Conditionals

; maybe-wanted : image -> image

WANTED

(maybe-wanted) →

(maybe-wanted) →
Conditionals in Algebra

General format of conditionals in algebra:

\[
\begin{cases}
\text{answer} & \text{question} \\
\ldots \\
\text{answer} & \text{question}
\end{cases}
\]

Example:

\[
\text{abs}(x) = \begin{cases}
x & \text{if } x > 0 \\
-x & \text{otherwise}
\end{cases}
\]

\[
\text{abs}(10) = 10 \\
\text{abs}(-7) = 7
\]
Conditionals in Racket

\[
\{ \begin{align*}
\text{answer} & \quad \text{question} \\
\vdots \\
\text{answer} & \quad \text{question}
\end{align*} \rightarrow \ (\text{cond} \ \begin{align*}
[\text{question} \ \text{answer}] \\
\ldots \\
[\text{question} \ \text{answer}]
\end{align*})
\]
Conditionals in Racket

\[
\text{(cond}
\quad \text{[question answer]}
\quad \ldots
\quad \text{[question answer]})
\]

• Any number of \textbf{cond} “lines”

• Each line has one \textit{question} expression and one \textit{answer} expression

• Last \textit{question} can be \textbf{else} for “otherwise”

\[
\text{(define (absolute x)}
\quad \text{(cond}
\quad \quad \text{[(> x 0) x]}
\quad \quad \text{[(else (- x))])}
\quad \quad \text{(absolute 10) \rightarrow 10}
\quad \quad \text{(absolute -7) \rightarrow 7}
\]

\]
Evaluation Rules for cond

First question is literally \texttt{true}:

\[
\text{(cond}
\begin{align*}
\text{[true answer]} & \quad \rightarrow \quad \text{answer} \\
\ldots & \\
\text{[question answer]} & 
\end{align*}
\]

i.e., keep only the first answer

Example:

\[
(* \ 1 \ (\text{cond} \quad \rightarrow \quad (* \ 1 \ 0) \rightarrow 0 \\
\text{[true 0]}))
\]
Evaluation Rules for cond

First question is literally \texttt{false}:

\[
\begin{align*}
\text{(cond} \\
\quad \text{[false answer]} \\
\quad \text{[question answer]} \quad \rightarrow \\
\quad \ldots \\
\quad \text{[question answer]})
\end{align*}
\]

i.e., throw away the first line

Example:

\[
\begin{align*}
(+ 1 \text{ (cond} \\
\quad \text{[false 1]} \\
\quad \text{[true 17]}) \\
\quad \text{[true 17]}) \\
\quad \rightarrow (+ 1 17) \rightarrow 18
\end{align*}
\]
Evaluation Rules for cond

First question isn’t a value, yet:

\[
\text{(cond [question answer] ... [question answer])} \quad \rightarrow \quad \text{(cond [nextques answer] ... [question answer])}
\]

where \text{question} \rightarrow \text{nextques}

i.e., evaluate first question as sub-expression

Example:

\[
(+ 1 \ (\text{cond} \ [(< 1 2) 5] \ [\text{else} 8]) \rightarrow (+ 1 \ (\text{cond} \ [\text{true} 5] \ [\text{else} 8])) \rightarrow (+ 1 5) \rightarrow 6
\]
Evaluation Rules for cond

No true answers:

\[(\text{cond}) \rightarrow \text{error}\]

Just an else:

\[(\text{cond} \quad [\text{else } \text{answer}]) \rightarrow \text{answer}\]
(define clyde)

; maybe-wanted : image -> image
(define (maybe-wanted who)
  (cond
   [(image=? who clyde)
    (above (text "WANTED" 32 "red") who)]
   [else
    who]))

(maybe-wanted)
Programming with Conditionals

\[
\text{(define } \text{clyde } \text{)}
\]

; maybe-wanted : image -> image
\[
\text{(define } \text{(maybe-wanted who)}
\]
\[
\text{(cond}
\]
\[\text{[(image=? who clyde)}
\]
\[\text{(above (text "WANTED" 32 "red") who)]]}
\]
\[\text{[else}
\]
\[\text{who][])}\]

\[
\text{(maybe-wanted } \text{)} \rightarrow
\]