# Testing Latexify.jl cases, and/or, negation

## 1. Cases

$$R\left(p, e, d\right) = \begin{cases} 0 & \text{if } e \\ \log\left(p\right) - d & \text{otherwise} \end{cases}$$

```
function R(p,e,d) # standard: with explicit return
    if e
        return 0
    else
        return log(p) - d
    end
end
```

$$R\left(p, e, d\right) = \begin{cases} 0 & \text{if } e \\ \log\left(p\right) - d & \text{otherwise} \end{cases}$$

```
function R(p,e,d) # without explicit `return`
    if e
        0
    else
        log(p) - d
    end
end
```

$$R\left(p, e, d\right) = \begin{cases} 0 & \text{if } \left(2d \neq 10\right) \end{cases}$$

```
function R(p,e,d) # without else
    if 2d != 10
        0
    end
end
```

$$R\left(p, e, d\right) = \begin{cases} 0 & \text{if } \left(\sqrt{d} < 777\right) \\ 999 & \text{if } e \vee \neg t \end{cases}$$

```
function R(p,e,d) # without else, but with elseif
    if sqrt(d) < 777
        0
    elseif e || !t
        999
    end
end
```

### 1.1 Ternary Ifs

$$R\left(p, e, d\right) = \begin{cases} 0 & \text{if } e \\ \log\left(p\right) - d & \text{otherwise} \end{cases}$$

```
R(p,e,d) = e ? 0 : log(p) - d
```

$$R\left(p, e, d, t\right) = \begin{cases} 0 & \text{if } e \vee t \\ \log\left(p\right) - d & \text{otherwise} \end{cases}$$

```
R(p,e,d,t) = e || t ? 0 : log(p) - d
```

$$R\left(p, e, d, t\right) = \begin{cases} 0 & \text{if } t \wedge e \\ \begin{cases} -d & \text{if } t \wedge \neg e \\ \log\left(p\right) & \text{otherwise} \end{cases} & \end{cases}$$

```
R(p,e,d,t) = (t && e) ? 0 : ((t && !e) ? -d : log(p)) # nested (same result w/out parens)
```

$$R\left(p, e, d, t\right) = \begin{cases} 0 & \text{if } t \wedge e \\ -d & \text{if } t \wedge \neg e \\ \log\left(p\right) & \text{otherwise} \end{cases}$$

```
R(p,e,d,t) = if (t && e); 0 elseif (t && !e); -d else log(p) end # one-line conditional
```

### 1.2 Larger Cases

$$R\left(p, e, d, t\right) = \begin{cases} 0 & \text{if } t \wedge e \\ -d & \text{if } t \wedge \neg e \\ -2d & \text{if } 2t \wedge e \\ -3d & \text{if } 3t \wedge e \\ \log\left(p\right) & \text{otherwise} \end{cases}$$

```
function R(p,e,d,t) # lots of elseifs
    if t && e
        return 0
    elseif t && !e
        return -d
    elseif 2t && e
        return -2d
    elseif 3t && e
        return -3d
    else
        return log(p)
    end
end
```

$$R\left(p,e,d,t\right) = \begin{cases} 0 & \text{if } t \wedge e \\ -d & \text{if } t \wedge \neg e \\ \begin{cases} -10d & \text{if } \left(t = 10\right) \\ -2d & \text{otherwise} \end{cases} & \text{if } 2t \wedge e \\ -3d & \text{if } 3t \wedge e \\ \log\left(p\right) & \text{otherwise} \end{cases}$$

```julia
function R(p,e,d,t) # lots of elseifs (with some nesting)
    if t && e
        return 0
    elseif t && !e
        return -d
    elseif 2t && e
        if t == 10
            return -10d
        else
            return -2d
        end
    elseif 3t && e
        return -3d
    else
        return log(p)
    end
end
```

$$\text{reward}\left(s,a,sp,o\right) = \begin{cases} -1.0 & \text{if } \left(a = 1\right) \\ -100.0 & \text{if } \left(s = a\right) \\ 10.0 & \text{otherwise} \end{cases}$$

```julia
function reward(s, a, sp, o) # note \mathrm for function names longer than one character
    if a == 1
        return -1.0
    elseif s == a
        return -100.0
    else
        return 10.0
    end
end
```

Embedded Julia source file.