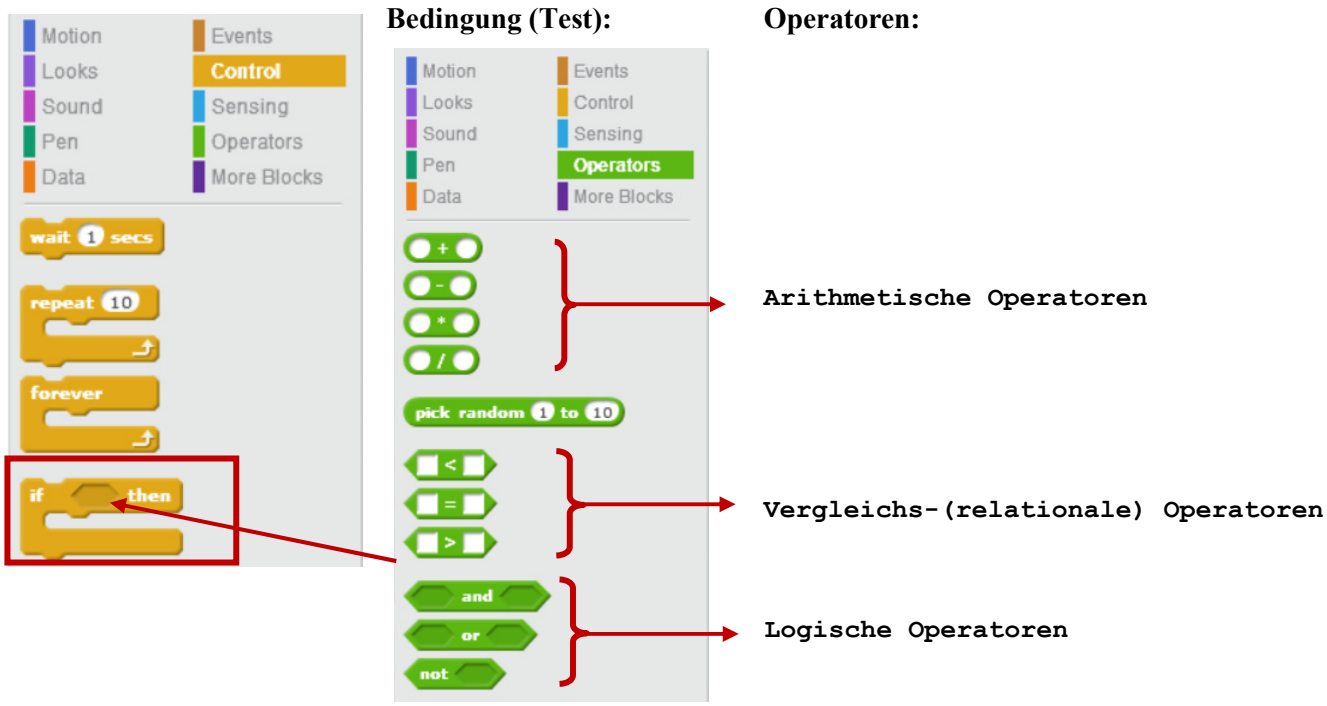
5 Auswahlbefehle

Aufgrund von Umgebungsinformationen soll der Roboter Entscheidungen treffen können, was er als nächstes tun soll. Hierfür gibt es die Auswahl (Selektion), die als IF Befehl im Roboter vorhanden ist. Es gibt zwei Versionen des IF Befehls: IF und IF/ELSE.

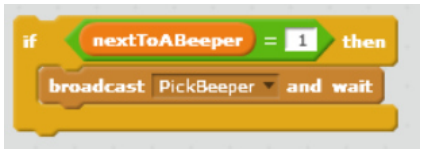
5.1 Der IF Befehl (IF-Anweisung)

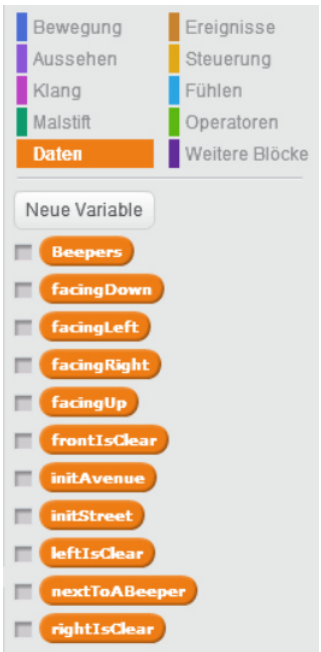
Der IF Befehl ist der einfachere Befehl der zwei Versionen. Er hat die folgende allgemeine Form:



5.2 Sensoren des Roboters (Bedingungen)

Der Bertl hat wie schon erwähnt (siehe 1.2 Robotereigenschaften) einige Sensoren, die im Hintergrund eine entsprechende Variable im Daten Bereich setzt. Diese Variable wiederum kann in einer **Bedingung** (z.B.: = auf Gleichheit) „abgefragt“ werden:





Die Einträge listen die Variablen auf, die ein Roboter mit seinen Sensoren testen kann und die entsprechende Variable daraufhin setzt oder löscht.

Im obigen Beispiel wird abgefragt ob Bertl auf einem Beeper steht, den er durch sein eingebautes Mikrofon hören kann.

```
if( nextToABeeper = 1 )
{
      // Bertl ist nahe eines Beepers ...
      PickBeeper();     // ... und hebt ihn auf
}
```

Der Test der **Bedingung** kann entweder Wahr (**true**) oder Falsch (**false**) sein und deshalb werden sie als **Boolean (bool)** bezeichnet. Der Befehle `PickBeeper()` wird nur ausgeführt wenn die **Bedingung true** ist.

5.3 Der IF/ELSE Befehl (IF/ELSE-Anweisung)

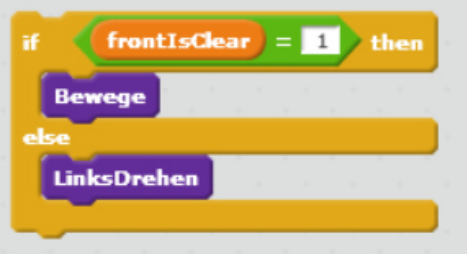
Der IF/ELSE Befehl ist der zweite Type des IF-Befehl. Er wird verwendet, wenn der Bertl abhängig vom Ergebnis der Bedingung eine von zwei alternativen Befehlen ausführen soll. Er hat die folgende allgemeine Form:

Scratch:

C, C++, Java, …

```
if( <Bedingung> )
{
        <Befehlsliste 1>
}
else
{
        <Befehlsliste 2>
}
```

Das folgende Beispiel überprüft ob der Weg für Bertl frei ist, also vor keiner Mauer (`FrontIsClear` =1) steht. Wenn das Ergebnis der Bedingung:



true ist: rufe die Funktion `Bewege` (`Move`) auf.

false ist: rufe die Funktion `LinksDrehen` (`TurnLeft`) auf.

Abhängig vom Ergebnis der Bedingung `FrontIsClear`, der Bertl also vor keiner Mauer steht, macht der Bertl einen Move oder, wenn eine Mauer vor ihm vorhanden ist den Befehl TurnLeft.

Obiger Befehl könnte in C kürzer geschrieben werden, da in den <Befehlslisten> jeweils nur ein Befehl enthalten ist können die geschwungenen Klammern weggelassen werden:

```
if( frontIsClear == 1 )
    Move();
else
    TurnLeft();
```

5.4 Komplexere Bedingungen (Nested IF)

Es ist nicht unbedingt eine triviale Angelegenheit, wenn ein Roboter zwei oder mehrere Entscheidungen zur gleichen Zeit zu treffen hat. Daher stellen die Programmiersprachen bei den IF und IF/ELSE Befehlen (*instructions*) Möglichkeiten zur gleichzeitigen Überprüfung zur Verfügung. Das folgende Beispiel zeigt einen komplexeren IF Test - überlegen Sie was dieser macht:


```
1   if( nextToABeeper == 0 )
2   {
3         if( frontIsClear == 1 )
4         {
5             Move();
6         }
7   }
```

Der Roboter macht den Befehl Bewege() (Move) nur wenn:

1. er auf KEINEN Beeper steht (Zeile 1)
2. keine Mauer vor ihm ist (Zeile 3)

Wie könnten diese zwei Tests in eine Bedingung für einen einzigen IF-Befehl zusammengefasst werden?