Generating Constraints (1/2)

intconst:	[[*intconst*]] = `int`
E_1 *op* E_2:	[[E_1]] = [[E_2]] = [[E_1 *op* E_2]] = `int`
E_1 `==` E_2:	[[E_1]] = [[E_2]] ∧ [[E_1`==`E_2]] = `int`
`input`:	[[`input`]] = `int`
id `=` *E*:	[[*id*]] = [[*E*]]
`output` *E*:	[[*E*]] = `int`
`if (`*E*`) {`*S*`}`:	[[*E*]] = `int`
`if (`*E*`) {`S_1`} else {`S_2`}`:	[[*E*]] = `int`
`while (`*E*`) {`*S*`}`:	[[*E*]] = `int`

Generating Constraints (2/2)

id`(`id_1`, ..., `id_n`) { ... return` *E*`; }`:	[[*id*]] = `(`[[id_1]]`, ..., `[[id_n]]`) ->` [[*E*]]
`(`*E*`) (`E_1`, ..., `E_n`)`:	[[*E*]] = ([[E_1]], ..., [[E_n]]) `->` [[`(`*E*`) (`E_1`, ..., `E_n`)`]]
`&`*id*:	[[`&`*id*]] = `&`[[*id*]]
`malloc`:	[[`malloc`]] = `&`α
`null`:	[[`null`]] = `&`α
`*`*E*:	[[*E*]] = `&`[[`*`*E*]]
`*`*id* `=` *E*:	[[*id*]] = `&`[[*E*]]