

**Parameterized** 

specification:

 $\forall i: \ G(r_i \rightarrow Fg_i)$ 

Synthesizers automatically construct an implementation from a specification. In *parameterized* synthesis [JB12], we search for a process such that a system made of *any* number of these processes satisfies a specification.



HOW?

Two ideas:

1) *cutoffs* [EN95]: if a small system satisfies the spec

then a larger system will satisfy the spec too

2) SMT-based bounded synthesis [FS07]:

bound a process size, synthesize,

if not found – increase the bound.

Parameterized specifications arise naturally in synthesis. Engineers need more than 7-clients arbiter. But monolithic synthesis tools do not scale. Parameterized synthesis allows 'once and for all' synthesis.

WHY?

(FSM)

Systems (async tok rings)



U RSE

**Synthesizer** 

(PARTY)

PARTY is a *semi-decision* tool. And it always finds a model (if exist) of a *minimal* size.



Besides a parameterized synthesis PARTY also supports a monolithic bounded synthesis. Available at <u>https://github.com/5nizza/Party</u>.

[KJB13a] Towards Efficient Parameterized Synthesis, A. Khalimov, S. Jacobs, R.Bloem, VMCAI13 [KJB13b] PARTY: Parameterized Synthesis of Token Rings, A. Khalimov, S. Jacobs, R.Bloem, CAV13 [JB12] Parameterized Synthesis, S. Jacobs, R. Bloem, TACAS12 [FS07] Bounded Synthesis, S. Schewe, B. Finkbeiner, ATVA07

[EN95] Reasoning about Rings, A. Emerson, K. Namjoshi, POPL95