QHypergeometricCT Package

```
> restart;
> read("QHypergeometricCT.mm");
> with(QHypergeometricCT);
```

$$[QDispersionSet, QHypergeomReduction, QHypergeomTelescoping, QOrderBounds, QPolynomialReduction, QRNF, QShellReduction, QShiftFactorization, QShiftQuotient, QSplitFactor, VerifyQHypergeomReduction, VerifyQHypergeomTelescoping, VerifyQShellReduction] \tag{1.1}$$

QHypergeomReduction

Calling sequence:

QHypergeomReduction(T,k,q)

Input: T, a q-hypergeometric term over F(q^k);
k, a variable name;
q, an indeterminate over QQ used for the parameter q.

Output: [[g,r],H], where
g,r - two rational functions in F(q^k),
H - a q-hypergeometric term over F(q^k) with Qk(H)/H = K being Qk-standard,
such that
(1) T = (Qk-1)(g*H)+r*H;
(2) r is a Qk-remainder w.r.t. K.

```
> T := q^k*(q^(2*k+3)-q^(k+2)-q^(k+1)-q^2+q+1)/(q^(k+1)-1)/(q^(k+2)
  -1)*QDifferenceEquations:-QPochhammer(q,q,k);
```

$$T := \frac{1}{\left(q^{k+1}-1\right)\left(q^{k+2}-1\right)}\left(q^{k}\left(q^{2k+3}-q^{k+2}-q^{k+1}-q^{2}+q+1\right) QDifferenceEquations{:-}QPochhammer(q,q,k)\right) \tag{2.1}$$

```
> res := QHypergeomReduction(T,k,q); #Apply the algorithm
  QHypergeomReduction to T w.r.t. k
```

$$res := \left[\left[\frac{-q^{k}q+q+1}{\left(q^{k}q-1\right)q}, \frac{q\,q^{k}}{q^{2}q^{k}-1}\right], \left(\left(q^{2k+3}-q^{k+2}-q^{k+1}-q^{2}+q+1\right) QDifferenceEquations{:-}QPochhammer(q,q,k)\left(q^{k}q-1\right)\left(q^{2}q^{k}-1\right)\right) \Big/ \left(\left(q^{k+1}-1\right)\left(q^{k+2}-1\right)\left(q^{3}\left(q^{k}\right)^{2}-q^{2}q^{k}-q^{2}-q^{k}q+q+1\right)\right)\right] \tag{2.2}$$

```
> VerifyQHypergeomReduction(res,T,k,q);
```

$$true \tag{2.3}$$

QHypergeomTelescoping

Calling sequence:

HypergeomTelescoping(T,x,y,Sx,'cert')

Input: T, a hypergeometric term over QQ(x,y);
x,y, two variable names;
Sx, an operator name used for the shift operator w.r.t. x;
cert, (optional) an unevaluated name used for storing the certificate.

Output: L, a nonzero linear difference operator in QQ(x)[Sx]
such that
L(T) = (Sy-1)(G),
where G is a hypergeometric term over QQ(x,y), if such an L exists.
The optional 5th argument 'cert' is assigned [g,H] with G = g*H,
where g is a rational function in QQ(x,y) and H is a hypergeometric
term over QQ(x,y) whose Sy-quotient is Sy-reduced.
"No telescoper exists!",
if there is no such L.

```
> T := QDifferenceEquations:-QBinomial(n,k,q);
```

$$T := QDifferenceEquations{:-}QBinomial(n, k, q) \tag{3.1}$$

```
> L := QHypergeomTelescoping(T,n,k,q,Qn,'cert'); #Apply the algorithm QHypergeomTelescoping to T w.r.t. n and k, yielding a minimal telescoper and a corresponding certificate
cert;
```

$$L := 1 - q^n q - 2\,Qn + Qn^2$$

$$\left[\frac{-(q^n)^2 (q^k)^2 q^3 + q^3 (q^n)^2 q^k + q^2 q^n (q^k)^2 - q^n q^k q^2}{(q^n q - q^k)(q^n q^2 - q^k)}, QDifferenceEquations{:-}QBinomial(n, k, q)\right] \tag{3.2}$$

```
> VerifyQHypergeomTelescoping([L,cert],T,n,k,q,Qn);
```

$$true \tag{3.3}$$

```
> QHypergeomTelescoping(T,n,k,q,Qn); #Apply the algorithm HypergeomTelescoping to T w.r.t. x and y, yielding a minimal telescoper without computing a certificate
```

$$1 - q^n q - 2\,Qn + Qn^2 \tag{3.4}$$

```
> QDifferenceEquations:-Zeilberger(T,n,k,q,Qn); #Apply Zeilberger's algorithm to T w.r.t. x and y, yielding a minimal telescoper and a corresponding certificate
```

$$\left[1 - q^n q - 2\,Qn + Qn^2, -\frac{q^n (q^n q - 1)(-1 + q^k) q^k q^2\, QDifferenceEquations{:-}QBinomial(n, k, q)}{(-q^n q^2 + q^k)(-q^n q + q^k)}\right] \tag{3.5}$$