

# The 7th Reactive Synthesis Competition: SYNTCOMP'20

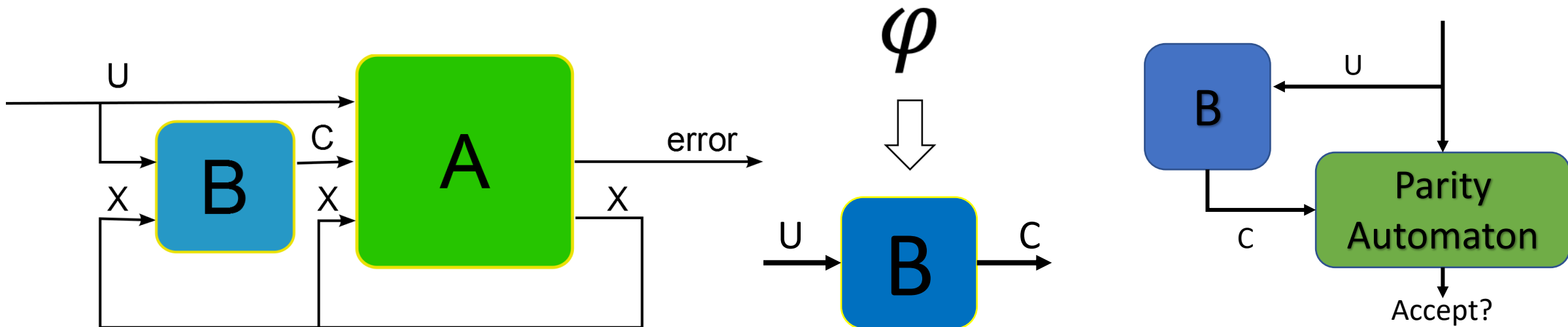
Swen Jacobs, CISPÀ

Guillermo A. Perez, University of Antwerp

Advisory committee: Roderick Bloem, Armin Biere, You?

# What is Reactive Synthesis?

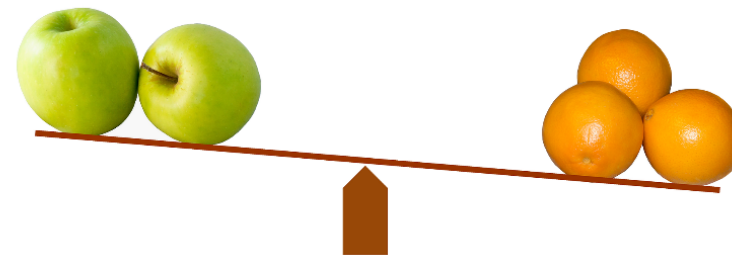
- Synthesize a **circuit B** for a given **circuit A** such that **error** is not raised for any sequence of Us
- Synthesize a **circuit B** that satisfies an **LTl formula  $\varphi$**  on its inputs and outputs
- Synthesize a **circuit B** for a given **automaton A** such that it accepts for any sequence of Us



# Why do we need SYNTCOMP?

Make it easier to compare synthesis tools

- Establish a benchmark format
- Collect a benchmark library
- Fair and comprehensive evaluation



Guide development of synthesis tools

- Encourage implementation of mature push-button tools
- Improve state of the art through challenging benchmarks

# Historical milestones and rules

- 2014 – First SYNTCOMP @ Vienna Summer of Logic
- 2016 – Third SYNTCOMP: LTL/TLSF tracks added
- 2019 – Sixth SYNTCOMP: migration to StarExec

This year we added parity-automata tracks!

**Input:** specification in AIGER, TLSF, or HOA

**Output:** Y/N answer or implementation in AIGER

**Ranking:** based on quantity and quality (size) of solutions

# SYNTCOMP 2020

New parity-automata tracks, Participants, Rankings

# Extended Hanoi Omega-Format

- The HOA format supports many types of infinite-word automata
- Extension allows to label atomic propositions as (un)controllable
- **Restriction for synthesis:** deterministic parity automata

## Tooling:

- `hoa2pg`: translator to PGSolver format
- `hoa2aig`: constructs a model-checking-friendly AIGER monitor
- `tlsf2pg`: translates LTL to a parity game via Spot

# Participants

## **Safety – mostly inactive**

- Simple BDD Solver (Walker, Ryzhyk)
- AbsSynthe (Basset, Brenguier, Perez, Raskin, Sankur)
- Demiurge (Konighofer, Seidl)

## **LTL:**

- Itlsynt (Duretz-Lutz, Colange, Michaud, Schlehuber-Caissier)
- Strix (Meyer, Sickert, Luttenberger)

## **Parity automata:**

- Strix (Meyer, Sickert, Luttenberger)
- Knor (van Dijk)

# Results: Realizability

## Safety (1 CPU hr)

1. simpleBDDSolver (sc2): 186
2. simpleBDDSolver (sc1): 177
3. AbsSynthe (sc3): 152

## LTL (1 CPU hr)

1. Strix (pq): 424
2. Strix (bfs): 422
3. Itlsynt (lar): 398

## Parity automata (all solved)

1. Strix (seq): 6.84s
  2. Strix (par): 8.12s
  3. Knor (npp): 12.71
- (hors concours)
- Knor-BDD (npp): 1.57s
  - Knor-BDD (fpj): 1.59s
  - Knor-BDD (fpi): 1.66s



# Results: Synthesis quality ranking

**LTL – size of circuit is compared to reference solution**

1. Strix conf1 (bfs): 599.56
2. Strix conf2 (pq): 594.56
3. Itlsynt (larabc): 395.15
4. Itlsynt (dsabc): 360.01
5. Itlsynt (sdabc): 350.16

# Summary and outlook

## Parity automata

- More benchmarks needed
  - Translate PGSolver benchmarks?
- Only Strix supports synthesis now
- Fixed HOA-to-PGSolver-trans sub-track?
- Minimal on-the-fly automaton usage
  - New quality score based on how much of the automaton is "queried"?
- Model checking parity automata directly
- Succinct format of parity games? Via AIGER?

## Safety

- Revive with a focus on safety-shield synthesis
  - Translate safety-shield benchmarks?

## Parallel?

SYNTCOMP'20 Winners

| Track                   | Tool            |
|-------------------------|-----------------|
| Safety (real)           | simpleBDDSolver |
| LTL (real)              | Strix           |
| LTL (synthesis-quality) | Strix           |
| Parity game (real)      | Strix           |