The reverse engineering field of disassemblers is, by its nature, an ever changing and evolving one. There is an ever-present need for state of the art programs that can guarantee efficient translation of machine code into assembly.

Having access to a large variety of processors and file formats is essential. At the moment, IDA Pro offers the most vast range, including support for certain exotic architectures like RISC-V and SPARC. Taking that into consideration, Ghidra comes close behind with support for most file formats, including modern ones, and for most processors including their variants, while allowing for the addition of new architectures or extending pre existing ones.  
Processors such as:

* Z80
* X86 16/32/64
* 6502
* ARM/AARCH64
* 8051
* PowerPC 32/64/VLE
* MSP430
* MIPS 16/32/64/micro
* AVR8
* 68xxx
* Java / DEX bytecode
* AVR32
* PA-RISC
* PIC 12/16/17/18/24
* Sparc 32/64
* CR16C

In terms of decompilers, IDA Pro’s "Hex-Rays" is best thanks to its accuracy being comparable to that of a human. The technological advancements implemented through IDA 9.0 also introduced new decompilers for 32- and 64-bit RISC-V code, designed to modernise processes and increase capacities. Ghidra’s decompiler is powerful, but not as refined as Hex-Rays.  
Furthermore, IDA Pro provides superior debugging and emulation for running and tracing target systems, particularly for Android and Windows.  
Currently IDA Pro delivers greater support for patching binaries, interactively changing characteristics of displayed data, renaming variables, functions, structures or changing data presentation and creating custom types.

Regarding graph generation, Ghidra offers a more informative, highly customizable and detailed visualisation of code, even marking “if … else” statements on the graph and making it more organised.  
For working teams, facilitating collaborative reverse engineering is important. Ghidra’s design supports working on the same project, while IDA Pro requires certain plugins and the database files are not designed to be shared.

The support for cross-platform, while versatile, is implemented by few tools, like Ghidra, Rafare2 and JaDx, so that they work on macOS, Windows and Linux.

Concerning binaries, Ghidra is faster and loads multiple binaries into projects simultaneously, resulting in an easier approach to following code between applications and libraries, by using multi-threading and multi-core properties of the CPU. For bigger files, 150 MB+, IDA Pro has a better optimization of the memory and CPU usage, it can lessen the processing time, being altogether faster than the other.

The most significant cutting-age technology implemented recently is helping with pattern recognition accuracy and processing very large files due to integration of AI and machine learning in disassemblers. Models are being trained to increase code recognition patterns and data structures.  
A disassembler based on deep learning, DeepDi, proved to be simultaneously fast, highly accurate and efficient against obfuscated binaries, managing to discern valid instructions from the raw binary. For capturing and propagating instruction relations, it uses a neural network graphing model. It has a very low false positive rate and false negative rate when identifying instructions, 0.02%/00.2%, and compared to IDA Pro, the CPU version is twice as fast and the GPU version is 350 times faster. Taking in consideration code section size-disassembly time correlations, the CPU version has a throughput of 146 KB/s and the GPU one is 170 times faster at around 24.5 MB/s.

### **Conclusion**

At the moment, state-of-the-art disassemblers are extraordinary, powerful tools used in domains such as reverse engineering, software development and debugging, but still suffer from challenges like the ambiguity of instructions, the absence of symbols and debugging information and mostly from the presence of obfuscation techniques, but these difficulties will begin to become a problem of the past with the evolution of machine learning.