CADP tools for LOTOS

BY:
Mohsen Biglari
Main Titles

- CAESAR
- CAESAR.ADT
- CAESAR.BDD
- CAESAR.INDENT
- CAESAR.OPEN
A compiler that translates the *behavioural* part of a LOTOS specification into

- a C program (to be executed or simulated)
- Or into
- an LTS (to be verified using bisimulation tools and/or temporal logic evaluators).
CAESAR - Synopsis

- -bcg
  - Create a bcg graph (.bcg) → Can be processing by bcg tools
- -graph
  - Create a automaton (.gph) → States & Arcs
- -simulator
  - Generate the simulator program filename.c
CAESAR

- **Inputs:**
  - `filename.lotos` → a LOTOS program
  - `filename.h` → (a implementation in C of the abstract data types defined in the LOTOS source text)

- **Outputs:**
  - C program
  - Bcg graph
  - Finite state automaton
CAESAR

LOTOS spec.

- analysis

LOTOS tree

- expansion

SUBLOTOS tree

- generation

network

- optimisation

simulation

- graph

compile-time

run-time
CAESAR

- Analysis
  - Perform syntactic + static semantic analysis
  - Result $\rightarrow$ abstract syntax tree (LOTOS tree)

- Expansion
  - Translate Lotos behavior expressions into SubLotos ones ($\left[\right] \rightarrow \text{par}, \mid\left[\ldots\right]\mid \rightarrow \text{chioce}, \ldots$)
  - Result $\rightarrow$ SUBLOTOS tree
CAESAR

- Generation
  - Translate SubLotos behavior expressions into network
  - Sample: $\Rightarrow \text{“G O1, …On [V0] ; B0”}$
  - Gates $\Rightarrow G \text{ O1, …On}$
  - Condition $\Rightarrow V0$
  - Behavior $\Rightarrow B0$
CAESAR

- Optimization
  - Networks generated from SubLotos behavior expressions are generally not optimal
  - Control flow → Reduce the number of places, transitions, and units
  - Data flow → removing unused variables, eliminating useless assignments, folding constants
CAESAR - Sample

Specification BITALT [PUT,GET] : noexit behaviour
    P [PUT, GET]

Where
    process P [PUT, GET] : noexit :=
        PUT; P [GET, PUT]
    endproc

endspec
CAESAR - Sample

- caesar –bcg –english –simulator simple.lotos
- simple.bcg + simple.c
CAESAR.ADT

- A compiler that translates
  - the abstract data part of LOTOS specifications into
  - libraries of C types and functions
CAESAR.ADT - Synopsis

- `caesar.adt [-english] [-map] filename[.lotos]`
- `-map`
  - Generate `filename.map` which gives correspondence between sort and operation names occurring in `filename.lotos` and C type and function
CAESAR.ADT

- Inputs:
  - `filename.lotos` → a LOTOS program

- Outputs:
  - `filename.h` → containing the C types and functions that implement the LOTOS sorts and operations defined in `filename.lotos`
specification LIST : noexit
library NATURAL endlib
library BOOLEAN endlib
type LIST is NATURAL
  sorts LIST (*! implemented by LIST compared by CMP_LIST printed by PRINT_LIST *)
opns
    <> (*! implemented by EMPTY constructor *) : -> LIST
    _+_ (*! implemented by ADD constructor *) : NAT, LIST -> LIST
    _++_ (*! implemented by CONCAT *) : LIST, LIST -> LIST
    REVERSE (*! implemented by REVERSE *) : LIST -> LIST
    FIRST (*! implemented by FIRST *) : LIST -> NAT
    LAST (*! implemented by LAST *) : LIST -> NAT
eqns
  forall s, t : LIST, x, y : NAT
  ofsort LIST
    <> ++ s = s;
    (y + s) ++ t = y + (s ++ t);
    REVERSE (<>) = <>;
    REVERSE (y + s) = REVERSE (s) ++ (y + <>);
  ofsort NAT
    LENGTH (<>) = 0;
    LENGTH (y + s) = SUCC (LENGTH (s));
    FIRST (y + s) = y;
    LAST (y + s) = LAST (s);
endtype
behavior
  stop
endspec
CAESAR.ADT - Sample

- caesar.adt –english –map list.lotos
- list.h + list.map

```c
#ifndef CAESAR_ADT_INTERFACE
LIST REVERSE (CAESAR_ADT_1)
LIST CAESAR_ADT_1;
{
    if (CAESAR_ADT_is_EMPTY (CAESAR_ADT_1))
        return CAESAR_ADT_1;
    else
        return CONCAT (REVERSE (CAESAR_ADT_Get_2_ADD(CAESAR_ADT_1))
                        , ADD (CAESAR_ADT_Get_1_ADD (CAESAR_ADT_1), EMPTY ()));
}
#endif
```
CAESAR.BDD

- performs structural analysis of hierarchical Petri nets (Basic Petri Net encoded in a textual format) using Binary Decision Diagrams
- It is used in particular to perform optimizations on the intermediate Petri net generated by CAESAR
CAESAR.INDENT

- is a LOTOS program formatter.
- It checks first the syntax of the LOTOS description and then
- indents it in a way that makes reading it easy.
CAESAR.INDENT - Synopsis

- `caesar.indent [-line_length num] [-tabs] [-notabs] [-every num] [-m num] filename`

- `-line_length num`
  - The maximum line length. The default value is 500

- `-tabs`
  - Replace spaces by tabs

- `-every num`
  - The number of spaces that make up a tabulation. The default value is 8.

- `-m num`
  - The maximum length of the margin to `num`. The default value is 460.
CAESAR.INDENT - Sample

- caesar.indent simple.lotos

```lotos
specification BITALT [PUT, GET] : noexit
  where
  process P [PUT, GET] : noexit :=
  endproc
endspec
```

```lotos
specification BITALT [PUT, GET] : noexit
  behaviour
  
  P [PUT, GET]
  where
  process P [PUT, GET] : noexit :=
  PUT;
  P [GET, PUT]
  endproc
endspec
```