Propagating imprecise probabilities through event trees

Gert de Cooman, Filip Hermans & Erik Quaeghebeur

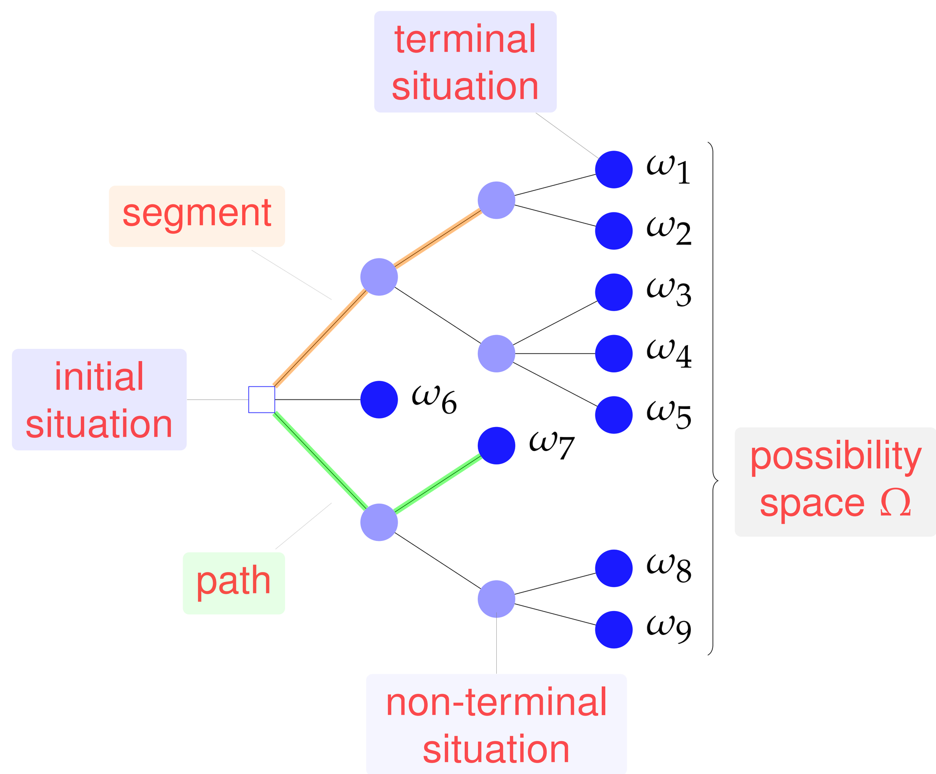
SYSTeMS Research Group
Department Of Electrical Energy, Systems & Automation
Faculty of Engineering, Ghent University
Technologiepark-Zwijnaarde 914, 9052 Zwijnaarde

{Gert.deCooman,Filip.Hermans,Erik.Quaeghebeur}@UGent.be

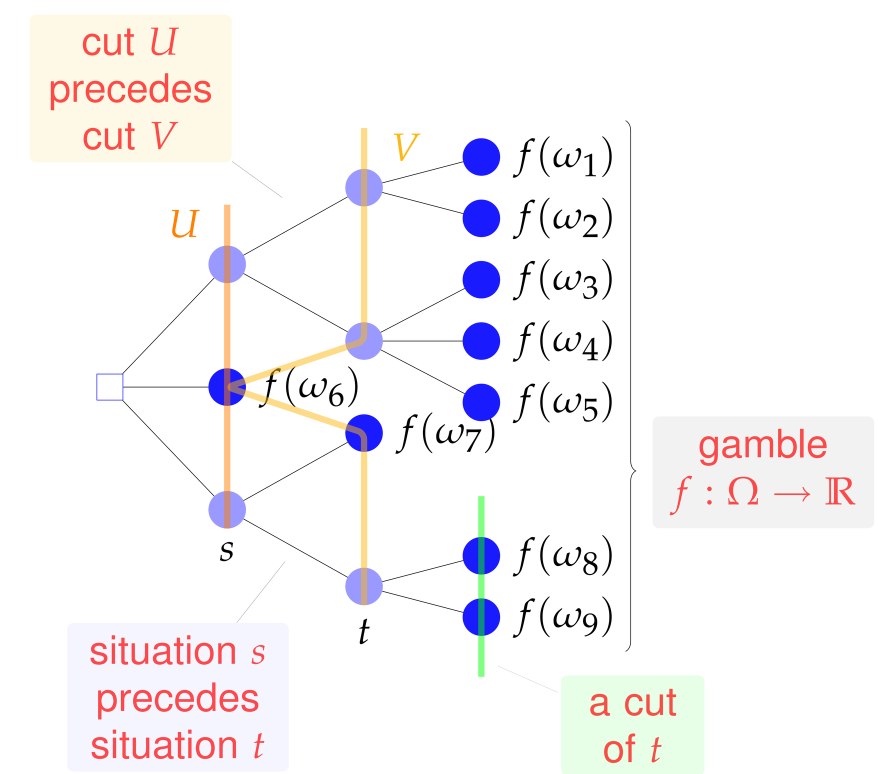
Abstract

Event trees are a graphical model of a set of possible situations and the possible paths going through them, from the initial situation to the terminal situations. With each situation, there is associated a local uncertainty model that represents beliefs about the next situation. The uncertainty models can be classical, precise probabilistic type, in which case they can be seen as sets of classical probabilities (yielding probability intervals). To work with such event trees, we must combine these local uncertainty models. We show this can be done efficiently by back-propagation through the tree, both for precise and imprecise probabilistic models, and we illustrate this using an imprecise probabilistic counterpart of the classical Markov chain. This allows us to perform a robustness analysis for Markov chains very efficiently.

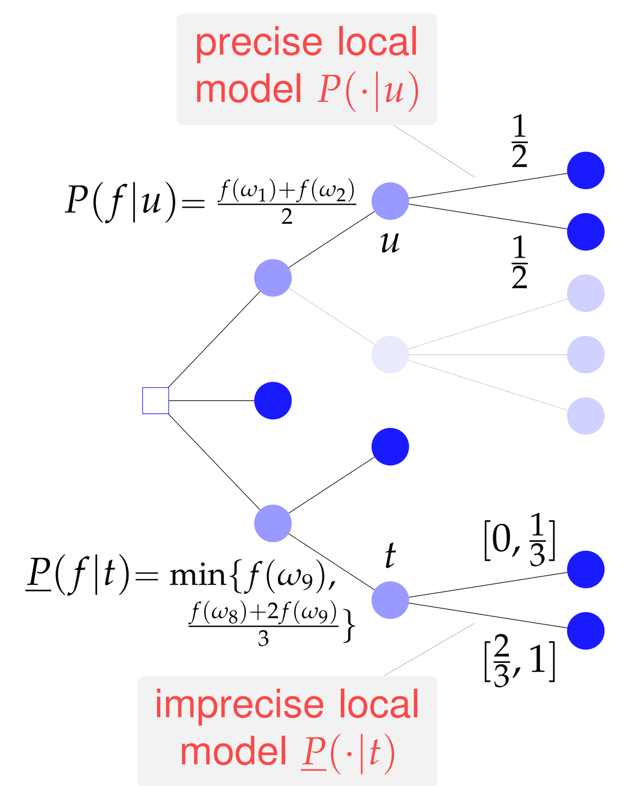
Event trees & possibility spaces



precedence, cuts & gambles



local models & previsions



Conservative coherent reasoning

Consider two random variables X and Y, a (not necessarily local) conditional model $\underline{P}(\cdot|Y)$ for X and the marginal model $\underline{P}$ for Y. These can be combined to get the smallest (most conservative) coherent joint model for (X,Y) by means of the Marginal Extension Theorem.

Theorem 1 (Marginal Extension Theorem) *Assume we have:*

1. a separately coherent conditional lower prevision $\underline{P}(\cdot|Y)$,

2. a coherent marginal lower prevision $\underline{P}$.

Then the smallest coherent joint lower prevision $\underline{M}$ is given by

$$\underline{M} = \underline{P}(\underline{P}(\cdot|Y)).$$

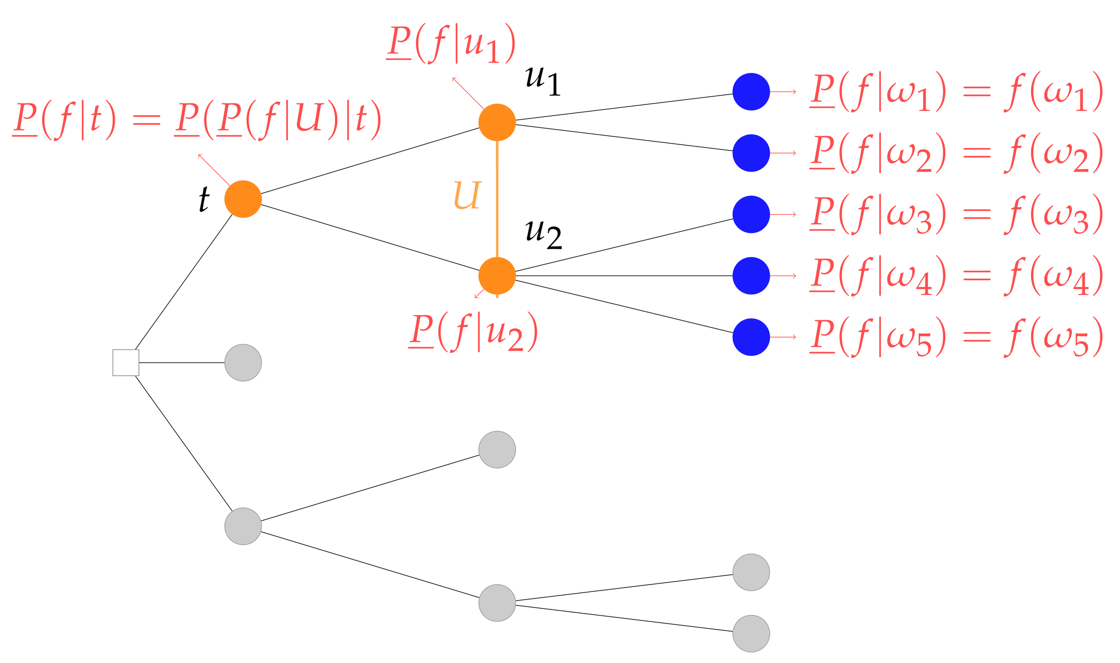
Implications of the conditional assesments

Due to the particular structure of event trees, the Marginal Extension Theorem can be reformulated into the computationally attractive Concatenation Formula.

Theorem 2 (Concatenation Formula) *Consider any cut U of a situation t. Then for all gambles f on Ω,*

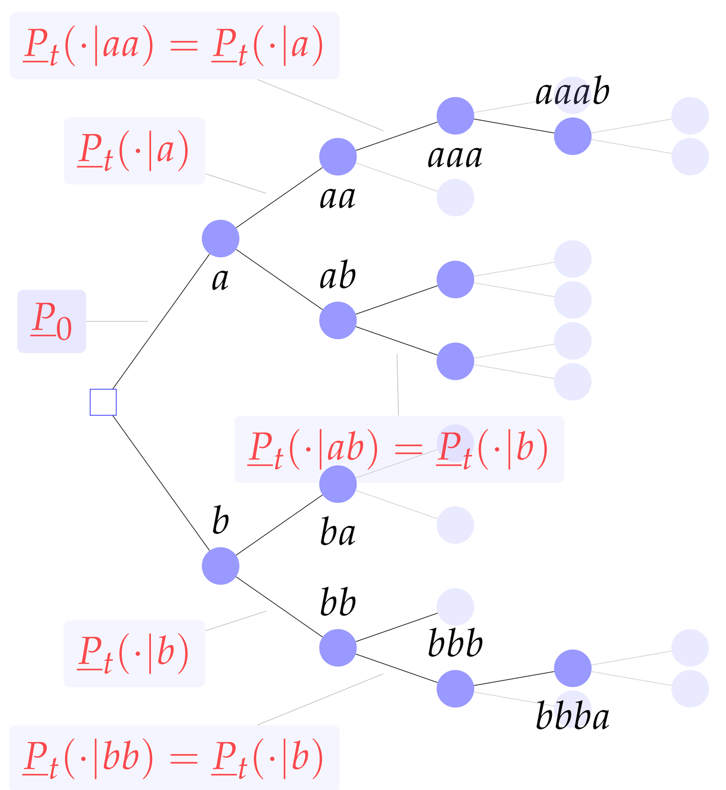
$$\underline{P}(f|t) = \underline{P}(\underline{P}(f|U)|t).$$

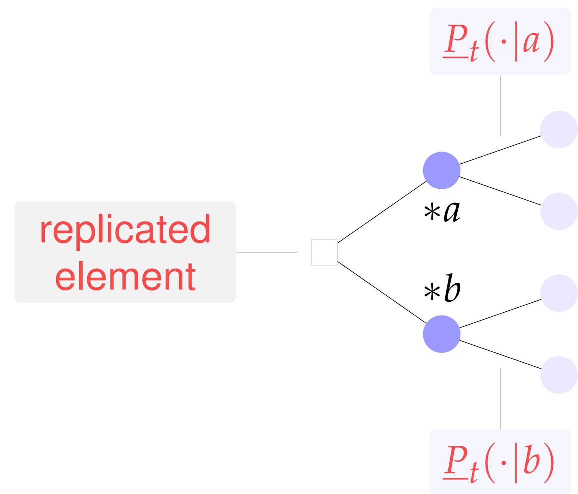
This theorem tells us that all predictive lower (and upper) previsions can be calculated using backwards recursion, by starting with the trivial predictive previsions $\underline{P}(f|\Omega) = \overline{P}(f|\Omega) = f$ for the terminal cut Ω, and using only the local models $\underline{P}(\cdot|t)$.



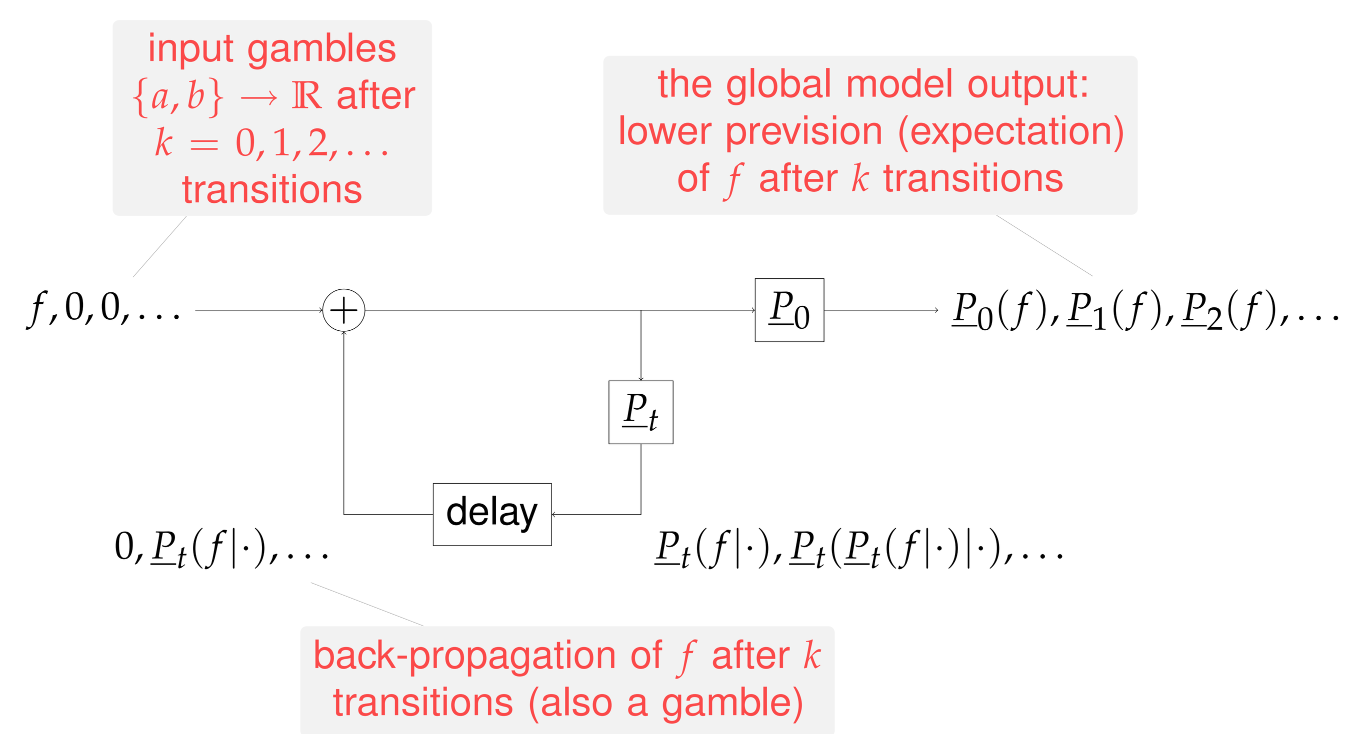

```
double getLowerPrevision(Gamble, Node);

double getLowerPrevisionInTree(Gamble f, Node t)
{
   if t.isTerminal()
     return f(t);

   // else, we are not in a terminal node

   // construct new Gamble g
   Gamble g;
   for(i=0,i<t.getNumberOfChildren(),i++)
   {
     Node childNode = t.getChild(i);
     g.addValue(childNode) = \
     getLowerPrevisionInTree(f,childNode);
   }
   // calculate the Natural Extension using g
   return getLowerPrevision(g,t);
}
```

Event trees and conditional independence: Markov-like behaviour





Back-propagation in block-diagram form



The block-diagram shows that the computational complexity is linear in the number of transitions k. Each back-propagation has quadratic complexity in the size of the possibility space. This holds for both precise *and* imprecise local models!

An illustrative result

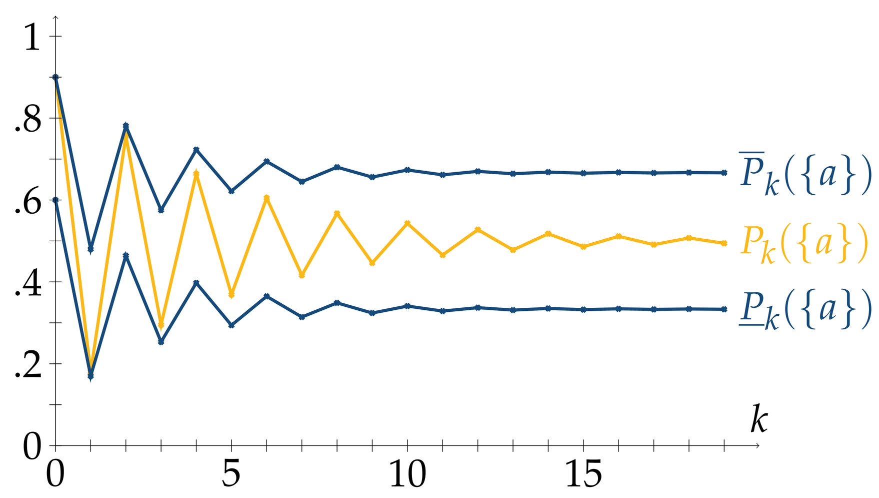
Given $\{a,b\}$, note that $\overline{P}(\{a\}) = 1 - \underline{P}(\{b\})$ and $\overline{P}(\{b\}) = 1 - \underline{P}(\{a\})$ for any imprecise model. Similarly, for any precise model, $P(\{a\}) = 1 - P(\{b\})$.

$$P_0(\{a\}) = .9$$

$$P_t\begin{bmatrix}(\{a\}|a)\\(\{a\}|b)\end{bmatrix} = \begin{bmatrix}.1\\.9\end{bmatrix}$$

$$[\underline{P}_0,\overline{P}_0](\{a\}) = [.6,.9]$$

$$[\underline{P}_t,\overline{P}_t]\begin{bmatrix}(\{a\}|a)\\(\{a\}|b)\end{bmatrix} = \begin{bmatrix}[.1,.2]\\[.8,.9]\end{bmatrix}$$



[The data for this plot was generated by a Matlab-program written by a Master's thesis student, Stefaan Dhaenens.]