tf.contrib.bayesflow.csiszar_divergence.kl_reverse

```
kl_reverse(
    logu,
    self_normalized=False,
    name=None
)
```

Defined in `tensorflow/contrib/bayesflow/python/ops/csiszar_divergence_impl.py`.

The reverse Kullback-Leibler Csiszar-function in log-space.

A Csiszar-function is a member of,

```
F = { f:R_+ to R : f convex }.
```

When `self_normalized = True`, the KL-reverse Csiszar-function is:

```
f(u) = -log(u) + (u - 1)
```

When `self_normalized = False` the `(u - 1)` term is omitted.

Observe that as an f-Divergence, this Csiszar-function implies:

```
D_f[p, q] = KL[q, p]
```

The KL is "reverse" because in maximum likelihood we think of minimizing `q` as in `KL[p, q]`.

Warning: when self_normalized = True`this function makes non-log-space calculations and may therefore be numerically unstable for`|logu| >> 0`.

Args:

- `logu`: `float`-like `Tensor` representing `log(u)` from above.
- `self_normalized`: Python `bool` indicating whether `f'(u=1)=0`. When `f'(u=1)=0` the implied Csiszar f-Divergence remains non-negative even when `p, q` are unnormalized measures.
- `name`: Python `str` name prefixed to Ops created by this function.

Returns:

- `kl_reverse_of_u`: `float`-like `Tensor` of the Csiszar-function evaluated at `u = exp(logu)`.

Raises:

- `TypeError`: if `self_normalized` is `None` or a `Tensor`.

Last updated November 2, 2017.