NewtonGF[BoltzmannParameter] - compute the point giving a desired expected size

Calling Sequence

BoltzmannParameter(**Sys**, **labelling**, **struct**, **size**)

Parameters

`Sys` – set of equations; a grammar in the combstruct syntax
labelling - one of labelled, labeled, unlabelled, unlabeled, as in combstruct
struct - one of the structures defined by Sys
`size` – the expected size

Description

- The **BoltzmannParameter** command returns a nonnegative real number where the expected size of the structure struct defined by Sys is size.
- It is computed by dichotomy using **NewtonGF[BoltzmannExpectedSize]** for the evaluation at nonnegative real values.
- This command is part of the **NewtonGF** package, so it can be used in the form **BoltzmannParameter(..)** only after executing the command **with(NewtonGF)**. However, it can always be accessed through the long form of the command by using **NewtonGF[BoltzmannParameter]**(..).

Examples

```
> with(NewtonGF);
```

$$[BoltzmannExpectedSize, BoltzmannParameter, GFSeries, NumericalNewtonIteration, Radius, SeriesNewtonIteration] \quad (2.1)$$

A grammar for binary sequences.

```
> sys:={B = Sequence(Union(Z, Z))};
```

$$sys := \{B = Sequence(Union(Z, Z))\} \quad (2.2)$$

Here we have explicit generating functions:
combstruct[gfeqns](sys, labeled, z);

$$\left[B(z) = \frac{1}{1-2z}, Z(z) = z\right]$$

```
> BoltzmannParameter(sys,labeled,B,10);
```

$$0.4545454545 \quad (2.3)$$

Check:

```
> BoltzmannExpectedSize(sys,labeled)(%);
```

$$[E(B) = 9.999999991, E(Z) = 1.000000000] \quad (2.4)$$

A grammar for set partitions.

```
> sys:={F = Set(Set(Z, 1 <= card))};
```

$$sys := \{F = Set(Set(Z, 1 \le card))\} \quad (2.5)$$

combstruct[gfeqns](sys, labeled, z);

$$\left[F(z) = e^{e^z - 1}, Z(z) = z\right]$$

```
> BoltzmannParameter(sys,labeled,F,1000);
```

$$5.249602852 \tag{2.6}$$

```
> BoltzmannParameter(sys,unlabeled,F,1000);
```

$$0.9604922247 \tag{2.7}$$

A grammar for ternary trees (Sloane A001764).

```
> sys:={F=Union(Epsilon,Prod(Z,T)), T = Prod(F,F,F)};
```

$$sys := \{F = Union(\mathrm{E}, Prod(Z, T)), T = Prod(F, F, F)\} \tag{2.8}$$

In this example, the generating function do not have a very nice closed form:
combstruct[gfeqns](sys, labeled, z); combstruct[gfsolve](sys, labeled, z);

$$[F(z) = 1 + z\,T(z), T(z) = F(z)^3, Z(z) = z]$$

$$\{F(z) = 1 + z\,RootOf(1 + z^3\,\_Z^3 + 3\,z^2\,\_Z^2 + (-1 + 3\,z)\,\_Z), T(z) = RootOf(1 + z^3\,\_Z^3 + 3\,z^2\,\_Z^2 + (-1 + 3\,z)\,\_Z), Z(z) = z\}$$

```
> BoltzmannParameter(sys,labeled,T,100);
```

$$0.1481372200 \tag{2.9}$$

```
> BoltzmannParameter(sys,labeled,F,100);
```

$$0.1481469204 \tag{2.10}$$

See Also

combstruct[gfseries], NewtonGF[SeriesNewtonIteration], NewtonGF[NumericalNewtonIteration], NewtonGF[BoltzmannExpectedSize], NewtonGF