Conditional instruction if/then

- Can Karel do something only in a special case?

  ```
  if <condition> then
    <instruction>;
  ```

- Example:
  ```
  if front-is-clear
    move;
  ```

- Note indentation!
Semantics of if/then instruction

if <condition> then
  <instruction-1> ;
  <next-instruction> ;

- Semantics of execution
  - If condition is true - instruction-1; after that - next-instruction
  - If condition is false - next-instruction

Flowchart of if/then
if/then instruction with a block

if <condition> then begin
  <instruction-1>;
  <instruction-2>;
  ...
  <instruction-k>;
end;
<next-instruction>;

Semantics of execution:
- If condition is true - instruction-1 ... instruction-k after that - next-instruction
- If condition is false - next-instruction

Karel’s conditions

- Walls
  - front-is-clear, left-is-clear, right-is-clear
  - front-is-blocked, left-is-blocked, right-is-blocked

- Direction
  - facing-north, facing-south, facing-east, facing-west
  - not-facing-north, not-facing-south, not-facing-east, not-facing-west

- Beepers
  - next-to-a-beeper, any-beepers-in-beeper-bag
  - not-next-to-a-beeper, no-beepers-in-beeper-bag
Why? Case 1: Cleaner Stairs

- Move Karel up the stairs picking beepers - but now a stair may not have a beeper!

Start:

Target:

Solution 1: Cleaner Stairs

beginning-of-program
define-new-instruction
  turnright as begin
  turnleft;
  turnleft;
  turnleft;
end;
define-new-instruction
climb-stair as begin
  turnleft;
  move;
  turnright;
  move;
end;
define-new-instruction pickbeeper-if-present as
  begin
    if next-to-a-beeper then
      pickbeeper
    end;
end;
beginning-of-execution
  climb-stair;
  pickbeeper-if-present;
  climb-stair;
  pickbeeper-if-present;
  climb-stair;
  pickbeeper-if-present;
  turnoff;
end-of-execution
end-of-execution
end-of-program
Case 2: The Bad Year Harvest

`define-new-instruction`
`pickbeeper-if-present as`
`begin`
`if next-to-a-beeper then`
`pickbeeper;`
`end;`

`define-new-instruction`
`harvest-1-row as`
`begin`
`pickbeeper-if-present;`
`move;`
`pickbeeper-if-present;`
`move;`
`pickbeeper-if-present;`
`move;`
`pickbeeper-if-present;`
`move;`
`pickbeeper-if-present;`
`end;`

Conditional instruction if–else

- Can Karel do different things in different situations?

```plaintext
if <condition> then
  <instruction-1>
else <instruction-2>;
```

- Example:

```plaintext
if front-is-clear then
  move;
else
  turnright;
```
Semantics of *if-else* instruction

```plaintext
if <condition> then
  <instruction-1>
else
  <instruction-2>;
<next-instruction>;
```

- Semantics of execution
  - If condition is true - *instruction-1*; after that - *next-instruction*
  - If condition is false - *instruction-2*; after that - *next-instruction*

Flowchart of *if-else*

```
true
       ▲
        ◀
condition
         ◀
      false
```

- Instruction-1
- Instruction-2
- next-instruction
Case 3: Hurdle Jumping Race

- Move Karel through a row of “hurdles”
- Each pair of Avenues may or may not have a hurdle between them

Solution 3: Hurdle Jumping Race

Main program:
beginning-of-execution
race-stride;
race-stride;
race-stride;
race-stride;
race-stride;
race-stride;
race-stride;
race-stride;

Main subtask:
define-new-instruction race-stride as begin
if front-is-clear then
  move
else
  jump-hurdle
end;

turnoff;
end-of-execution
Solution 3: Hurdle Jumping Race

Decomposing \textit{jump-hurdle}:

\begin{verbatim}
define-new-instruction jump-hurdle as begin
  jump-up;
  move;
  jump-down;
end;
\end{verbatim}

\begin{verbatim}
define-new-instruction jump-up as begin
  turnleft;
  move;
  turnright;
end;
\end{verbatim}

\begin{verbatim}
define-new-instruction jump-down as begin
  turnright;
  move;
  turnleft;
end;
\end{verbatim}

Stepwise refinement tree for Hurdle

```
hurdle problem
  ↓
race-stride
  ↓
jump-hurdle
  ↓
jump-up
  ↓
jump-down
  ↓
moves
```

Flowchart of else-if

Example of else-if

define-new-instruction inverse as begin
  if next-to-a-beeper then
    pickbeeper
  else if any-beepers-in-beeper-bag then
    putbeeper;
  else
    turnover;
end

What will happen if in bad year harvest we replace pick-beeper-if-present into inverse?
Checking Several Conditions

- How we can check that Karel is in a dead end of a maze?

```plaintext
if front-is-blocked then
  if left-is-blocked then
    if right-is-blocked then begin
      turnleft;
      turnleft;
      move;
    end;
```

Before next lecture:

- Do reading assignment
  - Pattis: Chapter 4
  - Tutorial: lessons 6, 7, 9
- Run Classroom Examples
- Check yourself by answering 3 questions (#1-#5) and doing 2 exercises (#7-#10) from Section 4.10.