tf.contrib.bayesflow.csiszar_divergence.symmetrized_csiszar_function

```
symmetrized_csiszar_function(
    logu,
    csiszar_function,
    name=None
)
```

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

Symmetrizes a Csiszar-function in log-space.

A Csiszar-function is a member of,

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

The symmetrized Csiszar-function is defined as:

```
f_g(u) = 0.5 g(u) + 0.5 u g (1 / u)
```

where `g` is some other Csiszar-function.

We say the function is "symmetrized" because:

```
D_{f_g}[p, q] = D_{f_g}[q, p]
```

for all `p << >> q` (i.e., `support(p) = support(q)`).

There exists alternatives for symmetrizing a Csiszar-function. For example,

```
f_g(u) = max(f(u), f^*(u)),
```

where `f^*` is the dual Csiszar-function, also implies a symmetric f-Divergence.

Example:

When either of the following functions are symmetrized, we obtain the Jensen-Shannon Csiszar-function, i.e.,

```
g(u) = -log(u) - (1 + u) log((1 + u) / 2) + u - 1
h(u) = log(4) + 2 u log(u / (1 + u))
```

implies,

```
f_g(u) = f_h(u) = u log(u) - (1 + u) log((1 + u) / 2)
       = jensen_shannon(log(u)).
```

Warning: 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.

- `csiszar_function`: Python `callable` representing a Csiszar-function over log-domain.
- `name`: Python `str` name prefixed to Ops created by this function.

Returns:

- `symmetrized_g_of_u`: `float`-like `Tensor` of the result of applying the symmetrization of `g` evaluated at `u = exp(logu)`.

Last updated November 2, 2017.