Psuedo-code

int true_pairs // in units of $precision^2$.
int false_pairs // in units of $precision^2$.
int maybe_pairs // in units of $precision^2$.
// Compute the time value that contains this coarseness.
function convertToTV(**Granule** α, **int** coarseness) : **TimeValue** {

return $\lceil (2 \cdot coarseness + 1)/(2 \cdot getMaxCoarseness(\alpha)) \cdot (\alpha^* - \alpha_*) \rceil + \alpha_*$

}
// Compute the inverse: from time value to coarseness.
function convertToC(**Granule** α, *TimeValue* tv) : **int** {

return $\lfloor (2 \cdot getMaxCoarseness(\alpha) \cdot (tv - \alpha_*)/(\alpha^* - \alpha_*) - 1 \rfloor /2$

}
function lessThan(**Granule** α, **Granule** β, **int** *plaus*) : **Extended-Boolean** {

// Easy cases: no overlap of periods of indeterminancy
if $(\alpha^* < \beta_*)$ **then**
  return True
else if $(\alpha_* > \beta^*)$ **then**
  return False
  // Easy cases: no need for probabilistic analysis
else if $(plaus = POSSIBLE)$ **then**
  return True
else if $(plaus = DEFINITE)$ **then**

return False
// Last easy case: missing mass functions
else if α is missing its mass function $\vee\beta$ is missing its mass function **then**
return Maybe
// Hard case: convolution
else
$\alpha_{prec} \leftarrow$ getMaxPrecision(α)
$\beta_{prec} \leftarrow$ getMaxPrecision(β)
$\alpha_{coars} \leftarrow$ getMaxCoarseness(α)
$\beta_{coars} \leftarrow$ getMaxCoarseness(β)

$true\_pairs \leftarrow \alpha_{prec} \cdot \beta_{prec} \cdot plaus/100$ // Goal of true pairs
$false\_pairs \leftarrow \alpha_{prec} \cdot \beta_{prec} \cdot (100 - plaus)/100$ // Goal of false pairs

$rodCounting(0, \alpha_{prec} - 1, 0, \beta_{prec} - 1)$

if $(true\_pairs \leq 0)$ **then**
return True // Found sufficent number of true pairs
else if $(false\_pairs \leq 0)$ **then**
return False // Found sufficent number of false pairs
else
return Maybe // Found insufficient evidence of either
end if
end if

}
procedure rodCounting(**int** α_{from}, **int** α_{to}, **int** β_{from}, **int** β_{to}) {
// All four parameters are precisions

if $(true\_pairs \leq 0 \vee false\_pairs \leq 0)$ **then**
return // We already found sufficient evidence
else if $(\alpha_{to} < \alpha_{from} \vee \beta_{to} \leq \beta_{from})$ **then**
return // α or β has been used up (need to count pivoting rod for α)
end if

int $\alpha_{pivot} \leftarrow \alpha_{from} + (\alpha_{to} - \alpha_{from})/2$ // Precision
// Amount of precision on the left of the pivot

int $\alpha_{left} \leftarrow \alpha_{pivot} - \alpha_{from}$
// Amount of precision on the right of the pivot
int $\alpha_{right} \leftarrow \alpha_{to} - \alpha_{pivot}$

// Locate time value within α of the pivot
$tv \leftarrow convertToTV(\alpha, getCoarseness(\alpha, \alpha_{pivot}))$
// Precision within β of tv
$\beta_{pivot} \leftarrow getPrecision(\beta, convertToC(\beta, tv))$

// Amount of precision on the left of the pivot
$\beta_{left} \leftarrow \beta_{pivot} - \beta_{from}$
// Amount of precision on the right of the pivot
$\beta_{right} \leftarrow \beta_{to} - \beta_{pivot}$

// Found some true pairs, including the α pivot
$true\_pairs \leftarrow true\_pairs - (\alpha_{left} + 1) \cdot \beta_{right}$
// Found some false pairs, cannot include the β pivot
$false\_pairs \leftarrow false\_pairs - \alpha_{right} \cdot \beta_{left}$

// Recurse on two subproblems
$rodCounting(\alpha_{from}, \alpha_{pivot}, \beta_{from}, \beta_{pivot})$
$rodCounting(\alpha_{pivot} + 1, \alpha_{to}, \beta_{pivot} + 1, \beta_{to})$

}
function equalTo(**Granule** α, **Granule** β, **int** $plaus$) : **ExtendedBoolean**
{

// Easy cases: no overlap of periods of indeterminancy
if $(\alpha^* < \beta_*)$ **then**
  return False
else if $(\alpha_* > \beta^*)$ **then**
  return False
  // Easy cases: no need for probabilistic analysis
else if $(plaus = POSSIBLE)$ **then**
  return True
else if $(plaus = DEFINITE)$ **then**
  return False

// Last easy case: missing mass functions
else if α is missing its mass function $\vee\beta$ is missing its mass function **then**
return Maybe
// Hard case: convolution
else
int $\alpha_{prec} \leftarrow$ getMaxPrecision(α)
int $\beta_{prec} \leftarrow$ getMaxPrecision(β)
int $\alpha_{coars} \leftarrow$ getMaxCoarseness(α)
int $\beta_{coars} \leftarrow$ getMaxCoarseness(β)

int $\alpha_{value} \leftarrow \alpha_*$ // Start at far left of α
int $\alpha_{rod} \leftarrow 0$
int $\beta_{value} \leftarrow \beta_*$ // Start at far left of β
int $\beta_{rod} \leftarrow 0$

$true\_pairs \leftarrow \alpha_{prec} \cdot \beta_{prec} \cdot plaus/100$ // Goal of true pairs
// Negation of goal of false pairs ($= true\_pairs + false\_pairs$)
$nonfalse\_pairs \leftarrow \alpha_{prec} \cdot \beta_{prec} \cdot plaus/100$

// Move α_{value} and β_{value} succesively to the right.
// Continue while there are more rods to count and not enough evidence
// has been collected
while $(\alpha_{rod} < \alpha_{prec} \wedge \beta_{rod} < \beta_{prec} \wedge true\_pairs > 0)$ **do**
// If the current α_{rod} starts at a lesser time value than the current
// β_{rod} then increment α_{rod}
if $(\alpha_{value} < \beta_{value})$ **then**
$\alpha_{value} \leftarrow convertToTV(\alpha, getCoarseness(\alpha, \alpha_{rod}))$
$\alpha_{rod} \leftarrow \alpha_{rod} + 1$
// If the current β_{rod} starts at a lesser time value than the current
// α_{rod} then increment β_{rod}
else if $(\alpha_{value} > \beta_{value})$ **then**
$\beta_{value} \leftarrow convertToTV(\beta, getCoarseness(\beta, \beta_{rod}))$
$\beta_{rod} \leftarrow \beta_{rod} + 1$
// α_{rod} and β_{rod} start at the same time value,
// count all rods within this value
else
int $\alpha_{count} \leftarrow 0$
int $\beta_{count} \leftarrow 0$

// Get the time value this α_{rod} ends at
int $\alpha_{nextvalue} \leftarrow convertToTV(\alpha, getCoarseness(\alpha, \alpha_{rod}))$
$\alpha_{rod} \leftarrow \alpha_{rod} + 1$
// Get the time value this β_{rod} ends at
int $\beta_{nextvalue} \leftarrow convertToTV(\beta, getCoarseness(\beta, \beta_{rod}))$
$\beta_{rod} \leftarrow \beta_{rod} + 1$

// While there are more α_{rods} that begin at the
// initial time value, count them
while $(\alpha_{nextvalue} = \alpha_{value} \wedge \alpha_{rod} < \alpha_{prec})$ **do**
  $\alpha_{count} \leftarrow \alpha_{count} + 1$
  $\alpha_{nextvalue} \leftarrow convertToTV(\alpha, getCoarseness(\alpha, \alpha_{rod}))$
  $\alpha_{rod} \leftarrow \alpha_{rod} + 1$
end while

// While there are more α_{rods} that begin at the
// initial time value, count them
while $(\beta_{nextvalue} = \beta_{value} \wedge \beta_{rod} < \beta_{prec})$ **do**
  $\beta_{count} \leftarrow \beta_{count} + 1$
  $\beta_{nextvalue} \leftarrow convertToTV(\beta, getCoarseness(\beta, \beta_{rod}))$
  $\beta_{rod} \leftarrow \beta_{rod} + 1$
end while

// If there are any rods that fit within one time value,
// then found some true pairs. These are not false pairs
if $(\alpha_{count} > 0 \wedge \beta_{count} > 0)$ **then**
  $true\_pairs \leftarrow true\_pairs - (\alpha_{count} + \beta_{count})$
  // Remove $true\_pairs$
  $nonfalse\_pairs \leftarrow total\_pairs - (\alpha_{count} + \beta_{count})$
end if

// Found at least one maybe pair, which is not a false pair
// Remove $maybe\_pairs$
$nonfalse\_pairs \leftarrow total\_pairs - (\alpha_{count} + \beta_{count} + 1)$

$\alpha_{value} \leftarrow \alpha_{nextvalue}$ // Set to next value
$\beta_{value} \leftarrow \beta_{nextvalue}$ // Set to next value
end if

```
  end while

  if (true_pairs ≤ 0) then
    return True // Found sufficient number of true pairs
  else if (nonfalse_pairs > 0) then
    return False // Total of true and maybe pairs did not reach goal
  else
    return Maybe
  end if

end if

}
function narrow(Granule α, int cred) : Granule {

if (cred is Indeterminate) then
  return α
else if (cred is Min) then
  return new Granule(α.granularity, α*)
else if (cred is Max) then
  return new Granule(α.granularity, α*)
else
  // cred is Expected
  int coarseness ← getCoarseness(α, getMaxPrecision(α)/2)
  int value ← convertToTV(α, coarseness)
  return new Granule(α.granularity, value)
end if

}
```