

INTRODUCTION

- **Janus** is an imperative reversible programming language, meaning that every computation and function can be reversed.
- **Hanus** is an extended implementation of Janus that can be compiled straight to Haskell. Because of this, Hanus contains many awesome Haskell features that are unthinkable in regular Janus!

REVERSE YOUR PROGRAM

data $_ \longrightarrow _ : Configuration\ ads\ cs\ ds \rightarrow Configuration\ ads\ cs\ ds \rightarrow Set$ **where**

D-AuthJoin : $\langle A, v \rangle^d \mid \langle A, v' \rangle^d \mid \Gamma \longrightarrow \langle A, v \rangle^d \mid \langle A, v' \rangle^d \mid A\ [0 \leftrightarrow 1] \mid \Gamma^{pppp} \longrightarrow \langle A, v + v' \rangle^d \mid \Gamma$

C-Advertise : $Any\ (_ \in Hon)\ (participants\ (G\ ad)) \rightarrow (\Gamma \longrightarrow ad \mid \Gamma)$

C-AuthCommit : $(secrets\ A\ (G\ ad) \equiv a_0 \dots a_n) \times (A \in Hon \rightarrow All\ (_ \neq nothing)\ a_i) \rightarrow 'ad \mid \Gamma \longrightarrow 'ad \mid \Gamma \mid \dots \langle A : a_i \# N_i \rangle \dots \mid A\ [\# \triangleright ad]$

SEMANTIC CHECKING (JANUS SIDE)

SEMANTIC CHECKING (HASKELL SIDE)

HASKELL POWER

