To Link or not to Link?

Robin Coutelier

TU Wien, Vienna, Austria
robin.coutelier@tuwien.ac.at

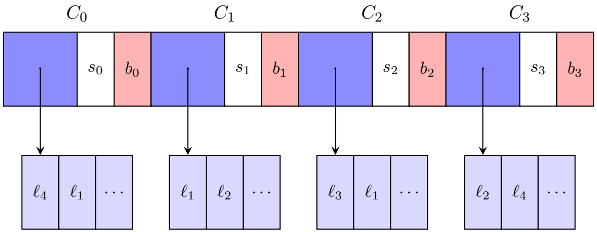
August 20th 2024

Acknowledgements

We thank the reviewers for their valuable feedback. As a reviewer pointed out, a similar approach was presented in [Bie08] and [MMZ+01]. This work is therefore not novel, but rather an independent re-discovery of an existing method.

The author acknowledges support from the ERC Consolidator Grant ARTIST 101002685; the TU Wien Doctoral College TrustACPS; the FWF SpyCoDe SFB projects F8504; the WWTF Grant ForSmart 10.47379/ICT22007.

Clauses in NapSAT



C_0
C_1
C_2
C_3
s_0
b_0
s_1
b_1
s_2
b_2
s_3
b_3
ℓ_4 ℓ_1 $\cdots$
ℓ_1 ℓ_2 $\cdots$
ℓ_3 ℓ_1 $\cdots$
ℓ_2 ℓ_4 $\cdots$

Figure: Representation of clauses in NapSAT. Clauses are stored as a fixed size structure containing a pointer to the literals, the size s of the clause, and a blocker literal b.

Array Representation of Watch Lists in NapSAT

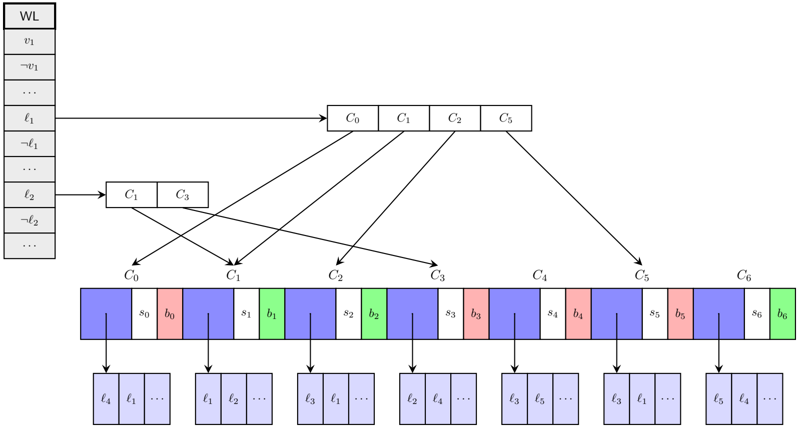

WL
v_1
$\neg v_1$
$\cdots$
ℓ_1
$\neg \ell_1$
$\cdots$
ℓ_2
$\neg \ell_2$
$\cdots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\cdots$
ℓ_1 ℓ_2 $\cdots$
ℓ_3 ℓ_1 $\cdots$
ℓ_2 ℓ_4 $\cdots$
ℓ_3 ℓ_5 $\cdots$
ℓ_3 ℓ_1 $\cdots$
ℓ_5 ℓ_4 $\cdots$

Figure: Array-based representation of watch lists. The watch list of ℓ_1 is $\{C_0, C_1, C_2, C_5\}$ and the watch list of ℓ_2 is $\{C_1, C_3\}$.

Array Representation of Watch Lists in MiniSAT

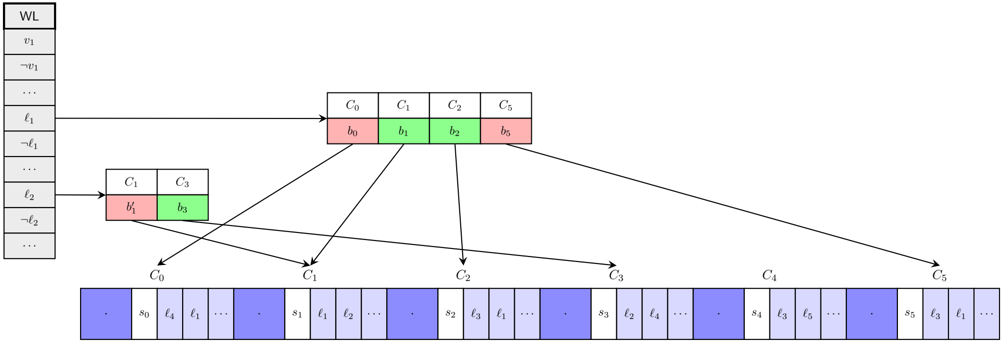

WL
v_1
$\neg v_1$
$\dots$
ℓ_1
$\neg \ell_1$
$\dots$
ℓ_2
$\neg \ell_2$
$\dots$
C_0 C_1 C_2 C_5
b_0 b_1 b_2 b_5
C_1 C_3
b'_1 b_3
C_0 C_1 C_2 C_3 C_4 C_5
s_0 ℓ_4 ℓ_1 $\dots$ s_1 ℓ_1 ℓ_2 $\dots$ s_2 ℓ_3 ℓ_1 $\dots$ s_3 ℓ_2 ℓ_4 $\dots$ s_4 ℓ_3 ℓ_5 $\dots$ s_5 ℓ_3 ℓ_1 $\dots$

Figure: Array-based representation of watch lists in MiniSAT. The blockers are attached to the watch list.

Linked List Representation of Watch Lists

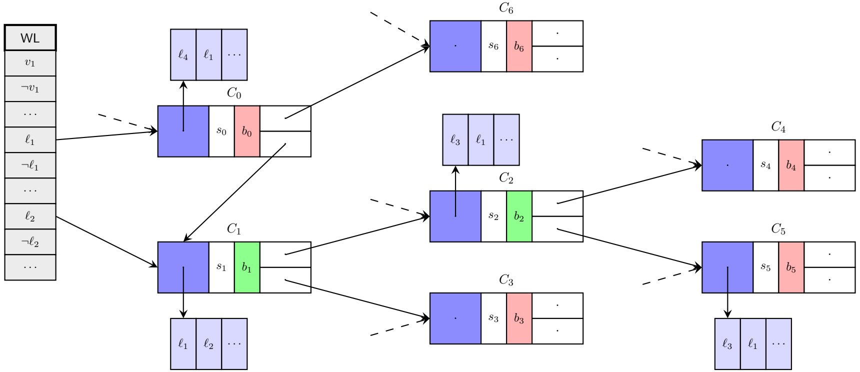


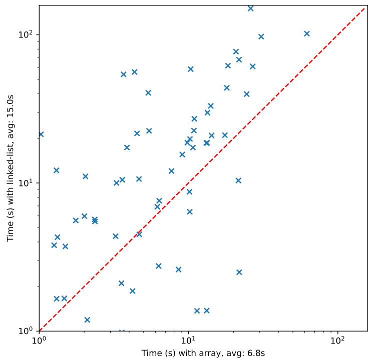
Figure: Representation of watch lists using the linked list data structure. The watch list of ℓ_1 is $\{C_0, C_1, C_2, C_5\}$ and the watch list of ℓ_2 is $\{C_1, C_3\}$.

Why Linked Lists?

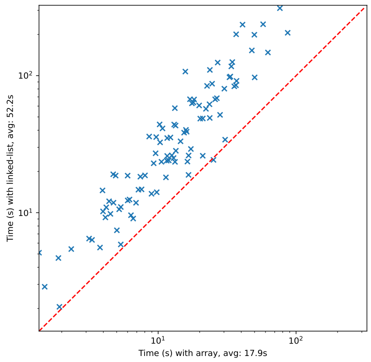
Table: Intuitive comparison of the array and linked list representation of watch lists.

Aspect	**Array**		**Linked list**	
Dereference level	2 levels	$(-)$	1 level	$(+)$
Memory usage	Extensible	$(-)$	Fixed	$(+)$
Insertion	$O(1)$ or $O(n)$	$(-)$	$O(1)$	$(+)$
Bookkeeping overhead	Low	$(+)$	High	$(-)$
Code complexity	Low	$(+)$	Medium	$(-)$

Experiments



(a) SAT instances.
$T(\text{linked list}) = 2.21 \times T(\text{array})$



(b) UNSAT instances.
$T(\text{linked list}) = 2.91 \times T(\text{array})$

Figure: Random 3-SAT instances of the SATLIB with 250 variables.

Going through the Array-based Watch List

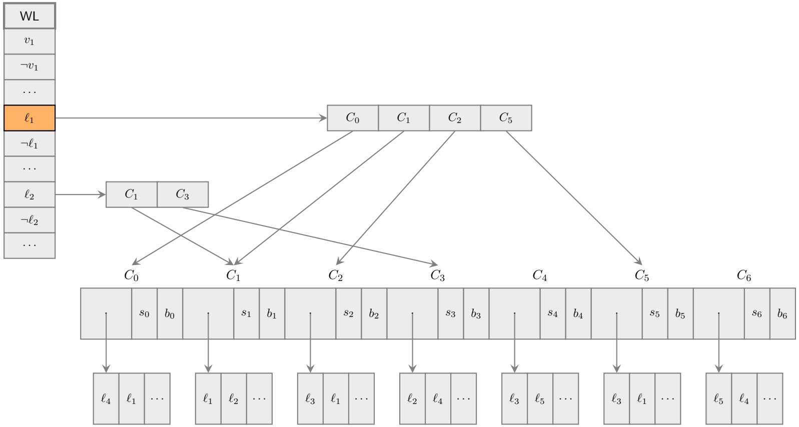
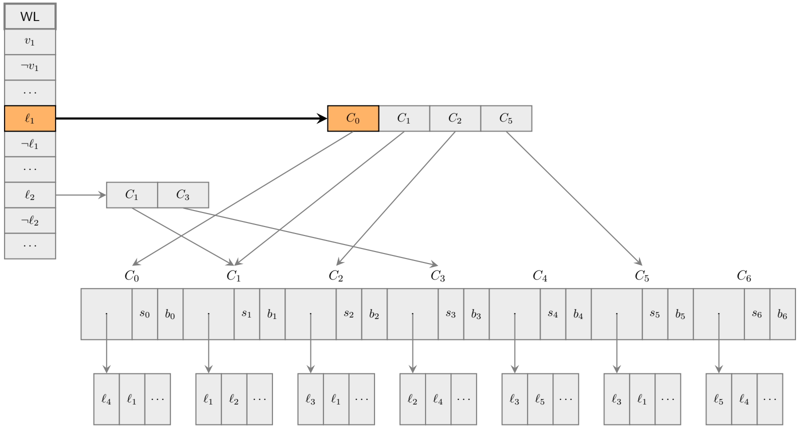

WL
v_1
$\neg v_1$
$\dots$
ℓ_1
$\neg\ell_1$
$\dots$
ℓ_2
$\neg\ell_2$
$\dots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\dots$
ℓ_1 ℓ_2 $\dots$
ℓ_3 ℓ_1 $\dots$
ℓ_2 ℓ_4 $\dots$
ℓ_3 ℓ_5 $\dots$
ℓ_3 ℓ_1 $\dots$
ℓ_5 ℓ_4 $\dots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation.

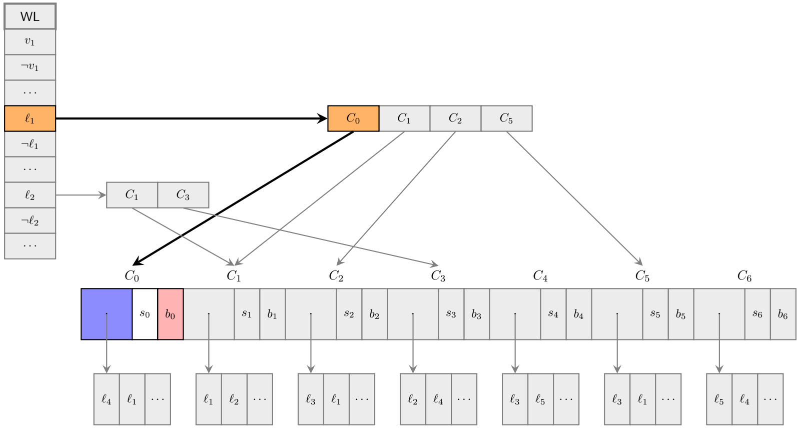
Going through the Array-based Watch List



WL
v_1
$\neg v_1$
$\cdots$
ℓ_1
$\neg \ell_1$
$\cdots$
ℓ_2
$\neg \ell_2$
$\cdots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\cdots$
ℓ_1 ℓ_2 $\cdots$
ℓ_3 ℓ_1 $\cdots$
ℓ_2 ℓ_4 $\cdots$
ℓ_3 ℓ_5 $\cdots$
ℓ_3 ℓ_1 $\cdots$
ℓ_5 ℓ_4 $\cdots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation.

Going through the Array-based Watch List



WL
v_1
$\neg v_1$
$\dots$
ℓ_1
$\neg \ell_1$
$\dots$
ℓ_2
$\neg \ell_2$
$\dots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\dots$
ℓ_1 ℓ_2 $\dots$
ℓ_3 ℓ_1 $\dots$
ℓ_2 ℓ_4 $\dots$
ℓ_3 ℓ_5 $\dots$
ℓ_3 ℓ_1 $\dots$
ℓ_5 ℓ_4 $\dots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation.

Going through the Array-based Watch List

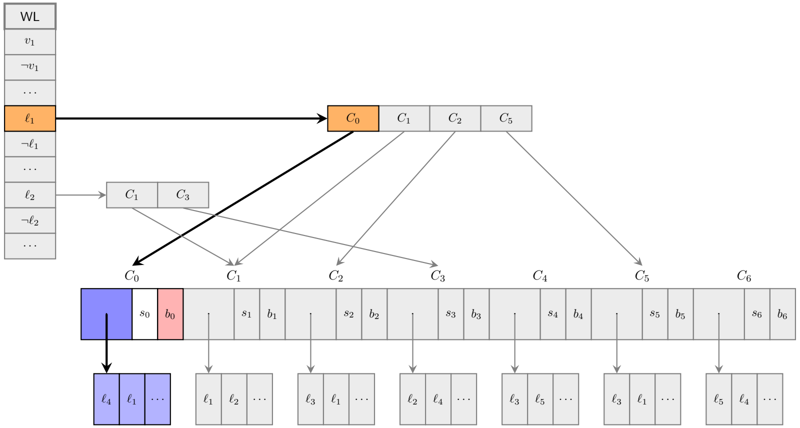

WL
v_1
$\neg v_1$
$\cdots$
ℓ_1
$\neg \ell_1$
$\cdots$
ℓ_2
$\neg \ell_2$
$\cdots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\cdots$
ℓ_1 ℓ_2 $\cdots$
ℓ_3 ℓ_1 $\cdots$
ℓ_2 ℓ_4 $\cdots$
ℓ_3 ℓ_5 $\cdots$
ℓ_3 ℓ_1 $\cdots$
ℓ_5 ℓ_4 $\cdots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation.

Going through the Array-based Watch List

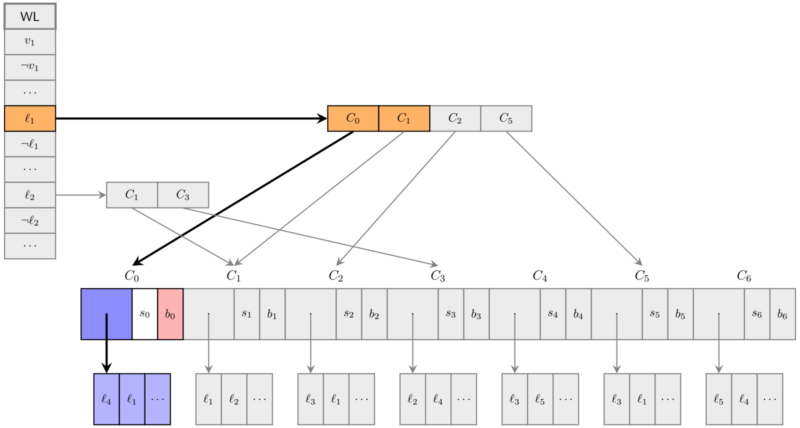
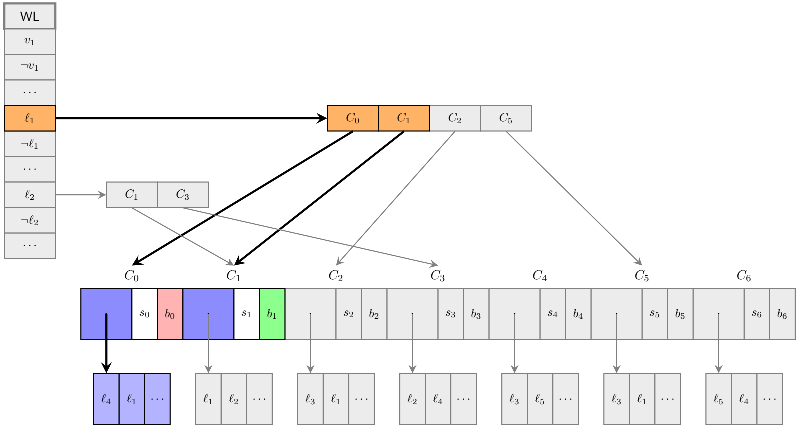

WL
v_1
$\neg v_1$
$\ldots$
ℓ_1
$\neg \ell_1$
$\ldots$
ℓ_2
$\neg \ell_2$
$\ldots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\ldots$
ℓ_1 ℓ_2 $\ldots$
ℓ_3 ℓ_1 $\ldots$
ℓ_2 ℓ_4 $\ldots$
ℓ_3 ℓ_5 $\ldots$
ℓ_3 ℓ_1 $\ldots$
ℓ_5 ℓ_4 $\ldots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation.

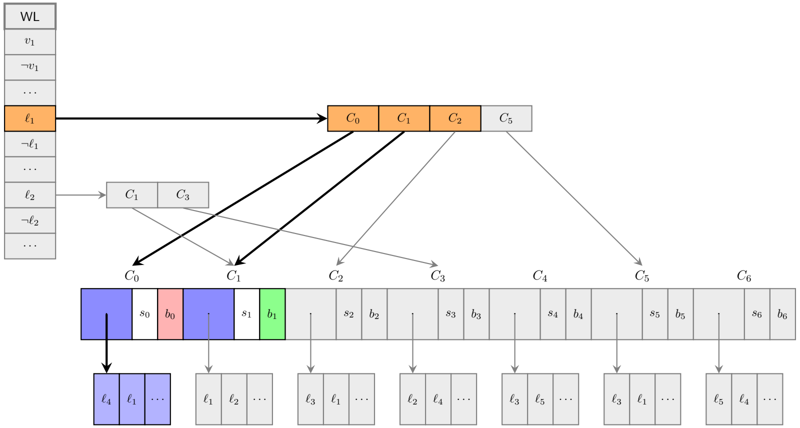
Going through the Array-based Watch List



WL
v_1
$\neg v_1$
$\dots$
ℓ_1
$\neg \ell_1$
$\dots$
ℓ_2
$\neg \ell_2$
$\dots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\dots$
ℓ_1 ℓ_2 $\dots$
ℓ_3 ℓ_1 $\dots$
ℓ_2 ℓ_4 $\dots$
ℓ_3 ℓ_5 $\dots$
ℓ_3 ℓ_1 $\dots$
ℓ_5 ℓ_4 $\dots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation.

Going through the Array-based Watch List



WL
v_1
$\neg v_1$
$\dots$
ℓ_1
$\neg\ell_1$
$\dots$
ℓ_2
$\neg\ell_2$
$\dots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\dots$
ℓ_1 ℓ_2 $\dots$
ℓ_3 ℓ_1 $\dots$
ℓ_2 ℓ_4 $\dots$
ℓ_3 ℓ_5 $\dots$
ℓ_3 ℓ_1 $\dots$
ℓ_5 ℓ_4 $\dots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation.

Going through the Array-based Watch List

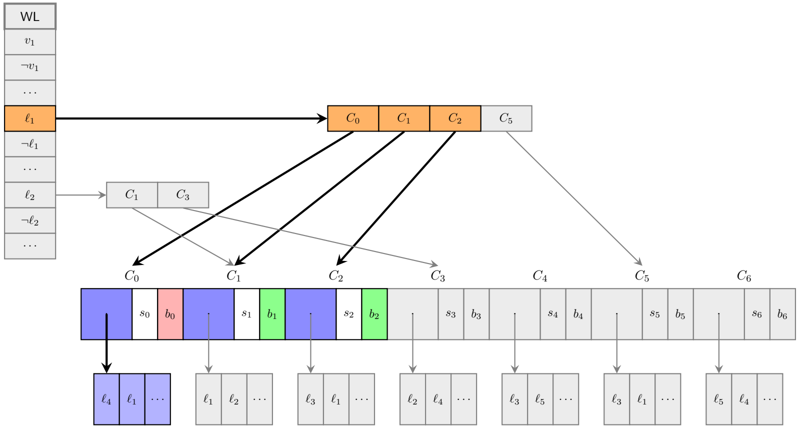

WL
v_1
$\neg v_1$
$\ldots$
ℓ_1
$\neg \ell_1$
$\ldots$
ℓ_2
$\neg \ell_2$
$\ldots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\ldots$
ℓ_1 ℓ_2 $\ldots$
ℓ_3 ℓ_1 $\ldots$
ℓ_2 ℓ_4 $\ldots$
ℓ_3 ℓ_5 $\ldots$
ℓ_3 ℓ_1 $\ldots$
ℓ_5 ℓ_4 $\ldots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation.

Going through the Array-based Watch List

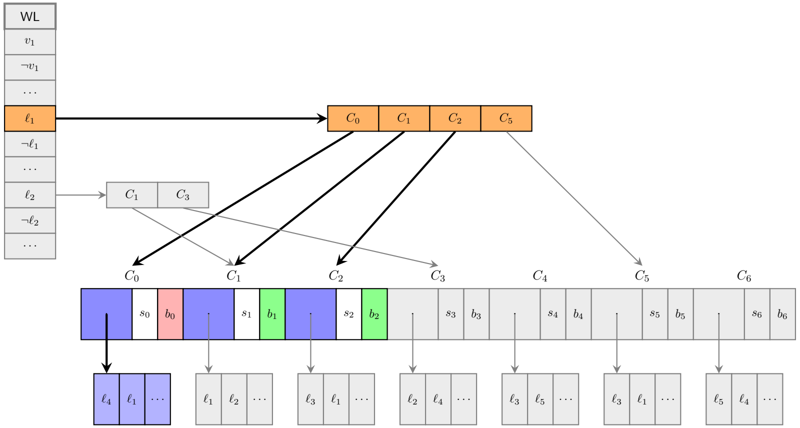

WL
v_1
$\neg v_1$
$\dots$
ℓ_1
$\neg \ell_1$
$\dots$
ℓ_2
$\neg \ell_2$
$\dots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\dots$
ℓ_1 ℓ_2 $\dots$
ℓ_3 ℓ_1 $\dots$
ℓ_2 ℓ_4 $\dots$
ℓ_3 ℓ_5 $\dots$
ℓ_3 ℓ_1 $\dots$
ℓ_5 ℓ_4 $\dots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation.

Going through the Array-based Watch List

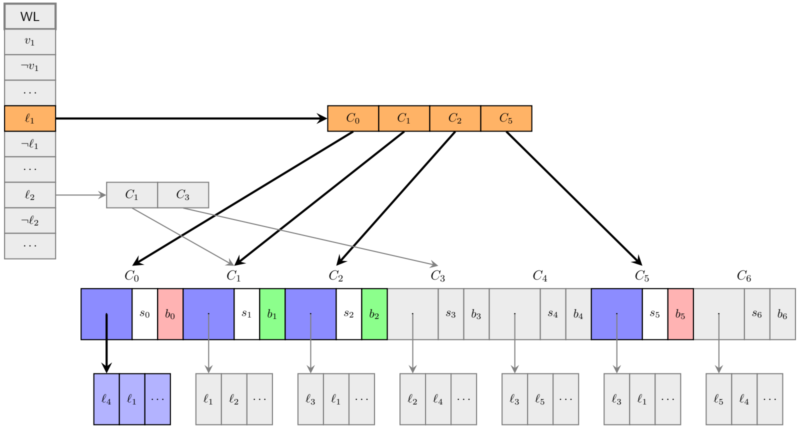

WL
v_1
$\neg v_1$
$\ldots$
ℓ_1
$\neg \ell_1$
$\ldots$
ℓ_2
$\neg \ell_2$
$\ldots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\ldots$
ℓ_1 ℓ_2 $\ldots$
ℓ_3 ℓ_1 $\ldots$
ℓ_2 ℓ_4 $\ldots$
ℓ_3 ℓ_5 $\ldots$
ℓ_3 ℓ_1 $\ldots$
ℓ_5 ℓ_4 $\ldots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation.

Going through the Array-based Watch List

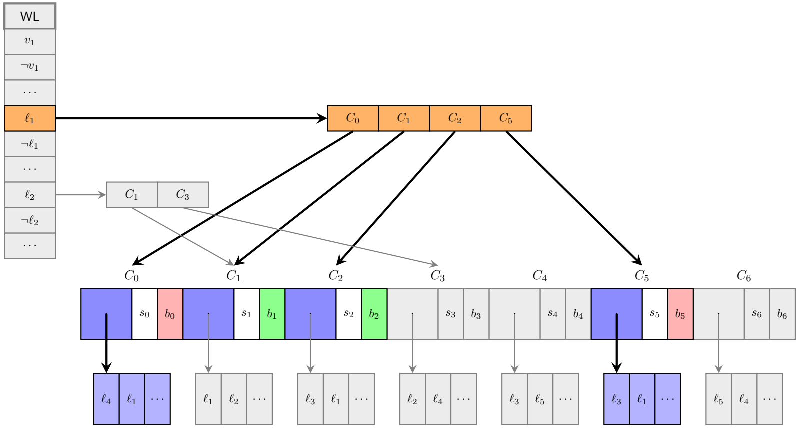

WL
v_1
$\neg v_1$
$\cdots$
ℓ_1
$\neg \ell_1$
$\cdots$
ℓ_2
$\neg \ell_2$
$\cdots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\cdots$
ℓ_1 ℓ_2 $\cdots$
ℓ_3 ℓ_1 $\cdots$
ℓ_2 ℓ_4 $\cdots$
ℓ_3 ℓ_5 $\cdots$
ℓ_3 ℓ_1 $\cdots$
ℓ_5 ℓ_4 $\cdots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation.

Going through the Linked List-based Watch List

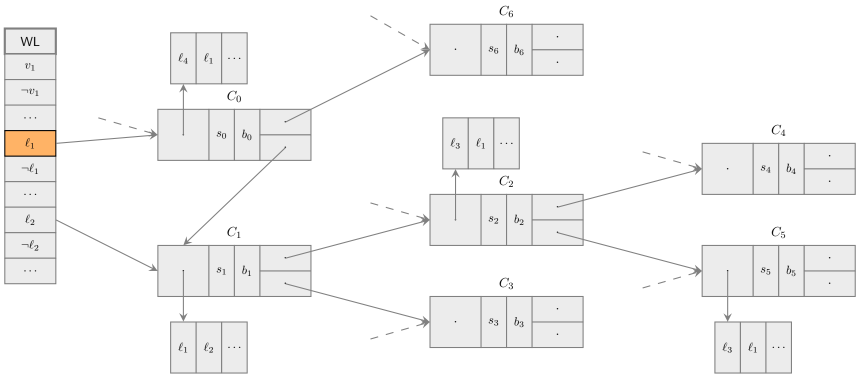


Figure: Iteration over the watch list of ℓ_1 in the linked list-based representation.

Going through the Linked List-based Watch List

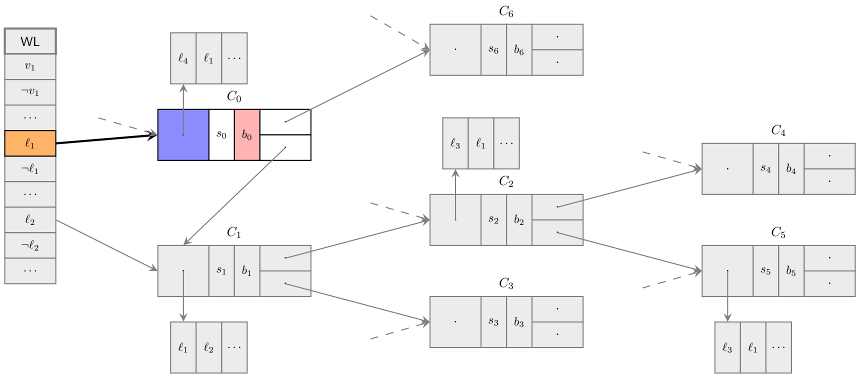


Figure: Iteration over the watch list of ℓ_1 in the linked list-based representation.

Going through the Linked List-based Watch List

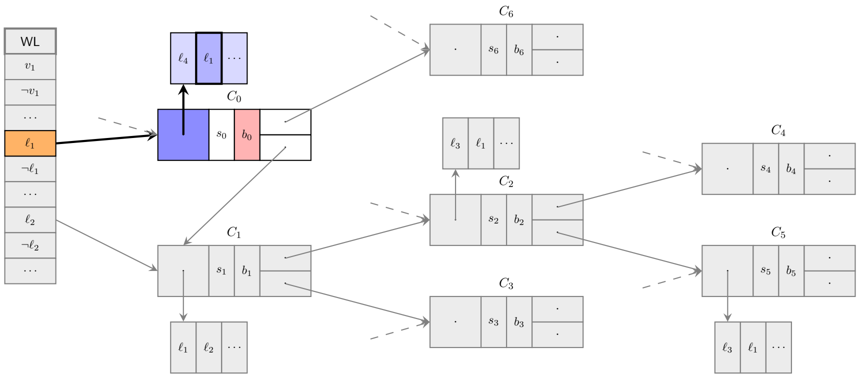


Figure: Iteration over the watch list of ℓ_1 in the linked list-based representation.

Going through the Linked List-based Watch List

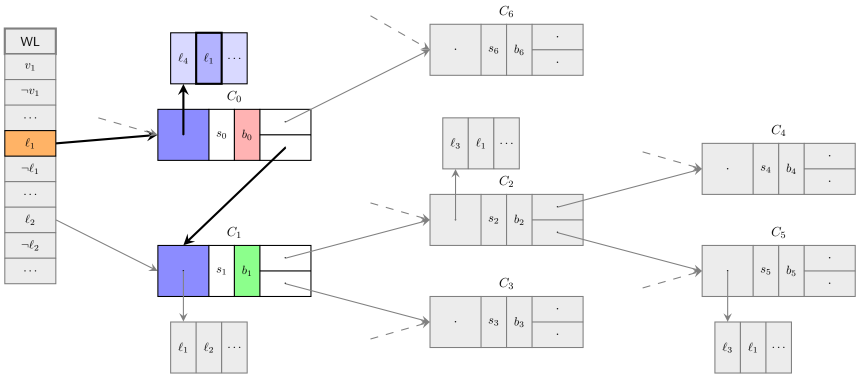


Figure: Iteration over the watch list of ℓ_1 in the linked list-based representation.

Going through the Linked List-based Watch List

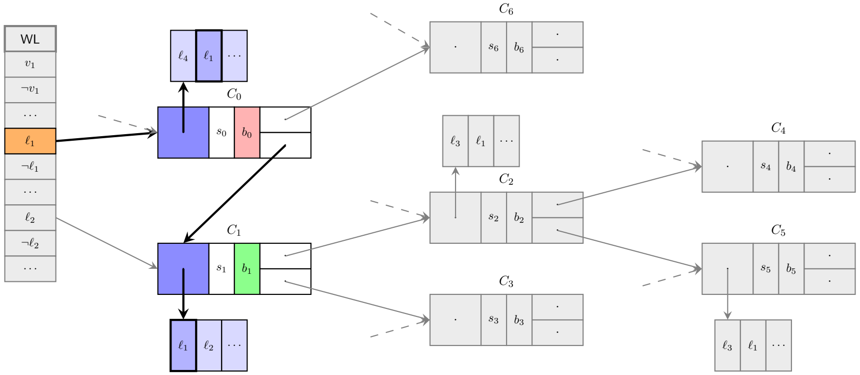


Figure: Iteration over the watch list of ℓ_1 in the linked list-based representation.

Going through the Linked List-based Watch List

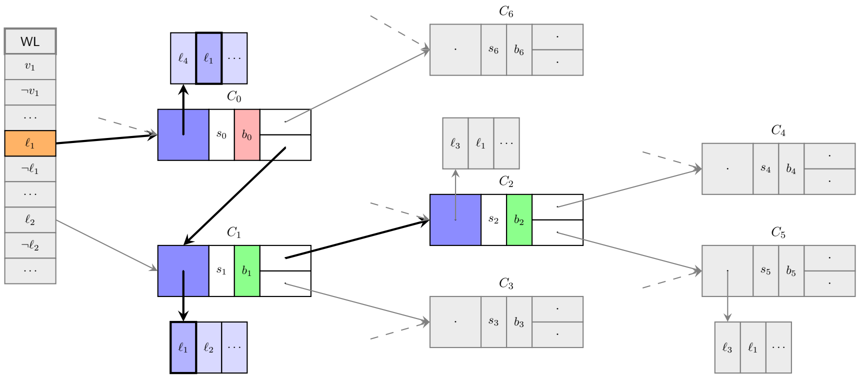


Figure: Iteration over the watch list of ℓ_1 in the linked list-based representation.

Going through the Linked List-based Watch List

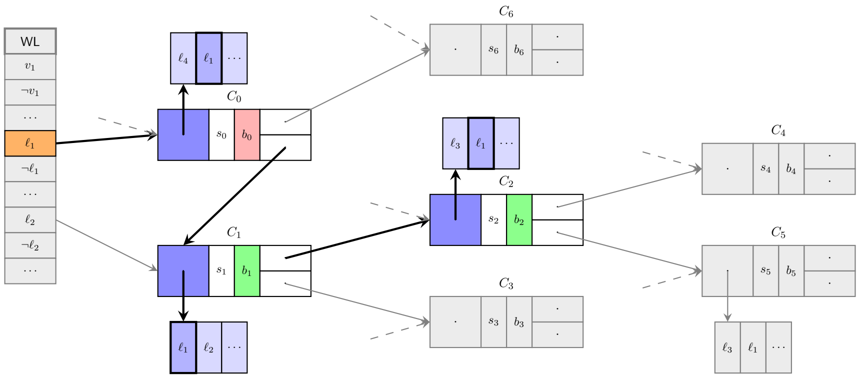


Figure: Iteration over the watch list of ℓ_1 in the linked list-based representation.

Going through the Linked List-based Watch List

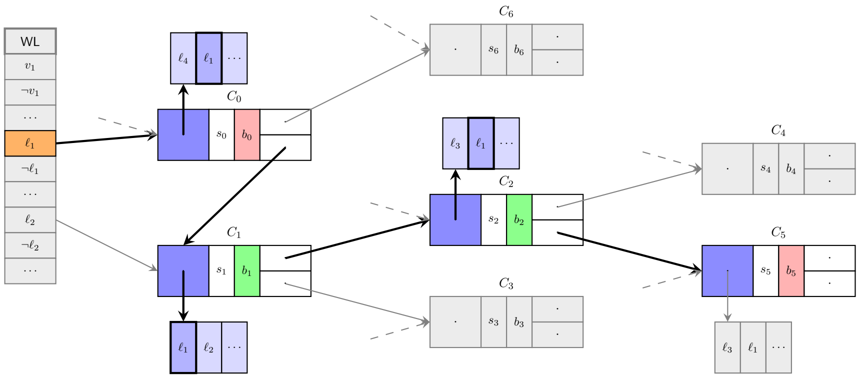


Figure: Iteration over the watch list of ℓ_1 in the linked list-based representation.

Going through the Linked List-based Watch List

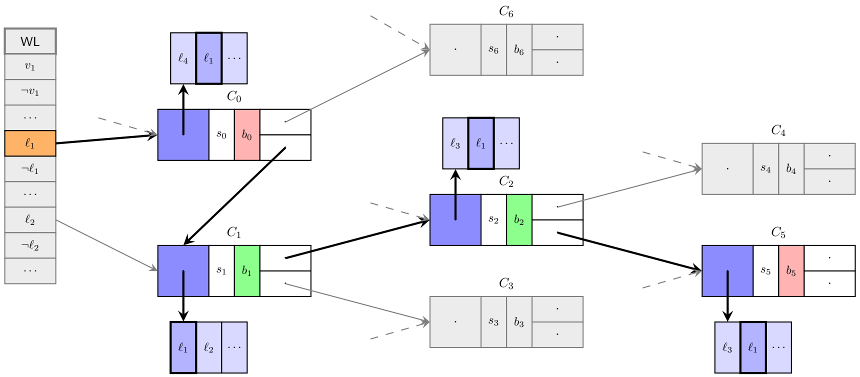


Figure: Iteration over the watch list of ℓ_1 in the linked list-based representation.

(Reminder) Going through the Array-based Watch List

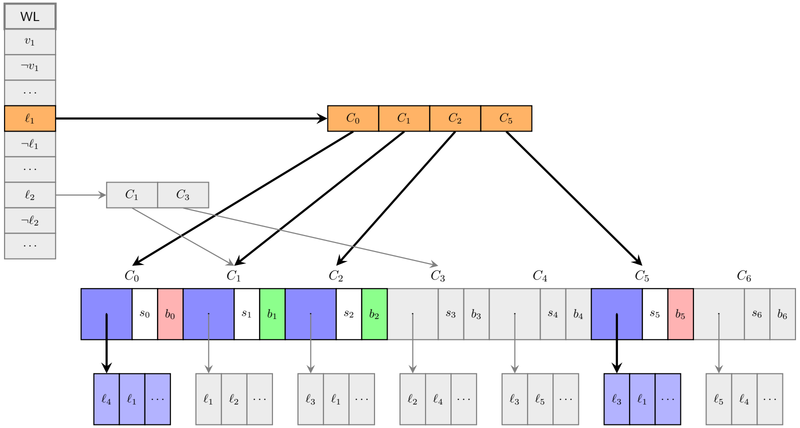

WL
v_1
$\neg v_1$
$\cdots$
ℓ_1
$\neg\ell_1$
$\cdots$
ℓ_2
$\neg\ell_2$
$\cdots$
C_0 C_1 C_2 C_5
C_1 C_3
C_0 C_1 C_2 C_3 C_4 C_5 C_6
s_0 b_0 s_1 b_1 s_2 b_2 s_3 b_3 s_4 b_4 s_5 b_5 s_6 b_6
ℓ_4 ℓ_1 $\cdots$
ℓ_1 ℓ_2 $\cdots$
ℓ_3 ℓ_1 $\cdots$
ℓ_2 ℓ_4 $\cdots$
ℓ_3 ℓ_5 $\cdots$
ℓ_3 ℓ_1 $\cdots$
ℓ_5 ℓ_4 $\cdots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation.

Does it generalize to MiniSAT?

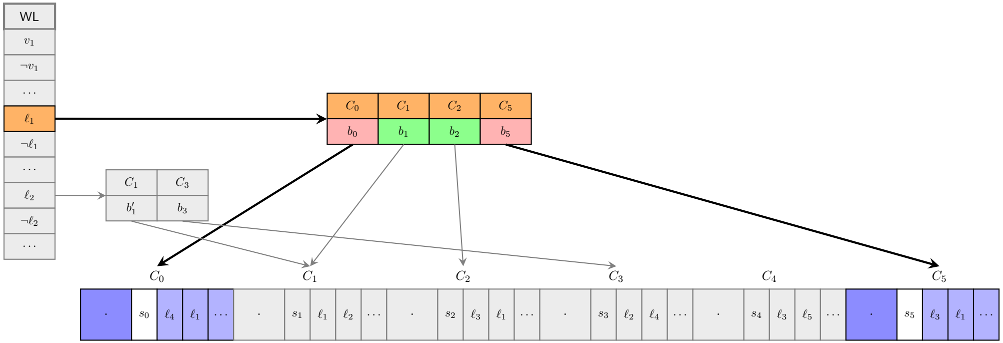

WL
v_1
$\neg v_1$
$\dots$
ℓ_1
$\neg \ell_1$
$\dots$
ℓ_2
$\neg \ell_2$
$\dots$
C_0 C_1 C_2 C_5
b_0 b_1 b_2 b_5
C_1 C_3
b'_1 b_3
C_0 C_1 C_2 C_3 C_4 C_5
s_0 ℓ_4 ℓ_1 $\dots$ s_1 ℓ_1 ℓ_2 $\dots$ s_2 ℓ_3 ℓ_1 $\dots$ s_3 ℓ_2 ℓ_4 $\dots$ s_4 ℓ_3 ℓ_5 $\dots$ s_5 ℓ_3 ℓ_1 $\dots$

Figure: Iteration over the watch list of ℓ_1 in the array-based representation (MiniSAT).

Is dereferencing so important?

Table: Comparison of the average runtime of the different watch list representations on the uniform random 3-SAT instances of the SATLIB.

	uf200	uuf200	uf225	uuf225	uf250	uuf250
Linked list	0.28 s	0.75 s	1.78 s	5.10 s	15.00 s	52.20 s
Array	0.20 s	0.44 s	1.10 s	2.63 s	6.80 s	17.92 s
Array with dereference	0.17 s	0.46 s	1.16 s	2.92 s	8.52 s	24.52 s

Related Work

Thank you to the reviewers for pointing out the following related work:

- The original implementation of watched lists in [MMZ$^+$01]
- Implementation and detailed analysis of linked lists in PicoSAT [Bie08]

This paper is an independent rediscovery of the same ideas.

Contrast with PicoSAT [Bie08]

Table: (Simplified) Results of the experiments ran with PicoSAT and presented in [Bie08]

Version	Solved	Unsolved	Sum Time (s)	Sum Space (MB)
Linked list	78	22	38240	5793
Array	76	24	40334	6768

Contrast with PicoSAT [Bie08]

Table: (Simplified) Results of the experiments ran with PicoSAT and presented in [Bie08]

Version	**Solved**	**Unsolved**	**Sum Time (s)**	**Sum Space (MB)**
Linked list	78	22	38240	5793
Array	76	24	40334	6768

Discussion

Why are the results so different?

Conclusion

Summary

- We reimplemented the linked list-based watch list idea on modern hardware.
- We empirically showed that the array-based watch list is faster.
- We discussed the importance of dereferencing pointers.
- We conclude that the linked list-based watch list is not a good idea.

References

Armin Biere.

Picosat essentials.

J. Satisf. Boolean Model. Comput., 4(2-4):75–97, 2008.

Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik.

Chaff: Engineering an Efficient SAT Solver.

In *DAC*, pages 530–535. ACM, 2001.