



Facultad de Matemática y Computación

PROYECTO DE PROGRAMACIÓN II

H U L K

Havana University Language for Kompilers

Estudiante: *Claudia Hernández Pérez*

Grupo: *C-113*

HULK es un lenguaje de programación imperativo, funcional, estática y fuertemente tipado. Casi todas las instrucciones en HULK son expresiones. En particular, el subconjunto de HULK que usted implementar se compone solamente de expresiones que pueden escribirse en una línea.

◦ Tipos Básicos

El lenguaje HULK contiene tres tipos básicos: 'string', 'boolean' y 'number'.

```
public static bool IsString(string s) {
    // Método para saber si una expresión es un string
    s = Aux.StringOut(s);
    s = s.Replace(" ", "");
    return s == "\\\"\\\" || s.Contains("@");
}

public static List<char> symbols = new() {'*', '/', '^', 'X', '+', '-', '(', ')'};
2 references
public static bool IsBinary(string s) {
    // Método para determinar si una expresión es binaria (numéricos)
    string n = Aux.StringOut(s);
    return Aux.IsNumber(s) || symbols.GetRange(0, 6).Any(n.Contains);
}

public static bool IsBoolean(string s) {
    // Método para determinar si una expresión es booleana
    s = Aux.StringOut(s);
    s = s.Replace(" ", "");
    s = s.Replace(" ", "");
    return bool.TryParse(s, out bool _) || s.Contains("<") || s.Contains(">") ||
        s.Contains("==") || s.Contains("!=") || s.Contains("<=") || s.Contains(">=") ||
        s.Contains("&") || s.Contains("|");
}

public static bool IsNumber(string s) {
    // Método para conocer si la expresión es un número
    if (s.Contains(",")) return false;
    return double.TryParse(s, out _) || s.Trim() == "PI" || s.Trim() == "E";
}
```

Fig 1.1 Identificación de tipos básicos del lenguaje

La jerarquía de símbolos es:

- Símbolos de tipo 'number' : '^'; '*' y '/' y '%'; '+' y '-'
- Símbolos de tipo 'boolean': '!', '>', '<', '>=' y '<='; '==' y '!='; '&' y '|'
- Símbolos de tipo 'string' : '@'

De forma recursiva se procede en sentido contrario a la jerarquía, de forma que se va descomponiendo la expresión desde la operación más interna a la más externa. La expresión se va evaluando en dos miembros, excepto el '!' que solo admite miembro derecho, hasta hacerla tan pequeña

◦

◦

◦

◦

◦