Report on Subproject S-1: Existing Software

J. Mairesse and the LIAFA (LITP) team

Contents

1 Introduction **2**

2 Automata **2**
2.1 Definition of an automaton 2
2.2 Existing software 3
2.2.1 AMORE 3
2.2.2 AUTOMATA (Pittsburgh) 4
2.2.3 AUTOMATA (Milan) 4
2.2.4 GRAIL 4
2.2.5 AUTOMATE 5

3 Queueing networks **5**
3.1 Definition of a queueing network 6
3.2 Existing software 7
3.2.1 MODLINE (QNAP) 7
3.2.2 TELPACK 7
3.2.3 QPACK 8

4 Petri nets **8**
4.1 Definition of a Petri net 8
4.2 Existing software 9
4.2.1 GreatSPN 9
4.2.2 ERS 9

5 Idempotent semirings **10**
5.1 Definition of an idempotent semiring 10
5.2 Existing software 11
5.2.1 MAX package 11

6 MAPLE **11**

7 SCILAB **11**

8 Expectations for the future **12**
8.1 LIAFA 12
8.2 INRIA 13
8.3 ARMINES 13

1 Introduction

Description of the subproject, quoted from the *Alapedes Project Programme*:

> Investigation and critical analysis of existing software.

The participants are LIAFA (LITP) and ARMINES.

One of the difficulty in investigating software associated with the Alapedes program is to decide what the scope of the investigation should be.
There are many domains, and associated software, which could be of interest for some of the subprojects, even though they are not primarily designed to cope with 'Performance Evaluation of Discrete Event Systems'. To avoid drifting away too far from the main preoccupations of the project, we have insisted on the software which were mainly designed for Discrete Event Systems (DES). We have given a particular attention to softwares based on algebraic properties of DES as opposed to pure simulation packages. We have also insisted on those which were easiest to obtain, i.e. the non-commercial ones, and on those developed by one of the partners of the project.
Another rather artificial distinction is that between 'existing software' (subproject S-1) and 'software in development' (subproject S-2). Some of the software developed by Alapedes members already existed before the beginning of the project (S-1 !) but are still currently developed and are being substantially improved (S-2 !). In such cases (see AUTOMATE, ERS, MAX), we have decided to include a description of these softwares in the report.

Organization of the document In the first three sections, we describe successively the softwares dealing with three different description formalisms for DES: automata, queueing networks and Petri nets. To fix the ideas, we have recalled the basic definition for these three models.
Starting from the description formalism of the model, these software usually propose solver and simulation based on a mathematical formalization. One of the recurrent mathematical framework is that of idempotent semirings. In §5, we present software for the computation in idempotent semirings.
In §6 and §7, we describe 2 global packages: MAPLE and SCILAB. They are not specifically designed for DES, but are used as a programming framework by several of the software described in the document.

2 Automata

The study of automata was initiated in the 50's by the work of Kleene. A finite automaton provides the simplest model of a computing device. It has proved to be a basic and central model in computer science, see [8, 15].
It is also one of the basic models for the algebraic study of discrete event systems. Hence, it is natural to start this report by describing the software dealing with finite automata.

2.1 Definition of an automaton

Here is a formal definition of an automaton:

Definition 2.1. *A finite automaton (resp. deterministic finite automaton) is a 5-tuple* (Q, A, δ, I, F) *where:*

- Q *is a finite set of states;* A *is a finite alphabet (set of input symbols);*
- δ *is a function* $\delta : Q \times A \to \mathcal{P}(Q)$ *(resp.* $\delta : Q \times A \to Q$*), where* $\mathcal{P}(Q)$ *is the set of subsets of* Q*;*
- $I \subset Q$ *is the set of initial states (resp.* $I \in Q$ *is the initial state);* $F \subset Q$ *is the set of final states.*

The function δ *is extended to the domain* $Q \times A^*$ *as follows:* $\delta(q, xa) = \delta(\delta(q, x), a)$.
The language L *recognized by the automaton is defined as the set of words:* $L = \{x \mid \delta(I, x) \in\}$.

2.2 Existing software

There exists many software manipulating automata. A complete list, with short comments, is maintained by Darrell Raymond.

1. http://www.csd.uwo.ca/staff/drraymon/.grail/grail.html
2. e-mail contact: "drraymon@solo.uwaterloo.ca".

This web page is a good starting point to investigate software dealing with automata. Some of the information below was gathered with the help of the above page but the links have been updated.

2.2.1 AMORE

The AMORE (computing Automata, MOnoids, and Regular Expressions) system was developed at the University of Kiel under the direction of W. Thomas. It is a program for the computation of finite automata, syntactic monoids of regular languages, and (possibly star-free) regular expressions. The system is implemented in C under Unix or DOS. A graphical component generates state graphs of finite automata on the screen from given transition tables (under Unix only). It can be obtained by anonymous ftp.

1. http://www.informatik.uni-kiel.de/inf/Thomas/amore.html
2. FTP: ftp.informatik.uni-kiel.de (134.245.15.113) in pub/kiel/amore/.
3. e-mail contact: amore@informatik.uni-kiel.de.
4. snail-mail contact: Prof. Dr. Wolfgang Thomas, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik, Olshausenstrasse 40, D-24098 Kiel, Denmark.

2.2.2 AUTOMATA (Pittsburgh)

AUTOMATA is a Mathematica based package that manipulates finite state machines and their syntactic semigroups. A number of operations on 1-dim cellular automata are also available. All the algorithms are implemented in Mathematica, and a good number of them is also implemented externally in a C++ library (e.g., power automaton construction, minimization using Hopcroft's algorithm, $\mathcal{D}$-class decomposition). The C++ code can be used directly or, via MathLink, from within Mathematica. On a standard workstation, the external code allows one to generate machines and/or semigroups with several 10000 elements. The program can be downloaded from the web page.

1. http://www.cs.cmu.edu/afs/cs.cmu.edu/user/sutner/www/auto.html
2. e-mail contact: "sutner@cs.cmu.edu"
3. snail-mail contact: Klaus Sutner, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA.

2.2.3 AUTOMATA (Milan)

From the COLOS project (European project, started in 1988 and supported by Hewlett-Packard) at the University of Milan. AUTOMATA is a tool for modelling automata and experimenting with grammars. The environment deals with: finite state automata (FSA, both deterministic and non deterministic), push-down automata - PDA (both deterministic and non deterministic), finite state transducers (Moore and Mealy machines), grammars, regular expressions, pattern matching. It provides the visualization of some important algorithms, such as the transformation of non deterministic automata into deterministic ones or the minimization. The automata are defined by entering their state diagram in a graphic editor or by giving a grammar. It can be downloaded from the web page.

1. http://www.dsi.unimi.it/Users/Labs/aa2/public_html/colos/
2. e-mail contact: " alberti@dsi.unimi.it"

2.2.4 GRAIL

Grail is developed by Darrell Raymond and Derick Wood. Grail is a symbolic computation environment for finite-state machines, regular expressions, and other formal language theory objects. Using Grail, one can input machines or expressions, convert them from one form to the other, minimize, make deterministic, complement, and perform many other operations. Grail is intended for use in teaching, for research into the properties of machines, and for efficient computation with machines. Grail is written in C++. It can be accessed either through a process library or through a C++ class library. The source code can be downloaded from the web page.

1. http://www.csd.uwo.ca/staff/drraymon/.grail/grail.html
2. e-mail contact: "drraymon@csd.uwo.ca"
3. snail-mail contact: Darrell Raymond, Department of Computer Science, 20 Erb Street West, University of Waterloo, Waterloo, Ontario, Canada, N2L 1T2.

2.2.5 AUTOMATE

The AUTOMATE project is being developed by one of the ALAPEDES TEAMS: LIAFA (LITP). For this reason, we will go into more details to describe it. The original software was written in C by Champarnaud and Hansel [6]. The project is now pursued both in the University of Rouen (LIR), and in LIAFA. The project in Rouen is conducted by J.-M. Champarnaud and in LIAFA by J.-E. Pin. For additional information:

1. http://www-litp.ibp.fr:80/ jep/node13.html
2. e-mail contact: "Jean-Marc.Champarnaud@dir.univ-rouen.fr" and "pin@litp.ibp.fr"
3. snail-mail contact: Jean-Marc Champarnaud, LIR, Faculté des Sciences, Université de Rouen, 76821 Mont-Saint-Aignan cedex, France; and Jean-Eric Pin, LIAFA, Boite 168, 2 Place Jussieu, 75251 PARIS CEDEX 05, France.

Here is a description of the software AUTOMATE. The objects manipulated are automata which can be given under the following forms:

- Labeled graph, transition monoid, regular (rational) language.

The main functionalities implemented are:

- Computation of the minimal deterministic automaton of a regular language;
- for a transition monoid, computation of all the elements, the relations and decomposition into Green's classes (regular $\mathcal{D}$-classes only). Given a word, computation of its representative and type (regular, idempotent).

A programming language associated with AUTOMATE has been developed, [5], and the program also exists under the form of a MAPLE package, see §6.

A program dealing with finite semigroups and complementing AUTOMATE is currently being developed in LIAFA, see §8.1.

3 Queueing networks

Queueing networks were first studied in the beginning of the century in connection with telecommunication. They have become one the most popular modelling framework in communication, traffic control or manufacturing.
An interesting web page on queueing theory, including a list of software can be found at the following address:

1. http://www2.uwindsor.ca/people/hlynka/queue.html

3.1 Definition of a queueing network

The huge variety of queueing problems have given rise to an also huge variety of mathematical models for queues, each with its specificities, see [13, 1, 2].

Generally speaking, a queue consists of a *service unit* (processing unit) and a *buffer* (waiting line). The mathematical description of the queue is made through: (i) the *arrival process* of *customers* (which are going to wait in the buffer until being served by the processing unit), (ii) the *service process*, i.e. the type and quantity of service required by the customer from the unit and (iii) the *service discipline*, i.e. the algorithm determining the order in which customers are served by the processing unit.
A shorthand notation introduced by Kendall, and which has become classical, enables to classify the most common types of models of queues.

Definition 3.1. *A queue is described under the form $A/S/c/d$ DIS, where*

- *A corresponds to the arrival process. It is described by a point process $\{T_n, n \in \mathbb{N}$ or $\mathbb{Z}\}$, where T_n is the instant of arrival of the n-th customer. The following conventions are used (1) $A = M$ if the inter-arrival times $\{t_n = T_{n+1} - T_n\}$ are i.i.d. (independent identically distributed) and exponential; (2) $A = D$ if $t_n = c \in \mathbb{R}^+, \forall n$; (3) $A = GI$ if $\{t_n\}$ is i.i.d. with a general distribution;* (4) *$A = G$ if $\{t_n\}$ is stationary and ergodic (M stands for Markovian, D for Deterministic, GI for General Independent and G for General).*

- *S corresponds to the service process. It is described by a point process $\{s_n, n \in \mathbb{N}$ or $\mathbb{Z}\}$, where s_n is the service required by the n-th customer. The conventions are the same as for A (e.g. $S = M$ if the random variables $\{s_n\}$ are i.i.d. and exponential).*

- *$c \in \mathbb{N}\backslash\{0\}$ corresponds to the number of servers in the processing unit, i.e. the maximal number of customers which can be served at the same time.*

- *$d \in \mathbb{N} \cup \{+\infty\}$ corresponds to the size of the buffer, i.e. the maximal number of customers which can wait to be served.*

- *DIS corresponds to the service discipline. The following conventions are used* (1) *FIFO = First In First Out,* (2) *LIFO = Last In First Out, and so on.*

One should remark that with the exception of the simplest queues (e.g. $M/M/1/\infty$ FIFO or $M/D/1/\infty$ FIFO), the Kendall notation is not enough to completely describe a queue. For instance, for a $GI/GI/1/c$ FIFO queue, it is necessary to specify the distribution of the arrival and service processes and also to describe what happens to a customer arriving in front of a full buffer.

A *queueing network* consists of a finite set of queues, with external arrival processes, and which are inter-connected in such a way that a customer departing from one queue may become an arrival customer for an other queue. Several types of inter-connections are possible: for instance a queue i could be linked to at most one queue j, each customer leaving i being routed to j; or a queue i could be linked to several queues, each departing customer deciding where to go in a random way.

3.2 Existing software

An interesting web page on queueing story, including a list of software can be found at the following address:

1. http://www2.uwindsor.ca/people/hlynka/queue.html

Furthermore, the following two web pages devoted to Operations Research have very good sections on software. Some of them are of direct interest to Alapedes: codes for network optimization, job-shop simulation.

1. http://mat.gsia.cmu.edu/
2. http://www.maths.mu.oz.au/ worms/

Let us mention rapidly some of the programs.

3.2.1 MODLINE (QNAP)

MODLINE is commercialized by Simulog. It is originally an output of the ESPRIT European research program (IMSE project), where INRIA Sophia was one of the participants. MODLINE is an open environment for modeling and simulation of discrete events systems. It integrates the QNAP2 software package. QNAP2 is a generic language for simulation and performance evaluation of queuing network systems. In particular, here are the available resolution methods: Exact or approximate analytical models for fast analysis of steady states. The most accurate method is automatically selected by the system.
Markov solver computing an exact solution in steady cases that include complex synchronization processes.

1. http://www.simulog.fr/US/html/depts/sep.html
2. e-mail contact: "info@simulog.fr".
3. snail-mail contact: Simulog, 1, Rue James Joule, 78280 Guyancourt Cedex, FRANCE .

3.2.2 TELPACK

TELPACK is a teletraffic analysis software which allows users to solve structured Markov chains motivated by probabilistic modeling and analysis for design and engineering of computer and communication networks. Similar problems also arise in many other different areas. The software utilizes a small subset of CLAPACK routines, the most advanced and efficient numerical algebra routines available.
The program can be downloaded from the web-page.

1. http://www.cstp.umkc.edu/org/tn/telpack/home.html
2. e-mail contact: telpack@cstp.umkc.edu

3.2.3 QPACK

QPACK has been developed by M.L. Chaudhry. It consists of a series of user-friendly software packages for several different bulk and non-bulk queueing models. The software package finds numerically the steady-state probabilities and moments for the number of customers in the system at various time epochs (i.e. Pre-arrival, Random, and Post-departure). It also, finds moments for waiting times, and waiting time densities in some cases. The models presently available in the form of a package are:

$M/G/1$, $M/G/1/N+1$ $(GI/M/1/N+1)$, $M(X)/G/1$, $M/G(Y)/1$, $M/G(a,b)/1$ $(Eb/G/1)$, $GI(X)/M/1$, $(GI/Er/1)$, $GI/M(a,b)/1$, $M/M/c$, $M(X)/M/c$, $M/D/c$, $M(X)/D/c$.

This program is not free: University Faculties: 1st package $70.00 (U.S.); each additional package $60.00 (U.S.). More information to be found at:

1. http://www.rmc.ca/ chaudhry/qpack.html
2. e-mail contact: "chaudhry@mail.rmc.ca"
3. snail-mail contact: A & A Publications, 395 Carrie Crescent, Kingston, Ontario, K7M 5X7, Canada

4 Petri nets

Timed and untimed Petri nets have been actively studied, since their introduction in the 60's, in particular as a modelling and analysis tool for Discrete Event Systems, see [3, 4, 14].
In the case of untimed Petri nets, the basic object of study is the *reachability graph* describing all the possible markings. In the case of a bounded Petri net, the reachability graph can be viewed as a deterministic automaton, see §2.2.5.
Here we are interested in software dealing with the behaviour of timed Petri nets.

4.1 Definition of a Petri net

There exist many variants of Petri nets. To fix the ideas, we present the most standard one:

Definition 4.1. *A Petri net is a 4-tuple* $(\mathcal{P}, \mathcal{T}, \mathcal{F}, M)$*, where :*

- $\mathcal{P}$ *is a finite set, whose elements are called places.*
- $\mathcal{T}$ *is a finite set, whose elements are called transitions.*
- $\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ *is a relation between places and transitions.*
- M *is a map* $\mathcal{P} \to \mathbb{N}$*. The integer* $M(p)$ *is called the marking of place* p*.*

The marking M evolves according to the *firing rule*.

1. Transition a is *enabled* at M if there is at least one token in each of its input places.
2. An enabled transition a can *fire*. The firing transforms M into M' by removing one token from each of the input places and adding one token in each of the output places of a.

A Timed Petri net (TPN) is a net with *firing times* associated with transitions and/or *holding times* associated with places.

4.2 Existing software

There exists many software dealing with Petri nets, most of them being purely simulation software.
A very good access point is the web page:

1. http://www.dsi.unimi.it/Users/Tesi/trompede/petri/home.html

It contains a classification of the dozens of different kinds of Petri nets which coexist in the litterature as well as links to many existing software (62 of them are listed!).
We now detail two of these software, which are of particular interest.

4.2.1 GreatSPN

GreatSPN (GRaphical Editor and Analyzer for Timed and Stochastic Petri Nets) is a free software developed in the University of Torino. It is a package for the modelling, validation, and performance evaluation of distributed systems using Generalized Stochastic Petri Nets and their coloured extensions.
Along with a simulation module, there are analytic tools available:

- structural properties: deadlocks, traps, implicit places, conflicts, confusion structures;
- $\mathcal{P} - \mathcal{T}$-invariants;
- computation of the reachability graph;
- Markovian solvers for steady-state as well as transient performance evaluation exploiting sparse matrix numerical techniques;

Instruction on how to get the software (free for universities) can be found on the web page.

1. http://www.di.unito.it/WWW/PEgroup/GreatSPN/
2. e-mail contact: “marina@di.unito.it”
3. snail-mail contact: Marina Ribaudo, Università degli Studi di Torino, Dipartimento di Informatica, Corso Svizzera 185, 10149 Torino, Italia.

4.2.2 ERS

ERS is a developed by one of the Alapedes team, INRIA (Sophia-Antipolis, Mistral project). One of its specificities is to rely heavily on the algebraic formalization of Petri nets to propose exact solvers and efficient simulations.
ERS is a toolbox containing several programs for the performance evaluation of DES. Several formalisms are used, including Petri nets, Queueing networks and Task graphs. They are included in an environment allowing the user to go from model specification to result analysis. There exist bridges between ERS and QNAP2, see §3.2.1.
Here is a more detailed description of the functionalities provided:

- *Description languages*: the ‘Event Graph language’ (EGL), the ‘Petri net language’ (PNL) and the ‘Task graph language’ (TGL);

- *graphical interfaces*: 'ERS' an editor for Petri nets; 'pnanim' a trace based animator for Petri nets, 'TGED' an editor for task graphs.

- *code generation*: 5 different languages: 'EGL', 'TGL', 'MAGMAS language (see below), C and QNAP language;

- *simulation*: 5 different simulators, one for each language. MAGMAS is a parallel simulator working on a Connection Machine and based on the (max,+) linear representation of timed event graphs;

- *performance evaluation*: 'Marktool' is a Markov chain solver using MAPLE; 'Event Graph analysis tools', 'Petri Net analysis tools' and 'Task Graph analysis tools' contain both structural analysis and performance evaluation tools.

Other tools, for the distributed simulation of queuing networks are developed in INRIA (PARSEVAL, PROSIT, PASTEQ). It is intended to also integrate these programs in the ERS toolbox.

1. e-mail contact: "Alain.Jean-Marie@sophia.inria.fr"

2. snail-mail contact: Alain Jean-Marie, Projet Mistral, INRIA, BP 93, 06902 Sophia-Antipolis Cedex, France.

5 Idempotent semirings

In the algebraic study of DES, idempotent semirings play a central role, see [3, 7, 12]. We now present a software, MAX, which provides tools for working in idempotent semirings. Several of the programs currently developed within Alapedes follow this approach, see 8.

5.1 Definition of an idempotent semiring

Definition 5.1. *A set K equipped with two operations $\oplus$ and $\otimes$ is a semiring if $\oplus$ is associative and commutative, $\otimes$ is associative and distributive with respect to $\oplus$, there is a zero element 0 ($a \oplus 0 = a, a \otimes 0 = 0 \otimes a = 0$) and a unit element 1 ($a \otimes 1 = 1 \otimes a = a$). A semiring K is idempotent if $\forall a \in K,\ a \oplus a = a$.*

Some idempotent semirings are very classical and useful in the study of Discrete Event Systems:

- $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +)$, the so-called (max,+) algebra;

- $\mathbb{N}_{\min} = (\mathbb{N} \cup \{+\infty\}, \min, +)$, the so-called tropical semiring;

- $\mathcal{M}_{ax}^{in}[[\gamma, \delta]]$, the idempotent semiring of commutative formal series in two indeterminates (γ and δ) over the Boolean semiring and quotiented by the simplification rules: $\gamma^n \oplus \gamma^m = \gamma^{\min(n,m)}$ and $\delta^n \oplus \delta^m = \delta^{\max(n,m)}$.

5.2 Existing software

5.2.1 MAX package

MAX is a MAPLE package developed by Stéphane Gaubert from the Alapedes team of INRIA (Rocquencourt, Meta2 Project). The main source of documentation is Gaubert's thesis [10], Chapters VIII and IX. The package can be obtained by e-mail demand. For more information:

1. http://amadeus.inria.fr/gaubert/PAPERS/MAX.html
2. e-mail contact: "Stephane.Gaubert@inria.fr"
3. snail-mail contact: Stéphane Gaubert, INRIA, Domaine de Voluceau, 78153 Le Chesnay Cedex, France.

Here is a more precise description of the functionalities provided.

- $\mathbb{R}_{\max}$: basic algebraic operations;
- $\mathbb{R}_{\max}^{k\times k}$: basic algebraic operations, star operation, eigenvalue;
- rational series over $\mathcal{M}_{ax}^{in}[[\gamma,\delta]]$: basic algebraic operations, canonical form, star operation.

In a Timed Event Graph (with some input and output transitions) where all the timings are integer valued, the transfer function can be described as a rational series over $\mathcal{M}_{ax}^{in}[[\gamma,\delta]]$. For two specific kinds of Event Graphs (flow-shop and Kanban transfer line), MAX is equipped with commands which, given the Event Graph, provide its transfer function.
When some resources (i.e. number of tokens in some places) are unknown in a Timed Event Graph (with integer timing), the transfer function can be described as a formal rational series over $\mathcal{M}_{ax}^{in}[[\gamma,\gamma_1,\gamma_2,\ldots,\delta]]$ (where the γ_i's correspond to the unknowns). MAX enables one to manipulate images of these series in order to compute symbolically the throughput as a function of the unknown quantities (for optimisation purposes for instance).

6 MAPLE

Maple was originally developed by the University of Waterloo (Canada). It is a commercial symbolic computation system. It also includes classical numerical calculus functionalities.
All the information needed can be found at the home page:

1. http://www.maplesoft.com/
2. e-mail contact: "info_web@maplesoft.com"

7 SCILAB

It is a free software whose functionalities are close to those of the commercial program MATLAB. It is developed by INRIA (the Scilab group: INRIA-Rocquencourt Meta2 Project and Cergrene ENPC). Here is a description from *Introduction to Scilab* (document to be found at http://www-rocq.inria.fr/scilab/doc/intro-html/intro-html.html).

> Developed at INRIA, Scilab has been developed for system control and signal processing applications. It is freely distributed in source code format. (...) Scilab provides a variety of powerful primitives for the analysis of non-linear systems. Integration of explicit and implicit dynamic systems can be accomplished numerically. The scicos toolbox allows the graphic definition and simulation of complex interconnected hybrid systems. There exist numerical optimization facilities for non linear optimization (including non differentiable optimization), quadratic optimization and linear optimization.

Scilab can be obtained by anonymous ftp. There exist versions for Unix-Xwindow systems, Windows 95 and Macintosh. A lot of information can be found on the Scilab home page.

1. FTP: ftp.inria.fr (Internet #192.93.2.54) in "INRIA/Projects/Meta2/Scilab"
2. http://www-rocq.inria.fr/scilab/
3. e-mail contact: "Scilab@inria.fr".

There exists a Maple to Scilab interface, transforming Maple objects into Scilab functions.

8 Expectations for the future

The development of new software is the object of subproject S-2. Here is a preview of S-2, listing some software developments which have been initiated during the first year of operation of the Alapedes project.

To facilitate the integration of the software developed within Alapedes, several of them have been written using the software package SCILAB, see §7. Such an effort should be encouraged.

8.1 LIAFA

J.E. Pin and V. Froidure are currently developing a program, called 'SEMIGROUPE', and which complements AUTOMATE, see §2.2.5. Its specificity is to be able to deal with semigroups which are not necessarily defined as transformation semigroups (as opposed to AUTOMATE for instance). In SEMIGROUPE, one can consider semigroups of matrices with coefficients in some commutative semirings: Boolean semiring, $\mathbb{N}_{\max}$, $\mathbb{N}_{\min}$ and $(\mathbb{N}, +, \times)$ (see 5).
The functionalities already implemented (with some original algorithms) are (given the generators of the semigroup): list of the elements, relations, idempotents, computation of the minimal ideal, of Green's classes, of the inverses, of local subsemigroups, of the representative of an element, variety tests: commutativity, nilpotence, idempotence, aperiodicity, and so on. For more details, see [9].

M. Kanta is developing a software (TROPRAT) to work with power series in one variable over the tropical semiring $\mathbb{N}_{\min}$. It is known that in this case, all classical decidability problems are decidable (as opposed to the case of formal power series in several variables over the tropical semiring). It enables to make computations with such series. The functionalities implemented are: basic algebraic operations, normal form computations, verification of the equality of 2 series.

8.2 INRIA

E. Hautecloque, a masters student from ENSTA, has been working under the supervision of S. Gaubert (INRIA) and J. Mairesse (LIAFA) on implementing a software based on the results of [11]. The dynamic of jobshops is shown to be represented by a (max,+) automaton. This algebraic representation makes it possible to evaluate the different schedulings in a jobshop in an efficient way. The program, written in C and interfaced with Scilab, takes as an input a jobshop and computes the makespan of all admissible schedules.

8.3 ARMINES

M. McGettrick is working on adding functionalities to SCILAB to enable calculations in idempotent semirings. The functions already implemented or being currently developed are: $\mathbb{R}_{\max}$ and $\mathbb{R}_{\min}$ basic operations for (sparse) matrices; residuation for (sparse) matrices; star operation for matrices; karp/howard algorithms for eigenvalue.

References

[1] S. Asmussen. *Applied Probability and Queues.* John Wiley & Sons, New York, 1987.

[2] F. Baccelli and P. Brémaud. *Elements of Queueing Theory.* Number 26 in Applications of Mathematics. Springer Verlag, Berlin, 1994.

[3] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity.* John Wiley & Sons, New York, 1992.

[4] W. Brauer, W. Reisig, and G. Rozenberg, editors. *Petri Nets: Central Models and Their Properties*, volume 254 of *Lect. Notes in Computer Sciences.* Springer-Verlag, 1987.

[5] J.M. Champarnaud. A programming language for symbolic computation of regular languages, automata and semigroups. In *Proc. 9th Annual Symposium on Theoretical Aspects of Computer Science*, number 577 in Lect. Notes Comput. Sci., pages 605–606. Springer, 1992.

[6] J.M. Champarnaud and G. Hansel. AUTOMATE, a computing package for automata and finite semigroups. *J. Symbolic Computation*, 12:197–220, 1991.

[7] P. Dudnikov and S. Samborskiĭ. Endomorphisms of finitely generated free semimodules. In V. Maslov and S. Samborskiĭ, editors, *Idempotent analysis*, volume 13 of *Adv. in Sov. Math.* AMS, 1992.

[8] S. Eilenberg. *Automata, languages and machines*, volume A. Academic Press, New York, 1974.

[9] V. Froidure and J.E. Pin. Algorithms for computing finite semigroups. In F. Cucker and M. Shub, editors, *Foundations of Computational Mathematics*, pages 112–126. Springer-Verlag, 1997.

[10] S. Gaubert. *Théorie des systèmes linéaires dans les dioïdes.* PhD thesis, École des Mines, Paris, 1992.

[11] S. Gaubert and J. Mairesse. Modeling and analysis of timed Petri nets using heaps of pieces. In *Proceedings of the Europ. Conf. Control 97*, 1997. Submitted to IEEE-TAC.

[12] J. Gunawardena, editor. *Idempotency.* Publications of the Newton Institute. Cambridge University Press, 1997.

[13] F.P. Kelly. *Reversibility and Stochastic Networks.* Wiley, New-York, 1979.

[14] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[15] A. Salomaa and M. Soittola. *Automata Theoretic Aspects of Formal Powers Series.* Springer Verlag, Berlin, 1978.