Collectif francophone pour l’enseignement libre de l’informatique

Bases de données historicisées

Modélisation discrète du temps

MCED_BDH_00

Christina KHNAISSER (*christina.khnaisser@usherbrooke.ca*)

—

CoFELI/«depot»/BDH_00_Modele-du-temps, version 1.0.0.a, en date du 2024-04-19

— document de travail, ne pas citer —

Table des matières

1. Introduction

Perspectives

Du temps absolu au temps relatif :

- Le temps selon Newton
- Le temps selon Leibniz
- La réflexion de Kant
- La démonstration d’Einstein…

Du temps continu au temps discret :

- Le temps macroscopique perçu
- De la théorie atomique à la théorie quantique
- La non-démonstration de la continuité
- L’hypothèse discrète
- Les interrogations du quantique
- Les impératifs de l’automatisation du calcul
- L’épuration de Lorentzos…

Synthèse des lectures proposées par [Manthey2018].

Le temps absolu selon Newton est

- vrai et mathématique,
- en soi et de sa propre nature,
- sans référence à rien d'extérieur,
- s'écoule uniformément.

Le temps relatif selon Einstein est

- une mesure sensible et externe de la durée d'un phénomène comparée à celle d'un autre phénomène (généralement un mouvement).

Phénomène de référence et unité de mesure

Les phénomènes de référence ont beaucoup changé à travers l'histoire : le temps écoulé entre deux levers de soleil consécutifs, entre deux positions zénithales consécutives du soleil, entre deux pleines lunes consécutives, etc.).

La définition scientifique contemporaine de la seconde, l'unité de temps fondamentale, telle qu'établie par le Bureau international des poids et mesures (BIPM) est la suivante :

> La seconde, symbole s, est l'unité de temps du SI. Elle est définie en prenant la valeur numérique fixée de la fréquence du césium, $\Delta\nu$Cs , la fréquence de la transition hyperfine de l'état fondamental de l'atome de césium 133 non perturbé, égale à 9 192 631 770 lorsqu'elle est exprimée en Hz, unité égale à s^{-1}.
>
> — Le Système international d'unités, 9e édition — 2019

2. Modèlisation du temps

La perception du temps dépend de plusieurs facteurs (astronomiques, religieux, etc.). Elle est le plus souvent continue, bornée ou non, parfois cyclique. Sa définition et les représentations qui en découlent sont donc souvent très complexes.

2.1. Modèle classique

- Le temps s'écoule en continu, à vitesse constante.
- Le temps s'écoule du passé vers le futur, il ne recule ni ne s'arrête jamais.
- Chaque observation ou action en cours a lieu à un moment précis, appelé *Maintenant*.
- *Maintenant* n'a pas de durée.
- Ce qui était *Maintenant* il y a un instant appartient déjà au passé; *Maintenant* est comme un point d'observation fixe sous lequel le temps passe.
- Le temps est illimité. Nous n'avons aucun moyen de dire si, ou quand, il y avait un début de temps, ou si, et quand, il y aura une fin de temps.

Le temps est illimité. Nous n'avons aucun moyen de dire si, où, ni quand il y a eu un début de temps, ni si, où ni quand il y aura une fin de temps — percevant ainsi le temps comme un processus infini semble « raisonnable ».

2.2. Modèle discret

- Le temps est discret, il « saute » d'un chronon à un autre (comme d'un niveau quantique à un autre).
- Le temps est unidimensionnel, ordonné, mais n'a pas de sens (orientation).

- Le temps est fini, compris entre deux bornes alpha et oméga.
- *Maintenant* correspond à un chronon (pas un intervalle) et conséquemment n'a pas de durée.

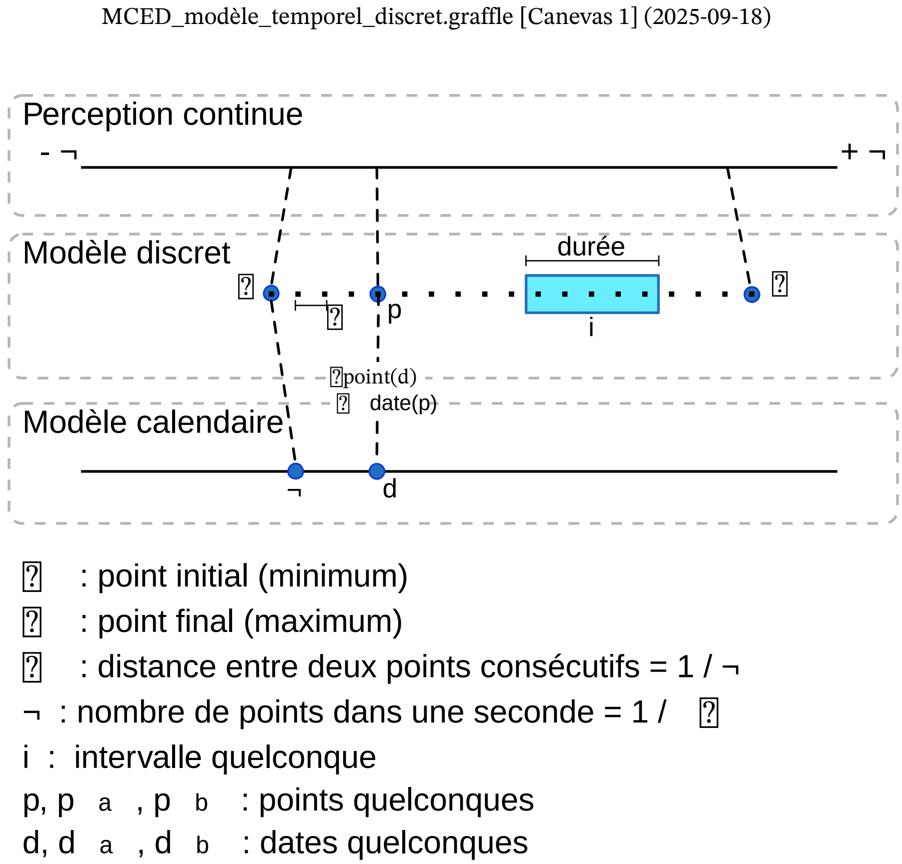

Figure 1. Illustration du modèle discret

2.2.1. Point

Un point [P] est tout type ordinal (donc discret et ordonné) borné. La valeur minimale est conventionnellement désignée par alpha (α) et la valeur maximale par oméga (ω).

Exemple 1. Exemples

> POINT (α= 01, ω = 99, γ = 1)
>
> INTEGER (α= -2147483648, ω = +2147483647, γ = 1)
>
> CHAR (α= A, ω = Z, γ = 1) une lettre selon l'ordre alphabétique latin tardif

Un modèle, ou un langage, pourront, notamment pour des raisons pratiques, ajouter des contraintes à cette définition ; en particulier, un point pourrait être limité aux seuls types scalaires.

2.2.2. Intervalle

Soit P un type point, un type intervalle de P, noté INTERVAL[P], est l'ensemble des valeurs d'intervalles légitimes définies sur P.

INTERVAL est le constructeur de type intervalle. Un intervalle peut être construit à l'aide de n'importe quel type point (un entier ou un type défini par énumération, par exemple).

Une valeur d'intervalle définie sur P est un ensemble de points tel que

$\{x \in P \mid b \leq x \leq e\}$

où b (pour « begin ») est le point de début de l'intervalle, e (pour « end ») est le point de fin de l'intervalle avec b ≤ e.

Corolairement, une valeur d'intervalle ne peut être (un ensemble) vide. Un intervalle de cardinalité 1 est appelé singleton ou encore intervalle unitaire.

Par ailleurs, notamment lorsqu'une sémantique physique est associée au point, il est convenu d'ajouter, à ce modèle un attribut gamma (γ) déterminant la distance entre deux points consécutifs selon une unité de mesure choisie.

Exemple 2. Exemples

INTERVAL[POINT], un type intervalle de POINT. Une valeur possible est [1:5], les points qui la constituent sont 1, 2, 3, 4 et 5.

INTERVAL[INTEGER], un type intervalle d'entiers relatifs. Une valeur possible est [-1:3], les points qui la constituent sont -1, 0, 1, 2 et 3.

Soit INTERVAL[CHAR], un type intervalle de caractères. Une valeur possible est ['B':'G'], les points qui la constituent sont 'B', 'C', 'D', 'E', 'F' et 'G'.

L'intervalle {x ∊ P | b ≤ x ≤ e} est conventionnellement noté de 5 façons, définies au tableau suivant. Soit f pour « from » (f=b-1), t pour « to » (t=e+1) et i pour « increment » (i=e-b+1). On remarque que, dans les cas discrets, ces notations sont équivalentes :

[b:e] = (b-1:e] = [b:e+1) = (b-1:e+1) = (b,e-b+1)

Enfin, presqu'équivalentes : Les valeurs [α:e] ne peuvent être dénotées à l'aide des notations ouvert-fermé et ouvert-ouvert. Les valeurs [b:ω] ne peuvent être dénotées à l'aide des notations fermé-ouvert et ouvert-ouvert. En conséquence, les notations fermé-fermé et fermé-incrément sont préférables lorsque vient le temps de formuler des règles générales (de façon à réduire le nombre de cas particuliers).

Tableau 1. Notation des intervalles

Type de bornes	Notation	Définition
fermé-fermé	[b:e]	b ≤ x ≤ e
fermé-ouvert	[b:t)	b ≤ x < t
ouvert-fermé	(f:e]	f < x ≤ e
ouvert-ouvert	(f:t)	f < x < t
fermé-incrément	(b,i)	b ≤ x ≤ b+i-1

2.2.3. Opérateurs d'intervalle

Tableau 2. Opérateurs de base

Définition	gauche	droite	retour	Fonction	PostgreSQL
Constructeur	Point (début)	Point (fin)	Intervalle	[d,f]	int4range(d,f, '[]')
Borne inférieure		Intervalle	Point	begin(i)	lower(i)
Borne supérieure		Intervalle	Point	end(i)	upper(i)
Prédécesseur		Intervalle	Point	pred(i)	lower(i) - 1
Successeur		Intervalle	Point	post(i)	lower(i) + 1
Cardinalité		Intervalle	entier	card(i)	-
Appartenance	Point	Intervalle	booléen	p ∈ i	p <@ i

Notes relatives à PostgreSQL

1. L'expression des opérateurs est valable pour les types d'intervalle d'entiers (int4range, int8range), mais

devrait être adapté pour les autres types d'intervalle… bien que cela ne soit pas toujours possible en particulier pour les intervalles dits continus.

2. Les intervalles de PostgreSQL, contrairement à ceux de l'algèbre d'Allen, comportent la valeur d'ensemble vide (sans compte la possibilité d'annulation comme pour tout autre type SQL). Il en découle plusieurs comportements mathématiquement incorrects qui ne sont pas pris en compte ici. Par exemple, attendu un intervalle d'entiers, la fonction de cardinalité card(i) pourrait être exprimée comme *end(i)-begin(i)+1* dans l'algèbre de Allen, ce qui donnerait `upper(i)-lower(i)+1` en PostgreSQL. Cela est cependant inexact dans le cas de l'ensemble vide.
3. Pour toutes ses raisons, il est préférable en PostgreSQL de se limiter aux intervalles d'entiers en les complétant au besoin par des fonctions de converssion vers d'autres types d'intérêt (Estampille, Date).

Opérateurs dd l'algèbre d'Allen

L'algèbre de d'Allen comporte 7 opérations de base dont six ont une version symétrique.

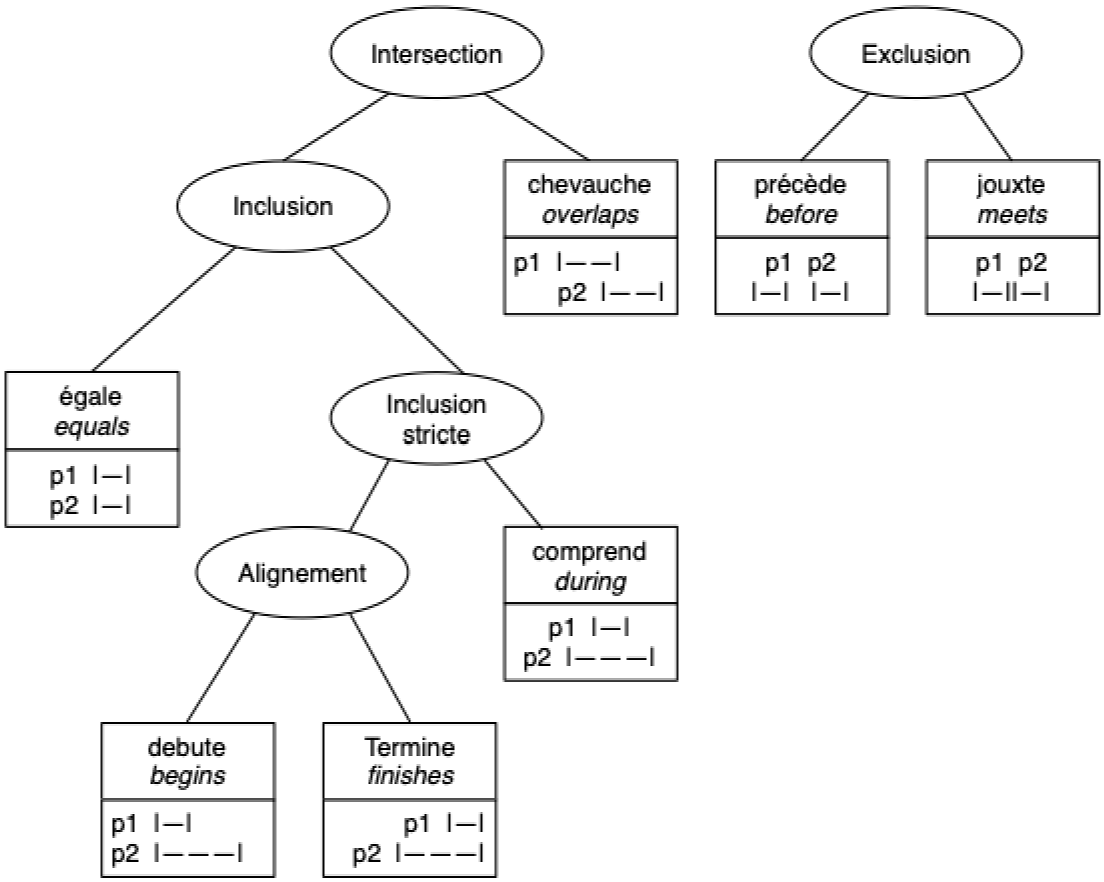

Figure 2. Illustration des opérateurs d'Allen

2.2.4. Opérateurs sur les ensembles d'intervalles

Équivalence

Deux ensembles d'intervalles A et B sont équivalents **si et seulement si** tout point d'un intervalle de A est compris dans un intervalle de B et réciproquement.

Corollaire : Deux ensembles d'intervalles A et B sont équivalents **si et seulement si** l'union des intervalles de A est égale à l'union des intervalles de B.

Expansion

Un ensemble d'intervalles A est l'expansion d'un ensemble d'intervalles B **si et seulement si** (A est équivalent à B) et (A ne contient que des intervalles unitaires).

Exemple : Expansion ({[2,4], [3,5]}) = {[2,2], [3,3], [4,4], [5,5]}

Compression

Un ensemble d'intervalles A et la compression d'un ensemble d'intervalles B **si et seulement si** (A est équivalent à B) et (a ne contient aucune paire d'intervalles pouvant être fusionnées).
Exemple : Compression ({[2,4], [3,5]}) = {[2,5]}

2.2.5. Opérateurs relationnels EXPAND et COLLAPSE

Attendu que *i* est un attribut de type intervalle de la relation *R*.

R EXPAND i

Calculer l'expansion de l'ensemble d'intervalles obtenu par la projection de R sur {i}.

R COLLAPSE i

Calculer la compression de l'ensemble d'intervalles obtenu par la projection de R sur {i}.

Exemple

R

a	i
g	[2,4]
h	[3,5]

R EXPAND i

i
[2,2]
[3,3]
[4,4]
[5,5]

R COLLAPSE i

i
[2,5]

2.2.6. Opérateurs relationnels UNFOLD et FOLD

UNFOLD R ON (i)

WITH (r1 := R GROUP {i} AS x, r2 := EXTEND r1 : {x := x EXPAND i}) : r2 UNGROUP x ;

FOLD R ON (i)

WITH (r1 := R GROUP {i} AS x, r2 := EXTEND r1 : {x := x COLLAPSE i}) : r2 UNGROUP x ;

Exemple

R		
idPatient	ville	i
P1	Montreal	[d02:d04]
P1	Montréal	[d03:d05]
P2	Québec	[d02:d05]
P2	Québec	[d04:d06]
P3	Ottawa	[d04:d06]
P3	Sherbrooke	[d03:d08]

r1 := R GROUP {i} AS x		
idPatient	ville	x
P1	Montreal	[d02:d04] [d03:d05]
P2	Québec	[d02:d05] [d04:d06]
P3	Ottawa	[d04:d06]
P3	Sherbrooke	[d03:d08]

r2 := EXTEND r1 : {x := x EXPAND i}		
idPatient	ville	x
P1	Montreal	[d02:d02] [d03:d03] [d04:d04] [d05:d05]
P2	Québec	[d02:d02] [d03:d03] [d04:d04] [d05:d05] [d06:d06]
P3	Ottawa	[d04:d04] [d05:d05] [d06:d06]
P3	Sherbrooke	[d03:d03] [d04:d04] [d05:d05] [d06:d06] [d07:d07] [d08:d08]

r := r2 UNGROUP x

idPatient	ville	i
P1	Montreal	[d02:d02]
P1	Montreal	[d03:d03]
P1	Montreal	[d04:d04]
P1	Montreal	[d05:d05]
P2	Québec	[d02:d02]
P2	Québec	[d03:d03]
P2	Québec	[d04:d04]
P2	Québec	[d05:d05]
P2	Québec	[d06:d06]
P3	Ottawa	[d04:d04]
P3	Ottawa	[d05:d05]
P3	Ottawa	[d06:d06]
P3	Sherbrooke	[d03:d03]
P3	Sherbrooke	[d04:d04]
P3	Sherbrooke	[d05:d05]
P3	Sherbrooke	[d06:d06]
P3	Sherbrooke	[d07:d07]
P3	Sherbrooke	[d08:d08]

Note

Les opérateurs FOLD et UNFOLD (noms choisis à l'origine par Lorentzos) sont respectivement dénotés PACK et UNPACK par de nombreux auteurs et dans le langage Tutorial D utilisés dans les livres de Date, Darwen… et Lorentzos [Date2004a, Date2014a].

2.3. Calendrier

Portée et couverture

Le modèle discret et le modèle calendaire n'ont pas forcément la même couverture. Il est donc en général utile de fournir les bornes de couvertures du modèle calendaire en termes du modèle discret dont l'étendue peut facilement être adaptée en faisant varier la représentation interne au besoin (typiquement, le nombre bits requis).

Granularité

Il n'est pas toujours possible de faire correspondre la granularité des deux, notamment lorsque celle du modèle calendaire n'est pas uniforme. Il est alors important de définir des fonctions de conversion qui respectent les conditions suivantes :

- $date(\alpha) = \varphi$
- $point(\varphi) = \alpha$
- $pa < pb \Rightarrow date(pa) \leq date(pb)$, mais $point(date(p)) = p$ n'est pas garanti.
- $da < db \Rightarrow point(da) \leq point(db)$, mais $date(point(d)) = d$ n'est pas garanti.

2.4. Notion de durée

Dans la plupart des modèles, la notion de durée s'applique à un intervalle (période), pas à un point (instant, chronon).

Dans un modèle discret, il est convenu de définir la durée comme

- $durée(i) = (card(i)\text{-}1) * \gamma$

Dans un modèle continu (et dans la plupart des modèles calendaires), elle est plutôt définie comme suit

- *durée(i) = (card(i)) * γ*

puisque le temps «s'écoule» entre deux instants distincts, alors qu'il «saute» d'un chronon à l'autre dans le modèle discret.

On comprend dès lors que lanotion de durée appartient au domaine d'application (comme toutes les mesures) et non à la théorie relationnelle (ni à ses modèles).

Références

[Date2014a]
Chris J. DATE, Hugh DARWEN, Nikos A. LORENTZOS ;
Time and Relational Theory: Temporal Databases in the Relational Model and SQL ;
Morgan Kaufmann, Waltham (MA, US), 2014 ;
ISBN 978-0-12-800631-3.

[Manthey2018a]
Rainer MANTHEY ;
Temporal Information Systems ;
Institut für Informatik, Universität Bonn, 2018 ;
https://pages.iai.uni-bonn.de/manthey_rainer/TIS2018 (consulté le 2024-04-19).

[Khnaisser2019a]
Christina KHNAISSER ;
Construction de modèles de données relationnels temporalisés guidée par les ontologies ;
Université de Paris en cotutelle avec l'Université de Sherbrooke, 2019 ;
http://www.theses.fr/s177273

Produit le 2025-09-18 13:05:41 -0400

Collectif francophone pour l'enseignement libre de l'informatique

