

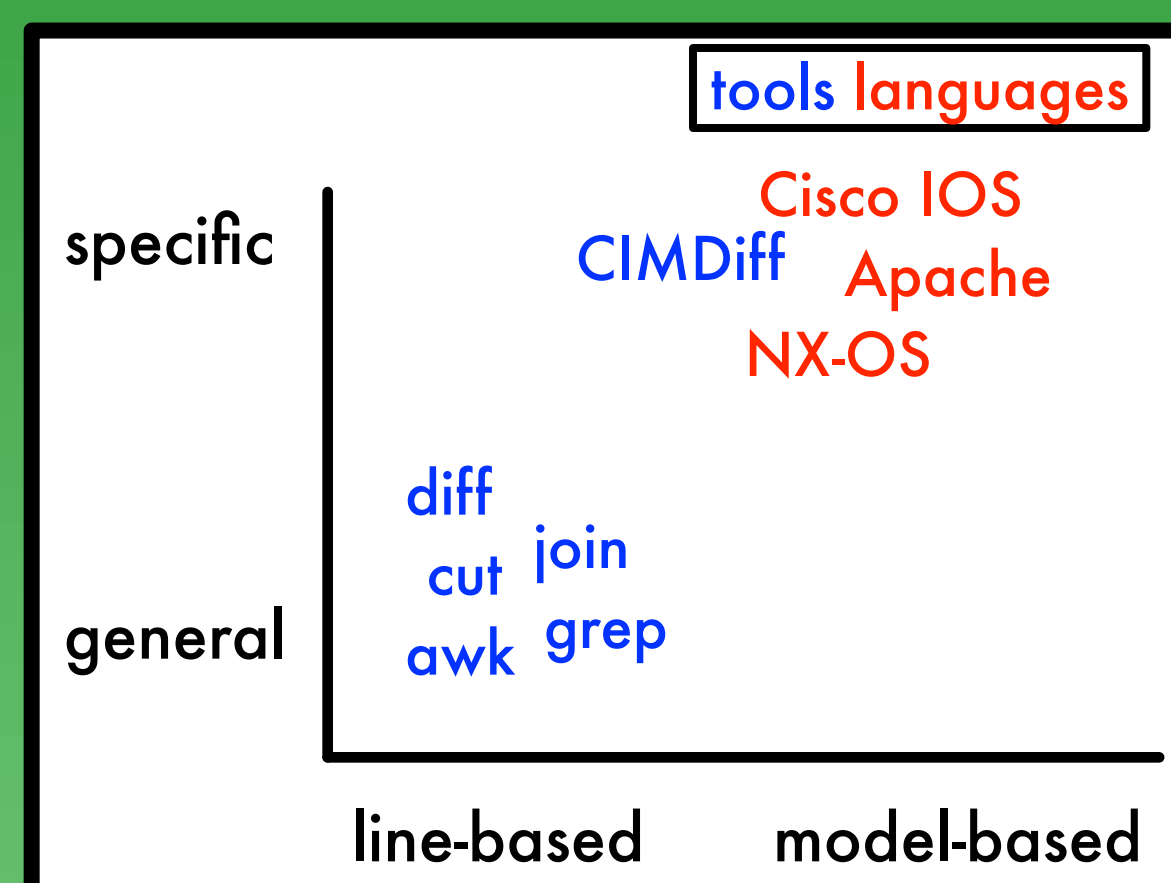
Context-Free Grep and Hierarchical Diff

Gabriel A. Weaver and Sean W. Smith

Department of Computer Science, Dartmouth College

1. Problem

Traditional UNIX commands diff and grep are geared towards old file paradigms.



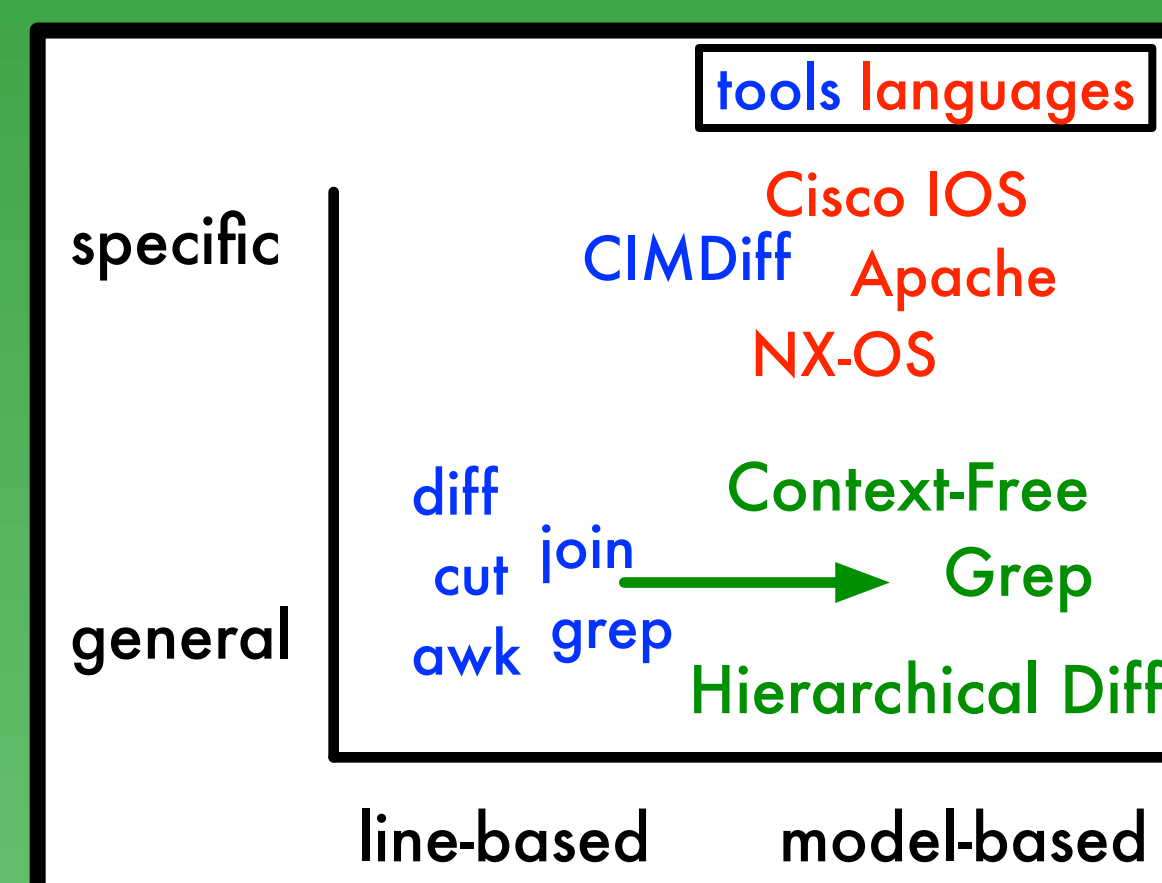
In network administration however, there is a rise in high-level languages.

2. Our Approach

We propose tools to match the rise in high-level languages.

We have gotten feedback on our tools from real-world

network administrators that such tools would be of practical use.



3. Current Status

We have designed our Context-Free Grep. Challenges include how best to output matches?

Reference	Description	wordED	treeED
SDG.1_5_1:6.1.1	In Sec 6.1.1, added more description.	12	0
AIST.1_1:1.4.3	Added Section 1.4.3	21	1
IUCC.1_5:4.6.1	Changed 4.6.1 to add logging of login, logout,...	0	0

We prototyped Hierarchical Diff and have initial results.

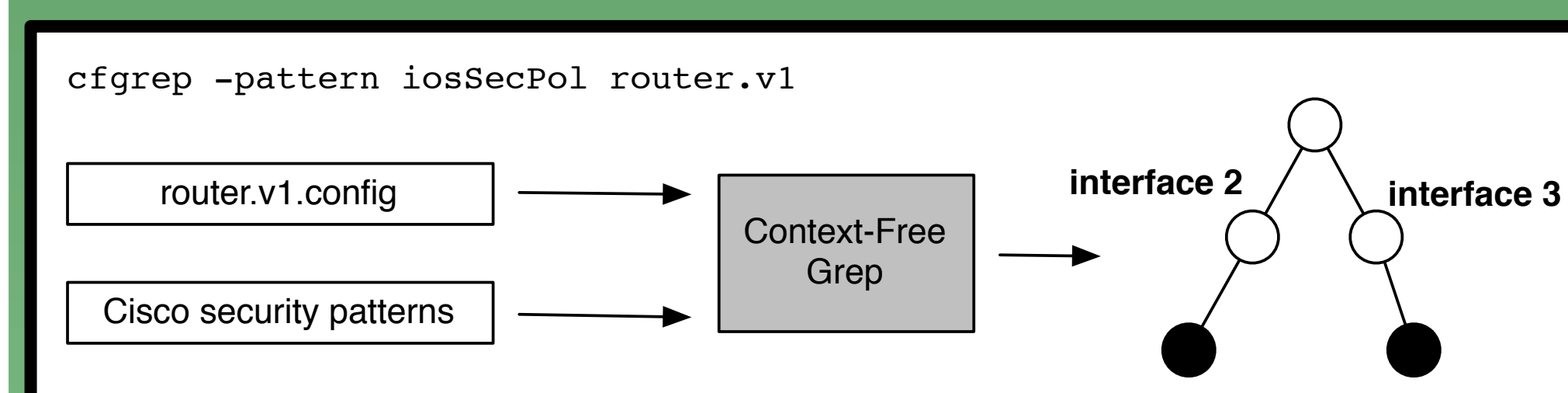
Grep

Traditional grep extracts *lines* that match one or more *regular expressions*.

But meaningful constructs can span multiple lines!

But many modern languages are block-based and more context-free than regular!

A network administrator may use our Context-Free Grep to extract security-relevant blocks from a Cisco IOS configuration file.



Here the administrator finds instances of the *service-policy* command within two of the router's interfaces.

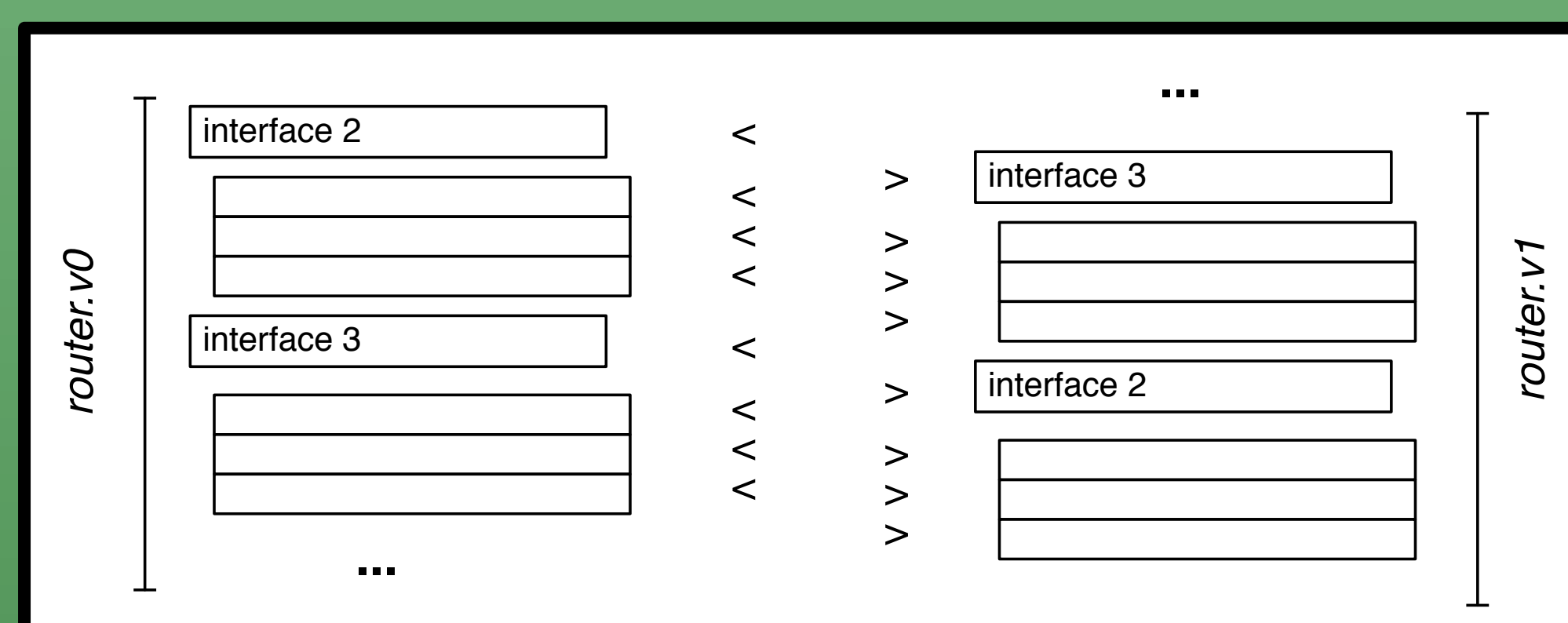
4. Impact

Our Context-Free Grep will allow network administrators to:

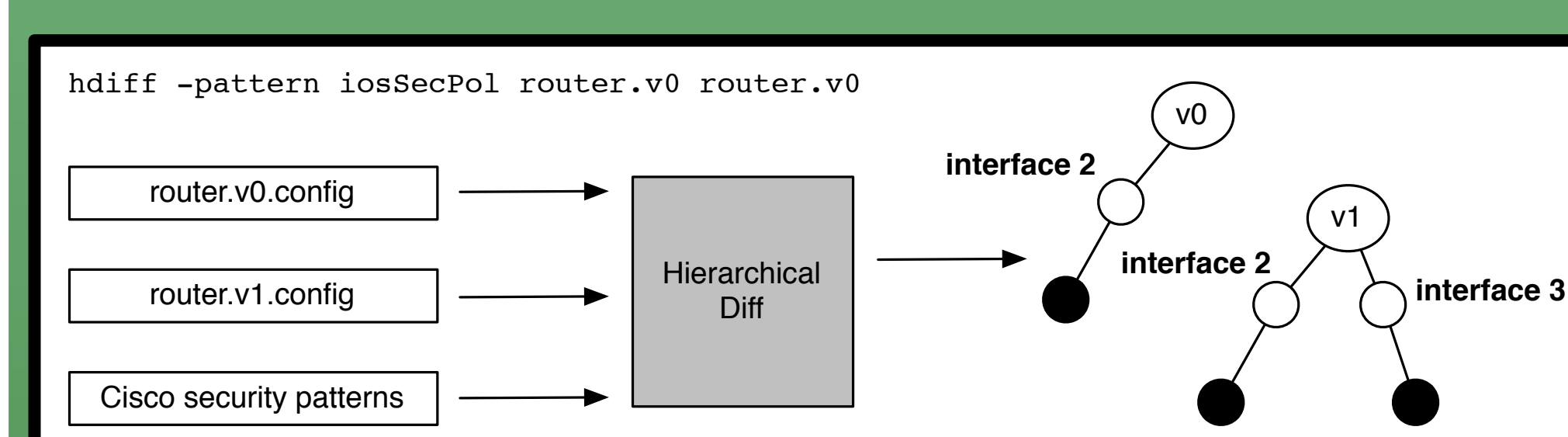
1. quickly extract *meaningful constructs* from within a configuration file.
2. generate standard parsers for meaningful, admin-defined subsets of a language.

Diff

Traditional diff compares files line-by-line. It *ignores syntactic structure* and may be too low-level.



A network administrator may use our Hierarchical Diff to track differences in the implementation of security policy over time.



Our Hierarchical Diff will allow network administrators to:

1. identify meaningful changes to configurations
 2. generate change documentation directly from block-structured, low-level configuration data.
- Some versions of tree diff are NP-hard, therefore we need to use and develop good heuristics!

5. References

- P. Bille. A survey on tree edit distance and related problems. *Theoretical Computer Science*, 2005.
- S.S. Chawathe, A Rajaraman, J. Garcia-Molina, and J. Widom. Change detection in hierarchically structured information. In *Proceedings of the ACM International Conference of Management of Data (SIGMOD)*. pages 493-504. ACM, 1996.
- G. Cobena, S. Abiteboul, and A. Marian. Detecting changes in XML documents. *International Conference on Data Engineering*. 2002.
- M. Patterson and Len Sassaman. Exploiting the forest with trees. In *BlackHat*, 2010.
- R. Routray and S. Nadgowda. CIMDIFF: Advanced difference tracking tool for CIM compliant devices. In *Proceedings of the 23rd Conference on Large Installation System Administration (LISA)*. USENIX Association, 2009.
- X. Sun, Y.W. Sung, S.D. Krothapalli, and S.G. Rao. A systematic approach for evolving VLAN designs. In *INFOCOM*, pages 1-9. IEEE Computer Society, 2010.
- Y.W. Sung, S. Rao, S. Sen, and S. Leggett. Extracting network-wide correlated changes from longitudinal configuration data. In *Proceedings of the 10th Conference on Passive and Active Measurement (PAM)*, 2009.
- A. Tsalolikhin. Configuration management summit. *USENIX login*, 35:104-105,2010.
- G. Weaver, N. Foti, S. Bratus, D. Rockmore, and S.W. Smith. Using Hierarchical Change Mining to Manage Network Security Policy Evolution. In *Proceedings of the 11th USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (HotICE)*. USENIX Association, 2011.

6. Conclusion

If successful, system and network administrators will use our tools to extract and compare information from high-level configuration languages.

This work was supported in part by the TCIPG project from DOE (under grant DE-OE0000097). Views are the authors' alone.