Algorithmic How-Possibly Explanation

Michael Cuffaro[1,2]

[1]Rotman Institute of Philosophy, University of Western Ontario
[2]Munich Center for Mathematical Philosophy, LMU Munich

May 8, 2017

Exploring Scientific Method: Evidence, Explanation, and Unification in Science, LMU Munich

Landscape of explanatory concepts

Pluralism

- Causal explanation
- Mechanistic explanation
- Mathematical explanation
- Structural explanation
- How-actually explanation
- How-possibly explanation

Etc.

Landscape of explanatory concepts

Pluralism

- Causal explanation
- Mechanistic explanation
- Mathematical explanation
- Structural explanation
- How-actually explanation
- How-possibly explanation

Etc.

Landscape of explanatory concepts

Pluralism

- Causal explanation
- Mechanistic explanation
- Mathematical explanation
- Structural explanation
- How-actually explanation
- How-possibly explanation

Etc.

Outline

1. Different senses of how-possibly explanation
2. Algorithmic how-possibly explanation
3. Mechanistic explanation, structural explanation
4. Relation between mechanistic, structural, and algorithmic how-possibly explanation

Different senses of explaining how-possibly

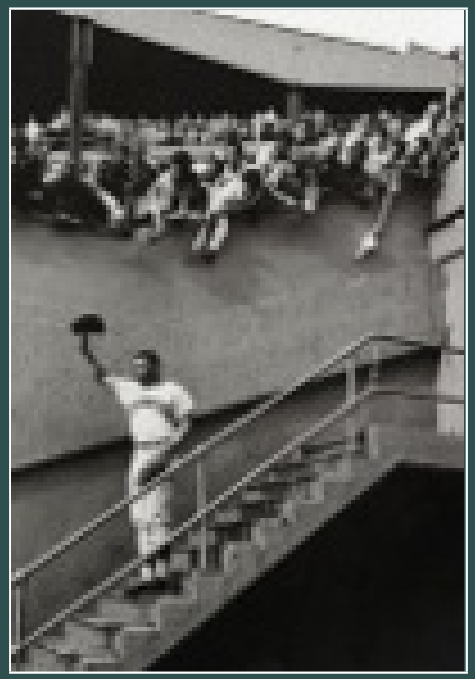
Different senses of explaining how-possibly

- Dray (1957): HPEs dispel "puzzlement"

" 'It's a long fly ball to centre field, and it's going to hit high up on the fence. The centre fielder's back, he's under it, he's caught it, and the batter is out.' " [Radio] Listeners who knew the fence was twenty feet high couldn't figure out how the fielder caught the ball" (p. 158 Dray, 1957).

Different senses of explaining how-possibly

- Dray (1957): HPEs dispel "puzzlement"



"Spectators could have given the unlikely explanation. At the rear of centre field was a high platform for the scorekeeper. The centre fielder ran up the ladder and caught the ball twenty feet above the ground" (Dray, 1957, p. 158).

Different senses of explaining how-possibly

- Dray (1957): HPEs dispel "puzzlement"
- Hempel (1965): HPEs help define and motivate how-actually questions

Different senses of explaining how-possibly

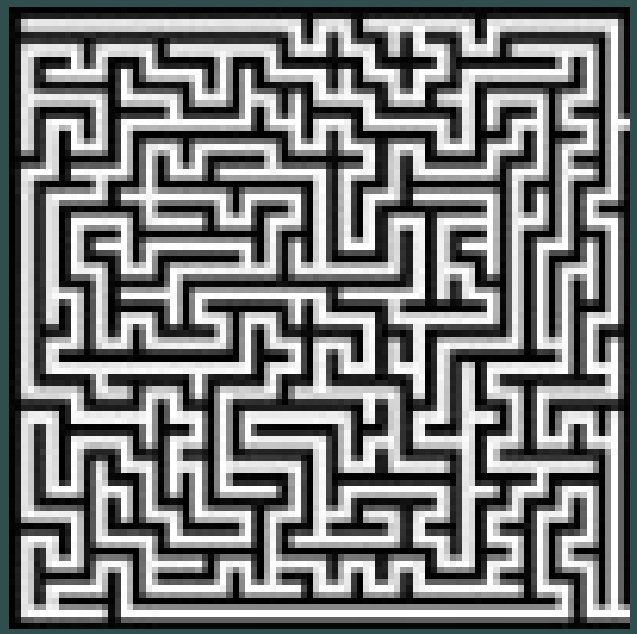
- Dray (1957): HPEs dispel "puzzlement"
- Hempel (1965): HPEs help define and motivate how-actually questions
- Resnik (1991): HPEs are unconfirmed how-actually explanations

Different senses of explaining how-possibly

- Dray (1957): HPEs dispel "puzzlement"
- Hempel (1965): HPEs help define and motivate how-actually questions
- Resnik (1991): HPEs are unconfirmed how-actually explanations
- Forber (2010): HPE is a formal mode of inquiry which carves out possibility space

Different senses of explaining how-possibly

- Forber (2010): HPE is a formal mode of inquiry which carves out possibility space



Different senses of explaining how-possibly

- Dray (1957): HPEs dispel "puzzlement"
- Hempel (1965): HPEs help define and motivate how-actually questions
- Resnik (1991): HPEs are unconfirmed how-actually explanations
- Forber (2010): HPE is a formal mode of inquiry which carves out possibility space

Different senses of explaining how-possibly

- Dray (1957): HPEs dispel "puzzlement"
- Hempel (1965): HPEs help define and motivate how-actually questions
- Resnik (1991): HPEs are unconfirmed how-actually explanations
- Forber (2010): HPE is a formal mode of inquiry which carves out possibility space
- Persson (2012): HPEs fill in the gaps in partial sketches of actual mechanisms

Different senses of explaining how-possibly

- Dray (1957): HPEs dispel "puzzlement"
- Hempel (1965): HPEs help define and motivate how-actually questions
- Resnik (1991): HPEs are unconfirmed how-actually explanations
- Forber (2010): HPE is a formal mode of inquiry which carves out possibility space
- Persson (2012): HPEs fill in the gaps in partial sketches of actual mechanisms
- Cuffaro (2015): Algorithmic HPEs

Complexity-theoretic distinctions

Complexity-theoretic distinctions

- Computability theory: Is P computable? (yes or no).
- Complexity theory: How many resources needed to solve P?

Complexity-theoretic distinctions

- Computability theory: Is P computable? (yes or no).
- Complexity theory: How many resources needed to solve P?
- Easy problems:

Complexity-theoretic distinctions

- Computability theory: Is P computable? (yes or no).
- Complexity theory: How many resources needed to solve P?
- Easy problems:
  - solvable in under n^k steps (worst case).

Complexity-theoretic distinctions

- Computability theory: Is P computable? (yes or no).
- Complexity theory: How many resources needed to solve P?
- Easy problems:
  - solvable in under n^k steps (worst case).
- Hard problems:

Complexity-theoretic distinctions

- Computability theory: Is P computable? (yes or no).
- Complexity theory: How many resources needed to solve P?
- Easy problems:
  - solvable in under n^k steps (worst case).
- Hard problems:
  - Problems that are not easy.
    - E.g., $\approx k^n$ steps in the worst case.

Complexity-theoretic distinctions

- Computability theory: Is P computable? (yes or no).
- Complexity theory: How many resources needed to solve P?
- Easy problems:
  - solvable in under n^k steps (worst case).
- Hard problems:
  - Problems that are not easy.
    - E.g., $\approx k^n$ steps in the worst case.
- Finer-grained complexity classes
  - linear: $\approx n$ steps in the worst case
  - quasi-linear: $\approx n \log n$ steps in the worst case
  - etc.

`SelectionSort`:

- $n(n-1)/2$ comparisons
- I.e., "$O(n^2)$".

`SelectionSort`:

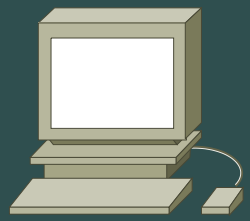
- $n(n-1)/2$ comparisons
- I.e., "$O(n^2)$".

`MergeSort`:

- $O(n \log n)$.

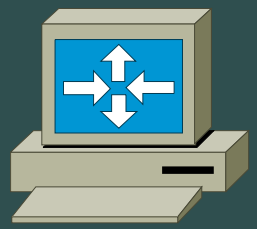
How is it that my computer sorts integers faster than yours?

How is it that my computer sorts integers faster than yours?



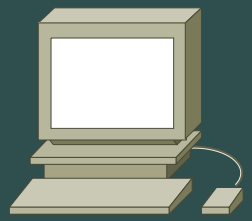
Selection-Sorter Mark I

Introducing the ...



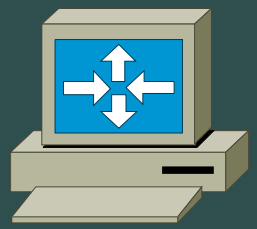
Merge-Sorter 9000 !

How is it that my computer sorts integers faster than yours?



Selection-Sorter Mark I

Introducing the ...

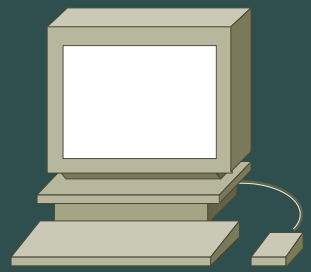


Merge-Sorter 9000 !

How-actually explanation

"My account has the somewhat counterintuitive consequence that one can move from a rather well-confirmed how-actually explanation ... at a high level of abstraction ... to a how-possibly model explanation as one tries to fill in some of the further details of that mechanism" (Bokulich, 2014, p. 335).

Coarse-grained description:



How-actually

"My account has the somewhat counterintuitive consequence that one can move from a rather well-confirmed how-actually explanation ... at a high level of abstraction ... to a how-possibly model explanation as one tries to fill in some of the further details of that mechanism" (Bokulich, 2014, p. 335).

Detailed description?

"My account has the somewhat counterintuitive consequence that one can move from a rather well-confirmed how-actually explanation ... at a high level of abstraction ... to a how-possibly model explanation as one tries to fill in some of the further details of that mechanism" (Bokulich, 2014, p. 335).

Detailed description?

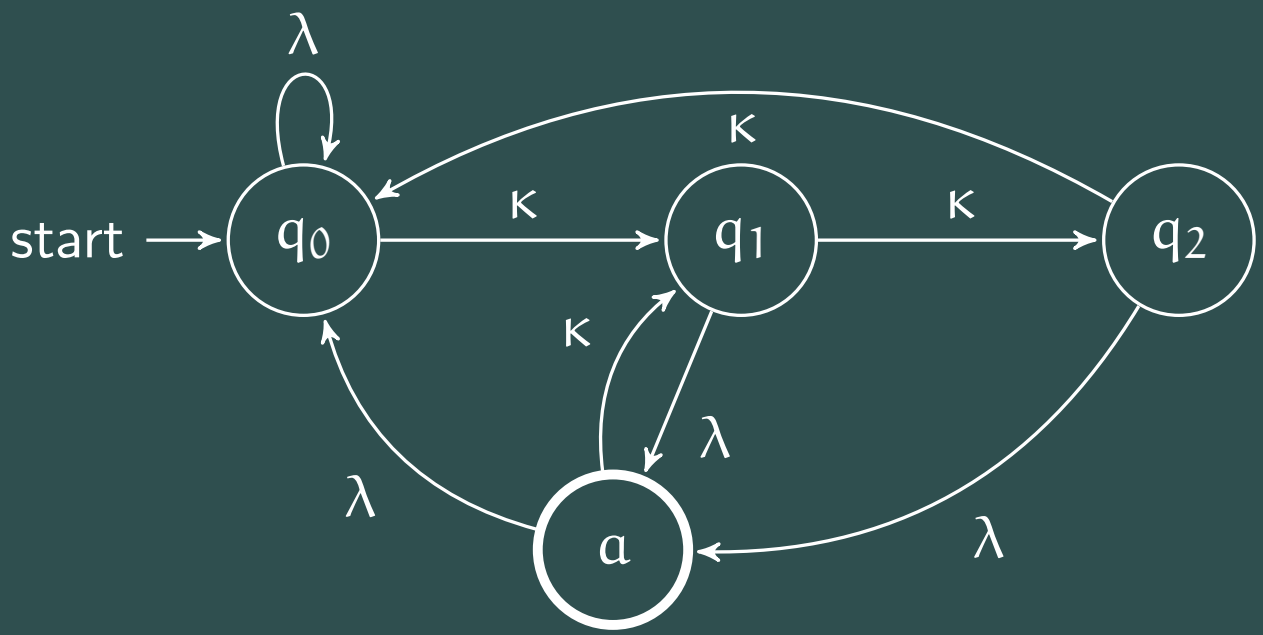
- Look at the code

```
void SelectionSort(int intsToSort[], int lengthOfList) {
   // Declare list indices:
   int i, j, indexOfLowestNum;
   // For each position in the list,
   for (i = 0; i < lengthOfList - 1; i++) {
      // provisionally assert that it points to the lowest number,
      indexOfLowestNum = i;
      // and then for each of the other list positions,
      for (j = lengthOfList - 1; j > i; j--) {
         // if the number pointed to by it is less than the number
         // pointed to by indexOfLowestNum,
         if (intsToSort[j] < intsToSort[indexOfLowestNum]) {
            // then make this the new provisional minimum index.
            indexOfLowestNum = j;
         }
      }
      // At the end of the ith iteration, put the number that is in the
      // indexOfLowestNum position into the ith position (and vice versa).
      Swap(&intsToSort[i], &intsToSort[indexOfLowestNum]);
   }
}
```

```
// on the first call, low = 0, high = n - 1
void Partition(int arr[], int low, int high) {
  // divide the list in two
  int mid;
  if (low < high){
    mid = (low + high) / 2;
    // recursively call partition function on
    // both halves of the list
    Partition(arr, low, mid);
    Partition(arr, mid + 1, high);
    // once the list is partitioned, call the
    // main merge sort procedure
    MergeSort(arr, low, mid, high);
  }
}

void MergeSort(int arr[], int low,
                    int mid,int high) {
  int i, m, k, l, temp[MAX];

  l = low;
  i = low;
  m = mid+1;

  while ((l <= mid) && (m <= high)){
    if (arr[l] <= arr[m]){
      temp[i] = arr[l];
      l++;
    }
    else{
      temp[i] = arr[m];
      m++;
    }
    i++;
  }
  if (l > mid){
    for(k = m; k <= high; k++){
      temp[i] = arr[k];
      i++;
    }
  }
  else {
    for (k = l; k <= mid; k++){
      temp[i] = arr[k];
      i++;
    }
  }
  for (k = low; k <= high; k++){
    arr[k] = temp[k];
  }
}
```

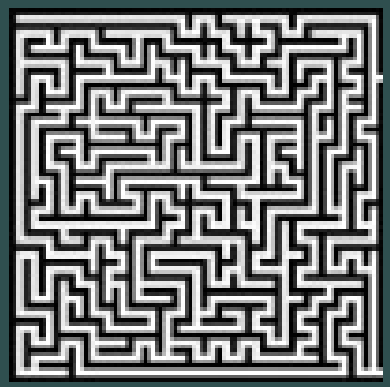
What does this code represent?

```
// on the first call, low = 0, high = n - 1
void Partition(int arr[], int low, int high) {
  // divide the list in two
  int mid;
  if (low < high){
    mid = (low + high) / 2;
    // recursively call partition function on
    // both halves of the list
    Partition(arr, low, mid);
    Partition(arr, mid + 1, high);
    // once the list is partitioned, call the
    // main merge sort procedure
    MergeSort(arr, low, mid, high);
  }
}

void MergeSort(int arr[], int low,
                    int mid,int high) {
  int i, m, k, l, temp[MAX];

  l = low;
  i = low;
  m = mid+1;

  while ((l <= mid) && (m <= high)){
    if (arr[l] <= arr[m]){
      temp[i] = arr[l];
      l++;
    }
    else{
      temp[i] = arr[m];
      m++;
    }
    i++;
  }
  if (l > mid){
    for(k = m; k <= high; k++){
      temp[i] = arr[k];
      i++;
    }
  }
  else {
    for (k = l; k <= mid; k++){
      temp[i] = arr[k];
      i++;
    }
  }
  for (k = low; k <= high; k++){
    arr[k] = temp[k];
  }
}
```

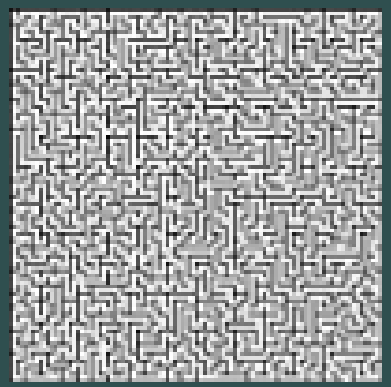

λ
κ
start
q_0
κ
q_1
κ
q_2
κ
λ
λ
a
λ

Space of possibilities

- The pathways available to `MergeSort` allow for quicker running times than the pathways available to `SelectionSort`.



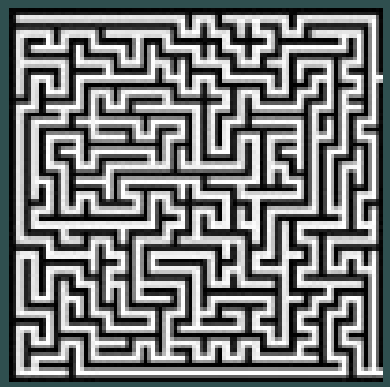
`MergeSort`



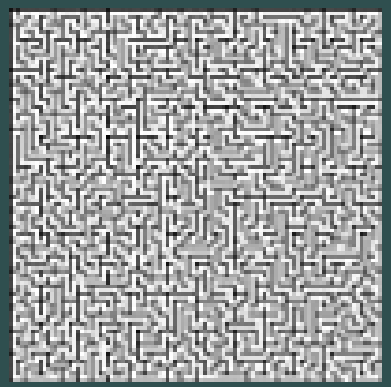
`SelectionSort`

Space of possibilities

- The pathways available to `MergeSort` allow for quicker running times than the pathways available to `SelectionSort`.



`MergeSort`



`SelectionSort`

- They explain how-possibly.

Do all appeals to algorithms count as how-possibly explanation?

Do all appeals to algorithms count as how-possibly explanation?

- Probably not
  - "Coarse-grained" appeals to algorithms
  - Algorithm as an explanation of a particular outcome

Do all appeals to algorithms count as how-possibly explanation?

- Probably not
  - "Coarse-grained" appeals to algorithms
  - Algorithm as an explanation of a particular outcome

Characteristic examples

- Comparisons between (not necessarily abstract) algorithmic processes
  - Why is $\mathbb{A}$ more ____ than $\mathbb{B}$?

Outline

1. Different senses of how-possibly explanation
2. Algorithmic how-possibly explanation
3. Mechanistic explanation, structural explanation
4. Relation between mechanistic, structural, and algorithmic how-possibly explanation

Mechanistic explanation

- Railton's DNP model

Mechanistic explanation

- Railton's DNP model
  - What is a mechanism?

Mechanistic explanation

- Railton's DNP model
  - What is a mechanism?

"The goal of understanding the world is a theoretical goal, and if the world is a machine—a vast arrangement of nomic connections—then our theory ought to give us some insight into the structure and workings of the mechanism, above and beyond the capability of predicting and controlling its outcomes" (Railton, 1978, p. 208).

Mechanistic explanation

- Railton's DNP model
  - What is a mechanism?
    - Describable in terms of lawlike statements / D-N style argument.

"The goal of understanding the world is a theoretical goal, and if the world is a machine—a vast arrangement of nomic connections—then our theory ought to give us some insight into the structure and workings of the mechanism, above and beyond the capability of predicting and controlling its outcomes" (Railton, 1978, p. 208).

Mechanistic explanation

- Railton's DNP model
  - What is a mechanism?
    - Describable in terms of lawlike statements / D-N style argument.
    - Deliberately vague

"Calling for an account of the mechanism leaves open the nature of that account, and as far as I can see, the model explanations offered in scientific texts are D-N when complete, D-N sketches when not" (Railton, 1978, p. 208).

Mechanistic explanation

- Salmon
  - Aim: to describe the “causal nexus”
    - Network of interacting causal mechanisms

Mechanistic explanation

- Salmon
  - Aim: to describe the “causal nexus”
    - Network of interacting causal mechanisms
  - What is a “causal mechanism”?

Mechanistic explanation

- Salmon
  - Aim: to describe the “causal nexus”
    - Network of interacting causal mechanisms
  - What is a “causal mechanism”?
    - Mark transmission (Salmon, 1984); invariant quantity (Salmon, 1994); conserved quantity (Salmon, 1997).

Complex systems approach ("New Mechanism"):

"A mechanism for a behavior is a complex system that produces that behavior by the interaction of a number of parts, where the interactions between parts can be characterised by direct, invariant, change-relating generalisations" (Glennan, 2002, p. S344).

Complex systems approach ("New Mechanism"):

"A mechanism for a behavior is a complex system that produces that behavior by the interaction of a number of parts, where the interactions between parts can be characterised by direct, invariant, change-relating generalisations" (Glennan, 2002, p. S344).

- Mechanical model (ibid p. S347)
  - description of a mechanism's behaviour
  - description of what accounts for this behaviour

Complex systems approach ("New Mechanism"):

"A mechanism for a behavior is a complex system that produces that behavior by the interaction of a number of parts, where the interactions between parts can be characterised by direct, invariant, change-relating generalisations" (Glennan, 2002, p. S344).

Mechanistic explanation

- Mechanical model (ibid p. S347)
    - description of a mechanism's behaviour
        - Explanandum
    - description of what accounts for this behaviour
        - Explanans

Complex systems approach ("New Mechanism"):

"A mechanism for a behavior is a complex system that produces that behavior by the interaction of a number of parts, where the interactions between parts can be characterised by direct, invariant, change-relating generalisations" (Glennan, 2002, p. S344).

Mechanistic explanation

- Mechanical model (ibid p. S347)
  - description of a mechanism's behaviour
    - Explanandum
  - description of what accounts for this behaviour
    - Explanans

Cf. Anderson (2014a,b), Piccinini (2007), Craver (2007), etc.

"New Mechanism"

- Anti-reductionist
- Metaphysically agnostic

"It is not explicitly anti-metaphysical but rather metaphysically agnostic. The anti-reductive character of Mechanism_1 allows us to make methodological recommendations about investigating the world ... without thereby committing ourselves to a single account of what that world is like" (Anderson, 2014a, p. 276).

Structural explanation (originally via Hughes 1989a)

Structural explanation (originally via Hughes 1989a)

- Non-causal: dynamics of systems are irrelevant

Structural explanation (originally via Hughes 1989a)

- Non-causal: dynamics of systems are irrelevant
- Microphysical details are irrelevant

Structural explanation (originally via Hughes 1989a)

- Non-causal: dynamics of systems are irrelevant
- Microphysical details are irrelevant
- Empirical phenomena explained by mathematical/formal features of the world on which they depend.
    - E.g. Minkowskian representation of spacetime as an explanation of relativistic effects

Structural explanation (originally via Hughes 1989a)

- Non-causal: dynamics of systems are irrelevant
- Microphysical details are irrelevant
- Empirical phenomena explained by mathematical/formal features of the world on which they depend.
    - E.g. Minkowskian representation of spacetime as an explanation of relativistic effects
- Anti-metaphysical
    - No presuppositions regarding underlying entities and dynamical processes

Mechanistic vs structural explanation

Mechanistic vs structural explanation

Hughes (1989b)

- SE, unlike ME, is anti-metaphysical
  - No presuppositions regarding the character of underlying entities and dynamical processes (and no account of them either)

Mechanistic vs structural explanation

Hughes (1989b)

- SE, unlike ME, is anti-metaphysical
  - No presuppositions regarding the character of underlying entities and dynamical processes (and no account of them either)

Felline (2015)

- SE, unlike ME, is non-mechanistic
  - No account of underlying entities and dynamical processes
  - SE not necessarily non-metaphysical?

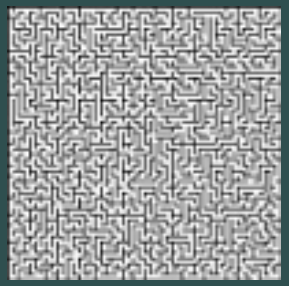
Outline

1. Different senses of how-possibly explanation
2. Algorithmic how-possibly explanation
3. Mechanistic explanation, structural explanation
4. Relation between mechanistic, structural, and algorithmic how-possibly explanation

Algorithmic How-Possibly Explanation

- Structural explanation?

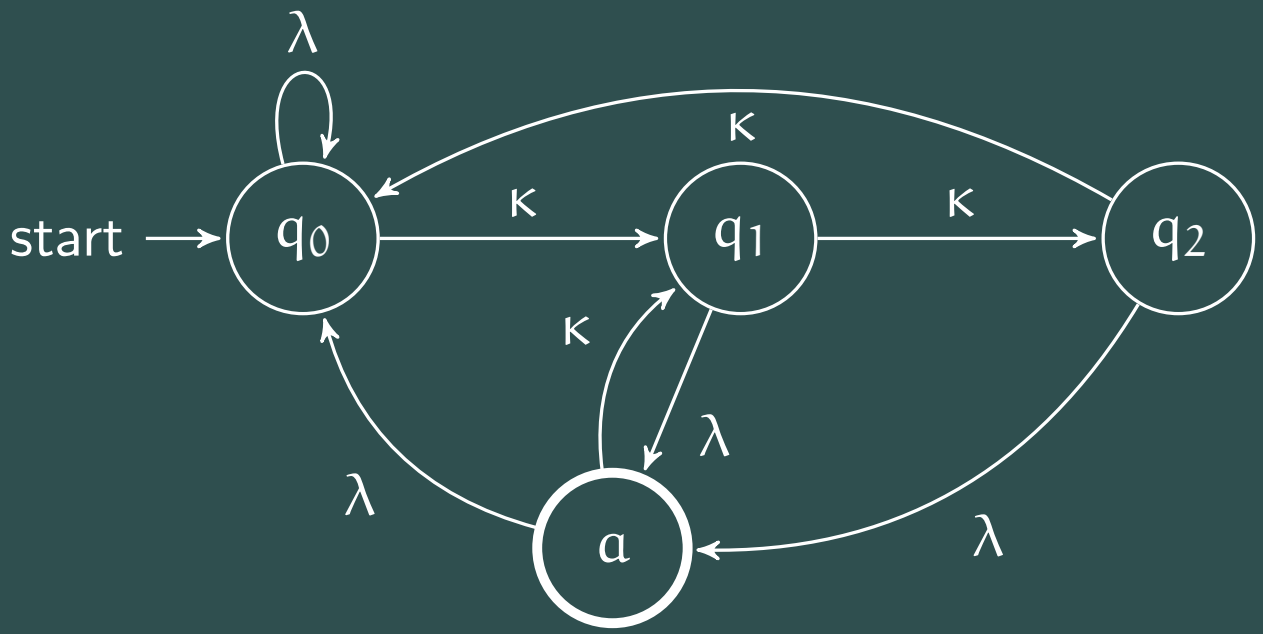
`MergeSort`



`SelectionSort`

Algorithmic How-Possibly Explanation

- Structural explanation?



Algorithmic How-Possibly Explanation

- Mechanistic explanation?

```
// on the first call, low = 0, high = n − 1
void Partition(int arr[], int low, int high) {
  // divide the list in two
  int mid;
  if (low < high){
    mid = (low + high) / 2;
    // recursively call partition function on
    // both halves of the list
    partition(arr, low, mid);
    partition(arr, mid + 1, high);
    // once the list is partitioned, call the
    // main merge sort procedure
    mergeSort(arr, low, mid, high);
  }
}

void MergeSort(int arr[], int low,
                    int mid,int high) {
  int i, m, k, l, temp[MAX];

  l = low;
  i = low;
  m = mid+1;

  while ((l <= mid) && (m <= high)){
    if (arr[l] <= arr[m]){
      temp[i] = arr[l];
      l++;
    }
    else{
      temp[i] = arr[m];
      m++;
    }
    i++;
  }
  if (l > mid){
    for(k = m; k <= high; k++){
      temp[i] = arr[k];
      i++;
    }
  }
  else {
    for (k = l; k <= mid; k++){
      temp[i] = arr[k];
      i++;
    }
  }
  for (k = low; k <= high; k++){
    arr[k] = temp[k];
  }
}
```

Algorithmic How-Possibly Explanation

- Not mechanistic explanation
  - No account of underlying entities
    - Even when process implemented by the algorithm is not abstract

Algorithmic How-Possibly Explanation

- Not mechanistic explanation
  - No account of underlying entities
    - Even when process implemented by the algorithm is not abstract
- Not structural explanation
  - Account of dynamics is essential to the explanation

Algorithmic How-Possibly Explanation

- Not mechanistic explanation
  - No account of underlying entities
    - Even when process implemented by the algorithm is not abstract
- Not structural explanation
  - Account of dynamics is essential to the explanation

Distinct type (appears to be):

- Shares features with both types of explanation
- But differs from each

Outline

1. Different senses of how-possibly explanation
2. Algorithmic how-possibly explanation
3. Mechanistic explanation, structural explanation
4. Relation between mechanistic, structural, and algorithmic how-possibly explanation

Works Cited

Anderson, H. (2014a). A field guide to mechanisms: Part I. Philosophy Compass, 9, 274–283.

Anderson, H. (2014b). A field guide to mechanisms: Part II. Philosophy Compass, 9, 284–293.

Bokulich, A. (2014). How the tiger bush got its stripes: 'how possibly' vs. 'how actually' model explanations. The Monist, 97, 321–328.

Craver, C. F. (2007). Explaining the Brain. Oxford: Oxford University Press.

Cuffaro, M. E. (2015). How-possibly explanations in quantum computer science. Philosophy of Science, 82, 737–748.

Dray, W. (1957). Laws and Explanation in History. Oxford: Oxford University Press.

Felline, L. (2015). Mechanisms meet structural explanation. Synthese. Advance access version available online.

Forber, P. (2010). Confirmation and explaining how possible. Studies in History and Philosophy of Biological and Biomedical Sciences, 41, 32–40.

Glennan, S. (2002). Rethinking mechanistic explanation. Philosophy of Science, 69, S342–S353.

Hempel, C. G. (1965). Aspects of Scientific Explanation And Other Essays in the Philosophy of Science. New York: The Free Press.

Hughes, R. I. G. (1989a). Bell's theorem, ideology, and structural explanation. In J. T. Cushing, & E. McMullin (Eds.) Philosophical Consequences of Quantum Theory, (pp. 195–207). Notre Dame: University of Notre Dame Press.

Works Cited

Hughes, R. I. G. (1989b). The Structure and Interpretation of Quantum Mechanics. Cambridge, MA.: Harvard University Press.

Persson, J. (2012). Three conceptions of explaining how possibly—and one reductive account. In H. W. de Regt, S. Hartmann, & S. Okasha (Eds.) EPSA Philosophy of Science: Amsterdam 2009, (pp. 275–286). Dordrecht: Springer.

Piccinini, G. (2007). Computing mechanisms. Philosophy of Science, 74, 501–526.

Railton, P. (1978). A deductive-nomological model of probabilistic explanation. Philosophy of Science, 45, 206–226.

Resnik, D. B. (1991). How-possibly explanations in biology. Acta Biotheoretica, 39, 141–149.

Salmon, W. C. (1984). Scientific Explanation and the Causal Structure of the World. Princeton: Princeton University Press.

Salmon, W. C. (1994). Causality without counterfactuals. Philosophy of Science, 61, 297–312.

Salmon, W. C. (1997). Causality and explanation: A reply to two critiques. Philosophy of Science, 64, 461–477.

Thanks!