thousands (k).

(a) Time (in seconds) taken to smooth circuits.

Size	Naive	Ours	Speedup ×
40k	0.82 ± 0.01	0.04 ± 0.01	21 ± 1
416k	50 ± 0.3	0.31 ± 0.01	161 ± 6
1,620k	293 ± 2	0.74 ± 0.04	390 ± 30
8,500k	6050 ± 20	4.13 ± 0.09	1470 ± 40

(b) Number of $\oplus, \ominus, \otimes, \oslash$ operations to compute All-Marginals when sampling the Segmentation-11 network.

Size	Naive	Ours	Impr %
100k	$28{,}494 \pm 598$	$20{,}207 \pm 411$	29 ± 3
200k	$55{,}875 \pm 1{,}198$	$36{,}101 \pm 1{,}522$	35 ± 5
400k	$86{,}886 \pm 6{,}330$	$56{,}094 \pm 817$	35 ± 6