PROCEEDINGS OF SCIENCE

The Analysis Description Language ecosystem: Latest developments and physics applications

Sezen Sekmen,[a,*] **Gökhan Ünel,**[b] **Harrison B. Prosper,**[c] **Grigory Fedyukovich,**[c] **Daniel Riley,**[c] **Burak Şen,**[d] **Wolfgang Waltenberger,**[e] **Junghyun Lee,**[a] **Aytül Adıgüzel,**[f] **Erkcan Özcan,**[g] **Ahmetcan Sansar,**[f] **Demircan Demirbağ,**[g] **Kağan Şahan**[f] **and Feyza Başpehlivan**[h]

[a] *Kyungpook National University, Department of Physics, Daegu, Republic of Korea*
[b] *University of California, Department of Physics and Astronomy, Irvine, CA, USA*
[c] *Florida State University, Department of Physics, Tallahassee, FL, USA*
[d] *Middle East Technical University, Department of Physics, Ankara, Turkey*
[e] *University of Vienna, Faculty of Physics, Wien, Austria*
[f] *Istanbul University, Department of Physics, Istanbul, Turkey*
[g] *Boğaziçi University, Department of Physics, Istanbul, Turkey*
[h] *TOBB University of Economics and Technology, Ankara, Turkey*

E-mail: ssekmen@cern.ch

We present latest developments in Analysis Description Language (ADL), a declarative domain-specific language describing the physics algorithm of a HEP data analysis decoupled from software frameworks. Analyses written in ADL can be integrated into any framework for various tasks. ADL is a multipurpose construct with uses ranging from analysis design to preservation, reinterpretation, queries, visualisation, combination, etc. The most advanced infrastructure to execute ADL on events is the CutLang runtime interpreter. Recent technical developments include an automated interface with different data types, generation of the abstract syntax tree, a visualization tool that that auto-converts analysis flows to graphs, incorporation of trained machine learning models and a Jupyter-based plotting tool. We also report physics implications including a large scale LHC analysis implementation and validation effort for beyond the standard model reinterpretation purposes and studies with ATLAS and CMS open data.

42nd International Conference on High Energy Physics (ICHEP2024)
18-24 July 2024
Prague, Czech Republic

*Speaker

PoS(ICHEP2024)1056

The LHC experiments have performed hundreds of physics analyses for standard model measurements and new physics searches, featuring a large variety of physics algorithms, which comprise of definitions of objects, event variables and event selections. Capturing and preserving the physics algorithms in an accessible manner, preferably independent of the software frameworks that execute the analyses, provide extensive benefits and physics opportunities beyond the lifetimes of the analyses. Analysis Description Language, or ADL, was invented to achieve this purpose.

ADL is a declarative, domain specific language (DSL) that describes the physics algorithm of a high-energy physics (HEP) analysis in a standard and unambiguous way [1]. It is an external DSL with a custom syntax for physics analysis concepts. It is a declarative language, which directly encodes what tasks to perform, but not how to explicitly perform them. It has a clear and self-describing syntax that is easy to read and communicate. ADL is inclusive, being designed for use by a diverse range of communities, including experimentalists, phenomenologists, students, and interested public.

ADL consists of a plain text ADL file describing the physics logic using an easy-to-read DSL with clear syntax rules. The file consists of blocks separating object, variable and event selection definitions. The blocks have a keyword-instruction structure. The syntax includes mathematical and logical operations, comparison and optimization operators, reducers, 4-vector algebra and HEP-specific functions such as $d\phi$ and dR. The ADL file is accompanied by a library of self-contained functions encapsulating variables that are non-trivial to express with the ADL syntax. These could be variables such as aplanarity, the stransverse mass M_{T2}, or machine learning model discriminants.

ADL is framework-independent, by design, decoupling physics information from the software framework the analysis is executed in. This feature leads to a wide-array of possibilities for multi-purpose use, as ADL can be translated, or integrated into any general purpose language or framework for various tasks. It also enables portability and preservation of the physics content.

Currently, the most advanced framework that can parse and execute ADL is the CutLang runtime interpreter [2, 3]. CutLang is a C++ runtime interpreter for ADL, based on the ROOT framework. It performs formal grammar parsing via the state-of-the-art tools Lex & Yacc. CutLang hosts by default many external functions relevant for HEP analyses, and also has an interface to machine learning models via ONNX. It automatically reads ROOT TTree-like formats, such as the Delphes output, CMS NanoAOD, or various LHC open data formats. CutLang runs in Linux and macOS, and is available in Docker and Conda. A Jupyter kernel exists and is accessible via binder or Conda. CutLang outputs cutflows, histograms, events, and the analysis description itself in order to trace provenance.

Recent technical advancements have significantly modernized the ADL and CutLang infrastructures. The ADL grammar implementation in CutLang, which previously required manual encoding of all input data attributes, has been updated to decouple the core ADL syntax from input data types and external functions. This update enables a semi-automated interface with physics data input types. Additionally, the automatic generation of an abstract syntax tree (AST) is now possible, serving as a foundation for various future applications. One immediate use case for the AST is the newly developed `adl2flowchart` tool, which visualizes ADL analyses as flowcharts, displaying the relationships between objects and regions as a connected graph [4].

ADL not only facilitates the design and execution of individual analyses but also excels in navigating and exploring multi-analysis landscapes due to its standardized structure. Many ATLAS

and CMS analyses are already implemented in ADL, and this growing database can serve as a foundation for various applications. These include analysis preservation, improved communication through ADL files or flowchart visualizations, and educating the next generation of students on analysis concepts. The database is also directly applicable for reinterpretation studies by the phenomenology community. For example, the right panel of Figure 1 demonstrates the reproduction of a CMS supersymmetry search in the photon + jets + p_T^{miss} final state using ADL/CutLang, where limits derived from privately generated signal samples align with those published by CMS. The left panel shows the reproduction of a search region from a CMS Run 1 vector-like quark search with boosted W and Higgs bosons, using ADL/CutLang on CMS Open Data. Additionally, ongoing work aims to leverage static analysis techniques to automate queries across multiple analyses, identify overlaps and disjoint analyses, and assist in performing combination studies.

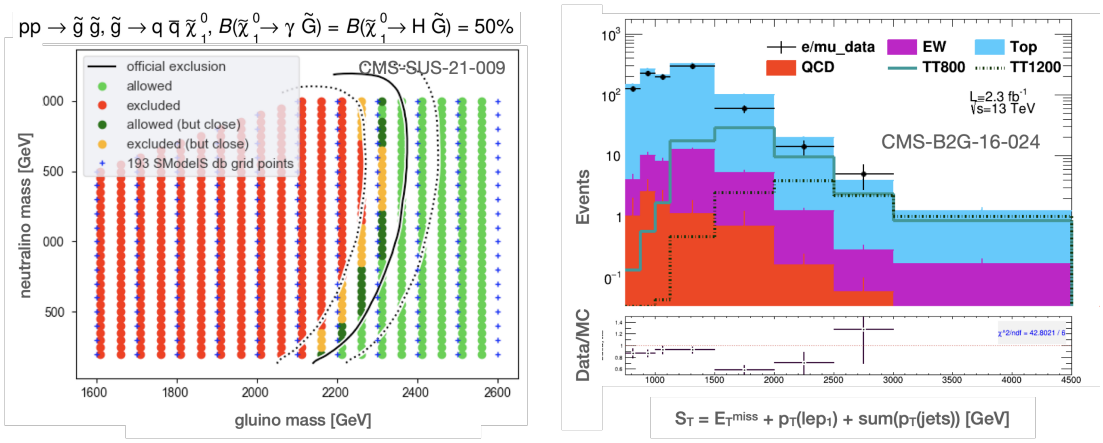

Figure 1: Right: Reproduction of a CMS supersymmetry search in the photon + jets + p_T^{miss} final using ADL/CutLang. Left: CMS Run 1 vector-like quark search with boosted W and Higgs bosons with ADL/CutLang on CMS Open Data.

Acknowledgements: SS is supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under contracts NRF-2021R1I1A3048138, NRF-2018R1A6A1A06024970, and NRF-2008-00460.

References

[1] H. B. Prosper, S. Sekmen and G. Unel, "Analysis Description Language: A DSL for HEP Analysis," [arXiv:2203.09886 [hep-ph]].

[2] S. Sekmen and G. Ünel, "CutLang: A Particle Physics Analysis Description Language and Runtime Interpreter," Comput. Phys. Commun. **233** (2018), 215-236 doi:10.1016/j.cpc.2018.06.023 [arXiv:1801.05727 [hep-ph]].

[3] G. Unel, S. Sekmen, A. M. Toon, B. Gokturk, B. Orgen, A. Paul, N. Ravel and J. Setpal, "CutLang v2: Advances in a Runtime-Interpreted Analysis Description Language for HEP Data," Front. Big Data **4** (2021), 659986 doi:10.3389/fdata.2021.659986 [arXiv:2101.09031].

[4] D. Riley, "adl2flowchart tool", https://github.com/danielmriley/adl2flowchart/