Methodology Matters—IV

Constructing Algorithm Flowcharts for Clinical Performance Measurement

An algorithm is a rule of procedure or set of instructions for solving a problem or accomplishing an objective. Algorithms are built on condition-based (branching) logic, in which the condition encountered at a decision node, or point of branching, determines the next step of the pathway. Algorithms can be represented in words only or in a flowchart format.

Clinical algorithms act as guides for the provision of patient care for specific problems, and such algorithms have been published in the English-language medical literature since 1968 [1]. In the case of these algorithms, flowcharts have been shown to be considerably clearer than algorithms in words only for communicating the conditional statements that constitute the underlying logic of most clinical algorithms [2], and have become the recommended format for representing clinical algorithms clearly and succinctly.

GUIDELINE AND CRITERIA ALGORITHMS

Algorithms provide an effective means of representing clinical practice guidelines and medical review criteria. A clinical practice guideline can be represented as an algorithm written in words only, by expressing its descriptive text in succinct, sequential, declarative statements, as shown in Fig. 1.

When presented in flowchart form, the guideline algorithm becomes a pictorial description of stepwise recommendations for patient care (Fig. 2).

Similarly, when medical review criteria are derived from practice guidelines, algorithms may be used as a means of providing clear instructions for evaluating the conformance of
Follow-up of a Low Hematocrit Found by Examining Lab Reports

1. Within 21 days of Hct 534.0, repeat hematocrit to confirm anemia
2. Within 2 weeks of confirmation of anemia, work up anemia, order
   - MCV and reticulocyte, or
   - serum iron/TIBC, or
   - trial of iron therapy and a repeat hematocrit
3. If iron deficiency anemia is confirmed and the source of blood loss is not known, order
   three occult blood tests to search for GI bleeding within 60 days of initial low hematocrit

Note: GI = gastrointestinal, Hct = hematocrit, MCV = mean corpuscular volume, TIBC = total iron-binding capacity

FIGURE 1. Word version of a clinical practice guideline algorithm (hematocrit example).

---

FIGURE 2. Flowchart of the algorithm for a guideline for follow-up of low hematocrit.

---

Follow-up of a Low Hematocrit Found by Examining Lab Reports

Criterion 1: Confirm anemia
1. If the patient is male, female over 50, or female S/P hysterectomy, the Hct requiring follow-up is 538.0, otherwise an index Hct 5 34.0.
2. If there was a prior abnormal Hct, the criterion need not be met, the criterion status = "acceptable alternative."
3. If a follow-up Hct was done within 21 days after an abnormal Hct, the criterion status = "met."
4. If the medical record documents any reason why the criterion could or should not be met, the criterion status = "acceptable alternative."
5. If there was no criterion compliance, the criterion status = "not met."

NOTE. S/P = status postoperative; Hct = hematocrit

FIGURE 3. Word version of the algorithm for assessing conformance to review criteria (hematocrit example).
a sample of clinical care to the guideline recommendations. The criteria assessment algorithm directs the reviewer to determine whether each of the conformance criteria within the set is met and, if not met, what the status of each criterion is. The status list includes five possible values—"met", "not met", "acceptable alternative", "criterion not applicable", or "case not reviewable". The status the reviewer assigns to each criterion in working through the steps of the algorithm indicates whether the care being evaluated adhered to the practice guideline.

Criteria assessment algorithms can also be expressed in words only or as a flowchart. Fig. 3 is an example of the words only form of a criterion assessment algorithm which is used to determine conformance to step 1 of the guideline represented as an algorithm in Fig. 1.

Fig. 4 shows the same criterion assessment algorithm in flowchart form.

![Flowchart](https://academic.oup.com/intqhc/article-abstract/8/4/395/1794828/fig4){:width=600px}

**NOTE:** HCT = hematocrit, S/P = status post, PT = patient

**FIGURE 4.** Flowchart of the algorithm for assessing conformance to review criteria.
RATIONALE FOR FLOWCHART USE

Algorithm flowcharts enhance the usefulness of the algorithm in a number of ways:
- a flowchart provides a visual display of the branching logic of the algorithm,
- working through the logic of the branching at the decision nodes reveals inconsistencies or omissions that may be obscured in an algorithm written in words only,
- all the pathways in the algorithm logic can be traced to determine the appropriateness of the end points.

FLOWCHART ELEMENTS

Shapes

A standard flowchart diagram connects a series of different shapes. Each shape has a specific meaning, as shown in Fig. 5. Although currently there are no universally accepted rules for constructing flowcharts, methods and standards have been proposed [3–6]. When the shapes of the symbols are consistent within a document, it is easier to understand the content of the algorithm. The names of the shapes and their uses are as follows.

**Oval**
- a terminal shape, which begins or ends the algorithm.

**Diamond or Hexagon**
- the decision shape; it represents a contingency situation or a question regarding data in the patient record under review. At these decision points, pertinent facts guide the flow to paths representing alternative decisions or actions.

**Rectangle**
- the process shape; it represents an action or process and shows the value ("met", "not met", etc.) assigned to a given criterion based on the path taken.

**Circle**
- a connector; it indicates where an algorithm continues when it cannot be completed in the space allowed.

Additional shapes

Other shapes may take on additional meanings when necessary:
- A parallelogram can be used to represent a temporary digression to another algorithm from the same practice guideline,
- A rounded rectangle can be used to clarify the patient's state or condition when the sample is divided by extensive branching in the algorithm pathway.

Fig. 5 lists the common flowchart symbols and includes examples of these additional shapes.
Arrows

Arrows from decision nodes usually flow to the right and down. Traditionally, in an individual flowchart, YES leads in one direction and NO in the other. While the direction of YES and NO may differ among documents, it should always be consistent within a document. When the direction of the arrows is inconsistent, the algorithm logic is difficult to follow.

Numbering

Numbering of shapes on algorithm flowcharts, as shown in Fig. 4, permits the user to identify an action or decision point precisely. In addition, the number may serve as a footnote for reference to notations that define key terms and as a way to link care recommendations to published justifications. Numbers should be assigned to shapes sequentially, following horizontal paths first (see Fig. 4, shapes 1 and 4). Terminal shapes and connectors are usually unnumbered.

Choosing Flowcharting Computer Programs

There are several commercially available computer software packages that draw flowcharts. The criteria development process involves numerous drafts of the algorithm flowchart. Using an automated flowcharting program to do this work is considerably common.

Common Errors in Constructing Flowcharts

Listed below are common flowchart errors and their solutions.

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of consistency within the same flowchart in the direction of the arrows representing YES and NO</td>
<td>Reword the text in the decision nodes or rearrange subsequent decision and action nodes so that all the YES answers lead from the node in the same direction.</td>
</tr>
<tr>
<td>Using a single flowchart to describe more than one algorithm—for example, one algorithm for both drug treatment decisions and drug inventory control processes</td>
<td>Construct separate algorithms and indicate the entry point from one to the other with connector shapes.</td>
</tr>
<tr>
<td>Giving decision nodes more than two outward paths for multiple choices</td>
<td>Construct separate decision nodes linked together. Determine one option at a time in logical sequence. For example:</td>
</tr>
<tr>
<td>Giving decision nodes only one exit path, which assumes that there is no other option or no interest in another option</td>
<td>Is Hct &lt; 34.0? (Answer: YES/NO)</td>
</tr>
<tr>
<td>Using the same shape for different meanings or different shapes for the same meaning</td>
<td>Is Hct ≥ 34.0–38.0? (Answer: YES/NO)</td>
</tr>
<tr>
<td>Looping arrows back or exiting to more than one subsequent series of steps, making sequencing hard to follow</td>
<td>Is Hct ≥ 38.0? (Answer: YES/NO)</td>
</tr>
<tr>
<td>Ending a path with a process shape</td>
<td>Provide an alternate path, ending with the terminal shape for “stop.”</td>
</tr>
<tr>
<td>Writing questions or statements on the connecting arrows</td>
<td>Review all meanings and standardize the shapes used to represent each meaning.</td>
</tr>
<tr>
<td></td>
<td>Use connectors to show a loop or path to other steps and draw these steps separately.</td>
</tr>
<tr>
<td></td>
<td>Use a terminal shape to signal the end of each pathway.</td>
</tr>
<tr>
<td></td>
<td>Write all text within the appropriate shape.</td>
</tr>
</tbody>
</table>
more efficient than drawing flowcharts by hand or with a general computer graphics program.

A flowchart program should be selected for its ability to perform basic flowcharting functions, for its ease of editing existing drawings and for the amount of effort required to learn its basic and advanced features. For projects in which the flowchart will be circulated widely and visual impact is important, an additional consideration in software selection is the quality of the printed graphic.

Basic features of flowcharting software programs should permit the user to place a desired shape and write text inside it, connect the shapes with arrows, write titles and move text outside the shapes, construct multiple-page flowcharts, move shapes and text, edit text and print the flowcharts. Additional features can permit changing the size of shapes, customizing fonts and type size, creating custom shapes, shading shapes and lines, routing lines automatically, changing line appearance, and linking flowchart files with each other or with text files.

REFERENCES


Naomi J. Banks, M.B.A., M.Ed.
Center for Quality of Care Research and Education
Harvard School of Public Health
677 Huntington Avenue
Boston, MA 02115, USA