Posits and Assertions

LARS RÖNNBÄCK

@anchormodeling

ANCHOR™

www.uptochange.com and www.anchormodeling.com

Public domain photo from pexels.com

definition of a posit and its dereferencing set

A *posit* is a triple, $[\{(i_1, r_1), \ldots, (i_n, r_n)\}, v, t]$, where the first element is a set of ordered pairs, the second a data value, and the third a time point. The set is called a *dereferencing set*, where each member is an ordered pair of a unique identifier and a string.

definition of an assertion

Concurrent–reliance–temporal:

An *assertion* is a predicate, $\mathrm{assert}(P, p, \alpha, T)$, taking four arguments, where the first argument is a unique identifier, the second a posit, the third a real number in the range $[-1, 1]$, and the fourth a time point.

Uni–temporal:

An *assertion* is a predicate, $\mathrm{assert}(p)$, taking a posit as its argument.

It’s Monday. Archie invites you to meet him and Bella for lunch on Friday.

It's *Monday*. Archie invites you to meet him and Bella for lunch on Friday.

POSITING TIME

CHANGING TIME

POSITOR

It's *Monday*. *Archie* invites you to meet him and Bella for lunch on Friday.

POSITING TIME

CHANGING TIME


POSITOR
It's *Monday*. *Archie* invites you to meet him and Bella for lunch on *Friday*.
POSITING TIME
CHANGING TIME
HAPPENING TIME


POSITOR
MEETING
MEETING TYPE
It’s Monday. Archie invites you to meet him and Bella for lunch on Friday.
POSITING TIME
CHANGING TIME
HAPPENING TIME


POSITOR
MEETING
MEETING TYPE
It's *Monday*. *Archie* invites *you* to *meet him* and *Bella* for *lunch* on *Friday*.
POSITING TIME
PARTICIPANTS
HAPPENING TIME
CHANGING TIME

POSITOR MEETING MEETING TYPE

It's *Monday*. *Archie* invites *you* to *meet him* and *Bella* for *lunch* on *Friday*.

POSITING TIME PARTICIPANTS HAPPENING TIME

CHANGING TIME

ASSERT (P, p, α, T)

$($ $\longleftrightarrow$ $)$

POSIT $p = [\{(i_1, r_1), \ldots, (i_n, r_n)\}, v, t?]$

$[$ $\longleftrightarrow$ $]$

```
{(Archie, [{(#42,      type)},    lunch          ], 1, MONDAY),
 (Archie, [{(#42,      date)},   Friday          ], 1, MONDAY),
 (The Database, [{(#42, attendees)},    2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
 (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
 (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

POSITOR **MEETING** **MEETING TYPE**

It's *Monday*. *Archie* invites *you* to *meet him* and *Bella* for *lunch* on *Friday*.

POSITING TIME **PARTICIPANTS** **HAPPENING TIME**

CHANGING TIME

IDENTITIES

```
              {(Archie, [{(#42,      type)},   lunch          ], 1, MONDAY),
               (Archie, [{(#42,      date)},  Friday          ], 1, MONDAY),
         (The Database, [{(#42, attendees)},         2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

POSITOR **MEETING** **MEETING TYPE**

It's *Monday*. *Archie* invites *you* to *meet him* and *Bella* for *lunch* on *Friday*.

POSITING TIME **PARTICIPANTS** **HAPPENING TIME**

CHANGING TIME

IDENTITIES **ROLES**

```
     {(Archie, [{(#42,      type)},    lunch          ], 1, MONDAY),
      (Archie, [{(#42,      date)},   Friday          ], 1, MONDAY),
(The Database, [{(#42, attendees)},         2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

POSITOR MEETING MEETING TYPE

It's *Monday*. *Archie* invites *you* to *meet him* and *Bella* for *lunch* on *Friday*.

POSITING TIME PARTICIPANTS HAPPENING TIME

CHANGING TIME

IDENTITIES ROLES VALUES

```
      {(Archie, [{(#42,       type)},   lunch            ], 1, MONDAY),
       (Archie, [{(#42,       date)},  Friday            ], 1, MONDAY),
 (The Database, [{(#42, attendees)},        2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```


POSITOR
MEETING
MEETING TYPE
It's *Monday*. *Archie* invites *you* to *meet him* and *Bella* for *lunch* on *Friday*.
POSITING TIME
CHANGING TIME
PARTICIPANTS
HAPPENING TIME
IDENTITIES
ROLES
VALUES
RELIABILITIES
{(Archie, [{(#42, *type*)}, lunch], 1, MONDAY),
(Archie, [{(#42, *date*)}, Friday], 1, MONDAY),
(The Database, [{(#42, *attendees*)}, 2, monday], 1, MONDAY)}
{(Archie, [{(#42, *meeting*), (#1, *participant*)}, accepted, monday], 1, MONDAY),
(Bella, [{(#42, *meeting*), (#2, *participant*)}, accepted, monday], 1, MONDAY),
(You, [{(#42, *meeting*), (#3, *participant*)}, pending, monday], 1, MONDAY)}


POSITOR
MEETING
MEETING TYPE
It's Monday. Archie invites you to meet him and Bella for lunch on Friday.
POSITING TIME
PARTICIPANTS
HAPPENING TIME
CHANGING TIME
historized
IDENTITIES
ROLES
VALUES
RELIABILITIES
{(Archie, [{(#42, type)}, lunch], 1, MONDAY),
(Archie, [{(#42, date)}, Friday], 1, MONDAY),
(The Database, [{(#42, attendees)}, 2, monday], 1, MONDAY)}
{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
(Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
(You, [{(#42, meeting), (#3, participant)}, pending, monday], 1, MONDAY)}

POSITOR MEETING MEETING TYPE

It's *Monday*. *Archie* invites *you* to *meet him* and *Bella* for *lunch* on *Friday*.

POSITING TIME PARTICIPANTS HAPPENING TIME

CHANGING TIME

IDENTITIES ROLES VALUES RELIABILITIES

{(Archie, [{(#42, *type*)}, lunch], 1, MONDAY),
(Archie, [{(#42, *date*)}, Friday], 1, MONDAY),
(The Database, [{(#42, *attendees*)}, 2, monday], 1, MONDAY)}

{(Archie, [{(#42, *meeting*), (#1, *participant*)}, accepted, monday], 1, MONDAY),
(Bella, [{(#42, *meeting*), (#2, *participant*)}, accepted, monday], 1, MONDAY),
(You, [{(#42, *meeting*), (#3, *participant*)}, pending, monday], 1, MONDAY)}

POSITOR MEETING MEETING TYPE

It's *Monday*. *Archie* invites *you* to *meet him* and *Bella* for *lunch* on *Friday*.

POSITING TIME PARTICIPANTS HAPPENING TIME

CHANGING TIME

Concurrent–reliance–temporal Anchor Modeling

Someone is at some time somewhat sure about something (or its opposite).

IDENTITIES ROLES VALUES RELIABILITIES

```
        {(Archie, [{(#42,       type)},   lunch           ], 1, MONDAY),
         (Archie, [{(#42,       date)},  Friday           ], 1, MONDAY),
   (The Database, [{(#42, attendees)},         2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```


POSITOR
MEETING
MEETING TYPE
It's *Monday*. *Archie* invites *you* to *meet him* and *Bella* for *lunch* on *Friday*.
POSITING TIME
PARTICIPANTS
HAPPENING TIME
CHANGING TIME
Uni–temporal Anchor Modeling
Everyone is always completely sure about all the same things (and no opposites).
IDENTITIES
ROLES
VALUES
RELIABILITIES
{(Archie, [{(#42, type)}, lunch], 1, MONDAY),
(Archie, [{(#42, date)}, Friday], 1, MONDAY),
(The Database, [{(#42, attendees)}, 2, monday], 1, MONDAY)}
{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
(Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
(You, [{(#42, meeting), (#3, participant)}, pending, monday], 1, MONDAY)}

An assertion is called *positive* when the reliability is above zero.

```
(Archie, [{(#42, date)}, Friday],  1, MONDAY)
```

Archie is completely certain that the lunch is on Friday.

```
{(Archie, [{(#42,       type)},    lunch          ], 1, MONDAY),
  (Archie, [{(#42,       date)},   Friday          ], 1, MONDAY),
  (The Database, [{(#42, attendees)},      2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

An assertion is called *negative* when the reliability is below zero.

```
(Archie, [{(#42, date)}, Friday], -1, MONDAY)
```

Archie is completely certain that
the lunch is not on Friday.

```
{(Archie, [{(#42,       type)},    lunch          ], 1, MONDAY),
  (Archie, [{(#42,       date)},   Friday          ], 1, MONDAY),
(The Database, [{(#42, attendees)},       2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

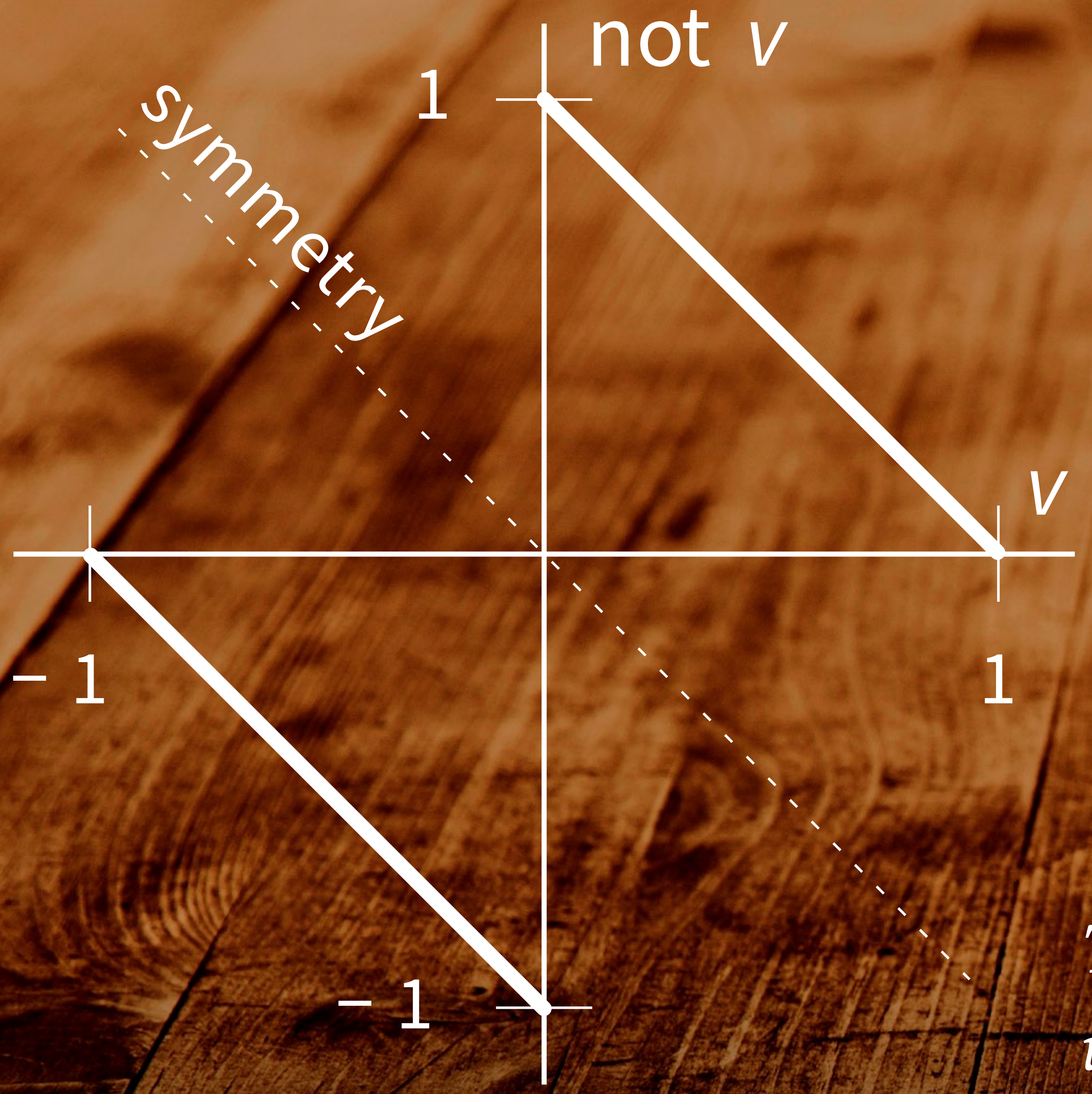
An assertion is called *completely uncertain* when the reliability is zero.

(Archie, [{(#42, *date*)}, Friday], 0, MONDAY)

Archie has absolutely no idea when the lunch is supposed to be held.

{(Archie, [{(#42, *type*)}, lunch], 1, MONDAY),
(Archie, [{(#42, *date*)}, Friday], 1, MONDAY),
(The Database, [{(#42, *attendees*)}, 2, monday], 1, MONDAY)}

{(Archie, [{(#42, *meeting*), (#1, *participant*)}, accepted, monday], 1, MONDAY),
(Bella, [{(#42, *meeting*), (#2, *participant*)}, accepted, monday], 1, MONDAY),
(You, [{(#42, *meeting*), (#3, *participant*)}, pending, monday], 1, MONDAY)}



Assertions without complements are in *canonical form*.

```
(Archie, [{(#42, date)}, not Friday],  1, MONDAY)
(Archie, [{(#42, date)},     Friday], -1, MONDAY)
```

We assume canonical form hereafter.

This symmetry shows how to translate assertions with complements to canonical form.

```
      {(Archie, [{(#42,      type)},    lunch          ], 1, MONDAY),
        (Archie, [{(#42,      date)},   Friday          ], 1, MONDAY),
  (The Database, [{(#42, attendees)},        2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

A *reassertion* asserts a posit and its reliability again at a later positing time.

```
(You, [{(#42, meeting), (#3, participant)}, pending, monday], 1, TUESDAY)
```

On Tuesday you state that you are still pending since Monday.

```
{(Archie, [{(#42,       type)},    lunch          ], 1, MONDAY),
 (Archie, [{(#42,       date)},   Friday          ], 1, MONDAY),
 (The Database, [{(#42, attendees)},      2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

A *revaluation* asserts a posit to a different reliability at a later positing time.

```
(Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 0.5, TUESDAY)
```

On Tuesday Bella states that what she said on Monday was that she only might attend on Friday.

```
{(Archie, [{(#42,       type)}, lunch         ], 1, MONDAY),
 (Archie, [{(#42,       date)}, Friday        ], 1, MONDAY),
 (The Database, [{(#42, attendees)},   2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
 (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
 (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

A *restatement* is when the preceding value over changing time is the same.

```
(The Database, [{(#42, attendees)}, 2, tuesday], 1, WEDNESDAY)
```

On Wednesday the Database states that there still were two attendees on Tuesday.

```
     {(Archie, [{(#42,      type)},    lunch       ], 1, MONDAY),
      (Archie, [{(#42,      date)},   Friday       ], 1, MONDAY),
(The Database, [{(#42, attendees)},        2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

A *change* is when the preceding value over changing time is different.

```
(You, [{(#42, meeting), (#3, participant)}, accepted, wednesday], 1, WEDNESDAY)
```

You accepted the meeting on Wednesday.

```
(The Database, [{(#42, attendees)}, 3, wednesday], 1, THURSDAY)
```

The Database states that the number of attendees has changed to three since Wednesday.

```
{(Archie, [{(#42,      type)},    lunch          ], 1, MONDAY),
 (Archie, [{(#42,      date)},   Friday          ], 1, MONDAY),
 (The Database, [{(#42, attendees)},       2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
 (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
 (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

A *retraction* is when a reliable posit is now considered completely unreliable.

(Archie, [{(#42, *type*)}, lunch], 0, THURSDAY)

On Thursday Archie cancels the lunch altogether.

```
{(Archie, [{(#42,       type)},    lunch          ], 1, MONDAY),
 (Archie, [{(#42,       date)},   Friday          ], 1, MONDAY),
(The Database, [{(#42, attendees)},       2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

A *correction* is a retraction of the erroneous posit and a simultaneous assertion.

these contain two different posits

```
(Archie, [{(#42, type)},  lunch], 0, THURSDAY)
(Archie, [{(#42, type)}, dinner], 1, THURSDAY)
```

On Thursday Archie realises that he intended to meet for dinner instead of lunch.

```
{(Archie, [{(#42,      type)},  lunch          ], 1, MONDAY),
 (Archie, [{(#42,      date)}, Friday          ], 1, MONDAY),
 (The Database, [{(#42, attendees)},   2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

With *decisiveness* only one value for each identity, role, and time may be asserted.

This excludes complete uncertainty, for which a positor may be oblivious to arbitrarily many posits.

```
        {(Archie, [{(#42,       type)},     lunch             ], 1, MONDAY),
         (Archie, [{(#42,       date)},    Friday             ], 1, MONDAY),
   (The Database, [{(#42, attendees)},          2, monday], 1, MONDAY)}

{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

With *indecisiveness* many values for each identity, role, and time may be asserted.

```
{(Archie, [{(#42, type)}, breakfast], 0.2, THURSDAY),
 (Archie, [{(#42, type)},     lunch], 0.3, THURSDAY),
 (Archie, [{(#42, type)},    dinner], 0.5, THURSDAY)}
```

Archie is not sure which meal the invitation was for.

As long as the reliabilities obey:

$$\frac{1}{2}\sum_{i=1}^{n}\left[1-\alpha_i|\alpha_i|^{-1}\right]+\sum_{i=1}^{n}\alpha_i \leq 1$$

the axiom of non-contradiction.

0.5(1–0.2/|0.2| +1–0.3/|0.3| + 1–0.5/|0.5|) + (0.2 + 0.3 +0.5) = 1

```
      {(Archie, [{(#42,      type)},   lunch        ], 1, MONDAY),
       (Archie, [{(#42,      date)},  Friday        ], 1, MONDAY),
 (The Database, [{(#42, attendees)},         2, monday], 1, MONDAY)}
```

```
{(Archie, [{(#42, meeting), (#1, participant)}, accepted, monday], 1, MONDAY),
  (Bella, [{(#42, meeting), (#2, participant)}, accepted, monday], 1, MONDAY),
    (You, [{(#42, meeting), (#3, participant)},  pending, monday], 1, MONDAY)}
```

definition of an anchor

Concurrent–reliance–temporal:

An *anchor* is a predicate, anchor(P, a, c, α, T), taking five arguments, where the first argument is a unique identifier, the second an assertion, the third a string, and the fourth a real number in the range $[-1, 1]$, and the fourth a time point.

Uni–temporal:

An *anchor* is a predicate, anchor(a, c), taking two arguments, where the first argument is an assertion and the second a string.


It's Monday. Archie invites you to meet him and Bella for lunch on Friday.
(One Modeler, a_1, *Appointment*, 1, MONDAY)
(One Modeler, a_2, *Appointment*, 1, MONDAY)
(One Modeler, a_3, *Appointment*, 1, MONDAY)
(Another Modeler, a_1, *Invitation*, 0.5, MONDAY)
(Another Modeler, a_2, *Invitation*, 0.5, MONDAY)
(Another Modeler, a_3, *Response*, 0.5, MONDAY)
Anchors provide a classification of posits and a set of anchors become a model.
ANCHOR (P, a, c, α, T)
ASSERT (P, p, α, T)
POSIT $p = [\{(i_1, r_1), \ldots, (i_n, r_n)\}, v, t?]$
a_1 = (Archie, [{(#42, *type*)}, lunch], 1, MONDAY)
a_2 = (Archie, [{(#42, *date*)}, Friday], 1, MONDAY)
a_3 = (The Database, [{(#42, *attendees*)}, 2, monday], 1, MONDAY)

It's Monday. Archie invites you to meet him and Bella for lunch on Friday.

```
(One Modeler,     a₁, Appointment,  1, MONDAY)
(One Modeler,     a₂, Appointment,  1, MONDAY)
(One Modeler,     a₃, Appointment,  1, MONDAY)
(Another Modeler, a₁,  Invitation, 0.5, MONDAY)
(Another Modeler, a₂,  Invitation, 0.5, MONDAY)
(Another Modeler, a₃,    Response, 0.5, MONDAY)
```

In uni-temporal Anchor modeling the theory becomes much simpler, with one implicit modeler and one certain model.

ANCHOR (a,c)

ASSERT (p)

POSIT $p = [\{(i_1, r_1), \ldots, (i_n, r_n)\}, v, t?]$

```
a₁ =       (Archie,       [{(#42,      type)},    lunch          ], 1, MONDAY)
a₂ =       (Archie,       [{(#42,      date)},   Friday          ], 1, MONDAY)
a₃ = (The Database,       [{(#42, attendees)},         2, monday], 1, MONDAY)
```