Properties of the Hebrew Calendar

David M. Russinoff
david@russinoff.com

May 27, 2022

Requirements of a Lunisolar Calendar

- Average length of a calendar month must approximate the average lunar (synodic) month of 29.53059 days.
- Average length of a year must approximate the average solar (tropical) year of 365.2422 days.
- This is achieved by arranging for the average number of months in a year to approximate the ratio

$$\frac{365.2422}{29.53059} \approx 12.36827.$$

Ancient Calendar was Observation-Based

- Sighting of the new moon was reported by witnesses to the high court in Jerusalem.
- Leap years were determined by agricultural and astronomical observations.

"Originally it was customary, when the court sanctified the new moon, to light fire signals on the tops of mountains, so that those who lived at a distance might learn of it. But when the Cutheans began to cause trouble by kindling fire signals in a mischievous way, in order to mislead the people, a law was enacted whereby messengers were sent out to inform the public."
– Moses Maimonides (a.k.a. Rambam). *Mishneh Torah, Book 3, Sanctification of the New Month*, 1174.

Months of the Year

- Tishri
- Cheshvan
- Kislev
- Tevat
- Shevat
- Adar I (leap year only)
- Adar (Adar II on leap year)
- Nisan
- Iyar
- Sivan
- Tammuz
- Av
- Elul

Fixed Calendar: Hillel ben Yehudah, 359 C.E.

Based on parameters established by Greeks and Babylonians:

- Mean synodic month: $\mathcal{L}$ = 29 days, 12 hours, 793 parts (1080 parts = 1 hour) $\approx$ 29.53059 days.
- Metonic cycle: 19-year cycle consisting of 12 common (12-month) years and 7 leap (13-month) years.

$$\frac{12 \cdot 12 + 7 \cdot 13}{19} = \frac{235}{19} \approx 12.36842 \text{ months/year}$$

Year y is a leap year $\Leftrightarrow y \bmod 19 \in \{0, 3, 6, 8, 11, 14, 17\}$.

Moladot

The calendar is based on an approximation of the lunar conjunction (*molad*) that marks the beginning of each month (based on the day beginning at 6 PM Jerusalem time).
Once the molad of any single month is established as a reference, every molad may be computed by adding a multiple of the molad interval $\mathcal{L}$.

- Hillel's reference molad: Solar eclipse of Nisan 4119 (?)
- Modern reference molad: Tishri of year 2, at 14 hours (8 AM) on a Friday (known as *Molad Adam*)

Beharad and the Absolute Calendar

As a convenient point of reference, the molad of Tishri of year 1, known as *Beharad*, is derived by subtracting $12 \cdot \mathcal{L}$ from Molad Adam, resulting in 5 hours, 204 parts on a Monday.

The preceding day (Sunday) is day 1 of the *absolute calendar*.

```
uint dayOfWeek(uint day) {return day % 7;}

struct Moment {uint day; uint hour; uint part;};
const Moment Beharad = {2, 5, 204};
const Moment Lunation = {29, 12, 793};
```

ACL2 translation:

```
(DEFUND LUNATION NIL (AS 'PART 793 (AS 'HOUR 12 (AS 'DAY 29 NIL))))

ACL2 !>(lunation)
((DAY . 29) (HOUR . 12) (PART . 793))
```

Computation of the Moladot

```
Moment molad(uint year) {
  uint priorMonths = 0;
  for (uint y=1; y<year; y++) {
    priorMonths += monthsInYear(y);}
  return addTime(Beharad, mulTime(priorMonths, Lunation));}
```

ACL2 translation:

```
(DEFUND MOLAD-LOOP-0 (Y YEAR PRIORMONTHS)
        (DECLARE (XARGS :MEASURE (NFIX (- YEAR Y))))
        (IF (AND (INTEGERP Y) (INTEGERP YEAR) (< Y YEAR))
            (LET ((PRIORMONTHS (+ PRIORMONTHS (MONTHSINYEAR Y))))
                 (MOLAD-LOOP-0 (+ Y 1) YEAR PRIORMONTHS))
           PRIORMONTHS))
(DEFUND MOLAD (YEAR)
        (LET* ((PRIORMONTHS 0)
               (PRIORMONTHS (MOLAD-LOOP-0 1 YEAR PRIORMONTHS)))
              (ADDTIME (BEHARAD)
                       (MULTIME PRIORMONTHS (LUNATION)))))
```

Rosh Hashanah on the Day of the Molad?

Let $\mathcal{M} = \{d, h, p\}$ and $\mathcal{M}' = \{d', h', p'\}$ be the moladot of years y and $y+1$, resp.

If y is a common year, then

$$\mathcal{M}' = \mathcal{M} + 12 \cdot \mathcal{L} = \{d, h, p\} + \{354, 8, 876\}$$

$$d' = \begin{cases} d + 354 & \text{if } (h : p) < (15 : 204) \\ d + 355 & \text{if } (h : p) \geq (15 : 204) \end{cases}$$

If y is a leap year, then

$$\mathcal{M}' = \mathcal{M} + 13 \cdot \mathcal{L} = \{d, h, p\} + \{383, 21, 589\}$$

$$d' = \begin{cases} d + 383 & \text{if } (h : p) < (2 : 491) \\ d + 384 & \text{if } (h : p) \geq (2 : 491) \end{cases}$$

Lengths of Months?

Month	Days
Tishri	30
Cheshvan	29, 30
Kislev	30, 29
Tevat	29
Shevat	30
Adar I (leap year only)	30
Adar (Adar II on leap year)	29
Nisan	30
Iyar	29
Sivan	30
Tammuz	29
Av	30
Elul	29
Common year total	354, 355
Leap year total	384, 383

BUT . . .

. . . there are certain days of the week on which Rosh Hashanah should not occur.

This means that RH may have to be postponed from the day of the molad.

In fact, RH is subject to 4 rules of postponement (*dechiyot*), which may result in a delay by up to 2 days.

This requires additional variation in year length.

Lengths of Months

Month	Days
Tishri	30
Cheshvan	29, 30
Kislev	30, 29
Tevat	29
Shevat	30
Adar I (leap year only)	30
Adar (Adar II on leap year)	29
Nisan	30
Iyar	29
Sivan	30
Tammuz	29
Av	30
Elul	29
Common year total	353, 354, 355
Leap year total	383, 384, 385

First Dechiyah

If the molad of Tishri occurs at or after noon, then Rosh Hashanah is postponed to the next day.

Origin unknown:

- To ensure that the new moon is visible at RH sunset?
- (Landau) To ensure that the molad of each month occurs before the end of the 1^{st} day of the month?

May be bypassed by replacing the molad with the *delayed molad*, 6 hours later (Gauss):

```
Moment dmolad(uint year) {
  Moment sixHours = {0, 6, 0};
  return addTime(molad(year), sixHours);}
```

Second Dechiyah

If the delayed molad occurs on a Wednesday, Friday, or Sunday, then Rosh Hashanah is postponed to the next day.

Dictated by religious constraints:

- Yom Kippur (10 Tishri) cannot occur on a Friday or Sunday
- Hoshanah Rabbah (21 Tishri) cannot occur on Shabbat

Third and Fourth Dechiyot

These are designed to ensure that the length of every year is admissible (353, 354, 355, 383, 384, or 385):

If the delayed molad of a common year occurs on a Tuesday at or later than $(15 : 204)$*, then Rosh Hashanah is delayed to the following Thursday.*

If the delayed molad following a leap year occurs on a Monday at or later than $(21 : 589)$*, then Rosh Hashanah is delayed to the next day.*

Determination of Rosh Hashanah

```
uint roshHashanah(uint year) {
  Moment dm = dmolad(year);
  uint day = dm.day;
  uint dw = dayOfWeek(day);
  if (dw == 1 || dw == 4 || dw == 6) {
    // 2nd dechiyah
    day++;
  }
  else if (dw == 3 && !earlier(dm, 15, 204) && !leap(year)) {
    // 3rd dechiyah
    day = day + 2;
  }
  else if (dw == 2 && !earlier(dm, 21, 589) && leap(year-1)) {
    // 4th dechiyah
    day++;
  }
  return day;
}
```

Lengths of Years and Months

```
uint yearLength(uint year) {
  return roshHashanah(year+1) - roshHashanah(year);}
```

Months of a common year are numbered 1–12.
Extra month of leap year (Adar 1) is number 13:

```
uint monthLength(uint month, uint yearLen) {
  uint monLen;
  switch(month) {
  case 2:
    monLen = yearLen == 355 || yearLen == 385 ? 30 : 29; break;
  case 3:
    monLen = yearLen == 353 || yearLen == 383 ? 29 : 30; break;
  default: monLen = month % 2 == 0 ? 29 : 30;
  }
  return monLen;}
```

Date Conversions

```
struct Date {uint day; uint month; int year;};
```

Conversion from Hebrew date to absolute date:

```
uint h2a(Date date) {
  // Compute number of days of date.year that precede date.month:
  uint priorDays = 0;
  for (uint m=1; m<date.month && (m<6 || date.month!=13); m++) {
    priorDays+= monthLength(m, yearLength(date.year));
  }
  if (leap(date.year) && date.month >= 6 && date.month != 13) {
    priorDays += 30;
  }
  // Add to that the absolute date of the last day of
  // the preceding year and the day of the month:
  return roshHashanah(date.year) - 1 + priorDays + date.day;}
```

Conversion from absolute date to Hebrew date:

```
Date a2h(uint d) { ...}
```

Some Properties of the Calendar

```
(defthmd legal-year-lengths
  (implies (posp y)
           (member (yearlength y)
                   (if1 (leap y)
                        '(383 384 385)
                      '(353 354 355)))))

(defthm landau-thm
  (implies (and (posp y) (posp month) (<= month (monthsinyear y)))
           (<= (day (monthlymolad month y))
               (h2a (firstofmonth month y)))))

(defthmd keviyot
  (implies
    (posp y)
    (let ((dw (dayofweek (roshhashanah y))) (len (yearlength y)))
      (or (and (= dw 3) (member len '(354 384)))
          (and (member dw '(0 2)) (member len '(353 355 383 385)))
          (and (= dw 5) (member len '(354 355 383 385)))))))
```