



# ADIC 2.0

## Status and Plans

---

Boyana Norris  
Beata Winnicka  
Argonne National Laboratory

April 15, 2005  
Nice



# Outline



- Quick ADIC 1.1 summary
- What's new ADIC 2.0?
- ADIC 2.0 architecture and implementation.
  - Current status
  - Future work
  - Example
- Further info



# ADIC 1.1 Summary



- Source transformation of ANSI C code
- Currently distributed, handles most of ANSI C, a little C++
- Based on old (abandoned) research C/C++ parser (Sage++)
- Available differentiation modules:
  - Jacobian (statement-level reverse/forward mode)
  - Hessian



# What is new in ADIC 2.0?



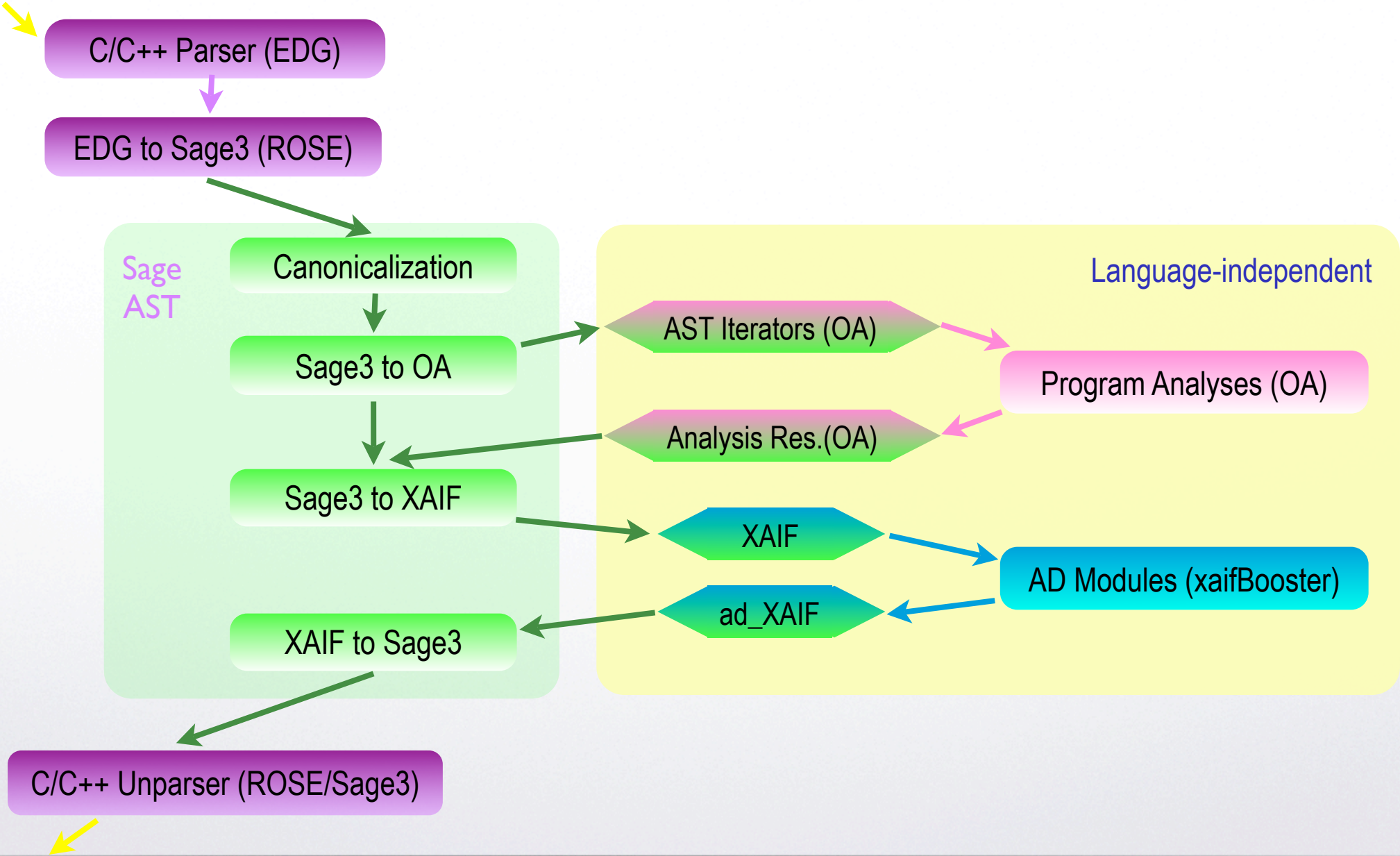
(Everything)

- **Underlying software**
  - New C/C++ parser and unparser: EDG-based
  - New AST toolkit: Sage3/ROSE
  - New differentiation modules
- **Internals**
  - XAIF representation
  - Integration with OpenAnalysis

# ADIC 2.0 Architecture



C/C++



C/C++

# ADIC 2.0 Architecture



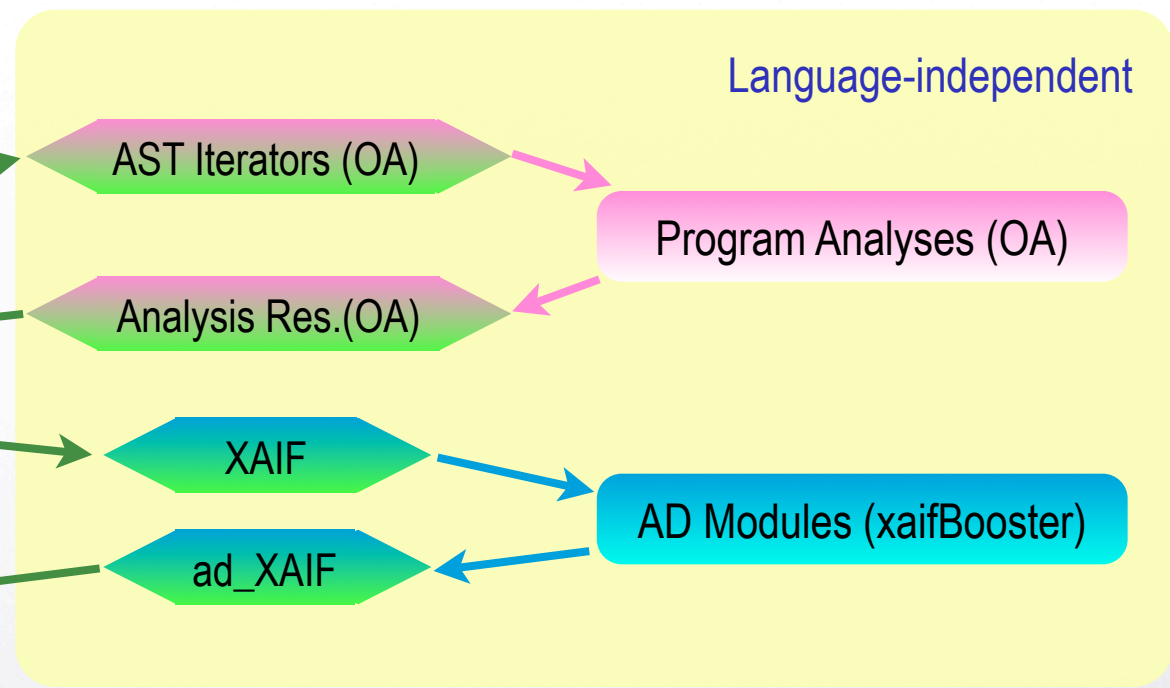
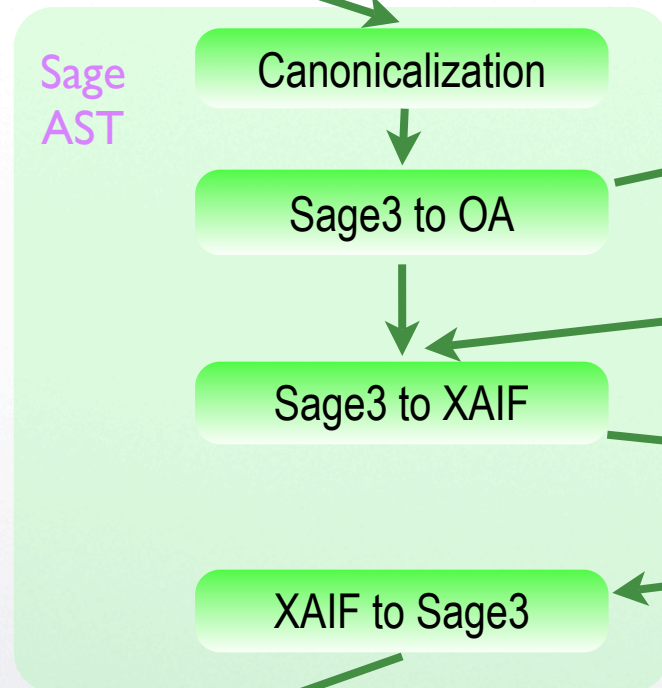
C/C++



C/C++ Parser (EDG)



EDG to Sage3 (ROSE)



C/C++ Unparser (ROSE/Sage3)



C/C++



- Current status:
  - C/C++ parser: EDG 3.3
  - Avoid lowering, e.g.,
    - replacing “sizeof x” expressions with constants
    - constant propagation
  - Preserve comments, preprocessor directives
- Future work
  - C++ template information preservation



- **Current status:**
  - expressions with side effects are hoisted, including noninlinable intrinsics
  - functions are converted to procedures
- **Future work:**
  - add customization hooks (user- or AD module-specified)
  - C++ mode of canonicalization, e.g., only convert functions to procedures when necessary, user polymorphism, etc.





- **Current status:**

- ADIC has implementations of the interfaces required by the following analyses:

- Call graph

- Control flow graph

- Alias

- ADIC incorporates the following analyses results into XAIF:

- Call graph

- Control flow

- **Future work:**

- Generate XAIF with results from alias analysis, duud chains, activity, etc (as OpenAnalysis evolves)



- XAIF 1.0: <http://www.mcs.anl.gov/xaif>
- Language-independent program representation (XML), nested graphs:
  - Call graph (including symbol tables)
  - Control flow graph
  - Expression DAG
- Current status:
  - ADIC produces XAIF for forward mode AD modules
  - Common inlinable intrinsics are also described in XAIF
- Future (currently ongoing) work:
  - Provide support for reverse mode differentiation (more canonicalization, reverse mode templates, checkpointing)
  - Add XAIF descriptions of all C/C++ intrinsics (C99 longer term)



- Language-specific intrinsic defined in XAIF
  - inlinable, e.g, +, -, \*, /, sin, cos, definition of many included in ADIC
  - noninlinable, e.g., functions with side effects
  - users can supply XAIF definitions of both inlinable and non-inlinable intrinsic
- Future work:
  - include noninlinable versions of the language intrinsic that provide exception handling at points of nondifferentiability



- Current status:
- Preserve (as much as possible)
  - preprocessor directives
  - comments
  - formatting
- Future work:
  - formatting enhancements
  - add option to include comments with original code



# Example



```
#include <math.h>

double func(double x) {
    if ( x > 0 ) {
        return sin(x * x);
    }
    return 0.0;
}
```



# Distribution



- Currently planned: binary on Linux, Sun, and MacOS
- Working on license that would allow source distribution (for ADIC 2.0 itself)
- Source distribution requirements (depth 1):
  - Autotools (autoconf, automake), GNU make
  - EDG 3.3
  - ROSE
  - OpenAnalysis
  - At least one XAIF-based differentiation module
  - Apache Xerces-C
- Porting plans: ADIC 2.0 itself is trivial to port, the libraries it depends on are not



- ADIC Web server provides simple access for testing smaller codes (ADIC 1.1 and 2.0):
  - <http://www.mcs.anl.gov/adicserver>
- ADIC 2.0 will be released in the next few weeks:
  - <http://www.mcs.anl.gov/adic>
- OpenAD project page contains references to almost everything else:
  - <http://www.mcs.anl.gov/~utke/OpenAD>
- Acknowledgments: ADIC 2.0 development is funded through DOE and NSF.