Probabilistic Reactive Programming
Louis Mandel, Guillaume Baudart, Avraham Shinnar, Kiran Kate, Martin Hirzel
IBM Research

Probabilistic Reactive Models
Extend ReactiveML with probabilistic constructs

Possible Applications
- Online time series prediction
- Agent based systems
- Infrastructure self-tuning

Example: Hidden Markov Model

```
let rec process hmm obs p_prev =
  await obs([o_t]) in (* Input from the environment *)
let p_t = sample (sph_gaussian p_prev speed) in
factor (score (sph_gaussian p_t noise) o_t);
propose p_t; (* Publish a probabilistic value *)
run hmm obs p_t
```

Inference: Particles filter on execution paths
- Non terminating function
- propose during execution
- No rollback

Hybrid Application

```
let process main =
  signal obs in
  signal p_dist in
  run sensor obs ||
  infer ~output:p_dist (hmm obs [0.;0.]) ||
  run display p_dist
```

References:
- Goodman, Stuhlmüller, 2014. The Design and Implementation of Probabilistic Programming Languages
- Ritchie, Stuhlmüller, Goodman, 2016. C2: Lightweight Incrementalized MCMC for Probabilistic Programs using Continuations and Callsite Caching

Alternative: Instantaneous Model
- Explicit inference at each step
- Sample from the previous distribution

```
let rec process hmm' obs p_dist last_dist =
  await obs(o_t) in
  let dist =
    infer (fun () ->
      let p_prev = sample (last_dist) in
      let p_t = sample (sph_gaussian p_prev speed) in
      factor (score (sph_gaussian p_t noise) o_t);
      p_t)
    in
    emit p_dist dist;
  run hmm' obs p_dist
```

Challenges
- Mixing deterministic and probabilistic parts
- Probabilistic: observe external inputs
- Deterministic: access inferred distributions
- Handle state in reactive control structures
- Parallel composition
- Internal communication
- Inference on non-terminating functions
- Sequential Monte-Carlo
- Real-time vs. non-real-time applications
- Relax the non-rollback constraint