UNIVERSIDADE FEDERAL DE ALAGOAS

INSTITUTO DE COMPUTAÇÃO

COMPILADORES - 2017.2

Resultados do Analisador Léxico da Linguagem Hapais

Autor 1
Dayvson SALES

Autor 2
Warley VITAL

May 31, 2018

Chapter 1

Alô Mundo

```
defmod AloMundo do
  def void main() do
    print("Alo mundo!");
  end
endmod
```

hello.hs

```
[001, 001] (0001,  TK_DEFMOD) {defmod}
[001, 008] (0000,      TK_ID) {AloMundo}
[001, 017] (0003,      TK_DO) {do}
[002, 003] (0002,     TK_DEF) {def}
[002, 007] (0011,    TK_VOID) {void}
[002, 012] (0000,      TK_ID) {main}
[002, 016] (0027,   TK_ABPAR) {(}
[002, 017] (0028,   TK_FCPAR) {)}
[002, 019] (0003,      TK_DO) {do}
[003, 005] (0025,   TK_PRINT) {print}
[003, 010] (0027,   TK_ABPAR) {(}
[003, 011] (0019,  TK_CTESTR) {Alo mundo!}
[003, 023] (0028,   TK_FCPAR) {)}
[003, 024] (0023,    TK_PVGL) {;}
[004, 003] (0004,     TK_END) {end}
[005, 001] (0005,  TK_ENDMOD) {endmod}
[006, 001] (0043,     TK_EOF) {}
```

Chapter 2

Fibonnaci

```
defmod Fibonacci do
  def void fib(limite : int) do
    int[limite+1] arr;
    arr[0] = 0;
    arr[1] = 1;
    int i = 0;

  when(limite <= 0) do
      print("0");
    end

    until(i < limite) do
      when (i > 2) do
        arr[i] = arr[i-1] + arr[i-2];
      end

      print(arr[i]);

      when(i != limite-1) do
        print(", ");
      end

      i = i + 1;
    end
  end


  def void main() do
    int limite = read();
    fib(limite);
  end

endmod
```

fib.hs

```
[001, 001] (0001,  TK_DEFMOD) {defmod}
[001, 008] (0000,      TK_ID) {Fibonacci}
[001, 018] (0003,      TK_DO) {do}
[002, 003] (0002,     TK_DEF) {def}
[002, 007] (0011,    TK_VOID) {void}
[002, 012] (0000,      TK_ID) {fib}
[002, 015] (0027,   TK_ABPAR) {(}
[002, 016] (0000,      TK_ID) {limite}
[002, 023] (0022,    TK_DPTS) {:}
[002, 025] (0012,     TK_INT) {int}
[002, 028] (0028,   TK_FCPAR) {)}
[002, 030] (0003,      TK_DO) {do}
[003, 005] (0012,     TK_INT) {int}
[003, 008] (0029,   TK_ABCOL) {[}
```

```
[003, 009] (0000,      TK_ID) {limite}
[003, 015] (0034,     TK_OPA) {+}
[003, 016] (0017,  TK_CTEINT) {1}
[003, 017] (0030,   TK_FCCOL) {]}
[003, 019] (0000,      TK_ID) {arr}
[003, 022] (0023,    TK_PVGL) {;}
[004, 005] (0000,      TK_ID) {arr}
[004, 008] (0029,   TK_ABCOL) {[}
[004, 009] (0017,  TK_CTEINT) {0}
[004, 010] (0030,   TK_FCCOL) {]}
[004, 012] (0036,     TK_ATR) {=}
[004, 014] (0017,  TK_CTEINT) {0}
[004, 015] (0023,    TK_PVGL) {;}
[005, 005] (0000,      TK_ID) {arr}
[005, 008] (0029,   TK_ABCOL) {[}
[005, 009] (0017,  TK_CTEINT) {1}
[005, 010] (0030,   TK_FCCOL) {]}
[005, 012] (0036,     TK_ATR) {=}
[005, 014] (0017,  TK_CTEINT) {1}
[005, 015] (0023,    TK_PVGL) {;}
[006, 005] (0012,     TK_INT) {int}
[006, 009] (0000,      TK_ID) {i}
[006, 011] (0036,     TK_ATR) {=}
[006, 013] (0017,  TK_CTEINT) {0}
[006, 014] (0023,    TK_PVGL) {;}
[008, 002] (0008,    TK_WHEN) {when}
[008, 006] (0027,   TK_ABPAR) {(}
[008, 007] (0000,      TK_ID) {limite}
[008, 014] (0037,     TK_REL) {<}
[008, 015] (0036,     TK_ATR) {=}
[008, 017] (0017,  TK_CTEINT) {0}
[008, 018] (0028,   TK_FCPAR) {)}
[008, 020] (0003,      TK_DO) {do}
[009, 006] (0025,   TK_PRINT) {print}
[009, 011] (0027,   TK_ABPAR) {(}
[009, 012] (0019,  TK_CTESTR) {0}
[009, 015] (0028,   TK_FCPAR) {)}
[009, 016] (0023,    TK_PVGL) {;}
[010, 005] (0004,     TK_END) {end}
[012, 005] (0006,   TK_UNTIL) {until}
[012, 010] (0027,   TK_ABPAR) {(}
[012, 011] (0000,      TK_ID) {i}
[012, 013] (0037,     TK_REL) {<}
[012, 015] (0000,      TK_ID) {limite}
[012, 021] (0028,   TK_FCPAR) {)}
[012, 023] (0003,      TK_DO) {do}
[013, 007] (0008,    TK_WHEN) {when}
[013, 012] (0027,   TK_ABPAR) {(}
[013, 013] (0000,      TK_ID) {i}
[013, 015] (0037,     TK_REL) {>}
[013, 017] (0017,  TK_CTEINT) {2}
[013, 018] (0028,   TK_FCPAR) {)}
[013, 020] (0003,      TK_DO) {do}
[014, 009] (0000,      TK_ID) {arr}
[014, 012] (0029,   TK_ABCOL) {[}
[014, 013] (0000,      TK_ID) {i}
[014, 014] (0030,   TK_FCCOL) {]}
[014, 016] (0036,     TK_ATR) {=}
```

```
[014, 018] (0000,      TK_ID) {arr}
[014, 021] (0029,   TK_ABCOL) {[}
[014, 022] (0000,      TK_ID) {i}
[014, 023] (0034,     TK_OPA) {-}
[014, 024] (0017,  TK_CTEINT) {1}
[014, 025] (0030,   TK_FCCOL) {]}
[014, 027] (0034,     TK_OPA) {+}
[014, 029] (0000,      TK_ID) {arr}
[014, 032] (0029,   TK_ABCOL) {[}
[014, 033] (0000,      TK_ID) {i}
[014, 034] (0034,     TK_OPA) {-}
[014, 035] (0017,  TK_CTEINT) {2}
[014, 036] (0030,   TK_FCCOL) {]}
[014, 037] (0023,    TK_PVGL) {;}
[015, 007] (0004,     TK_END) {end}
[017, 007] (0025,   TK_PRINT) {print}
[017, 012] (0027,   TK_ABPAR) {(}
[017, 013] (0000,      TK_ID) {arr}
[017, 016] (0029,   TK_ABCOL) {[}
[017, 017] (0000,      TK_ID) {i}
[017, 018] (0030,   TK_FCCOL) {]}
[017, 019] (0028,   TK_FCPAR) {)}
[017, 020] (0023,    TK_PVGL) {;}
[019, 007] (0008,    TK_WHEN) {when}
[019, 011] (0027,   TK_ABPAR) {(}
[019, 012] (0000,      TK_ID) {i}
[019, 014] (0038,    TK_REL2) {!=}
[019, 017] (0000,      TK_ID) {limite}
[019, 023] (0034,     TK_OPA) {-}
[019, 024] (0017,  TK_CTEINT) {1}
[019, 025] (0028,   TK_FCPAR) {)}
[019, 027] (0003,      TK_DO) {do}
[020, 009] (0025,   TK_PRINT) {print}
[020, 014] (0027,   TK_ABPAR) {(}
[020, 015] (0019,  TK_CTESTR) {, }
[020, 019] (0028,   TK_FCPAR) {)}
[020, 020] (0023,    TK_PVGL) {;}
[021, 007] (0004,     TK_END) {end}
[023, 007] (0000,      TK_ID) {i}
[023, 009] (0036,     TK_ATR) {=}
[023, 011] (0000,      TK_ID) {i}
[023, 013] (0034,     TK_OPA) {+}
[023, 015] (0017,  TK_CTEINT) {1}
[023, 016] (0023,    TK_PVGL) {;}
[024, 005] (0004,     TK_END) {end}
[025, 003] (0004,     TK_END) {end}
[028, 003] (0002,     TK_DEF) {def}
[028, 007] (0011,    TK_VOID) {void}
[028, 012] (0000,      TK_ID) {main}
[028, 016] (0027,   TK_ABPAR) {(}
[028, 017] (0028,   TK_FCPAR) {)}
[028, 019] (0003,      TK_DO) {do}
[029, 005] (0012,     TK_INT) {int}
[029, 009] (0000,      TK_ID) {limite}
[029, 016] (0036,     TK_ATR) {=}
[029, 018] (0024,    TK_READ) {read}
[029, 022] (0027,   TK_ABPAR) {(}
[029, 023] (0028,   TK_FCPAR) {)}
```

```
[029, 024] (0023,    TK_PVGL) {;}
[030, 005] (0000,      TK_ID) {fib}
[030, 008] (0027,   TK_ABPAR) {(}
[030, 009] (0000,      TK_ID) {limite}
[030, 015] (0028,   TK_FCPAR) {)}
[030, 016] (0023,    TK_PVGL) {;}
[031, 003] (0004,     TK_END) {end}
[033, 001] (0005,  TK_ENDMOD) {endmod}
[034, 001] (0043,     TK_EOF) {}
```

Chapter 3

ShellSort

```
defmod Shellsort do

  def int[] shellsort(valores : int[]) do

    int n = len(valores); # retorna tamanho do array valores, aqui nao eh comando
    para os tokens
    int h = 1;

    until(h < n) do
      h = h * 3 + 1;
    end

    h = h / 3;

    int c;
    int j;

    until(n > 2) do

      rep (i : int = h, i < n, i = i + 1) do
        c = valores[i];
        j = i;

        until(j >= h && valores[j - h] > c) do
                valores[j] = valores[j - h];
                j = j - h;
        end

        valores[j] = c;
      end

      h = h / 2;

    end

    return valores;
  end

  def void main() do

    int[] valores;

    until(read() != EOF) do
      valores[] = lastValueRead();
    end

    int[] arr = shellsort(valores);

    rep(i : int = 0, i < len(valores), i = i + 1) do
      print(arr[i]);
      print("\n");
    end

  end
```

```
54
55 endmod
```

shell.hs

```
[001, 001] (0001,  TK_DEFMOD) {defmod}
[001, 008] (0000,      TK_ID) {Shellsort}
[001, 018] (0003,      TK_DO) {do}
[003, 003] (0002,     TK_DEF) {def}
[003, 007] (0012,     TK_INT) {int}
[003, 010] (0029,   TK_ABCOL) {[}
[003, 011] (0030,   TK_FCCOL) {]}
[003, 013] (0000,      TK_ID) {shellsort}
[003, 022] (0027,   TK_ABPAR) {(}
[003, 023] (0000,      TK_ID) {valores}
[003, 031] (0022,    TK_DPTS) {:}
[003, 033] (0012,     TK_INT) {int}
[003, 036] (0029,   TK_ABCOL) {[}
[003, 037] (0030,   TK_FCCOL) {]}
[003, 038] (0028,   TK_FCPAR) {)}
[003, 040] (0003,      TK_DO) {do}
[005, 005] (0012,     TK_INT) {int}
[005, 009] (0000,      TK_ID) {n}
[005, 011] (0036,     TK_ATR) {=}
[005, 013] (0000,      TK_ID) {len}
[005, 016] (0027,   TK_ABPAR) {(}
[005, 017] (0000,      TK_ID) {valores}
[005, 024] (0028,   TK_FCPAR) {)}
[005, 025] (0023,    TK_PVGL) {;}
[006, 005] (0012,     TK_INT) {int}
[006, 009] (0000,      TK_ID) {h}
[006, 011] (0036,     TK_ATR) {=}
[006, 013] (0017,  TK_CTEINT) {1}
[006, 014] (0023,    TK_PVGL) {;}
[008, 005] (0006,   TK_UNTIL) {until}
[008, 010] (0027,   TK_ABPAR) {(}
[008, 011] (0000,      TK_ID) {h}
[008, 013] (0037,     TK_REL) {<}
[008, 015] (0000,      TK_ID) {n}
[008, 016] (0028,   TK_FCPAR) {)}
[008, 018] (0003,      TK_DO) {do}
[009, 007] (0000,      TK_ID) {h}
[009, 009] (0036,     TK_ATR) {=}
[009, 011] (0000,      TK_ID) {h}
[009, 013] (0035,     TK_OPM) {*}
[009, 015] (0017,  TK_CTEINT) {3}
[009, 017] (0034,     TK_OPA) {+}
[009, 019] (0017,  TK_CTEINT) {1}
[009, 020] (0023,    TK_PVGL) {;}
[010, 005] (0004,     TK_END) {end}
[012, 005] (0000,      TK_ID) {h}
[012, 007] (0036,     TK_ATR) {=}
[012, 009] (0000,      TK_ID) {h}
[012, 011] (0035,     TK_OPM) {/}
[012, 013] (0017,  TK_CTEINT) {3}
[012, 014] (0023,    TK_PVGL) {;}
[014, 005] (0012,     TK_INT) {int}
[014, 009] (0000,      TK_ID) {c}
[014, 010] (0023,    TK_PVGL) {;}
```

```
[015, 005] (0012,     TK_INT) {int}
[015, 009] (0000,      TK_ID) {j}
[015, 010] (0023,    TK_PVGL) {;}
[017, 005] (0006,   TK_UNTIL) {until}
[017, 010] (0027,   TK_ABPAR) {(}
[017, 011] (0000,      TK_ID) {n}
[017, 013] (0037,     TK_REL) {>}
[017, 015] (0017,  TK_CTEINT) {2}
[017, 016] (0028,   TK_FCPAR) {)}
[017, 018] (0003,      TK_DO) {do}
[019, 007] (0007,     TK_REP) {rep}
[019, 011] (0027,   TK_ABPAR) {(}
[019, 012] (0000,      TK_ID) {i}
[019, 014] (0022,    TK_DPTS) {:}
[019, 016] (0012,     TK_INT) {int}
[019, 020] (0036,     TK_ATR) {=}
[019, 022] (0000,      TK_ID) {h}
[019, 023] (0021,   TK_SPTOR) {,}
[019, 025] (0000,      TK_ID) {i}
[019, 027] (0037,     TK_REL) {<}
[019, 029] (0000,      TK_ID) {n}
[019, 030] (0021,   TK_SPTOR) {,}
[019, 032] (0000,      TK_ID) {i}
[019, 034] (0036,     TK_ATR) {=}
[019, 036] (0000,      TK_ID) {i}
[019, 038] (0034,     TK_OPA) {+}
[019, 040] (0017,  TK_CTEINT) {1}
[019, 041] (0028,   TK_FCPAR) {)}
[019, 043] (0003,      TK_DO) {do}
[020, 009] (0000,      TK_ID) {c}
[020, 011] (0036,     TK_ATR) {=}
[020, 013] (0000,      TK_ID) {valores}
[020, 020] (0029,   TK_ABCOL) {[}
[020, 021] (0000,      TK_ID) {i}
[020, 022] (0030,   TK_FCCOL) {]}
[020, 023] (0023,    TK_PVGL) {;}
[021, 009] (0000,      TK_ID) {j}
[021, 011] (0036,     TK_ATR) {=}
[021, 013] (0000,      TK_ID) {i}
[021, 014] (0023,    TK_PVGL) {;}
[023, 009] (0006,   TK_UNTIL) {until}
[023, 014] (0027,   TK_ABPAR) {(}
[023, 015] (0000,      TK_ID) {j}
[023, 017] (0037,     TK_REL) {>}
[023, 018] (0036,     TK_ATR) {=}
[023, 020] (0000,      TK_ID) {h}
[023, 022] (0039,     TK_AND) {&&}
[023, 025] (0000,      TK_ID) {valores}
[023, 032] (0029,   TK_ABCOL) {[}
[023, 033] (0000,      TK_ID) {j}
[023, 035] (0034,     TK_OPA) {-}
[023, 037] (0000,      TK_ID) {h}
[023, 038] (0030,   TK_FCCOL) {]}
[023, 040] (0037,     TK_REL) {>}
[023, 042] (0000,      TK_ID) {c}
[023, 043] (0028,   TK_FCPAR) {)}
[023, 045] (0003,      TK_DO) {do}
[024, 017] (0000,      TK_ID) {valores}
```

```
[024, 024] (0029,   TK_ABCOL) {[}
[024, 025] (0000,      TK_ID) {j}
[024, 026] (0030,   TK_FCCOL) {]}
[024, 028] (0036,     TK_ATR) {=}
[024, 030] (0000,      TK_ID) {valores}
[024, 037] (0029,   TK_ABCOL) {[}
[024, 038] (0000,      TK_ID) {j}
[024, 040] (0034,     TK_OPA) {-}
[024, 042] (0000,      TK_ID) {h}
[024, 043] (0030,   TK_FCCOL) {]}
[024, 044] (0023,    TK_PVGL) {;}
[025, 017] (0000,      TK_ID) {j}
[025, 019] (0036,     TK_ATR) {=}
[025, 021] (0000,      TK_ID) {j}
[025, 023] (0034,     TK_OPA) {-}
[025, 025] (0000,      TK_ID) {h}
[025, 026] (0023,    TK_PVGL) {;}
[026, 009] (0004,     TK_END) {end}
[028, 009] (0000,      TK_ID) {valores}
[028, 016] (0029,   TK_ABCOL) {[}
[028, 017] (0000,      TK_ID) {j}
[028, 018] (0030,   TK_FCCOL) {]}
[028, 020] (0036,     TK_ATR) {=}
[028, 022] (0000,      TK_ID) {c}
[028, 023] (0023,    TK_PVGL) {;}
[029, 007] (0004,     TK_END) {end}
[031, 007] (0000,      TK_ID) {h}
[031, 009] (0036,     TK_ATR) {=}
[031, 011] (0000,      TK_ID) {h}
[031, 013] (0035,     TK_OPM) {/}
[031, 015] (0017,  TK_CTEINT) {2}
[031, 016] (0023,    TK_PVGL) {;}
[033, 005] (0004,     TK_END) {end}
[035, 005] (0009,  TK_RETURN) {return}
[035, 012] (0000,      TK_ID) {valores}
[035, 019] (0023,    TK_PVGL) {;}
[036, 003] (0004,     TK_END) {end}
[038, 003] (0002,     TK_DEF) {def}
[038, 007] (0011,    TK_VOID) {void}
[038, 012] (0000,      TK_ID) {main}
[038, 016] (0027,   TK_ABPAR) {(}
[038, 017] (0028,   TK_FCPAR) {)}
[038, 019] (0003,      TK_DO) {do}
[040, 005] (0012,     TK_INT) {int}
[040, 008] (0029,   TK_ABCOL) {[}
[040, 009] (0030,   TK_FCCOL) {]}
[040, 011] (0000,      TK_ID) {valores}
[040, 018] (0023,    TK_PVGL) {;}
[042, 005] (0006,   TK_UNTIL) {until}
[042, 010] (0027,   TK_ABPAR) {(}
[042, 011] (0024,    TK_READ) {read}
[042, 015] (0027,   TK_ABPAR) {(}
[042, 016] (0028,   TK_FCPAR) {)}
[042, 018] (0038,    TK_REL2) {!=}
[042, 021] (0000,      TK_ID) {EOF}
[042, 024] (0028,   TK_FCPAR) {)}
[042, 026] (0003,      TK_DO) {do}
[043, 007] (0000,      TK_ID) {valores}
```

```
[043, 014] (0029,   TK_ABCOL) {[}
[043, 015] (0030,   TK_FCCOL) {]}
[043, 017] (0036,     TK_ATR) {=}
[043, 019] (0026, TK_LTVREAD) {lastValueRead}
[043, 032] (0027,   TK_ABPAR) {(}
[043, 033] (0028,   TK_FCPAR) {)}
[043, 034] (0023,    TK_PVGL) {;}
[044, 005] (0004,     TK_END) {end}
[046, 005] (0012,     TK_INT) {int}
[046, 008] (0029,   TK_ABCOL) {[}
[046, 009] (0030,   TK_FCCOL) {]}
[046, 011] (0000,      TK_ID) {arr}
[046, 015] (0036,     TK_ATR) {=}
[046, 017] (0000,      TK_ID) {shellsort}
[046, 026] (0027,   TK_ABPAR) {(}
[046, 027] (0000,      TK_ID) {valores}
[046, 034] (0028,   TK_FCPAR) {)}
[046, 035] (0023,    TK_PVGL) {;}
[048, 005] (0007,     TK_REP) {rep}
[048, 008] (0027,   TK_ABPAR) {(}
[048, 009] (0000,      TK_ID) {i}
[048, 011] (0022,    TK_DPTS) {:}
[048, 013] (0012,     TK_INT) {int}
[048, 017] (0036,     TK_ATR) {=}
[048, 019] (0017,  TK_CTEINT) {0}
[048, 020] (0021,   TK_SPTOR) {,}
[048, 022] (0000,      TK_ID) {i}
[048, 024] (0037,     TK_REL) {<}
[048, 026] (0000,      TK_ID) {len}
[048, 029] (0027,   TK_ABPAR) {(}
[048, 030] (0000,      TK_ID) {valores}
[048, 037] (0028,   TK_FCPAR) {)}
[048, 038] (0021,   TK_SPTOR) {,}
[048, 040] (0000,      TK_ID) {i}
[048, 042] (0036,     TK_ATR) {=}
[048, 044] (0000,      TK_ID) {i}
[048, 046] (0034,     TK_OPA) {+}
[048, 048] (0017,  TK_CTEINT) {1}
[048, 049] (0028,   TK_FCPAR) {)}
[048, 051] (0003,      TK_DO) {do}
[049, 007] (0025,   TK_PRINT) {print}
[049, 012] (0027,   TK_ABPAR) {(}
[049, 013] (0000,      TK_ID) {arr}
[049, 016] (0029,   TK_ABCOL) {[}
[049, 017] (0000,      TK_ID) {i}
[049, 018] (0030,   TK_FCCOL) {]}
[049, 019] (0028,   TK_FCPAR) {)}
[049, 020] (0023,    TK_PVGL) {;}
[050, 007] (0025,   TK_PRINT) {print}
[050, 012] (0027,   TK_ABPAR) {(}
[050, 013] (0019,  TK_CTESTR) {\n}
[050, 017] (0028,   TK_FCPAR) {)}
[050, 018] (0023,    TK_PVGL) {;}
[051, 005] (0004,     TK_END) {end}
[053, 003] (0004,     TK_END) {end}
[055, 001] (0005,  TK_ENDMOD) {endmod}
[056, 001] (0043,     TK_EOF) {}
```