CHRONOLOG: a tool for computer-assisted chronological research

Eythan Levy, Gilles Geeraerts and Frédéric Pluquet

Introduction

Archaeological research often involves complex debates regarding the chronology of archaeological strata, pottery types and royal figures. Yet, outside the field of radiocarbon calibration, computers are rarely used as a decision-making tool in these debates. We present a new tool called CHRONOLOG, which enables users to encode a wide set of chronological constraints, automatically checks their consistency and computes chronological estimates of dates and durations of chronological periods. The tool is based on algorithmic techniques from the field of temporal logics (Shostak 1981, Dill 1989) which, until recently, have never been applied to the field of archaeological chronology (see Geeraerts, Levy, and Pluquet 2017).

Data model

The CHRONOLOG data model features three types of objects:

- **Periods**: a Period is a continuous interval of time, without gaps. It is characterized by a *start date*, an *end date*, and a *duration*. It can represent a reign, an archaeological stratum, or a historical period, among others.
- **Sequences**: a Sequence is an ordered set of Periods where each Period starts exactly when the preceding one ends.
- **Synchronisms**: a Synchronism is a chronological constraint involving two Periods. The CHRONOLOG data model features several types of synchronisms, the most common one being *contemporaneity*, a constraint that imposes that the two Periods have at least one unit of time in common.

Additional constraints can be added on the start date, end date and duration of a Period:

- *Lower bound*: e.g. a date not earlier than 1984 CE or a duration of at least 5 years.
- *Upper bound*: e.g. a date not later than 1984 CE or a duration of at most 5 years.
- *Range*: e.g. a date between 1984 and 1990 CE or a duration between 5 and 10 years. This is equivalent to having both a lower and an upper bound.
- *Exact value:* e.g. the date 1984 CE or a duration of 5 years. This is equivalent to identical lower and upper bounds.

Main operations

CHRONOLOG automates two basic operations on chronological networks:

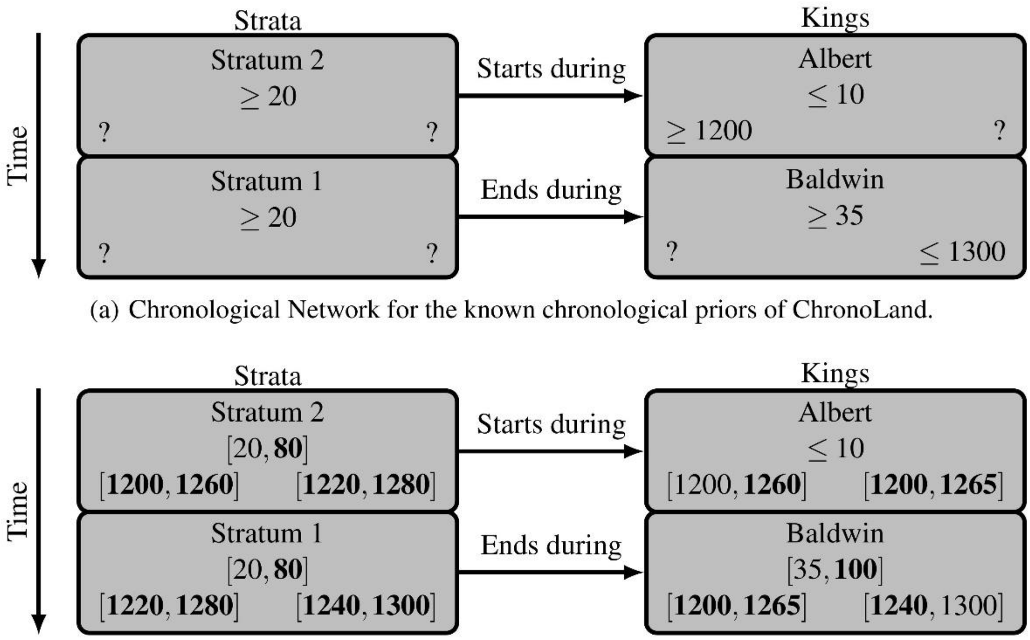
- **Consistency check**: this operation consists in checking whether all the encoded constraints are coherent, or whether they feature a contradiction.
- **Tightening**: this operation consists in computing the tightest possible ranges for each start date, end date and duration.

These operations enable two important applications:

- **Checking the impact of local changes**: every time a local change is made to the network, ChronoLog immediately launches a consistency check. If the model is found to be consistent, the tightening operation is applied, showing the impact of the local change on the overall model.
- **Testing chronological hypotheses:** one can want to check if a certain chronological hypothesis is true. For example: can two given kings have been contemporaries?

Example

In the kingdom of ChronoLand, king Albert and king Baldwin have reigned successively. Albert's reign starts no earlier than 1200 CE and lasts at most 10 years. Baldwin's reign ends no later than 1300 CE and lasts at least 35 years. The capital city of ChronoLand was excavated and revealed two strata. The earliest one (Stratum 2) was built during the reign of Albert, and the latest one (Stratum 1) was destroyed during the reign of Baldwin. We further assume that each stratum lasted for at least 20 years.



(b) Result of the tightening procedure. Enhanced upper and lower bounds are shown in bold.

Figure 1. The ChronoLand example. Duration constraints are given in the center of the box, start date constraints in the lower left side and end date constraints in the lower right side. Unknown dates are represented by question marks.

Fig. 1a shows the modelling of the above prior information within the ChronoLog data model. Fig. 1b provides the result of the tightening operation, i.e. the computed ranges for each start date, end date and duration. Note that the tightened ranges are not easy to guess, even on such a small example, let alone on a larger, real-life, archaeological case-study.

The ChronoLand example can also illustrate the two practical applications mentioned above. For example, what would be the impact of adding a 70 years upper bound to the duration of Baldwin's reign? This would limit the duration of each stratum to at most 60 years. This bound is due to the fact the time-span of our strata is bounded by the total duration of the dynasty, which is at most 80 years (10+70). Since each stratum has at least 20 years, the other one can have at most 60 years (80-20).

Another example is to check the following chronological hypothesis: is it possible that king Albert built Stratum 1? The answer is *no*, because, having already built Stratum 2, which lasts at least 20 years, and having reigned at most 10 years, his reign necessarily ended before the end of Stratum 2. The answers to these two questions are hard to obtain manually, yet can be obtained within a second with the use of CHRONOLOG.

Conclusion

The CHRONOLOG software enables users to build chronological model according to the data model described above. It implements fast algorithms in order to check consistency and to compute the tightened ranges of all dates and durations in real-time, even on models featuring hundreds of constraints. This allows users involved in chronological debates to test their models interactively and to check the global chronological impact of local chronological changes to the model. The objective of our software is to put chronological debates in archaeology on a more scientific and rigorous footing.

References

Dill, David L. 1989. “Timing assumptions and verification of finite-state concurrent systems.” In *Automatic Verification Methods for Finite State Systems, International Workshop, Grenoble, France, June 12-14, 1989, Proceedings*, edited by Joseph Sifakis, 197-212. Berlin: Springer Verlag.

Geeraerts, Gilles, Eythan Levy and Frédéric Pluquet. 2017. “Models and Algorithms for Chronology.” In *24th International Symposium on Temporal Representation and Reasoning (TIME 2017)*, edited by Sven Schewe, Thomas Schneider and Jef Wijsen, 13:1–13:18. Leibniz: Dagstuhl Publishing.

Shostak, Robert E. 1981. “Deciding linear inequalities by computing loop residues.” *Journal of the ACM* 28, no. 4: 769–779.