Implementation trade-offs

Approach	Portability	Execution speed	Compilation speed	Foreign language interface	Dynamicity (assert/retract)	
Interpreter in Prolog	yes	slow	fast	poor	yes	(1)
Interpreter in C or C++	yes	slow	fast	poor	yes	
Emulator in C or C++	yes	average	fast	poor	yes	
Generate Prolog	yes	average	depends *	depends *	yes	(2)
Generate C or C++	yes	fast †	slow	best	no	(3)
Assembly	no	fast	fast	average	no	
Machine code (binary)	no	fast	fast §	average	yes	

* Depends on the underlying Prolog system

† Using RISC-macro technique (see next slide)

§ Compilation becomes more complex with modern (superscalar) RISC

- Three good trade-offs:
  - (1) Smallest development time to get a system that works
  - (2) Smallest development time to get a system that is fast enough to be useful
  - (3) Best speed while maintaining excellent portability and interoperability