Synthesis of arbitrary audio signals via random DSP trees

Edward Ly (d8222106)

Winter Camp - 14 March 2022

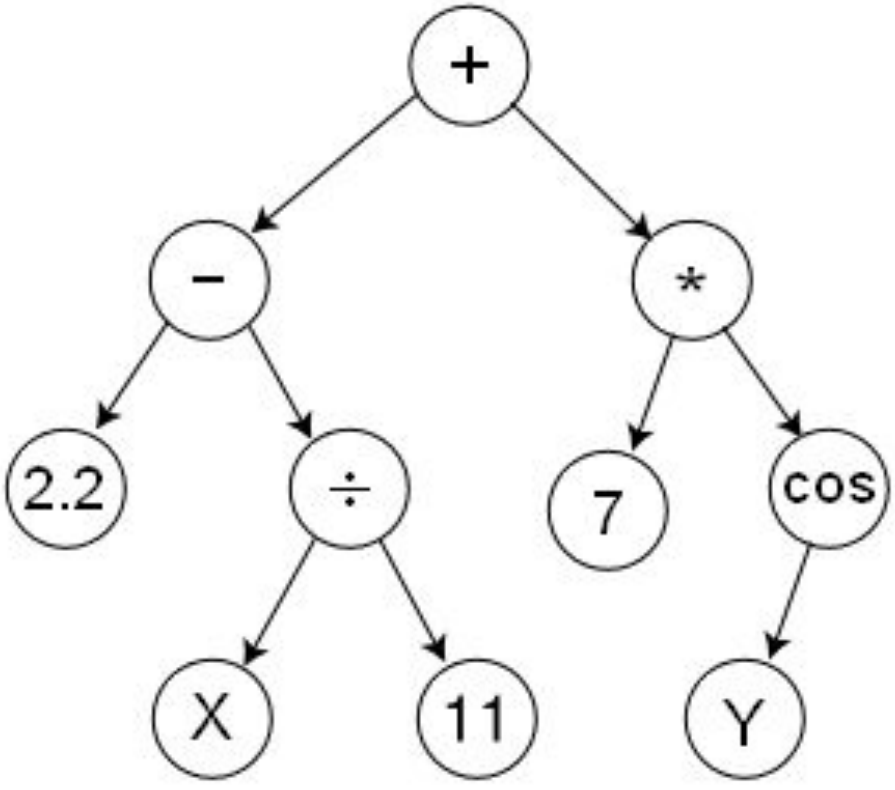
Outline

- Background
  - DSP Trees
  - FAUST Language
- Methods
- Demo
- Future Work

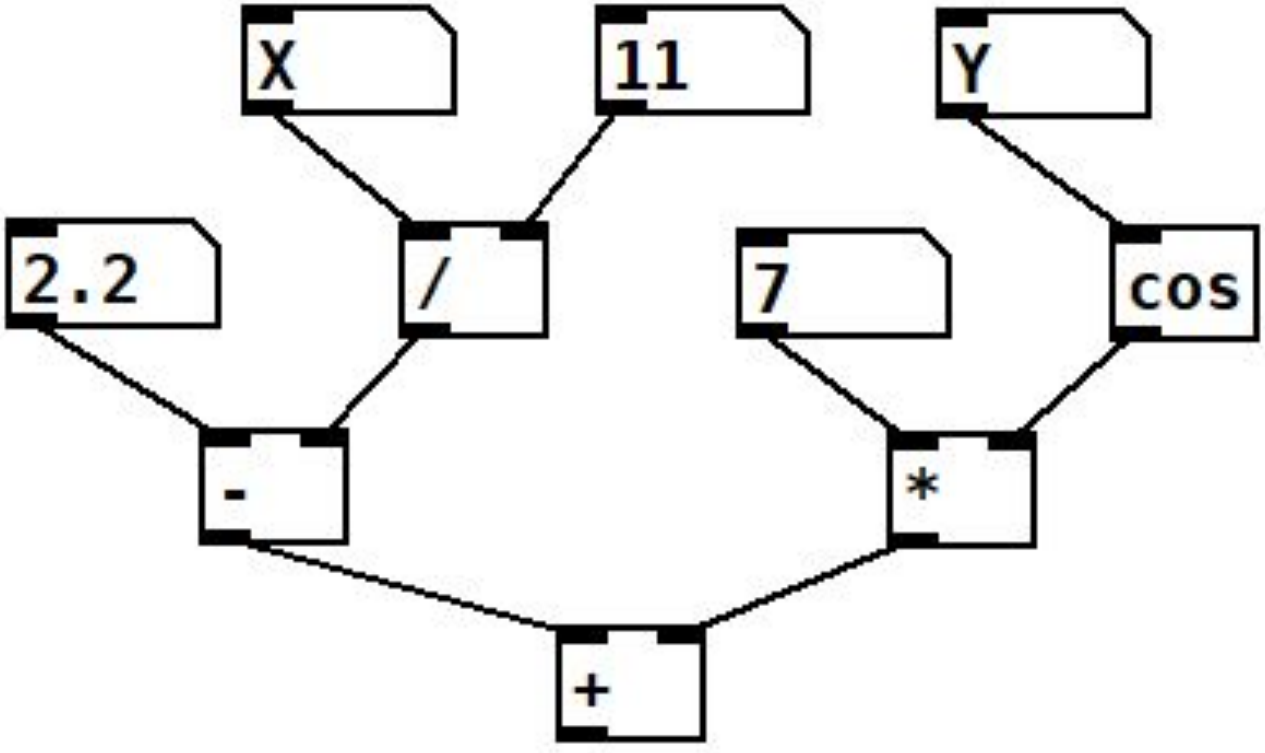
What is a computer program?

Functional Programming

- A paradigm where programs are constructed by **composition of functions**
- LISP is the earliest, most significant language
- Many LISP dialects:
  - Common Lisp, Scheme, Racket, …
- c.f. **imperative programming**
  - Popular languages: C++, Rust, Java, …
  - Programming via statements and state changes

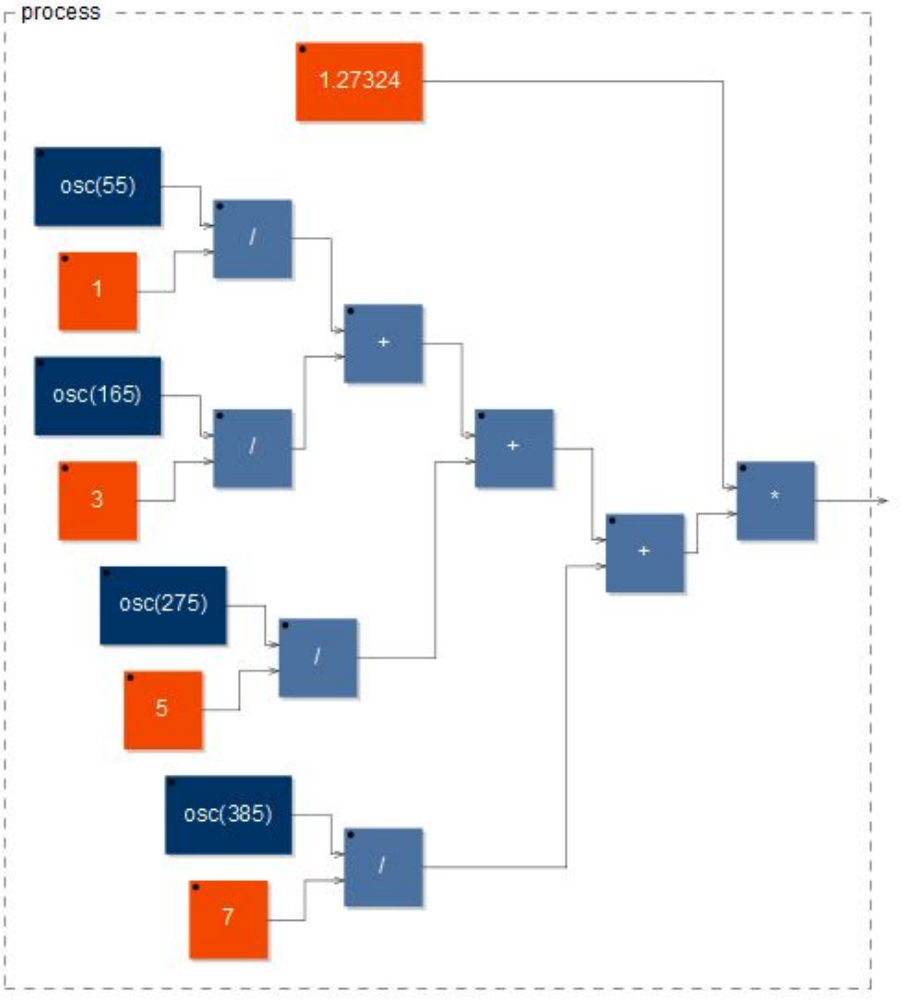


$$\left(2.2 - \left(\frac{X}{11}\right)\right) + \left(7 * \cos(Y)\right)$$


X
11
Y
2.2
/
7
cos
-
*
+

About FAUST

- **Functional** programming language for designing sound synthesizers and audio processors
- Suitable for tree-based representation



```
import("stdfaust.lib");

// Core Notation

process = (4, ma.PI : /), ((((os.osc(55), 1 : /),
(os.osc(165), 3 : /) : +), (os.osc(275), 5 : /) :
+),(os.osc(385), 7 : /) : +) : *;

// Infix Notation

process = (4 / ma.PI) * ((os.osc(55) / 1) + (os.osc(165) / 3)
+ (os.osc(275) / 5) + (os.osc(385) / 7));

// Prefix Notation

process = *(/(4, ma.PI), +(+(+(/(os.osc(55), 1),
/(os.osc(165), 3)), /(os.osc(275), 5)), /(os.osc(385), 7)));
```

Building Blocks

Functions/Operators

- +, -, *, /
- sin, cos, $\tan^{-1}$, …
- “@” (delay)
- Basic waves
  - sine, sawtooth, triangle, square
- …

Terminals

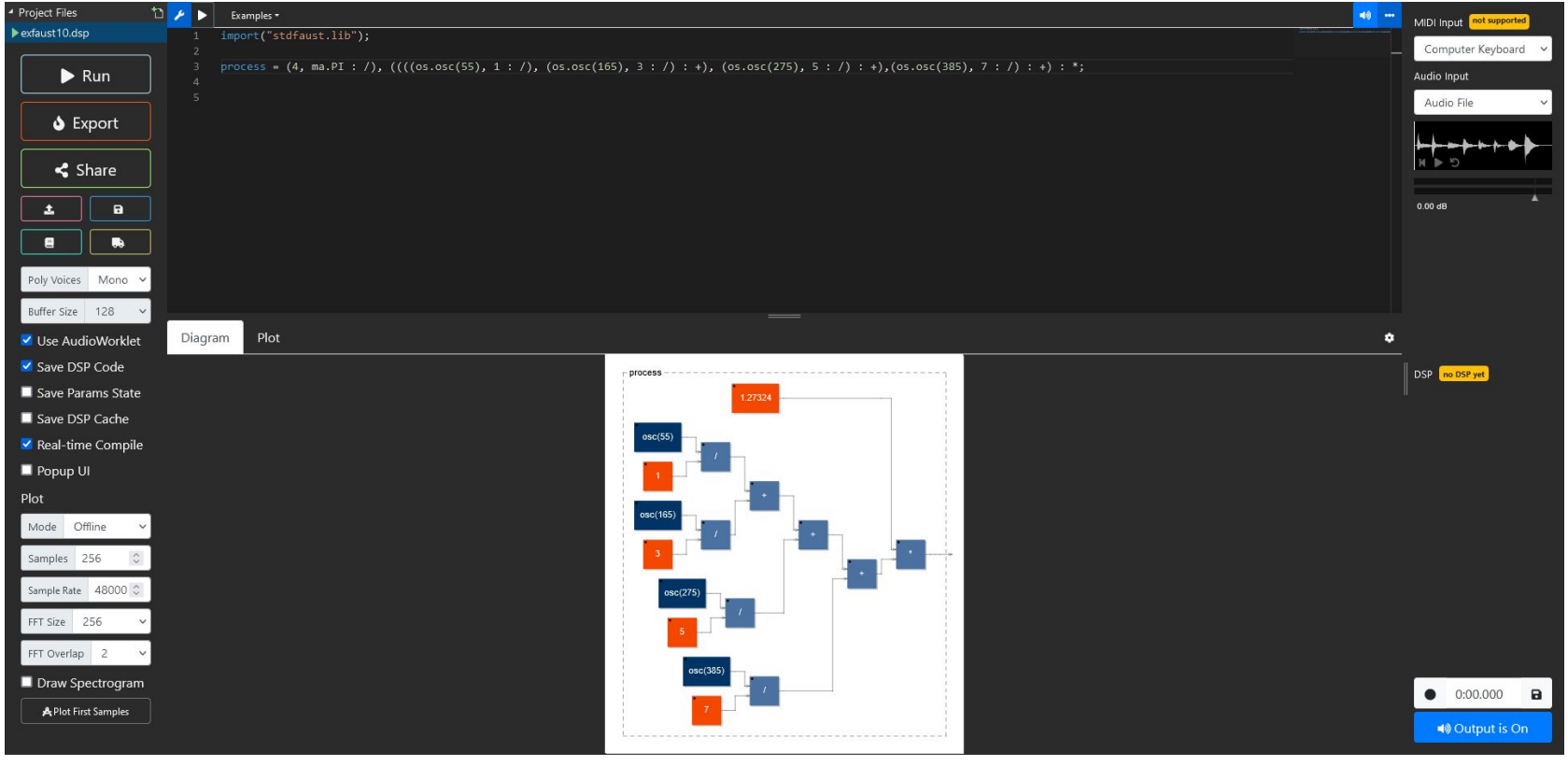
- $\mathbb{R}$ (0, 1, -½, π, …)
- random $\mathbb{R}$
  - $x \in [-1, 1]$
- random noise
  - white, pink, …
- “_” (input signal)
- …

Points to Note

- Maximum depth
- Perfect (or imperfect) trees
- Set of functions and terminals chosen
- Closure
  - *All functions must be able to accept any data type and value as arguments.*

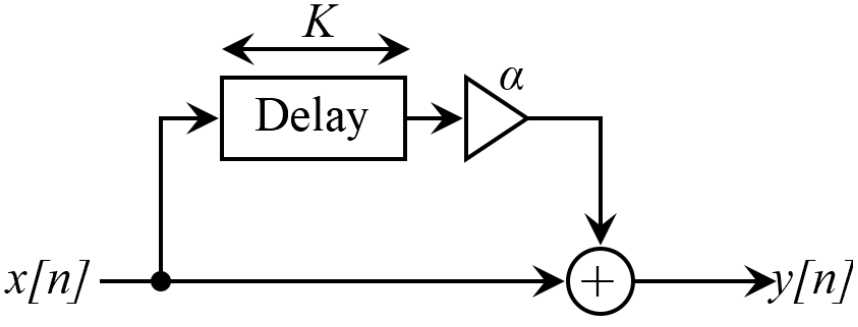
```
// Protected division in C [1]
float div(float num, float den) {
    if (!den) return 0;

    return num / den;
}


// These are all valid in FAUST:
process = os.osc(0);
process = os.osc(-440);
process = os.osc(os.osc(1));

process =
    no.pink_noise / no.noise;
```

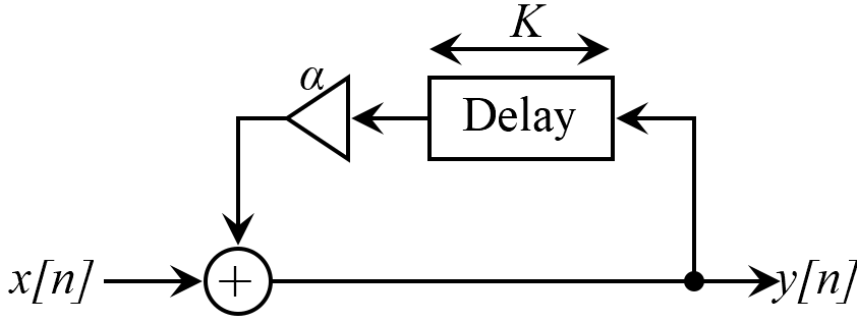
Demo


Project Files
exfaust10.dsp
Run
Export
Share
Poly Voices
Mono
Buffer Size
128
Use AudioWorklet
Save DSP Code
Save Params State
Save DSP Cache
Real-time Compile
Popup UI
Plot
Mode
Offline
Samples
256
Sample Rate
48000
FFT Size
256
FFT Overlap
2
Draw Spectrogram
Plot First Samples
Examples
import("stdfaust.lib");
process = (4, ma.PI : /), ((((os.osc(55), 1 : /), (os.osc(165), 3 : /) : +), (os.osc(275), 5 : /) : +),(os.osc(385), 7 : /) : +) : *;
Diagram
Plot
process
1.27324
osc(55)
osc(165)
osc(275)
osc(385)
MIDI Input
not supported
Computer Keyboard
Audio Input
Audio File
0.00 dB
DSP
no DSP yet
0:00.000
Output is On

Limitations

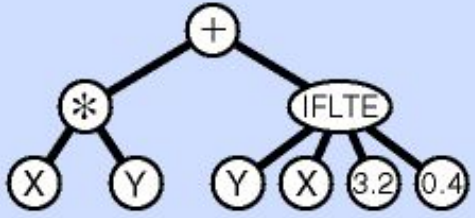
- Mono channel output
  - 2+ channel outputs only possible with disjoint trees
- No branching output or looping
  - e.g. feedforward/feedback comb filters

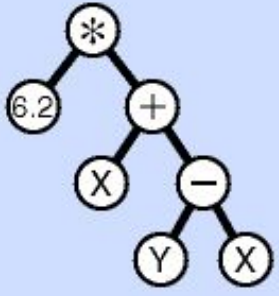




Future Work

- Genetic Programming [1]
  - Evolutionary algorithm for generating computer programs
- DSP Graphs (e.g. [2])
  - Cartesian Genetic Programming
- Research applications in music, speech, etc.
  - Modeling of new synthesizers, DSP effects
  - Reproduction of speech, acoustics, instruments, …





Thank you!

References

[1] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992. [Online]. Available: http://mitpress.mit.edu/books/genetic-programming

[2] M. Macret and P. Pasquier, “Automatic Design of Sound Synthesizers as Pure Data Patches Using Coevolutionary Mixed-Typed Cartesian Genetic Programming,” in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 2014, pp. 309–316. doi: 10.1145/2576768.2598303.