Addition Strategies: Rearranging to Make Bases (RMB)

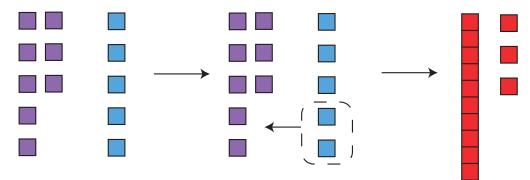
Compiled by: Theodore M. Savich

April 1, 2025

Transcript

Video from Carpenter et al. (1999). Strategy descriptions and examples adapted from Hackenberg (2025).

- Teacher: Lucy is eight fish. She buys five more fish. How many fish will Lucy have then?
- Sarah: 13.
- Teacher: How'd you get 13?
- Sarah: Well, because eight plus two is ten, but then two plus three is five. And she wants to buy five more fish. So you take care of two, and you need to add three more. And so I add three more, and you get 13.



Notation Representing Sarah's Solution:

$$8 + 5 = \square$$

$$8 + 2 = 10$$

$$2 + 3 = 5$$

$$8 + 5 = 10 + 3$$

$$8 + 5 = 13$$

Description of Strategy:

Objective: Rearranging to Make Bases (RMB) means shifting the extra ones from one addend over to the other so that one of the numbers becomes a complete multiple of the base (a whole "group" of that base). This rearrangement simplifies the addition process because there are established

patterns for adding an exact multiple of the base. In other words, when you add a full group of base units to a number, the ones digit stays the same while only the digit representing the base (like the tens place) increases.

Rearranging to Make Bases (RMB)

Description of Strategy

- **Objective:** Make one of the addends a whole number of bases by moving ones from the other addend.
- Example: 8+5
 - Move 2 ones from 5 to 8 to make 10.
 - Remaining ones in the second addend: 5-2=3.
 - Add the adjusted numbers: 10 + 3 = 13.

Automaton Type

Pushdown Automaton (PDA): Needed to handle digits and to remember the number of ones moved via the stack.

Formal Description of the Automaton

We define the PDA as the 7-tuple

$$M = (Q, \Sigma, \Gamma, \delta, q_{0/accept}, Z_0, F)$$

where

- $Q = \{q_{0/accept}, q_1, q_2, q_3, q_4, q_5\}$ is the finite set of states.
- $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +\}$ is the input alphabet (suitable for representing addends).
- $\Gamma = \{Z_0\} \cup \{x \mid x \in \mathbb{N}\}$ is the stack alphabet, where:
 - $-Z_0$ is the initial (bottom) stack symbol.
 - A symbol x represents the number of ones moved.
- $q_{0/accept}$ is the start state, which is also the accept state.
- Z_0 is the initial stack symbol.
- $F = \{q_{0/accept}\}$ is the set of accepting states.

The transition function

$$\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma^*)$$

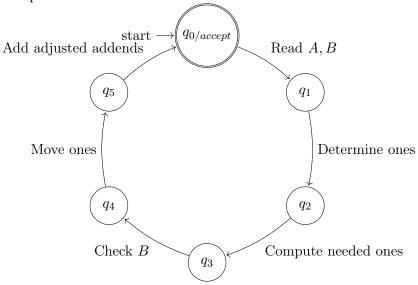
is defined by the following key transitions:

- 1. $\delta(q_{0/accept}, "A, B", Z_0) = \{(q_1, Z_0)\}$ (Read inputs A and B).
- 2. $\delta(q_1, \varepsilon, Z_0) = \{(q_2, Z_0)\}$ (Determine the ones digits of A and B).

- 3. $\delta(q_2, \varepsilon, Z_0) = \{(q_3, Z_0)\}$ (Compute the number of ones needed to make A a full base).
- 4. $\delta(q_3, \varepsilon, Z_0) = \{(q_4, k Z_0)\}$ (If B has at least k ones, push k onto the stack).
- 5. $\delta(q_4, \varepsilon, k) = \{(q_5, k)\}$ (Move k ones from B to A and adjust the addends).
- 6. $\delta(q_5, \varepsilon, k) = \{(q_{0/accept}, Z_0)\}$ (Add the adjusted numbers, output the result, and pop k from the stack).

Automaton Diagram for RMB

The following TikZ picture arranges the 6 states on a circle, with $q_{0/accept}$ serving as both the start and accept state.



HTML Implementation

```
<!DOCTYPE html>
   <html>
2
   <head>
       <title>Rearranging to Make Bases (RMB) Addition</title>
4
       <style>
           body { font-family: sans-serif; }
6
          #diagramRMBSVG { border: 1px solid #d3d3d3; } /* Style SVG like canvas */
          #outputContainer { margin-top: 20px; }
           .diagram-label { font-size: 14px; display: block; margin-bottom: 5px; } /*
               Improved label styling */
       </style>
   </head>
11
   <body>
13
       <h1>Addition Strategies: Rearranging to Make Bases (RMB)</h1>
14
15
       <div>
16
           <label for="addend1">Addend 1:</label>
17
           <input type="number" id="addend1" value="18">
18
       </div>
```

```
<div>
20
           <label for="addend2">Addend 2:</label>
           <input type="number" id="addend2" value="15">
24
       <button onclick="runRMBAutomaton()">Calculate and Visualize</button>
       <div id="outputContainer">
27
           <h2>Explanation:</h2>
28
           <div id="rmbOutput">
              <!-- Text output will be displayed here -->
           </div>
32
       </div>
       <h2>Diagram:</h2>
34
       <svg id="diagramRMBSVG" width="600" height="700"></svg> <!-- Increased height -->
36
       <!-- New button for viewing PDF documentation -->
37
       <button onclick="openPdfViewer()">Want to learn more about this strategy? Click here
38
           .</button>
       <script>
40
   document.addEventListener('DOMContentLoaded', function() {
41
       const rmbOutputElement = document.getElementById('rmbOutput');
42
       const rmbAddend1Input = document.getElementById('addend1');
43
       const rmbAddend2Input = document.getElementById('addend2');
       const diagramRMBSVG = document.getElementById('diagramRMBSVG');
45
46
       if (!rmbOutputElement | !diagramRMBSVG) {
47
           console.warn("Element_rmbOutput_or_diagramRMBSVG_not_found");
           return;
49
       }
       window.runRMBAutomaton = function() {
           try {
53
              const addend1 = parseInt(rmbAddend1Input.value);
54
              const addend2 = parseInt(rmbAddend2Input.value);
              if (isNaN(addend1) || isNaN(addend2)) {
57
                  rmbOutputElement.textContent = "Please_enter_valid_numbers_for_both_addends
58
                  return;
              }
61
              let output = '';
62
              output += '<h2>Rearranging to Make Bases (RMB)</h2><br>';
63
              output += '<strong>Problem:</strong> ${addend1} + ${addend2}<br>';
64
65
              const toMakeBase = (10 - (addend1 % 10)) % 10;
67
              // Strategy variables
68
              let newAddend1, newAddend2, result, transferAmount;
69
              let fromFirst = false; // Whether we're transferring from addend1 to addend2
70
71
```

```
// Case 1: When addend1 is already a multiple of 10
72
               if (toMakeBase === 0) {
73
                   // Instead of direct calculation, decompose addend2 into tens and ones
                   const a2_tens = Math.floor(addend2 / 10);
                   const a2_ones = addend2 % 10;
                   output += '${addend1} is already a multiple of 10.<br>';
                   output += 'Step 1: Break down ${addend2} into tens and ones<br>';
79
                   output += ' ${addend2} = ${a2_tens * 10} + ${a2_ones} <br>';
80
                   output += 'Step 2: Add the parts to ${addend1}<br>';
81
                   output += ' ${addend1} + ${a2_tens * 10} = ${addend1 + a2_tens * 10} <br/>
                   output += '\{addend1 + a2_{tens} * 10\} + \{a2_{ones}\} = \{addend1 + addend2\}<
83
                       br><br>';
                   output += 'Result: ${addend1} + ${addend2} = ${addend1 + addend2}';
84
85
                   newAddend1 = addend1;
86
                   newAddend2 = addend2;
87
                   transferAmount = 0;
88
                   result = addend1 + addend2;
89
90
                   rmbOutputElement.innerHTML = output;
91
                   drawRMBDiagram('diagramRMBSVG', addend1, addend2, transferAmount,
                       newAddend1, newAddend2, result, fromFirst);
93
                   return;
               }
94
               // Case 2: When addend2 is too small to provide needed units
96
               if (addend2 < toMakeBase) {</pre>
97
                   // Instead of direct calculation, transfer from addend1 to complete addend2
98
                        to a base
                   fromFirst = true;
99
                   const a1_ones = addend1 % 10;
100
                   const toCompleteAddend2 = 10 - addend2;
                   // We'll move units from addend1 to addend2
103
                   transferAmount = Math.min(a1_ones, toCompleteAddend2);
                   newAddend1 = addend1 - transferAmount;
                   newAddend2 = addend2 + transferAmount;
106
                   result = newAddend1 + newAddend2; // Will equal addend1 + addend2
107
108
                   output += '${addend2} is too small to provide the ${toMakeBase} units
109
                       needed for ${addend1}.<br>';
                   output += 'Step 1: Move ${transferAmount} from ${addend1} to ${addend2}<br
110
                      >';
                   output += ' ${addend1} - ${transferAmount} = ${newAddend1}<br>';
                   output += ' ${addend2} + ${transferAmount} = ${newAddend2}<br>';
112
113
                   // If we made a complete base in addend2
114
                   if (newAddend2 % 10 === 0) {
                      output += 'Step 2: Now ${newAddend2} is a complete base (multiple of
                          10) <br>';
                   } else {
117
                      output += 'Step 2: Even after moving, we can'tumakeuaucompleteubase,u
118
                          butuweurearrangeduforueasierumentaluaddition<br/>';
```

```
____}
119
120
          ___output_+=_'Step_3:_Add_the_rearranged_numbers<br/>';
121
          122
          124
          \verb| uuuuuuuuuuuuurmbOutputElement.innerHTML| = \verb| uoutput; |
125
          uuuuuuuuuuudrawRMBDiagram('diagramRMBSVG',uaddend1,uaddend2,utransferAmount,u
                    newAddend1, _newAddend2, _result, _fromFirst);
          uuuuuuuuuuuuureturn;
127
          128
129
130
          uuuuuuuuuuu//u0riginalucase:uStandarduRMBustrategy
          ____toMakeBase;
131
          \verb| uuuuuuuuunewAddend1| = \verb| uaddend1| + \verb| utransferAmount;
132
          \verb| uuuuuuuuunewAddend2| = \verb| uaddend2| - \verb| transferAmount;
133
          ____result_=_newAddend1_+_newAddend2;
134
135
          136
          \verb| unique | output | 
137
                    ⊔10)<br>';
          138
          ____output_+=_'Step_2:_Add_the_rearranged_numbers<br/>';
139
          uuuuuuuuuuoutputu+=u'${newAddend1}u+u${newAddend2}u=u${result}<br><br>';
140
          141
142
          \verb| uuuuuuuuurmb0utputElement.innerHTML| = \verb| uoutput;
143
          UUUUUUUUUUU//UDrawuRMBUDiagram
145
          ununununundrawRMBDiagram('diagramRMBSVG', addend1, addend2, transferAmount, newAddend1,
                    □newAddend2,□result,□fromFirst);
          uuuuuuuuu}ucatchu(error)u{
147
          المال mbOutputElement.textContent و 'Error: المالة (error message):
148
149
          ____}
          ____};
150
          uuuuufunctionudrawRMBDiagram(svgId,uaddend1,uaddend2,utransferAmount,unewAddend1,u
153
                    newAddend2, _result, _fromFirst_=_false)_{_{}}
          154
          uuuuuuuuifu(!svg)ureturn;
155
          uuuuuuuusvg.innerHTMLu=u'';u//uClearuSVG
156
          158
          uuuuuuuuconstublockUnitSizeu=u15;u//uSizeuofuindividualuunitublock
160
         \verb| uuuuuuuconst_{\sqcup} tenBlockWidth_{\sqcup} = \verb| ublockUnitSize ; \verb| u// \_Width_{\sqcup} of_{\sqcup} 10 - block_{\sqcup} rectangle | uuuuuuuconst_{\sqcup} tenBlockWidth_{\sqcup} = \verb| ublockUnitSize ; \verb| u// \_Width_{\sqcup} of_{\sqcup} 10 - block_{\sqcup} rectangle | uuuuuuuconst_{\sqcup} tenBlockWidth_{\sqcup} = \verb| ublockUnitSize ; \verb| u// \_Width_{\sqcup} of_{\sqcup} 10 - block_{\sqcup} rectangle | uuuuuuuconst_{\sqcup} tenBlockWidth_{\sqcup} = \verb| ublockWidth_{\sqcup} of_{\sqcup} 10 - block_{\sqcup} rectangle | uuuuuuuuconst_{\sqcup} tenBlockWidth_{\sqcup} = \verb| ublockWidth_{\sqcup} of_{\sqcup} 10 - block_{\sqcup} rectangle | uuuuuuconst_{\sqcup} tenBlockWidth_{\sqcup} = \verb| ublockWidth_{\sqcup} of_{\sqcup} 10 - block_{\sqcup} rectangle | uuuuuconst_{\sqcup} tenBlockWidth_{\sqcup} of_{\sqcup} 10 - block_{\sqcup} rectangle | uuuuconst_{\sqcup} tenBlock_{\sqcup} rectangle | uuuuconst_{\sqcup} tenBlock_{\sqcup} rectangle | uuuuconst_{\sqcup} tenBlock_{\sqcup} rectangle | uuuuconst_{\sqcup} tenBlock_{\sqcup} rectangle | uuuuconst_{\sqcup} rectangle 
          uuuuuuuconstutenBlockHeightu=ublockUnitSizeu*u10;u//uHeightuofu10-blockurectangle
162
          unununconstusectionSpacingYu=u120;u//uVerticaluspacingubetweenusections
164
         LILIULIULICONStustartXu=u50;
         166
         uuuuuuuconstucolorAddend1u=u'purple';
167
168 | ULUULUULUConstucolorAddend2u=u'blue';
```

```
uuuuuuuuconstucolorBaseu=u'red';
169
              uuuuuuuuconstucolorTransferu=u'orange';
170
171
              LUUUUUUU//u---UriginaluAddendsu(HorizontaluLayout)u---
172
              ULULULULUCreateText(svg, ustartX, ucurrentY, u'Original uAddends: u${addend1}u+u${addend2}');u
                            // Label
              uuuuuuuucurrentYu+=u30;u//uSpaceuafterulabel
174
175
              UUUUUUUU//UDrawuAddendu1u(purple)uonuleft
176
              uuuuuuuuletuaddend1Xu=ustartX;
177
              ____const_a1_tens_=_Math.floor(addend1_/_10);
178
              uuuuuuuuconstua1_onesu=uaddend1u%u10;
179
              _{\cup\cup\cup\cup\cup\cup\cup\cup} for _{\cup} (let _{\cup} i _{\cup} 0; _{\cup} i _{\cup} < _{\cup} a1_tens; _{\cup} i++) _{\cup} {
              المال drawTenBlock(svg, addend1X, currentY, tenBlockWidth, tenBlockHeight, المال المالية الما
181
                            colorAddend1);
              \verb| uuuuuuuuuuaddend1X_u+=_utenBlockWidth_u+_ublockSpacing; \\
182
              ____}
183
              ___let_a1_onesX_=_addend1X;
184
              uuuuuuuuletumovedFromFirstBlockPositionsu=u[];
185
              = 0; i < a1_ones; i + +) 
              ____const_jisTransferBlock_=_fromFirst_&&_ii_>=_a1_ones_-_transferAmount;
187
              \verb||_{\sqcup \sqcup \sqcup} const_{\sqcup}blockColor_{\sqcup} = \verb||_{\sqcup}isTransferBlock_{\sqcup}?_{\sqcup}colorTransfer_{\sqcup}:_{\sqcup}colorAddend1;
188
              ____currentY_+_i*(blockUnitSize_+_blockSpacing);
189
              ununununundrawBlock(svg,ual_onesX,ublockY,ublockUnitSize,ublockUnitSize,ublockColor);
190
191
              \verb|uuuuuuuuu| if \verb|u| (isTransferBlock)| | \{
192
              uuuuuuuuuuumovedFromFirstBlockPositions.push({
193
              uuuuuuuuuuuux:ua1_onesXu+ublockUnitSize/2,
              \verb"uuuuuuuuuuuuuuy:" \verb"blockY" + \verb"blockUnitSize/2"
195
              uuuuuuuuuuu);
196
              _____}
197
198
             ____}
              199
                            _+_(a1_ones_>_0_?_blockUnitSize_:_0);
200
              \verb| uuuuuuuu|/| \verb| DrawuAddendu2| (blue) | touthe| right| of \verb| Addendu1| 
201
              202
                            addend_{\sqcup}groups
              uuuuuuuuconstua2_tensu=uMath.floor(addend2u/u10);
203
              unununuconstua2_ones_=uaddend2u%u10;
204
              _{\cup\cup\cup\cup\cup\cup\cup\cup} for _{\cup} (let _{\cup} i _{\cup} 0; _{\cup} i _{\cup} < _{\cup} a 2_tens; _{\cup} i ++) _{\cup} {
205
              المال drawTenBlock(svg, addend2X, currentY, tenBlockWidth, tenBlockHeight, المال المالية الما
206
                            colorAddend2);
              uuuuuuuuuuaddend2Xu+=utenBlockWidthu+ublockSpacing;
207
              \verb|uuuuuuu| constuaddend20nesXu=uaddend2X;
209
              uuuuuuuletumovedFromSecondBlockPositionsu=u[];
              uuuuuuuuforu(let_ui_u=u_0;ui_u<u_a_2_ones;ui++)u{
211
              _____:fromFirst_&&_iu_<_transferAmount;
212
              \verb| uuuuuuuuuuconst_{\sqcup}blockColor_{\sqcup} = \verb| uisTransferBlock_{\sqcup}?_{\sqcup}colorTransfer_{\sqcup}: \verb| ucolorAddend2|; 
213
              UNDUMPRIOR TO THE PROPERTY OF 
214
              ununununundrawBlock(svg,uaddend20nesX,ublockY,ublockUnitSize,ublockUnitSize,ublockColor
215
216
```

```
217
                         \verb| uuuuuuuuuuuuuumovedFromSecondBlockPositions.push(\{a,b,b\}) | a to be a substitute of the conditions of the condition
218
                         \verb"uuuuuuuuuuux:uaddend20nesX"_+ \verb"blockUnitSize/2",
219
                        \verb"uuuuuuuuuuuuuuy:"blockY" + \verb"blockUnitSize/2"
220
                        uuuuuuuuuuu);
222
                         ____}
                         ____}
                         \verb| uuuuuuuucurrentY_{\sqcup} + = \verb| utenBlockHeight_{\sqcup} + \verb| usectionSpacingY;_{\sqcup} / / \verb| uMove_{\sqcup} down_{\sqcup} for_{\sqcup} the_{\sqcup} rearranged_{\sqcup} rear
224
                                                  addends_{\sqcup}section
225
                         UUUUUUUU//U---∪RearrangedUAddendsU---
                          227
                                                  newAddend2}'); □//□Label
                          uuuuuuuucurrentYu+=u30;u//uSpaceuafterulabel
228
229
                         LULULULU // Draw Rearranged Addend 1 Blocks
230
                         ____let_currentX_newAddend1_=_startX;
231
                          232
233
                         ____const_newAddend1_ones_=_newAddend1_%_10;
234
                         UUUUUUUU//UFirstudrawutens
235
                         ___let_tensPositions_=_[];
236
                         uuuuuuuuforu(letuiu=u0;uiu<unewAddend1_tens;ui++)u{
237
                         ____const_useColorBase_=_!fromFirst_&&_newAddend1_tens_>_a1_tens_&&_i_===_
238
                                                  newAddend1_tens___1;
                         \verb| uuuuuuuuuconst_ublockColor_u=_uuseColorBase_u?_ucolorBase_u: \verb| ucolorAddend1|;
239
                          الماليات drawTenBlock(svg, الماليات ال
240
                                                  tenBlockHeight, blockColor);
241
                         UUUUUUUUUUIfu(useColorBase)u{
242
                         uuuuuuuuuutensPositions.push({
243
                        \verb| uuuuuuuuuuuux: ucurrentX_newAddend1_u+utenBlockWidth/2, \\
244
                         \verb"uuuuuuuuuuuuuy: \verb"ucurrentY" + \verb"utenBlockHeight/2"
245
                        246
                         ____}
247
248
                         uuuuuuuuucurrentX_newAddend1u+=utenBlockWidthu+ublockSpacing;
249
                         250
251
                         ⊔⊔⊔⊔⊔⊔⊔//⊔Then⊔draw⊔ones
252
                         253
                         \verb| uuuuuuuuuudrawBlock(svg, ucurrentX_newAddend1, ucurrentY_u + ui*(blockUnitSize_u + 
254
                                                 blockSpacing),
                          uuuuuuuuuuuuuuuublockUnitSize,ublockUnitSize,ucolorAddend1);
255
                         ____}
257
                         UUUUUUUUU//UDrawuRearrangeduAddendu2uBlocks
                         ____Math.floor(newAddend2_/10);
259
                         \verb|u|| = \verb|u|
261
                                                  blockUnitSize_:_0); _//_Spacing_after_newAddend1
262
                        ⊔⊔⊔⊔⊔⊔⊔//⊔Draw⊔tens
263
```

```
ىرىن constuuseColorBaseر=ufromFirstu&&unewAddend2_tensu>ua2_tensu&&uiu===
265
                    newAddend2_tens_-1;
           uuuuuuuuuuuconstublockColoru=uuseColorBaseu?ucolorBaseu:ucolorAddend2;
266
           _{	ext{u}}
267
                    tenBlockHeight, blockColor);
268
          UUUUUUUUUUUIfu(useColorBase)u{
          uuuuuuuuuutensPositions.push({
270
          uuuuuuuuuuuux:ucurrentX_newAddend2u+utenBlockWidth/2,
271
          \verb"uuuuuuuuuuuuuuy: \verb"ucurrentY" + \verb"utenBlockHeight/2"
272
          uuuuuuuuuuu);
274
          ____}
           uuuuuuuuuuucurrentX_newAddend2u+=utenBlockWidthu+ublockSpacing;
276
277
          ____}
278
          UUUUUUUU//UDrawuones
279
          uuuuuuuuletuonesPositionsu=u[];
280
          uuuuuuuuforu(letuiu=u0;uiu<unewAddend2_ones;ui++)u{
281
          \verb|uuuuuuuuuuconst_uisTransferredBlock_u=ufromFirst_u\&\&ui_u>=ua2\_ones;
282
          283
           ____const_blockY_=_currentY_+_i*(blockUnitSize_+_blockSpacing);
284
          ununununundrawBlock(svg,ucurrentX_newAddend2,ublockY,ublockUnitSize,ublockUnitSize,u
285
                    blockColor);
286
          \verb| uuuuuuuuuuifu| (isTransferredBlock)_{\sqcup} \{
          uuuuuuuuuuonesPositions.push({
288
          uuuuuuuuuuuux:ucurrentX_newAddend2u+ublockUnitSize/2,
          uuuuuuuuuuuuy:ublockYu+ublockUnitSize/2
290
          uuuuuuuuuuu);
291
          _____}
292
293
          ____}
294
          LULULULU // L--- Draw Arrows Based On Strategy ---
295
          UUUUUUUUifu(transferAmountu>u0)u{
296
          297
           _______//_Case_2: Draw_arrows_from_addend1_to_addend2
298
          \verb| uuuuuuuuuuuuuufor| (let_{U}i_{U}=_{U}0;_{U}i_{U}<_{U}Math.min(movedFromFirstBlockPositions.length,_{U}i_{U}) |
                    onesPositions.length); i++) [
          \verb"uuuuuuuuuuuuuuuconst" start" = \verb"umovedFromFirstBlockPositions" [i];
300
          \verb"uuuuuuuuuuuuuuuuuuconst" e nd \verb"u=u" ones Positions" [i];
301
          \verb"uuuuuuuuuuuuuuuconst" controlX_{\sqcup} = \verb"u(start.x_{\sqcup} + \verb"uend.x")_{\sqcup} / {}_{\sqcup} 2;
302
          \verb| uuuuuuuuuuuuuuuuuuuuuuconstucontrolY_0 = \verb| uMath.min(start.y, uend.y)_0 - \verb| u40; u//uControl_point_0 = u40; u//uControl_po
                    above_both
          undergraphy of the control of the co
                    controlY):
          ____}
306
          uuuuuuuuuuuu//uIfuweuformeduaufulluten,udrawuarrowutoutensublock
          308
          ____const_start_=_movedFromFirstBlockPositions[0];
          \verb"uuuuuuuuuuuuuuuuconst" e nd \verb"u=utensPositions" [0];
310
          \verb| uuuuuuuuuuucreateText(svg, uend.x_u+u15, uend.y_u-u20, u'Formed_ua_ubase_u(10)'); \\
311
```

```
_{\cup\cup\cup\cup\cup\cup\cup\cup\cup\cup}\}_{\cup}else_{\cup}\{
313
          314
           \texttt{ULLICATION} \texttt{for}_{U}(\texttt{let}_{U} \texttt{i}_{U} = \texttt{U} \texttt{0}; \texttt{U} \texttt{i}_{U} < \texttt{U} \texttt{Math.min}(\texttt{movedFromSecondBlockPositions.length}, \texttt{ULLICATION}) 
315
                   tensPositions.length); i++) [{
          \verb| uuuuuuuuuuuuuuuuuuuconst_ustart_u = \verb| umovedFromSecondBlockPositions[i]; \\
          \verb"uuuuuuuuuuuuuuuuuconst" end \verb"u=utensPositions" [i];
317
          \verb| uuuuuuuuuuuuuuuuuuconst_ucontrolX_u = \verb| u(start.x_u + \verb| uend.x)_u/_u2;
          \verb| under u
319
                   above_{\sqcup}both
          under the control of 
320
                   controlY);
321
          ____}
          ____createText(svg,_tensPositions[0]?.x_+_15__||_startX_+_100,_tensPositions
                    [0]?.y_{\cup}-_{\cup}20_{\cup}||_{\cup}currentY_{\cup}-_{\cup}20,
          uuuuuuuuuuuuuuu'${transferAmount}umovedutouformubaseu(10)');
323
          ____}
324
         _{\cup\cup\cup\cup\cup\cup\cup\cup}_uelse_if_u(addend1__{\cup}_10_===_00)_u{
325
          UUUUUUUUUUUUU/UCase_1:uAlreadyuaumultipleuofu10,ushowutheudecomposition
326
327
          uuuuuuuuuconstua2_tensu=uMath.floor(addend2u/u10);
          uuuuuuuuuuifu(a2_tens_u>_00)_{u}
          ____s{addend2}_=_${
329
                   a2\_tens_{\square}*_{\square}10\}_{\square}+_{\square} {addend2_{\square}%_{\square}10}');
330
          ____}
331
          ____}
332
          ___Helper_SVG_drawing_functions_---
333
          ____function_drawBlock(svg,_x,_y,_width,_height,_fill)_{
334
          ____const_rect_=_document.createElementNS("http://www.w3.org/2000/svg",_'rect');
          uuuuuuuuurect.setAttribute('x', ux);
336
          uuuuuuuuurect.setAttribute('y',uy);
          uuuuuuuurect.setAttribute('width',uwidth);
338
         uuuuuuuuurect.setAttribute('height',uheight);
339
         uuuuuuuuurect.setAttribute('fill',ufill);
340
341
          "'black');
          uuuuuuuurect.setAttribute('stroke-width',u'1');
          343
          344
345
         ____function_drawTenBlock(svg,_x,_y,_width,_height,_fill)_{ [}
346
         underdeduction constagroup = document.createElementNS("http://www.w3.org/2000/svg",u'g');u//
347
                   _Group_for_10-block
          348
                   ⊔'rect');
          uuuuuuuuuuubackgroundRect.setAttribute('x', ux);
349
          uuuuuuuuubackgroundRect.setAttribute('y', uy);
          ____backgroundRect.setAttribute('width',_width);
351
         ____backgroundRect.setAttribute('height',_height);
          ____backgroundRect.setAttribute('fill',_fill);
353
          uuuuuuuubackgroundRect.setAttribute('stroke',u'black');
          uuuuuuuuubackgroundRect.setAttribute('stroke-width', u'1');
355
          ____group.appendChild(backgroundRect);
356
357
358
         _____/_Draw_10_unit_blocks_inside_-vertical_column
```

```
_{	ext{LULLULULULULULULULUC}}const_{	ext{L}}unit	ext{Block}_{	ext{L}}=_{	ext{L}}document.create	ext{ElementNS}("http://www.w3.org/2000/svg",_{	ext{L}}
360
             'rect');
      361
      ununununununununununun unitBlock.setAttribute('y',uyu+uiu*ublockUnitSize);u//uStackeduvertically
362
      unitBlock.setAttribute('width', blockUnitSize);
      uuuuuuuuuuuuuuuuuuunitBlock.setAttribute('height',ublockUnitSize);
364
      unitalick.setAttribute('fill',ufill);u//uSameufilluasuouterurect
      unitBlock.setAttribute('stroke', 'lightgrey'); // Lighter border for المارة ال
366
             units
      367
      uuuuuuuuuuugroup.appendChild(unitBlock);
369
      ____}
      {ىںںںںںںں}
371
372
      ____function_drawGroupRect(svg, _x, _y, _width, _height)__{{□}}
373
      LILILILICONSt_rect_=_document.createElementNS("http://www.w3.org/2000/svg",_'rect');
374
      uuuuuuuuuurect.setAttribute('x',ux);
375
      uuuuuuuuurect.setAttribute('y',uy);
376
      uuuuuuuuurect.setAttribute('width',uwidth);
      uuuuuuuuurect.setAttribute('height',uheight);
378
      ____rect.setAttribute('fill',_''none');_//_No_fill_for_group_rect
379
      uuuuuuuuurect.setAttribute('stroke',u'black');
380
      ununununurect.setAttribute('stroke-dasharray', '5 5'); // Dashed border for grouping
381
      uuuuuuuuurect.setAttribute('stroke-width',u'1');
382
      ערטיייטייטייט svg.appendChild(rect);
383
      384
      386
      ____const_text_=_document.createElementNS("http://www.w3.org/2000/svg",u'text');
      uuuuuuuuutext.setAttribute('x',ux);
388
      uuuuuuuuutext.setAttribute('y',uy);
389
      uuuuuuuuutext.setAttribute('class',u'diagram-label');
390
      ununununtext.setAttribute('text-anchor', 'start');
391
      uuuuuuuuutext.setAttribute('font-size',u'14px');
392
      uuuuuuuuutext.textContentu=utextContent;
393
      ערטריים svg.appendChild(text);
394
      395
396
      LILLILLILLI function create Curved Arrow (svg, Lx1, Ly1, Lx2, Ly2, Lcx, Lcy) (
397
      ULULULULULULUCONStupathu=udocument.createElementNS("http://www.w3.org/2000/svg",u'path');
398
      _____path.setAttribute('d',_''M_${x1}_${y1}_Q_${cx}_${cy}_${x2}_${y2}');
399
      uuuuuuuuupath.setAttribute('fill',u'none');
      uuuuuuuuuupath.setAttribute('stroke',u'black');
401
      uuuuuuuuupath.setAttribute('stroke-width',u'2');
      ערטייייט svg.appendChild(path);
403
404
      uuuuuuuuuu//uArrowhead
405
      ____const_arrowHead_=_document.createElementNS("http://www.w3.org/2000/svg",_''
            path');
      UUUUUUUUUUConstuarrowSizeu=u5;
      unununununuarrowHead.setAttribute('d',u'Mu${x2}u${y2}uLu${x2u-uarrowSize}u${y2u-u
408
             arrowSize}_L_${x2_+\_arrowSize}_\${y2_-\_arrowSize}\_Z');
      ____arrowHead.setAttribute('fill',_'black');
```

```
410
411
    ____}
    {ىںں }
412
413
    \square\square\square\square//\squareNew\squarefunction\squarefor\squareopening\squarePDF\squaredocumentation
414
    \square\square\square\squarefunction\squareopenPdfViewer()\square{
415
    ____window.open('../SAR_ADD_RMB.pdf',_'_blank');
417
    ____}
418
419
    });
420
    uuuu</script>
421
    </body>
423
    </html>
```

References

Carpenter, T. P., Fennema, E., Franke, M. L., Levi, L., & Empson, S. B. (1999). *Children's mathematics: Cognitively guided instruction* [Includes supplementary material: Children's mathematics: Cognitively guided instruction – videotape logs]. Heinemann; The National Council of Teachers of Mathematics, Inc.

Hackenberg, A. (2025). Course notes [Unpublished course notes].