KU LEUVEN


# Applied ordering of Sentential Decision Diagrams

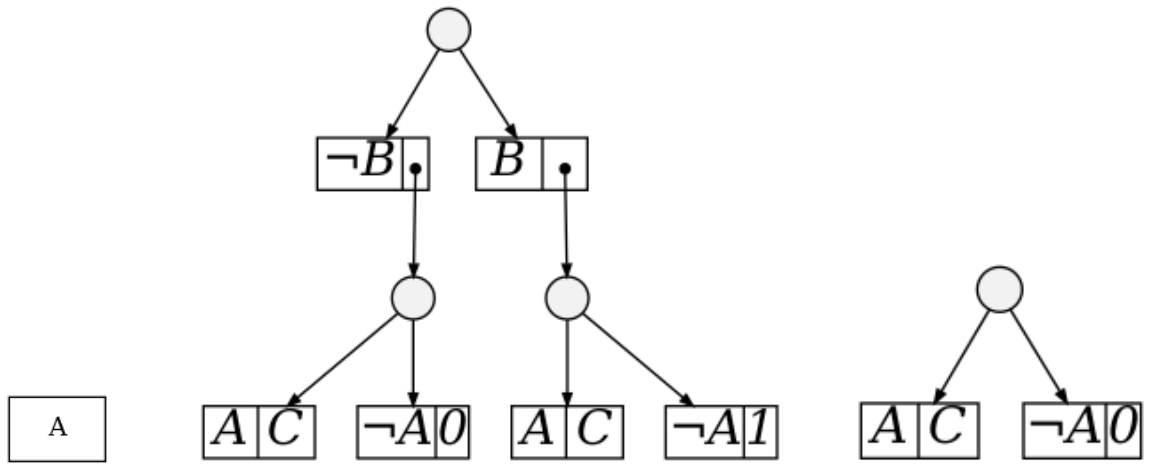
Gijs Vliegen - r0810400

March 27, 2024

# Onderzoeksprobleem - apply ordening

Is er een slimme ordening van de apply operaties bij de bottom-up compilatie, en kan die ordening efficiënt gevonden worden?

$$\underset{f_1}{(A \vee B)} \quad \wedge \quad \underset{f_2}{(A)} \quad \wedge \quad \underset{f_3}{(\neg A \vee C)}$$




$f_1 \wedge f_2$ $\qquad$ $f_1 \wedge f_3$ $\qquad$ $f_1 \wedge f_2 \wedge f_3$

# Formele probleemstelling

**Probleem**: $\alpha = \bigcirc_{i=1}^{N} \alpha_i$.
**Compilatie van** $\alpha$: sequentie van SDD's $\gamma_1, \cdots \gamma_M$, met $\gamma_M \equiv \alpha$ en voor elke $i \in [M]$ :

1. $\gamma_i \equiv \alpha_j$, of
2. $\gamma_i = \mathit{Apply}(\gamma_j, \gamma_k, \circ)$ met $\gamma_j$ en $\Sigma_k$ hebben dezelfde vtree, of
3. $\gamma_i \equiv \gamma_j$ met $j < i$ en de vtrees van $\gamma_i$ en $\gamma_j$ verschillen.

**Performantiemetrieken**
Het geheugengebruik $g$:

$$g = \max_{1 \leq i \leq N} \{|\gamma_i|\}$$

De totale compilatietijd $ct$:

$$ct = \sum_{i=1}^{M} t(\gamma_j)$$

Met $t(\gamma_j)$ de tijd nodig om $\gamma_j$ te berekenen.

# Experimenten

**Probleem**: $\alpha = \bigcirc_{i=1}^{N} \alpha_i$.

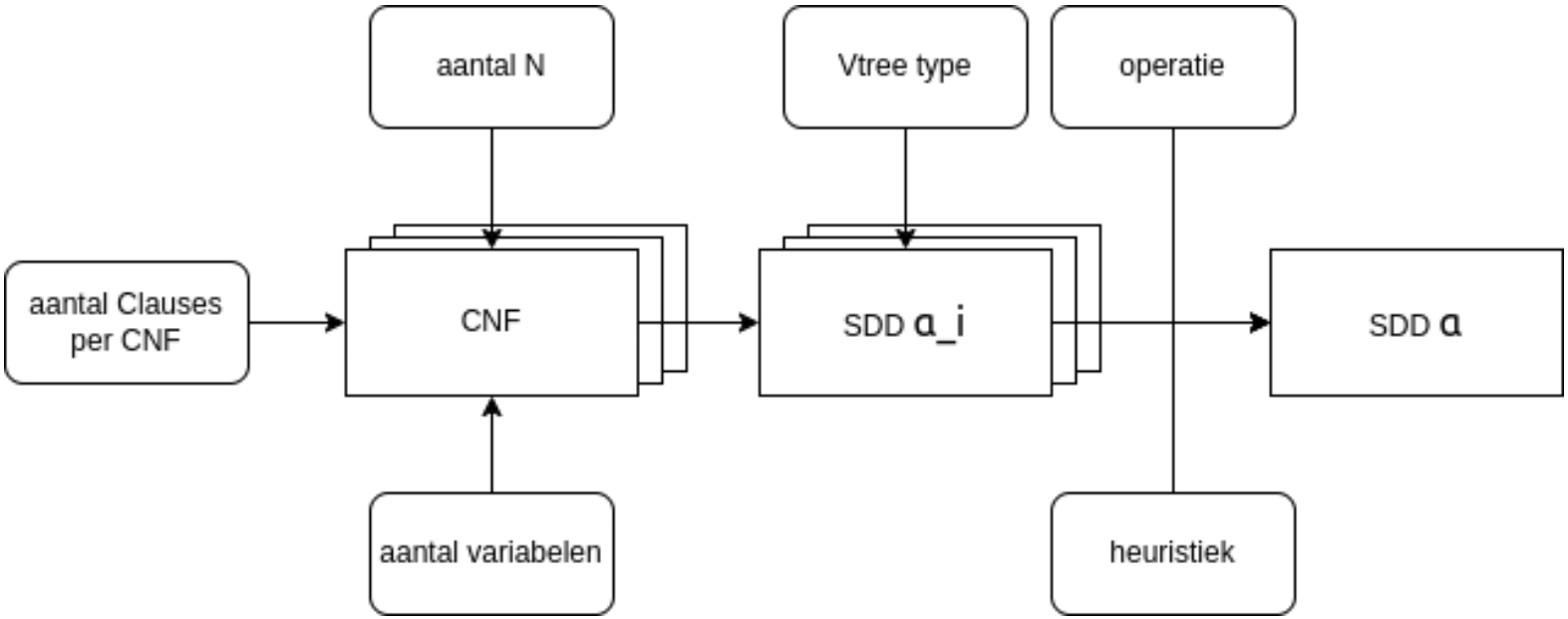



Figure: Experiment-diagram

# Variatie van willekeurige volgordes

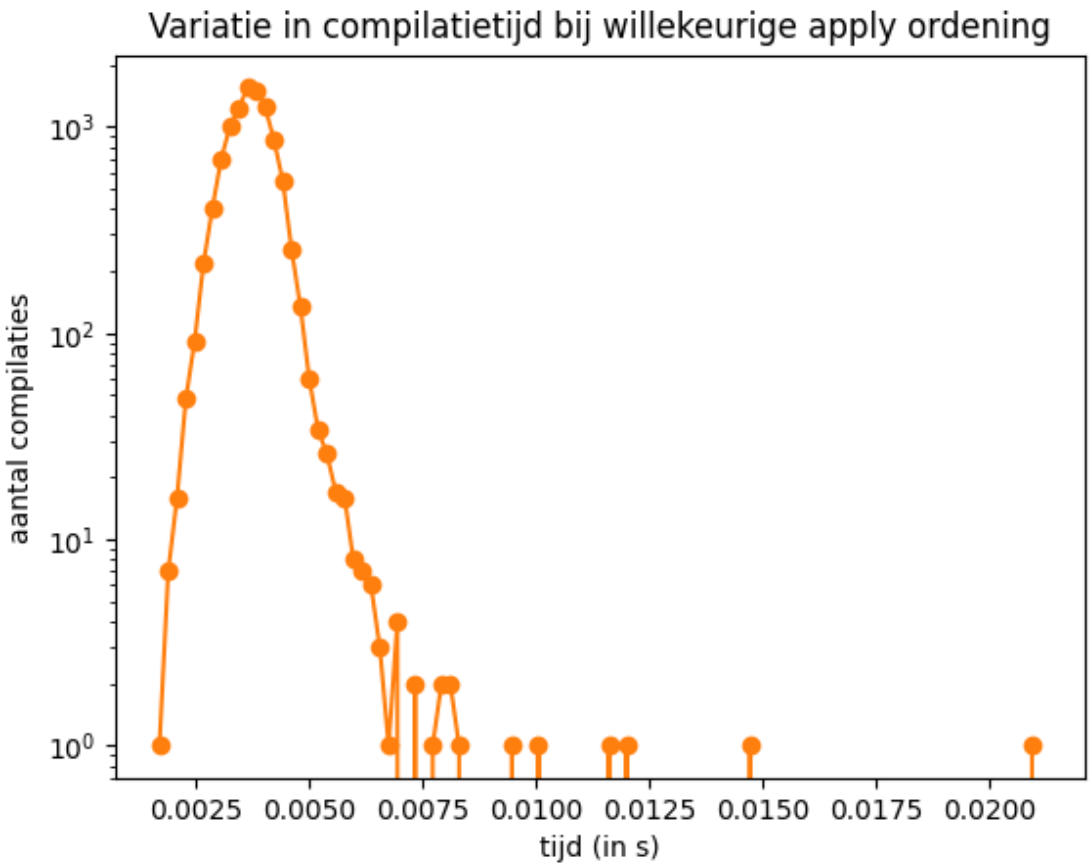



Figure: Variatie voor 5 clauses per CNF

# Variatie van willekeurige volgordes

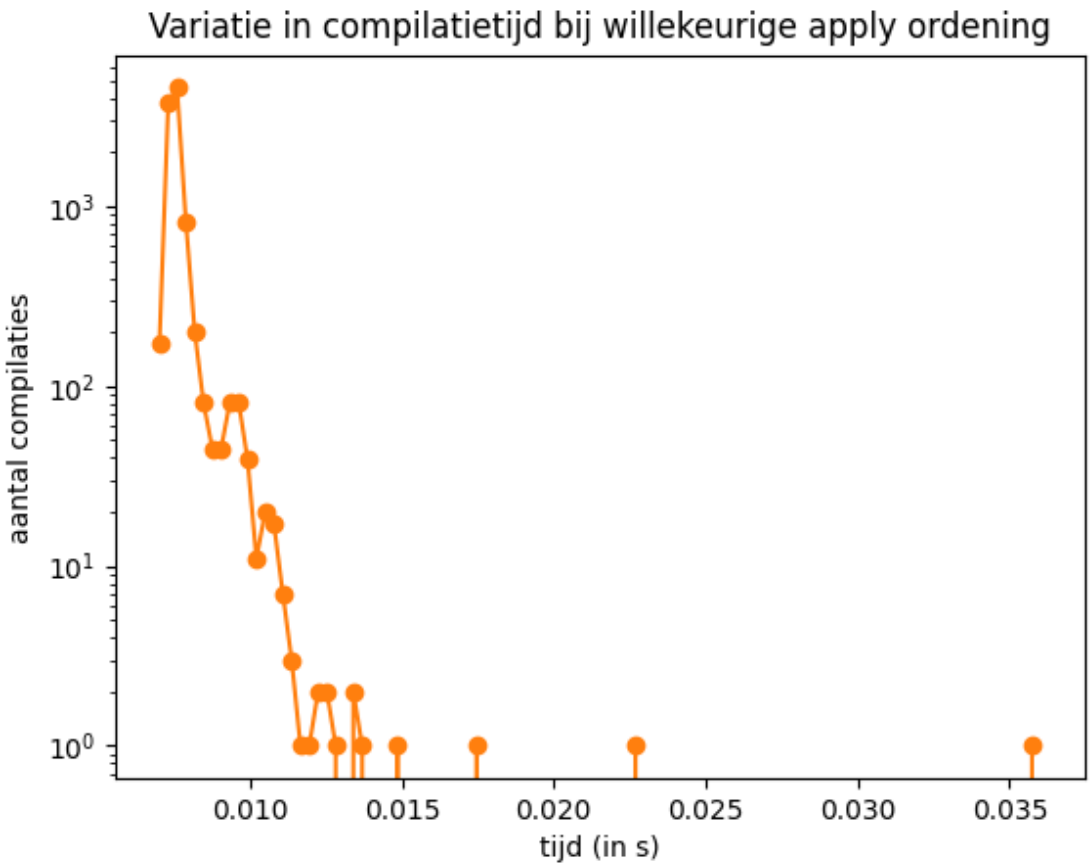



Figure: Variatie voor 20 clauses per CNF

# Variatie van willekeurige volgordes

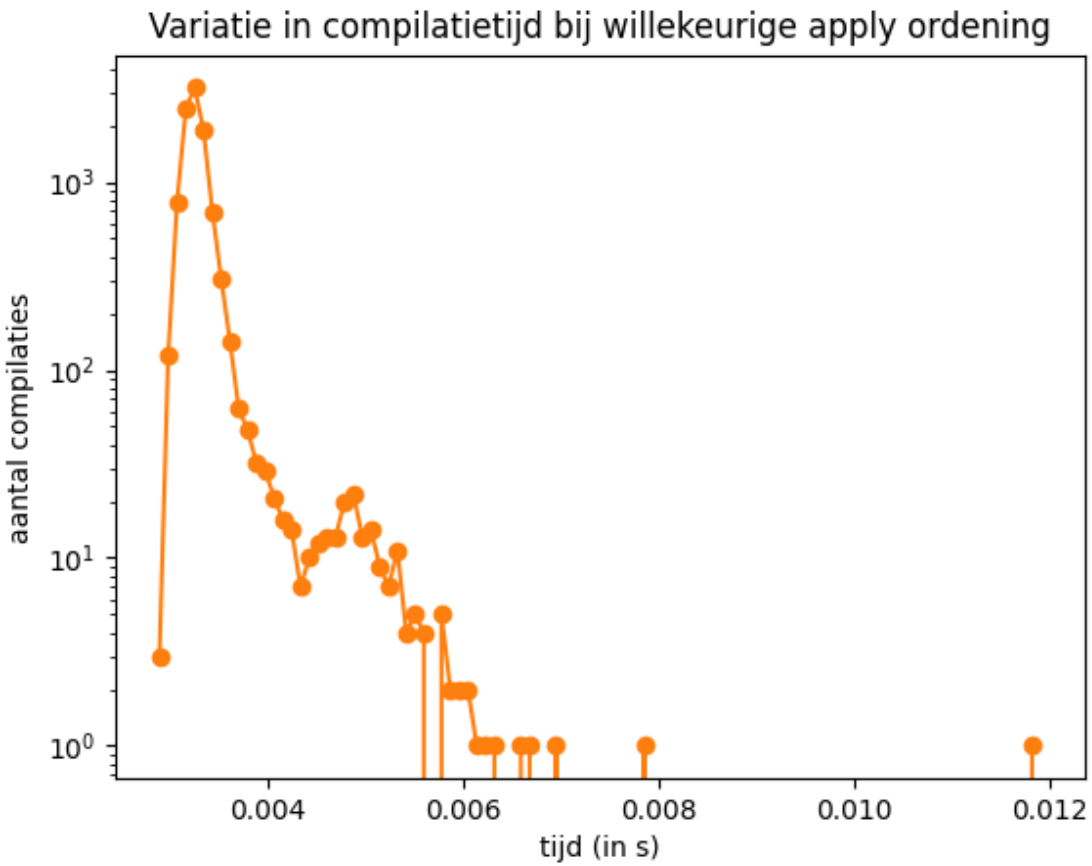



Figure: Variatie voor 35 clauses per CNF

# Heuristieken: vorige presentatie

- Heuristiek 1: kleinste SDD eerst.
  geïnspireerd op bovengrens na apply: $|A \circ B|$ is $O(|A| * |B|)$
- Heuristiek 2: recursief splitsen in left, right & middle adhv. vtree [DKL+21]

# Heuristiek 3

Apply-operatie van SDD $\alpha$ met variabele $x$.

$$\alpha = \bigvee_{i=1}^{n}(p_i \wedge s_i) \qquad \begin{array}{l} x \in v_l \rightarrow x = (x \wedge \top) \vee (\neg x \wedge \bot) \\ x \in v_r \rightarrow x = (\top \wedge x) \end{array}$$

| $\alpha \circ x$ | $x \in v_l$ | $x \in v_r$ |
|---|---|---|
| AND | $\bigvee_{i=1}^{n}((p_i \wedge x) \wedge s_i) \vee (\neg x \wedge \bot)$ | $\bigvee_{i=1}^{n} p_i \wedge (s_i \wedge x)$ |
| OR | $\bigvee_{i=1}^{n}((p_i \wedge \neg x) \wedge s_i) \vee (x \wedge \top)$ | $\bigvee_{i=1}^{n} p_i \wedge (s_i \vee x)$ |

# Heuristiek 3

Variabelenvolgorde breedte-eerst over vtree. Sdd's applyen adhv. welke variabelen ze bevatten.

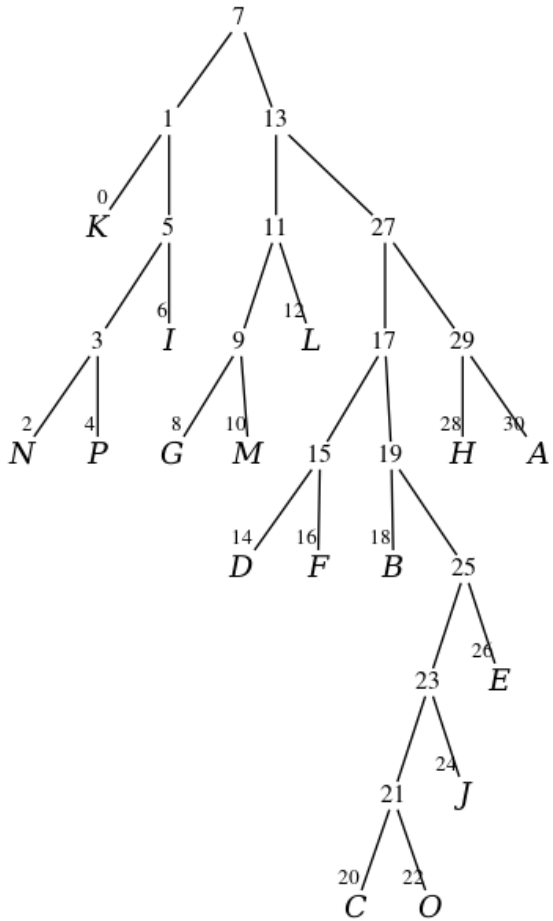



Figure: Willekeurige vtree

# Heuristiek 4

Betere bovengrens:

$$|\alpha| = \sum_{v \in V} \#el(\alpha, v)$$

$$\#el(\gamma_i \circ \gamma_j, v) \leq \#el(\gamma_i, v) \times \#el(\gamma_j, v)$$

Met $\#el(X, v)$ het aantal elementen in SDD $X$ voor vtreeknoop $v$. Extra bovengrens dmv. in termen van de primes en in termen van de subs.

# Resultaten

## Resultaten van heuristiek 1+2 en 3 met OR operatie.

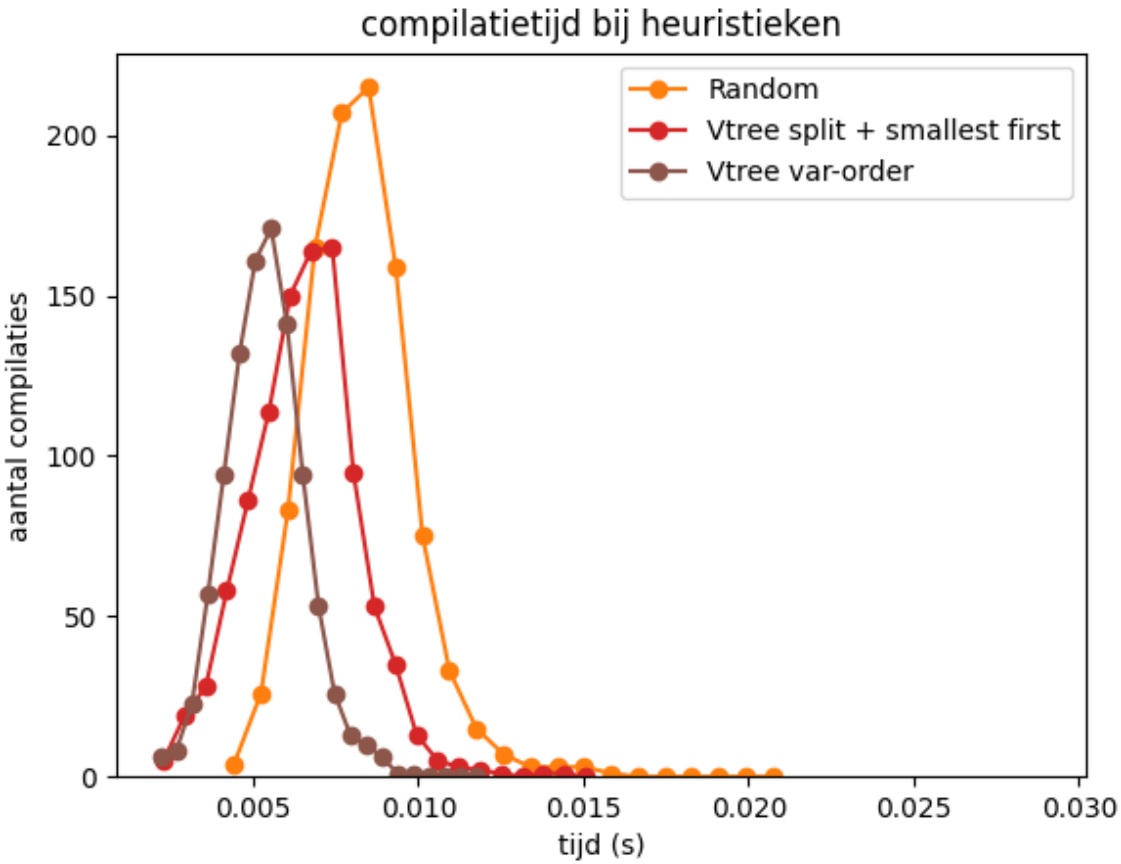



Figure: Compilatietijd voor 5 clauses per CNF

# Resultaten

## Resultaten van heuristiek 1+2 en 3 met OR operatie.

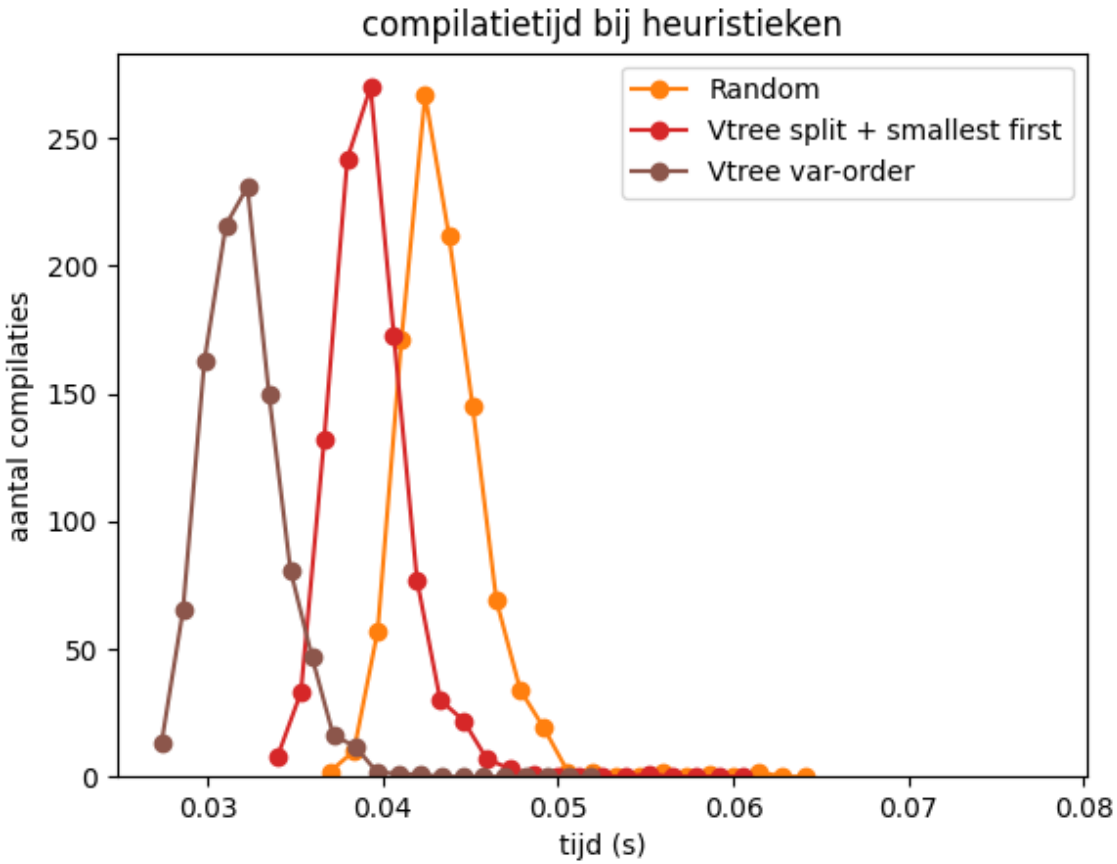



Figure: Compilatietijd voor 20 clauses per CNF

# Resultaten

## Resultaten van heuristiek 1+2 en 3 met OR operatie.

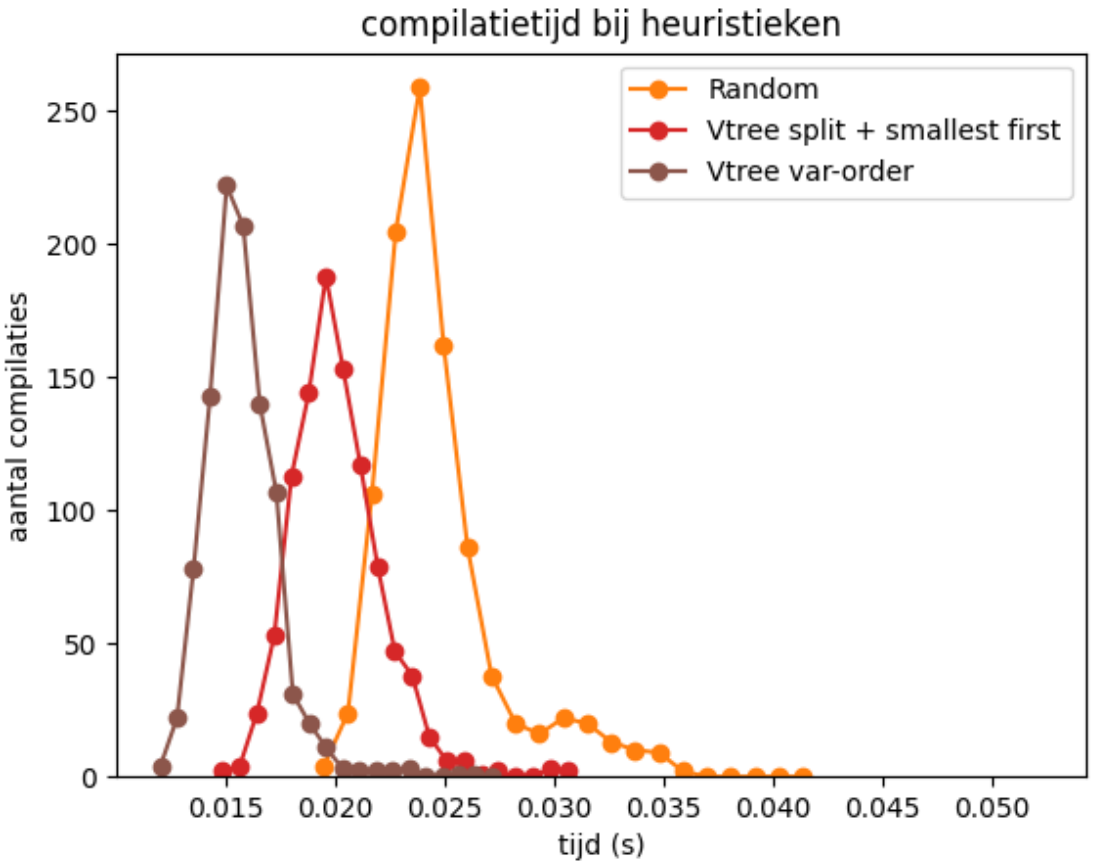



Figure: Compilatietijd voor 35 clauses per CNF

# Resultaten

## Resultaten van heuristiek 1+2 en 3 met OR operatie.

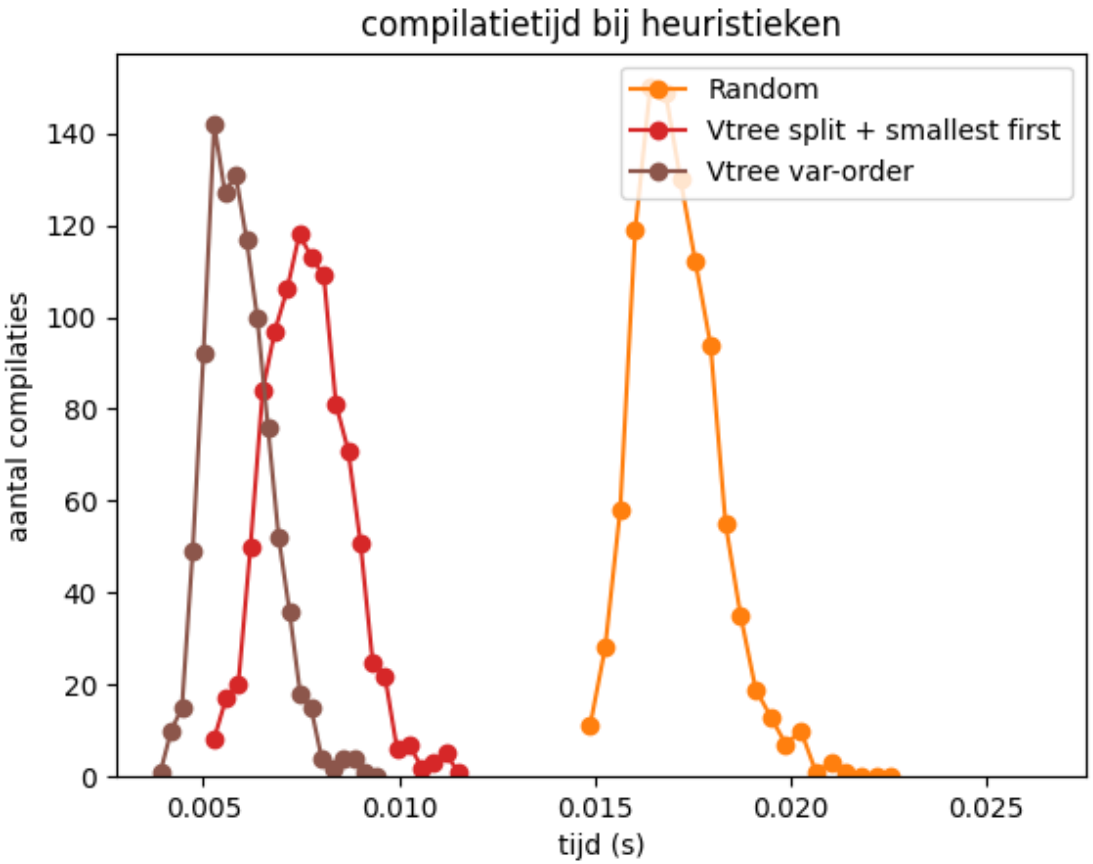



Figure: Compilatietijd voor 50 clauses per CNF

# Resultaten

## Resultaten van heuristiek 1+2 en 3 met OR operatie.

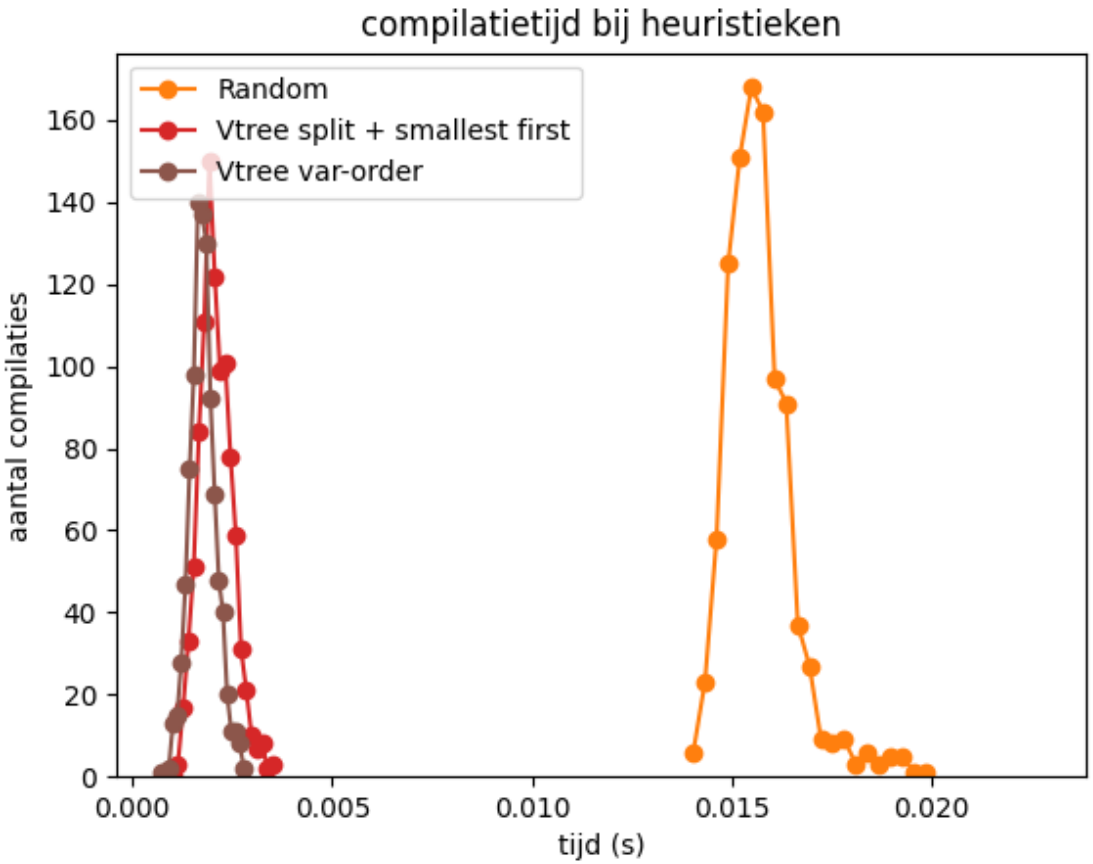



Figure: Compilatietijd voor 65 clauses per CNF

# Planning resterende maanden

- Overhead heuristiek 4 verminderen
- Extra bovengrens voor heuristiek 4
- Experimenten op grotere schaal uitvoeren
- Thesistekst afwerken


Fadi A. Aloul, Igor L. Markov, and Karem A. Sakallah.
Mince: A static global variable-ordering heuristic for sat search and bdd manipulation.
*J. Univers. Comput. Sci.*, 10:1562–1596, 2004.


Arthur Choi and Adnan Darwiche.
Dynamic minimization of sentential decision diagrams.
*Proceedings of the AAAI Conference on Artificial Intelligence*, 2013.


Arthur Choi and Adnan Darwiche.
Sdd beginning-user manual.
2018.


Arthur Choi, Doga Gizem Kisa, and Adnan Darwiche.
Compiling probabilistic graphical models using sentential decision diagrams.
In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, 2013.


Adnan Darwiche.
Sdd: A new canonical representation of propositional knowledge bases.

In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 819–826, 2011.


Alexis de Colnet and Stefan Mengel.
Lower bounds on intermediate results in bottom-up knowledge compilation.
*ArXiv*, abs/2112.12430, 2021.


Vincent Derkinderen.
The role of counting in probabilistic reasoning, 2023.


Meihua Dang, Pasha Khosravi, Yitao Liang, Antonio Vergari, and Guy Van den Broeck.
Juice: A julia package for logic and probabilistic circuits.
In *AAAI Conference on Artificial Intelligence*, 2021.


Nina Narodytska and Toby Walsh.
Constraint and variable ordering heuristics for compiling configuration problems.
In *International Joint Conference on Artificial Intelligence*, 2007.


Umut Oztok and Adnan Darwiche.
A top-down compiler for sentential decision diagrams.

In *International Joint Conference on Artificial Intelligence*, 2015.

Jonas Vlasselaer, Joris Renkens, Guy Van den Broeck, and Luc De Raedt.
Compiling probabilistic logic programs into sentential decision diagrams.
2014.