tf.sets.set_union

Aliases:

- `tf.contrib.metrics.set_union`
- `tf.sets.set_union`

```
set_union(
    a,
    b,
    validate_indices=True
)
```

Defined in `tensorflow/python/ops/sets_impl.py`.

See the guide: Metrics (contrib) > Set `Ops`

Compute set union of elements in last dimension of `a` and `b`.

All but the last dimension of `a` and `b` must match.

Example:

```
import tensorflow as tf
import collections

# [[{1, 2}, {3}], [{4}, {5, 6}]]
a = collections.OrderedDict([
    ((0, 0, 0), 1),
    ((0, 0, 1), 2),
    ((0, 1, 0), 3),
    ((1, 0, 0), 4),
    ((1, 1, 0), 5),
    ((1, 1, 1), 6),
])
a = tf.SparseTensor(list(a.keys()), list(a.values()), dense_shape=[2, 2, 2])

# [[{1, 3}, {2}], [{4, 5}, {5, 6, 7, 8}]]
b = collections.OrderedDict([
    ((0, 0, 0), 1),
    ((0, 0, 1), 3),
    ((0, 1, 0), 2),
    ((1, 0, 0), 4),
    ((1, 0, 1), 5),
    ((1, 1, 0), 5),
    ((1, 1, 1), 6),
    ((1, 1, 2), 7),
    ((1, 1, 3), 8),
])
b = tf.SparseTensor(list(b.keys()), list(b.values()), dense_shape=[2, 2, 4])

# `set_union` is applied to each aligned pair of sets.
tf.sets.set_union(a, b)

# The result will be a equivalent to either of:
#
# np.array([[{1, 2, 3}, {2, 3}], [{4, 5}, {5, 6, 7, 8}]])
#
# collections.OrderedDict([
#     ((0, 0, 0), 1),
#     ((0, 0, 1), 2),
#     ((0, 0, 2), 3),
#     ((0, 1, 0), 2),
#     ((0, 1, 1), 3),
#     ((1, 0, 0), 4),
#     ((1, 0, 1), 5),
#     ((1, 1, 0), 5),
#     ((1, 1, 1), 6),
#     ((1, 1, 2), 7),
#     ((1, 1, 3), 8),
# ])
```

Args:

- `a`: `Tensor` or `SparseTensor` of the same type as `b`. If sparse, indices must be sorted in row-major order.
- `b`: `Tensor` or `SparseTensor` of the same type as `a`. If sparse, indices must be sorted in row-major order.
- `validate_indices`: Whether to validate the order and range of sparse indices in `a` and `b`.

Returns:

A `SparseTensor` whose shape is the same rank as `a` and `b`, and all but the last dimension the same. Elements along the last dimension contain the unions.

Last updated November 2, 2017.