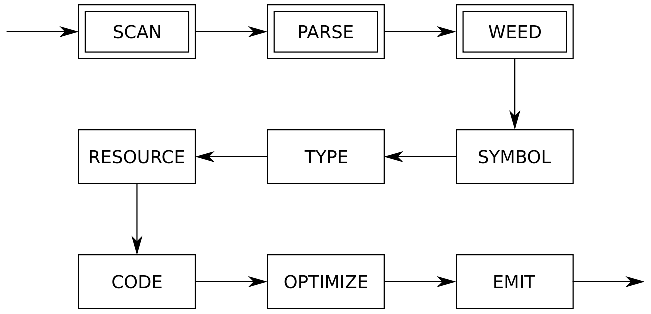
Abstract Syntax Trees (part 3)

COMP 520: Compiler Design (4 credits)

Professor Laurie Hendren

hendren@cs.mcgill.ca



Name the Dragon 2016

1. Parsimonia
2. Wendy the Whitespace-Intolerant Dragon
3. Mnemosyne
4. Context-Free Gary
5. Error-Gone (Eragon)

vikramPatientData.grp

```
/**
 * Contains all the information I might want to use across OncoTime progra
 */

group Id patientGroupOne = {1 to 250, 300 to 400, 1001}
group Birthyear patientBirthyearRange = {1950 to 1970}
```

vikramPatientData.grp

```
script DailyPatientHistory()
/**
 * Generates my patient Timelines.
 */

// ---- Change data in this group file ----
use vikramPatientData.grp

// ---- Filters ----
population is
      Id: <patientGroupOne>
      Birthyear: <patientBirthyearRange>
      Sex: M, F

// ---- Computations ----
{ foreach Patient p
   print p
}
```

Example 2 from Reference Compiler

```
script Barcharts()
/**
 * Generates barcharts.
 */

// ---- Change data in this group file ----
use vikramPatientData.grp

// ---- Filters ----
population is
      Id: <patientGroupOne>
      Birthyear: <patientBirthyearRange>
      Sex: M, F

{
      table t = count Patients by Birthyear
      table s = count Patients by Diagnosis
      table v = count Doctors by Id

      print t
      print s
      print v
}
```

Some Bits of OncoTime Grammar

```
Package otc;

/*****************************************************************
 * Helpers                                                       *
 *****************************************************************/
Helpers
   ascii_all = [0..127];
   alpha_lower = ['a' .. 'z'];
   alpha_upper = ['A' .. 'Z'];
   alpha     = [alpha_lower + alpha_upper];
   digit     = ['0'.. '9'];
   cr        = 13;                         /* Carriage Return */
   tb        = 9;                          /* Horizontal TAB */
   nl        = 10 | 13;                    /* New Line    */
   sp        = ' ';                        /* Space       */
   us        = 95;                         /* Underscore */

   /* Used for Documentation Comments, taken from Group 2015 Group 4 (R1)
   not_star = [ascii_all - '*'];
   not_star_or_slash = [ascii_all - ['*' + '/']];
```

```
/*******************************************************************
 * Tokens                                             *
 *******************************************************************/
Tokens

  /*********************
   * Keywords       *
   *********************/
    t_script =        'script';
    t_by   =          'by';
    t_of   =          'of';
    t_to   =          'to';
    ...
    t_female =        'female' |'f' | 'F' | 'Female';
```

```
/*********************
  * Types          *
  *********************/
   t_id_type =                'Id';
   t_sex_type =               'Sex';
   t_birthyear_type =         'Birthyear';
   t_patient_type =           'Patient' | 'Patients';
   t_doctor_type =            'Doctor' | 'Doctors';
   ...
   t_hours_type =             'Hours' | 'Hour';
```

```
/*********************
  * Char Tokens   *
  *********************/
   l_paren =   '(';
   r_paren =   ')';
   l_brace =   '{';
   r_brace =   '}';
   ...
```

```
/*********************
  * File Names    *
  *********************/
   t_group_file =    alpha (alpha | digit | us)* '.grp';

 /*********************
  * Values        *
  *********************/
   t_star =          '*';
   ...
   t_identifier =    alpha_lower (alpha | digit | us)*;
   t_script_name =   alpha_upper (alpha | digit | us)*;
   t_doc_comment =   ...
```

```
/*********************
  * Ignored      *
  *********************/
   empty_space =      cr | nl | tb | sp;
   line_comment =     '/''/' [ascii_all - [10+13]]* (10 | 13 | 10 13);


/*********************************************************************
 * Ignored Tokens                                        *
 *********************************************************************/
Ignored Tokens
   empty_space, line_comment;
```

```
/*********************************************************************
 * Productions                                               *
 *********************************************************************/
Productions

  /**********************
   * Root Program *
   **********************/
   program =
       {oncoprogram} header group_definitions* filter_definitions*
        computation_list
          {-> New program.oncoprogram(header, [group_definitions],
              [filter_definitions], computation_list)} |
       {groupfile} t_doc_comment group_definitions*
          {-> New program.groupfile(t_doc_comment,
                                    [group_definitions])};
```

```
/*********************
 * Header        *
 *********************/
 header =
    t_script t_script_name l_paren [params]:parameters? r_paren
      t_doc_comment dependencies*
     {-> New header(t_script_name,[params.typed_name],
        t_doc_comment, [dependencies])};
     ...

/*********************
 * Group Definitions *
 *********************/
 group_definitions =
    t_group typed_name equals l_brace typed_list r_brace
       {-> New group_definitions(typed_name, typed_list)};
```

```
/*********************
* Filter Definitions*
*********************/
 filter_definitions =
   {population_filter} t_population filter_list*
      {-> New filter_definitions.population_filter([filter_list])} |
   {period_filter} t_period filter_list*
      {-> New filter_definitions.period_filter([filter_list])} |
   {event_filter} t_events filter_list*
      {-> New filter_definitions.event_filter([filter_list])} |
   {doctor_filter} t_doctor_filter filter_list*
      {-> New filter_definitions.doctor_filter([filter_list])};

 filter_list =
    type colon typed_list
      {-> New filter_list(type, typed_list)};
```

```
/**********************
 * Computations *
 **********************/
 computation_list =
     l_brace computation* r_brace
        {-> New computation_list([computation])};
 ...
 computation =
 ...
```

```
/*******************************************************************
 * Abstract Syntax Tree                                  *
 *******************************************************************/
Abstract Syntax Tree

  /*********************
   * Root Program *
   *********************/
  program =
      {oncoprogram} header group_definitions* filter_definitions*
                    computation_list |
      {groupfile} t_doc_comment group_definitions*;


  /**********************
   * Header        *
   **********************/
  header =
      [name]:t_script_name [parameters]:typed_name*
            [script_comment]:t_doc_comment [uses]:dependencies*;

  dependencies =
      t_group_file*;

  /**********************
```

```
* Group Definitions *
**********************/
group_definitions =
    typed_name typed_list;
```

```
/***********************
 * Filter Definitions *
 ***********************/
 filter_definitions =
     {population_filter} filter_list* |
     {period_filter} filter_list* |
     {event_filter} filter_list* |
     {doctor_filter} filter_list*;

  filter_list =
     type typed_list;

/***********************
 * Computations  *
 ***********************/
 computation_list = computation*;
```