BERKELEY LAB

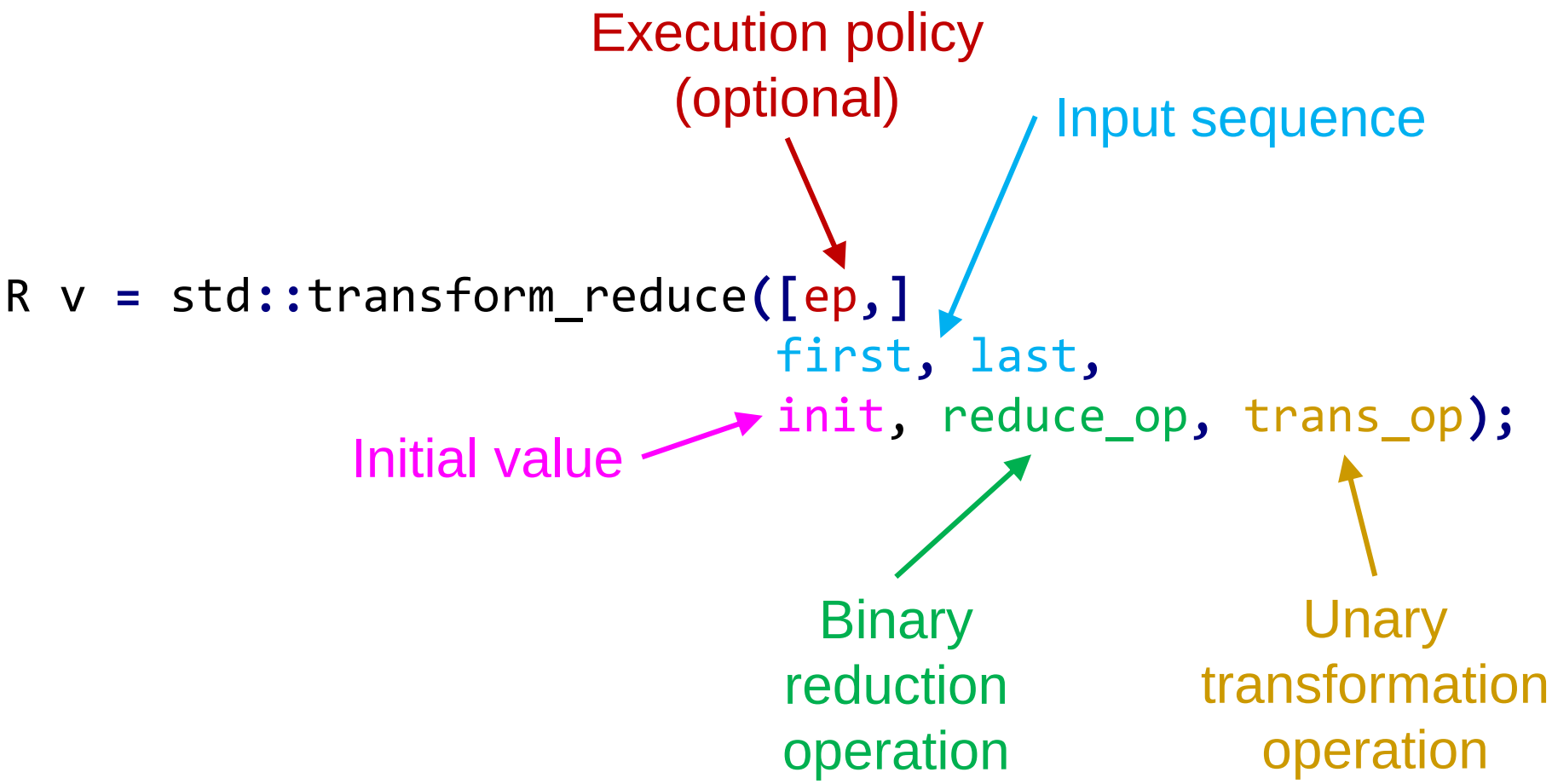
LAWRENCE BERKELEY NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

`std::transform_reduce`

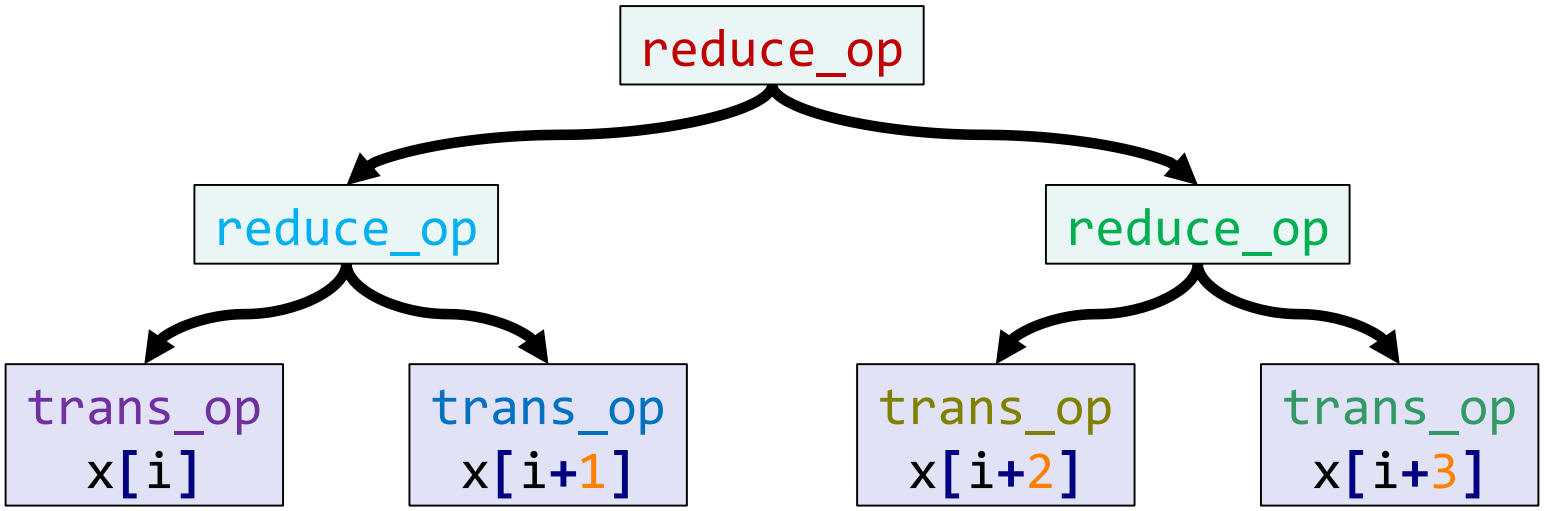
Bryce Adelstein Lelbach

Computer Architecture Group, Computing Research Division



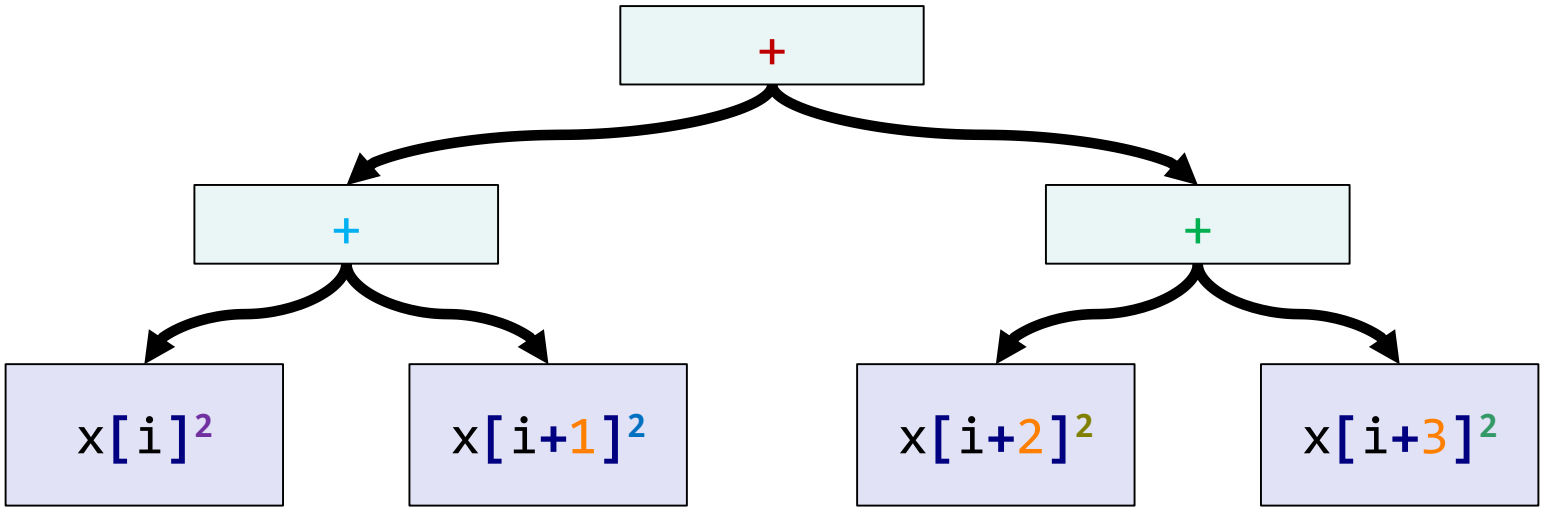
Copyright (C) 2016 Bryce Adelstein Lelbach

```
R v = reduce_op(
  reduce_op(trans_op(x[i]  ), trans_op(x[i+1])),
  reduce_op(trans_op(x[i+2]), trans_op(x[i+3])));
```



Copyright (C) 2016 Bryce Adelstein Lelbach

$$R\ v = (x[i]^2 + x[i+1]^2) + (x[i+2]^2 + x[i+3]^2) \ldots$$



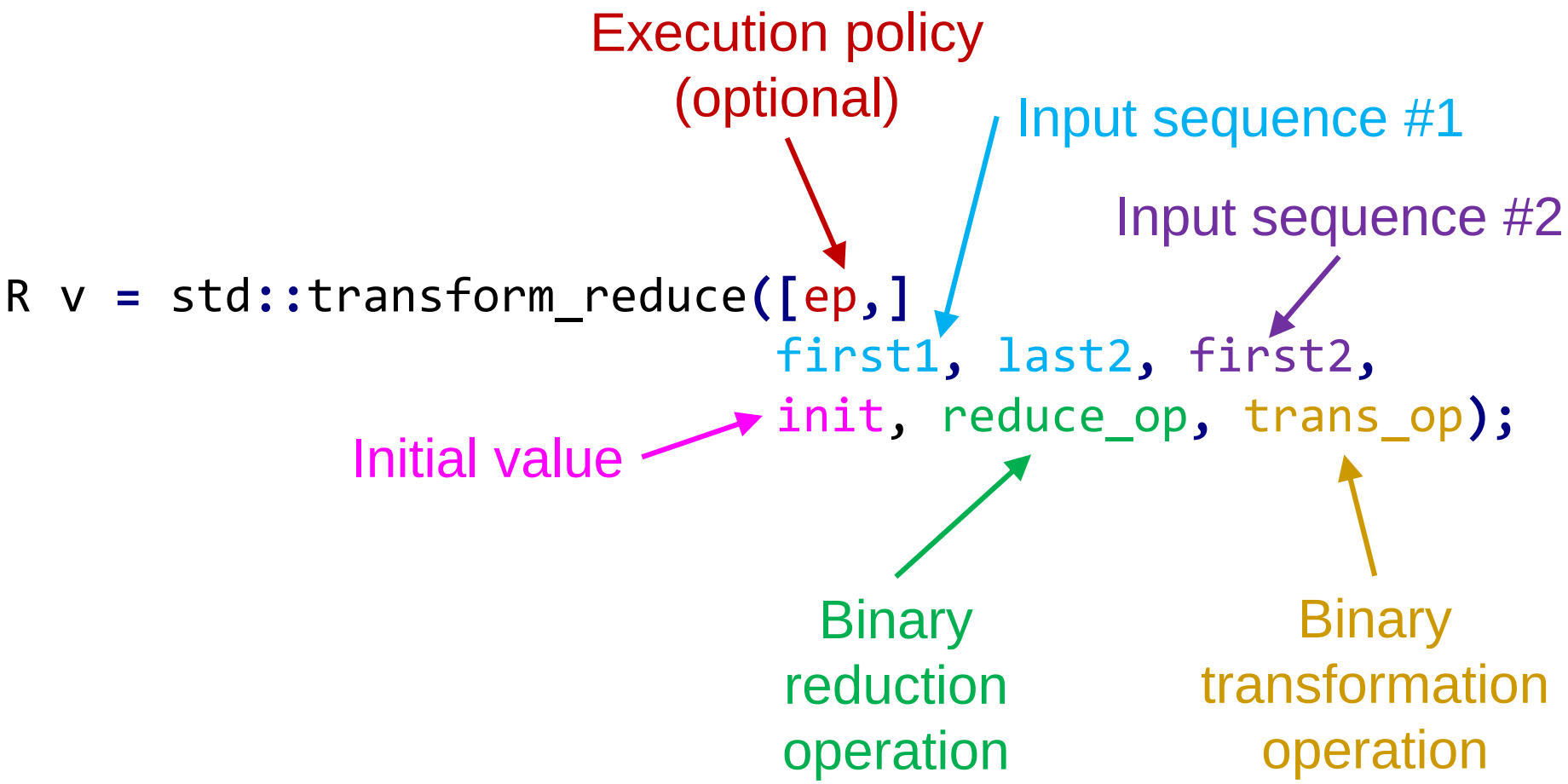
Copyright (C) 2016 Bryce Adelstein Lelbach

```
std::vector<double> x = // ...

double norm =
  std::sqrt((x[0] * x[0]) + (x[1] * x[1]) + /* ... */);
```


Copyright (C) 2016 Bryce Adelstein Lelbach

```
std::vector<double> x = // ...

double norm =
  std::sqrt(
    std::transform_reduce(
      std::execution::par_unseq,

      // Input sequence.
      x.begin(), x.end(),

      // Initial reduction value.
      double(0.0),

      // Binary reduction op.
      [] (double xl, double xr) { return xl + xr; },

      // Unary transform op.
      [] (double x) { return x * x; }
    )
  );
```


Copyright (C) 2016 Bryce Adelstein Lelbach



Copyright (C) 2016 Bryce Adelstein Lelbach

```
std::vector<double> x = // ...
std::vector<double> y = // ...

double dot_product =
  (x[0] * y[0]) + (x[1] * y[1]) + // ...
```


Copyright (C) 2016 Bryce Adelstein Lelbach

```
std::vector<double> x = // ...
std::vector<double> y = // ...

double dot_product = std::transform_reduce(
  std::execution::par_unseq, x.begin(), x.end(), y.begin());
```


Copyright (C) 2016 Bryce Adelstein Lelbach

```
std::size_t word_count(std::string_view s) {
  // Goal: Count the number of word "beginnings" in the
  // input sequence.

  // ...
```


Copyright (C) 2016 Bryce Adelstein Lelbach

```
bool is_word_beginning(char left, char right) {
  // If left is a space and right is not, we've hit a
  // new word.
  return std::isspace(left) && !std::isspace(right);
}
```


Copyright (C) 2016 Bryce Adelstein Lelbach

```
std::size_t word_count(std::string_view s) {
  if (s.empty()) return 0;

  // ...
```


Copyright (C) 2016 Bryce Adelstein Lelbach

```cpp
std::size_t word_count(std::string_view s) {
  if (s.empty()) return 0;

  // If the first character is not a space, then it's the
  // beginning of a word.
  std::size_t wc = (!std::isspace(s.front()) ? 1 : 0);

  // ...
```


Copyright (C) 2016 Bryce Adelstein Lelbach

```
std::size_t word_count(std::string_view s) {
  // ...

  // Count the number of characters that start a new word
  wc +=
    std::transform_reduce(
      std::execution::par_unseq,

      // "Left" input:  s[0], s[1], ..., s[s.size() - 2]
      s.begin(), s.end() - 1,
      // "Right" input: s[1], s[2], ..., s[s.size() - 1]
      s.begin() + 1,

      // ...
```


Copyright (C) 2016 Bryce Adelstein Lelbach

```
std::size_t word_count(std::string_view s) {
  // ...

  // Count the number of characters that start a new word
  wc +=
    std::transform_reduce(
      std::execution::par_unseq,

      // "Left" input:  s[0], s[1], ..., s[s.size() - 2]
      s.begin(), s.end() - 1,
      // "Right" input: s[1], s[2], ..., s[s.size() - 1]
      s.begin() + 1,

      std::size_t(0), // Initial value for reduction.

      std::plus<std::size_t>(), // Binary reduction op.
      is_word_beginning         // Binary transform op: Return
                                // 1 when we hit a new word.
);

    // ...
```


Copyright (C) 2016 Bryce Adelstein Lelbach

```
Input sequence:
  "Whose woods these are I think I know.\n"
  "His house is in the village though;  \n"
  "He will not see me stopping here     \n"
  "To watch his woods fill up with snow.\n"
```

First Stanza of Stopping by Woods on a Snowy Evening, Robert Frost

```
Post-transform pseudo-sequence:
    00000100000100000100010100000010100000
   10001000001001001000100000001000000000
   10010000100010001001000000001000000000
   10010000010001000010000100100001000000
```


Copyright (C) 2016 Bryce Adelstein Lelbach

```
Input sequence:
  "Whose woods these are I think I know.\n"
  "His house is in the village though;  \n"
  "He will not see me stopping here     \n"
  "To watch his woods fill up with snow.\n"
```

First Stanza of Stopping by Woods on a Snowy Evening, Robert Frost

```
Post-transform pseudo-sequence:
          1  +  1  +  1 + 1+1  +  1+1 +
   1 + 1  +  1 +1 +1 + 1   +   1     +
   1 +1  + 1 + 1 + 1 +1   +    1     +
   1 +1  +  1 + 1  +  1  + 1 +1  + 1
```


Copyright (C) 2016 Bryce Adelstein Lelbach

```cpp
bool is_word_beginning(char left, char right) {
  return std::isspace(left) && !std::isspace(right);
}

std::size_t word_count(std::string_view s) {
  if (s.empty()) return 0;

  std::size_t wc = (!std::isspace(s.front()) ? 1 : 0);

  wc +=
    std::transform_reduce(
      std::execution::par_unseq,
      s.begin(), s.end() - 1,
      s.begin() + 1,
      std::size_t(0),
      std::plus<std::size_t>(),
      is_word_beginning
    );

  return wc;
}
```


Copyright (C) 2016 Bryce Adelstein Lelbach

```
bool is_word_beginning(char left, char right) {
  return std::isspace(left) && !std::isspace(right);
}

std::size_t word_count(std::string_view s) {
  if (s.empty()) return 0;

  std::size_t wc =
    std::transform_reduce(
      std::execution::par_unseq,
      s.begin(), s.end() - 1,
      s.begin() + 1,
      std::size_t(!std::isspace(s.front()) ? 1 : 0),
      std::plus<std::size_t>(),
      is_word_beginning
    );

  return wc;
}
```


Copyright (C) 2016 Bryce Adelstein Lelbach