Synthesizing and Verifying Orchestration Algorithms

Simon Thrane Hansen
Ph.D. Student AU

June 17, 2021

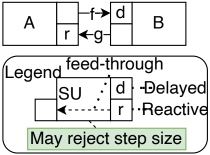
Overview

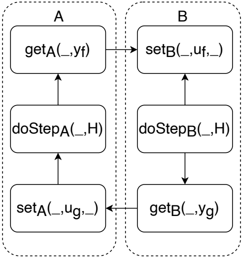
1. Recap Simple Scenarios

2. Complex Scenarios

Simulation of simple scenario

Last time we covered simple scenarios by describing the rules of the actions that formed the step operation graph - that for simple scenarios only contain trivial SCC.





Algorithm 1 Step function of scenario

```
$s_A^{(s+H)} \leftarrow \texttt{doStep}_A(s_A^{(s)}, H)$
$s_B^{(s+H)} \leftarrow \texttt{doStep}_B(s_B^{(s)}, H)$
$F_v \leftarrow \texttt{get}_B(s_B^{(s+H)}, y_F)$
$s_A^{(s+H)} \leftarrow \texttt{set}_A(s_a^{(s+H)}, u_F, F_v)$
$x_v \leftarrow \texttt{get}_A(s_A^{(s+H)}, y_x)$
$s_B^{(s)} \leftarrow \texttt{set}_B(s_B^{(s+H)}, u_x, x_v)$
```

Introduction to complex scenarios

Algebraic Loops

Happens if: The value on a ports depends on itself.
Example of the two kinds of Algebraic Loops:

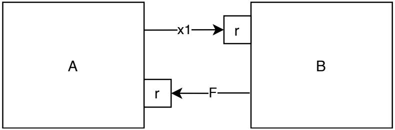


Figure: Reactivity Loop

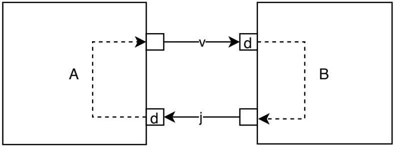


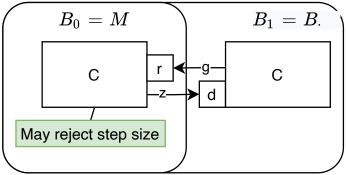
Figure: Feed-through Loop

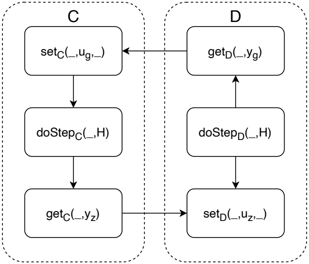
How do we simulate this: Fixed-point iteration - The simulation iteratively searches for a fixed-point on all the output ports between two successive iterations.
Problem: Introduces complex SCCs in the step operation graph - no topological order can be found.

Step negotiation

Happens if: A scenario contains SUs that may reject to take a step of arbitrary size.

Example:





How do we simulate this: Step finding procedure - The simulation iteratively searches for a step all SUs are able to perform.

Problem with the traditional method: Fails to identify these scenarios and can therefore not treat them.

Simulation of complex scenario

- Should be simulated using a correct configuration (Step size + Fixed-points).
- An incorrect configuration breaks the preconditions.
- The simulation strategy is to use an iterative search for a correct (stable) configuration.
- The simulation is restarted after unsuccessful tries (using restore and save).

Algorithm 2 Simulation of complex scenarios.

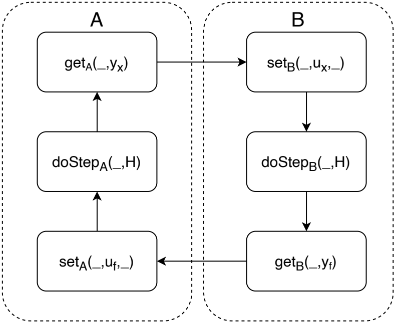
```
Save SUs
while !correctConfiguration do
    Iterative search
    if !correctConfiguration then
        Restore SUs
    end if
end while
```

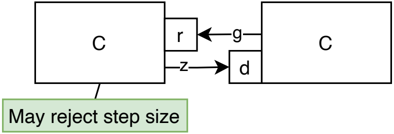
Challenges in Synthesizing Algorithms for Complex Scenarios

1. Detection of complex scenarios
2. Extracting the Algorithm
    - The graph contains cycles

- Complex Scenarios - having SCC in the graph.
    - Solving Algebraic Loops (cycles in the graph)
    - Performing Step Negotiation to ensure that SUs move in lockstep.

These are solved using specialize heuristics to break the SCCs.





May reject step size

Synthesizing Complex Scenarios - Algebraic Loop

The SCC is broken using reduction scheme:

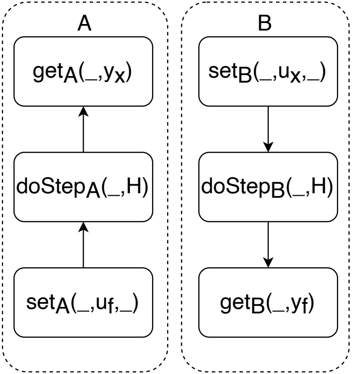


Figure: Maximal reduction.

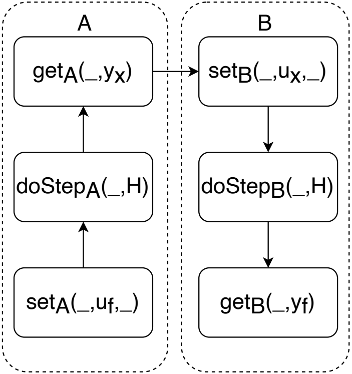


Figure: Minimal reduction.

Synthesizing Complex Scenarios - Step negotiation

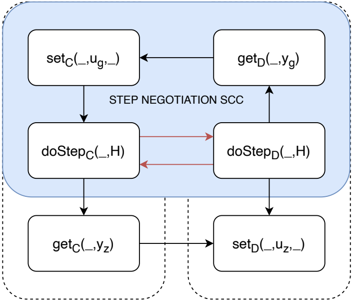


Figure: Step graph - the blue part shows the iterative search.

Algorithm 3 Step negotiation.

```
while !$Step\_found$ do
    $(s_D^{(s+h_D)}, h_D) \leftarrow \texttt{doStep}_D(s_D^{(s)}, h)$
    $g_v \leftarrow \texttt{get}_D(s_D^{(s+h_D)}, y_g)$
    $s_C^{(s)} \leftarrow \texttt{set}_C(s_C^{(s)}, u_G, G_v)$
    $(s_C^{(s+h_C)}, h_C) \leftarrow \texttt{doStep}_C(s_C^{(s)}, h_D)$
    $h \leftarrow min(h_C, h_D)$
    $Step\_found \leftarrow h == h_C \wedge h == h_D$
    if !$Step\_found$ then
        Restore SUs
    end if
end while
```