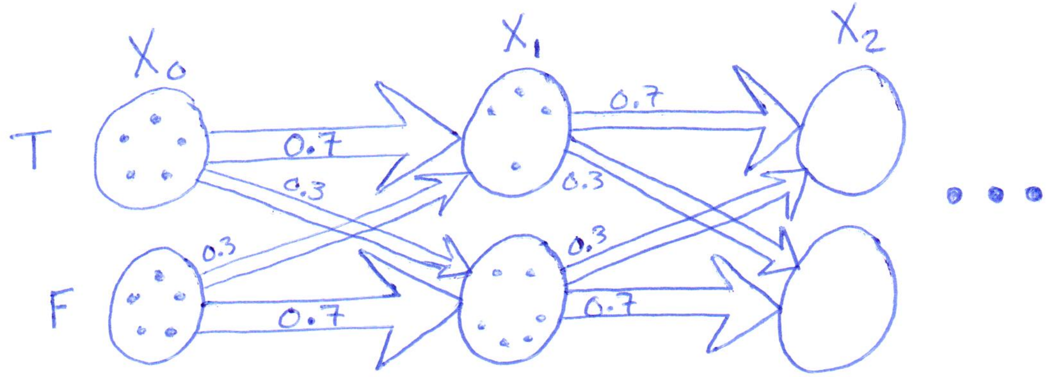
Approximate Inference (Particle Filtering)

Don't calculate the distribution of every combination of hidden RV over time.

Just take <u>samples</u>.

Transition model:

- as if pools connected by streams
- stream width is proportional to transition probability



represents

X_{t-1}	$P(X_t = T)$
T	0.7
F	0.3

Drop a quantity of particles in the pools at $t = 0$ and see where they end up.

like a uniform prior!

Example: drop 10 particles (5 in $X_0 = T$, 5 in $X_0 = F$)

- each has a 0.7 chance to flow in same lane (T/F) and 0.3 to cross over
- check particles at $t = 1$, perhaps 4 are in T, 6 in F (3 T→T, 2 T→F, 1 F→T, 4 F→F by random chance)

Sensor Model:

- if I flow the particles from $t = 1$ to $t = 2$ now, I am ignoring my observations
- after each step of transition flow, <u>weight</u> particles by likelihood of generating current evidence
- normalize weights to probability distribution of selecting each particle (sum to 1)
- resample same total number of particles <u>with replacement</u> (new samples unweighted)
- <u>now</u> flow new samples through transition model again

PARTICLE FILTERING 2

Sensor example:
- if particles from previous example are at X_1 $(4 = T, 6 = F)$
- and sensor model is

X_t	$P(E_t = T)$
T	0.9
F	0.2

- and we observe $e_1 = T$:

Then particles in $X_1 = T$ have weight 0.9 and particles in $X_1 = F$ have weight 0.2.

After normalizing, there is a $\frac{4(0.9)}{4(0.9)+6(0.2)} = 0.75$ chance to select a particle in $X_1 = T$, and $\frac{6(0.2)}{4(0.9)+6(0.2)} = 0.25$ chance to select a particle in $X_1 = F$.

Now generate (sample) 10 new, unweighted particles from that distribution: $\langle 0.75, 0.25 \rangle$.

Particle Filtering accounts for all information in DBN:
- $P(X_0)$ controls initial distribution of particles
- $P(X_{t+1} | X_t)$ controls "flow" of particles
- $P(E_t | X_t)$ controls weight of particles during resampling

Algorithm:
1. Create N particles at $t=0$ by sampling from prior
2. Propagate: "flow" samples forward in time using transition model
3. Resample: weight samples based on sensor model, normalize, and sample N new, unweighted particles
4. Repeat from 2 (particle distribution at t <u>is $P(X_t | e_t)$ estimate</u>)

Benefits:
- Never calculates the "hard" thing: $P(X_t | e_t)$
- Approximates $P(X_t | e_t)$ using discrete particles that are each always in a single, exact state.
- By increasing N, approximation becomes arbitrarily accurate (at cost of computation time)