Calcul des événements (event calculus): Quelques références

Le papier fondateur:

(Kowalski & Sergot, A Logic-based Calculus of Events,New Generation Computing, Vol 4, 1986, p. 67–95).

Depuis, le domaine est toujours actif, et de nombreuses variantes existent.Ce cours ne donnera qu'un aperçu, et est principalement fondé sur la version de Shanahan:

(The Event Calculus in Classical Logic - Alternative Axiomatisations, par Rob Miller et Murray Shanahan, Electronic Transactions on A.I., vol.3-A, pp.77–105,1999, http://www.ida.liu.se/ext/epa/ej/etai/1999/A/index.html)

(The Event Calculus Explained, par Murray Shanahan,Artificial Intelligence Today in LNAI 1600 – Springer-Verlag, pp. 409–430,

1999,http://www-ics.ee.ic.ac.uk/ mpsha/pubs.html)

Calcul des événements: Introduction

L'objectif de départ (assez ambitieux):

Pouvoir traduire naturellement, et traiter(en un temps raisonnable) des situations complexes faisant intervenir le temps, dans un langage de programmation logique.

Pour éviter les problèmes du cadre et de ramification(cf cours sur calcul situationnel), la notion de base est celle d' ***événement***: Un événement ne décrit qu'une relation à un instant donné, et non pas toute la situation à cet instant.

Une *hypothèse du monde clos* est faite: on a assez d'informations pour conclure. Si on s'aperçoit que ce n'est pas le cas (suite à un ajout d'informations), les conclusions seront modifiées, sans réel conflit (raisonnement *non monotone*).

Il s'agit d'une logique *réifiée*: afin d'éviter les ordres supérieurs, des propriétés sont exprimées par des termes et non par des prédicats.Cela permet de demeurer en premier ordre (à part de petites incursions en second ordre).

Il n' y a en fait que quelques prédicats, avec un rôle très particulier.

Calcul des événements: le temps

Le temps est intégré dans le formalisme (ce qui n'est pas le cas du calcul situationnel), et le traitement du temps de la version originale est assez général: instants, intervalles (à la Allen) plusieurs actions possibles au même instant.

Toutefois (comme dans les principales variantes "temporelles" du calcul situationnel), le temps sera considéré comme linéaire: représenté par des nombres, entiers ou réels ou..., et on suppose qu'on sait les comparer.

Même si les versions actuelles sont moins ambitieuses, pour des raisons d'efficacité, le traitement du temps demeure un point fort du calcul des événements.

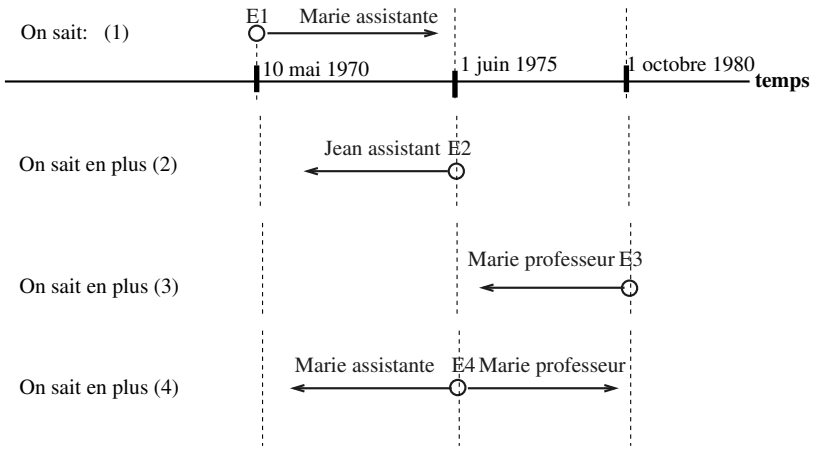
Calcul des événements *(versions initiales)*: exemple

- Marie a été recrutée comme assistante le 10 mai 1970.
- Jean a quitté sa fonction d'assistant le 1^{er} juin 1975.
- Marie a quitté sa fonction de professeur le 1^{er} octobre 1980.
- Marie a été promue d'assistante à professeur le 1^{er} juin 1975.

On apprend ces informations dans l'ordre donné. Cela se traduit ainsi:

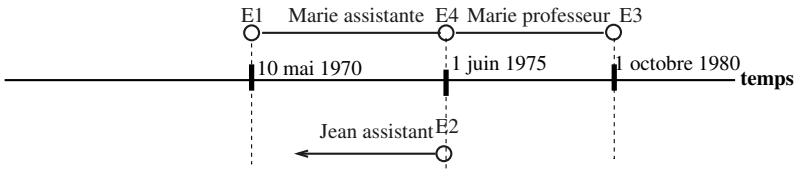
- $E1$ est un événement
  - ♦ dans lequel Marie a été recrutée comme assistante (elle n'était pas assistante ici avant, elle l'est après),
  - ♦ qui a pour date le 10 mai 1970.
- $E2$ ...
- $E3$ ...
- $E4$ ...

Calcul des événements: exemple (suite)



Calcul des événements: exemple (fin)

Il semble naturel de conclure après (4) que $E4$ termine la période entamée par l'événement $E1$:



La période initiée par l'événement $E1$ est égale à la période terminée par l'événement $E4$.

De même, il est naturel de conclure que $E3$ termine la période entamée par $E4$.

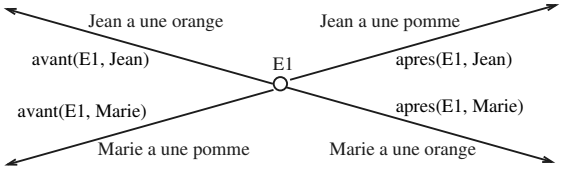
Calcul des événements: événements, “périodes”

Un *événement* initie (ou déclenche) et termine 0 ou plusieurs “*périodes*”.

Une “période” correspond à une relation

(représentée ici par un prédicat bien qu’en fait, cette relation sera réifiée et donc représentée par un terme, de plus, ce terme pourra mentionner aussi la propriété concernée – comme Jean a une pomme –, et la notion de période n’apparaîtra même plus explicitement dans la plupart des versions actuelles):

Exemple: Jean a échangé son orange contre la pomme de Marie



L’hypothèse du monde clos entraîne que
pendant la période $\boldsymbol{après(E1, Jean)}$, Jean n’a plus d’orange.

Quelques propriétés importantes (1)

Ce qui suit concerne les objectifs initiaux. quand le calcul des événements a été inventé. Ils sont loin d’être tous atteints par les versions actuelles les plus courantes, mais ils résument assez bien l’esprit du calcul des événements.

(1) Les mises à jour provoquées par de nouvelles informations sont additives:

Elles ajoutent des informations temporelles, mais n’en suppriment pas. Dans l’exemple, (4) ajoutait l’information $\boldsymbol{après(E1) = avant(E4)}$. Certaines conclusions peuvent se trouver effacées, mais le type d’information considéré comme fondamental ne peut pas être remis en cause.

Quelques propriétés importantes (2)

(2) Passé et futur sont traités de façon symétrique:

En particulier, cela permet de décrire une situation sans respecter l'ordre chronologique (cf l'exemple).

Dans l'exemple, on dit que Jean quitte un poste, sans avoir jamais appris explicitement qu'il avait ce poste. En style base de données, cela suppose qu'on a une contrainte d'intégrité disant qu'on ne peut pas quitter un poste sans l'avoir obtenu auparavant. Cela ne doit pas nous empêcher **d'apprendre** que quelqu'un quitte un poste avant d'avoir **appris** qu'il l'avait obtenu. Le traitement des informations choisi ici résout ce genre de problème de façon naturelle.

Toutefois, de nombreuses versions ne traitent pas passé et futur de façon totalement symétrique, même si, par exemple, beaucoup de versions permettent de décrire une situation sans respecter l'ordre chronologique.

Quelques propriétés importantes (3, 4)

(3) Le temps est numérique, mais c'est l'ordre qui compte.

On peut définir une relation $<$ de précédence sur les événements par:

$e1 < e2$ si $Temps(e1, t1)$, $Temps(e2, t2)$, et $t1$ se situe chronologiquement avant $t2$.

La relation est transitive et irréflexive (ordre strict).

(4) On peut avoir des événements concurrents:

sur l'exemple , $E4$ et $E2$.

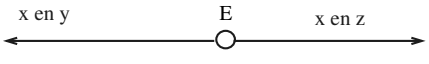
Quelques propriétés importantes (5)

(5) On permet aux événements d'avoir une durée.

Les périodes débutant ou finissant avec un événement ne contiennent pas celui-ci.
Cela ne signifie pas que ce sont des intervalles ouverts, mais qu'un événement correspond à une durée à l'intérieur de laquelle les instants ne sont pas discernables: les événements se produisent après les périodes qu'ils terminent et avant celles qu'ils commencent.

Soit l'événement "déplacer x de y vers z". On considérera 5 "sous-événements": *prendre*, *lever*, *déplacer*, *baisser*, *lâcher*. La période pour laquelle x est en y se termine quand x est levé, celle où x est en z débute quand x est baissé.

Au lieu de dessiner



deplacer x de y en z

on dessinera

deplacer x de y en z

Les fondateurs eux-mêmes sont revenus sur cet aspect (pour des raisons d'efficacité) mais il en reste toujours des traces dans des versions récentes.

Quelques propriétés importantes (6)

(6) Ce formalisme est plus une analyse formelle de concepts qu'un programme ou même une spécification de programme.

Cependant, comme il pouvait (pour la version originale) s'exprimer avec les clause de Horn augmentées de la négation par l'échec, il s'agissait d'une "analyse exécutable" qui, dans certaines circonstances, après des transformations adéquates préservant l'équivalence, s'exécute comme un programme PROLOG.

Quelques propriétés importantes (7)

(7) Par rapport au calcul situationnel:

Le traitement des informations incomplètes, et
des actions concurrentes
est beaucoup plus facile ici
(mais pas toujours si immédiat qu'on pourrait l'espérer: dans l'esprit des fondateurs, événement $\neq$ action, même si, pour Shanahan par exemple, événement $=$ action).

Le problème du cadre est apparemment plus facile à résoudre.

Il reste que le calcul situationnel est plus proche d'un vrai calcul, et qu'en fait les variantes du calcul des événements utilisées ne sont jamais aussi générales qu'exposé ici...

Quelques propriétés importantes (8)

(8) Le traitement du temps ressemble aux intervalles de Allen.

D'ailleurs, l'approche de Allen repose aussi principalement sur des événements associés aux périodes de temps qu'ils débutent et terminent. On évite ici la logique modale ce qui permet une meilleure expressivité(comme en situationnel, mais le temps "daté" est plus immédiat à utiliser ici).

Là encore, cet aspect demeure présent dans les versions actuelles,même s'il n'est pas aussi net qu'espéré au départ.

Rappelons que pour Allen, un intervalle n'est pas un ensemble de points ce qui permet de façon efficace de traiter de points et d'intervallessans invoquer des axiomes trop généraux (ensembles, réels).La version présentée ci-dessous, par exemple, permet de traiter de points et d'intervalles, mais de façon plus rustique.

Présentation d'une version du calcul des événements

On se fonde à partir de maintenant sur les papiers de Shanahan .

Un calcul des événements est un mécanisme logique qui permet d'inférer ce qui est vrai, sachant

- ce qui se passe, et quand, et
- ce que provoquent les événements.

La logique utilisée vient en général de la programmation logique.Pour les versions originales (et de nombreuses versions actuelles) c'est la logique de PROLOG.

Pour la version de Shanahan, c'est la logique du pemier ordre classique,avec un mécanisme de minimisation (la circonscription).

Cela peut encore être rendu par la programmation logique:celle des modèles stables et des "answer sets" (introduites par Gelfond et Lifschitz en 1988 et 1991) où l'ordre des clauses, et des atomes dans les clauses, n'intervient pas.

Stable sets et answer sets ("culture", hors de ce cours)

Disjonction dans les têtes de clause, et (answer sets) négation classique (en plus de par l'échec). Ordres des termes indifférents. Pour infos:

- le site de Smodels (Niemelä) qui donne une implémentation relativement efficace
  http://www.tcs.hut.fi/Software/smodels/.
- le papier de Baral et Gelfond, Logic programming and knowledge representation, Journal of logic programming, vol. 19,20, pp. 73–148, 1994 (http://www.public.asu.edu/ cbaral/papers/lp-kr.ps), ou
- le livre de Baral: Knowledge representation, reasoning and declarative problem solving with Answer sets, Cambridge University Press, 2003.
- Les deux papiers historiques, de Gelfond et Lifschitz:
  - The stable model semantics for logic programming, LP88: Int. Conf. and Symposium on Logic Programming, pp.1070–1080, 1988.
  - Classical Negation in Logic Programs and Disjunctive Databases, New generation computing, vol.9, pp.365–385, 1991.

Calcul des événements de Shanahan (2)

- Logique du premier ordre, avec trois (ou quatre) types d'objets:
  - $\mathcal{A}$ (variables $a, a_i, \cdots$): ***événements*** (ou actions);
  - $\mathcal{F}$ (variables $f, f_i, \cdots$): ***fluents*** (dont la valeur change avec le temps, propriété: "il pleut", quantité :"température", ...), les fluents sont donc réifiés;
  - $\mathcal{T}$ (variables $t, t_i, \cdots$): ***instants*** (des réels, on sait les comparer et effectuer les opérations habituelles);
  - $\mathcal{X}$ (variables $x, x_i, \cdots$): ***objets*** (particuliers au problème considéré);
- et des prédicats
  - qui expriment quels événements se produisent, et quand,
  - qui décrivent les effets de ces événements, ou
  - qui donnent des valeurs de fluents, selon le temps.

Calcul des événements de Shanahan (3)

Des prédicats de base:

formule	**sens**
$Initiates(\alpha, \varphi, \tau)$	L'événement α rend vrai le fluent φ à partir de l'instant τ
$Terminates(\alpha, \varphi, \tau)$	L'événement α rend faux le fluent φ à partir de l'instant τ
$\tau 1 < \tau 2$	L'instant $\tau 1$ précède l'instant $\tau 2$
$Happens(\alpha, \tau)$	L'événement α se produit à l'instant τ
$HoldsAt(\varphi, \tau)$	Le fluent φ est vrai à l'instant τ

Calcul des événements de Shanahan (4)

On utilise aussi les “abréviations” suivantes, que l'on nommera (EC1) et (EC2):

abréviation	définition	sens
$Clipped(t_1, f, t_2) =_{def}$	$\exists a, t\ [Happens(a, t) \wedge t_1 \leq t < t_2 \wedge Terminates(a, f, t)]$	le fluent f cesse d'être vrai entre les instants t_1 et t_2
$Declipped(t_1, f, t_2) =_{def}$	$\exists a, t\ [Happens(a, t) \wedge t_1 \leq t < t_2 \wedge Initiates(a, f, t)]$	le fluent f commence à être vrai entre les instants t_1 et t_2

Calcul des événements de Shanahan (5)

On introduit les deux axiomes de base suivants
(à ajouter aux définitions de $Clipped$ et $Declipped$):

$HoldsAt(f, t) \Leftarrow Happens(a, t_1) \wedge Initiates(a, f, t_1) \wedge t_1 < t \wedge \neg Clipped(t_1, f, t)$	(EC3)
$\neg HoldsAt(f, t) \Leftarrow Happens(a, t_1) \wedge Terminates(a, f, t_1) \wedge t_1 < t \wedge \neg Declipped(t_1, f, t)$	(EC4)

On a ainsi formalisé l'idée suivante: À l'instant t, un fluent f est

- vrai, s'il a été précédemment initié par l'occurrence d'un événement et s'il n'a pas été terminé entre temps par une autre occurrence d'un événement.
- faux, s'il a été précédemment terminé par l'occurrence d'un événement et s'il n'a pas été initié entre temps par une autre occurrence d'événement.

Calcul des événements de Shanahan (6)

Remarques:

- Avec les définitions et axiomes donnés,un fluent n'est pas vrai à l'instant τ_1 de l'événement qui l'a initié, mais il est vrai à l'instant τ_2 de l'événement qui le termine (intervalle $(\tau_1, \tau_2]$ ouvert à gauche et fermé à droite).
- Une autre conséquence plus subtile de ces définitions et axiomes est que juste après un instantτ où se produisent à la fois un événement qui initie un fluent φ et un événement qui termine φ,
  $HoldsAt(\varphi, \tau + \epsilon)$ et $\neg HoldsAt(\varphi, \tau + \epsilon)$ sont possibles.

  Ce "non déterminisme" pourrait être supprimé si, dans les définitions de $Clipped$ et $Declipped$, on remplaçait $t_1 \leq t < t_2$ par $t_1 < t < t_2$. Alors, initier et terminer simultanément un même fluent provoquerait une inconsistance:
  $HoldsAt(\varphi, \tau + \epsilon)$ et $\neg HoldsAt(\varphi, \tau + \epsilon)$ seraient établis.

Calcul des événements de Shanahan: exemple (1)

Un robot sort d'une pièce en ouvrant une porte qui peut être ouverte et fermée à l'aide d'une clé. On introduit trois fluents

- $Dedans$ (le robot est dans la pièce),
- $PossèdeClé$ (il a la clé), et
- $Fermé$ (la porte est fermée),

et trois événements

- $MetClé$ (le robot utilise la clé),
- $Franchit$ (il franchit la porte), et
- $PrendClé$ (il prend la clé).

On suppose l'unicité des noms (on a bien 6 constantes distinctes), et on sait que

- mettre la clé (si on l'a) ouvre ou ferme la porte selon son état,
- prendre la clé fait que le robot a la clé, et
- franchir la porte fait passer de dedans à dehors et réciproquement.

Événements de Shanahan: exemple (2)

On exprime ces données ainsi:

$Initiates(a,f,t) \Leftrightarrow [[a=PrendClé \wedge f=PossèdeClé]$ $\vee [a=MetClé \wedge f=Fermé$ $\wedge \neg HoldsAt(Fermé,t)$ $\wedge HoldsAt(PossèdeClé,t)]$ $\vee [a=Franchit \wedge f=Dedans$ $\wedge \neg HoldsAt(Fermé,t)$ $\wedge \neg HoldsAt(Dedans,t)]]$	(R1)
$Terminates(a,f,t) \Leftrightarrow [[a=Franchit \wedge f=Dedans$ $\wedge \neg HoldsAt(Fermé,t)$ $\wedge HoldsAt(Dedans,t)]$ $\vee [a=MetClé \wedge f=Fermé$ $\wedge HoldsAt(Fermé,t)$ $\wedge HoldsAt(PossèdeClé,t)]]$	(R2)

Événements de Shanahan: exemple (3)

- On suppose que la porte est fermée et le robot à l'intérieur au temps 0,
- et que le robot prend la clé, ouvre la porte et la franchit aux temps 2, 4 et 6 respectivement.

Ce que l'on traduit ainsi:

$HoldsAt(Fermé,0) \wedge HoldsAt(Dedans,0)$	(R3)
$Happens(a,t) \Leftrightarrow [[a=PrendClé \wedge t=2] \vee$ $[a=MetClé \wedge t=4] \vee$ $[a=Franchit \wedge t=6]]$	(R4)

Mais l'ensemble formé de (EC1)- -(EC4) et de (R1)–(R4) ne permet pas de démontrer que le robot n'est plus à l'intérieur au temps 8:

$$\neg HoldsAt(Dedans,8)??$$

- Rappel: On suppose la séparation des constantes
  (et les lois classiques sur les réels). Le problème ne vient donc pas de là.

Calcul des événements de Shanahan (“inertie”)

Le problème est qu’on n’a pas encore traduit la notion d’“inertie” ou de “persistance”.

Pour cela, on introduit deux nouveaux axiomes:

$\boldsymbol{HoldsAt(f,t)}$	$\Leftarrow$	$\boldsymbol{HoldsAt(f,t_1)} \wedge t_1 < t \wedge \neg \boldsymbol{Clipped(t_1,f,t)}$	(EC5)
$\neg \boldsymbol{HoldsAt(f,t)}$	$\Leftarrow$	$\neg \boldsymbol{HoldsAt(f,t_1)} \wedge t_1 < t \wedge \neg \boldsymbol{Declipped(t_1,f,t)}$	(EC6)

L’ensemble formé de (EC1)–(EC6) et de (R1)–(R4) permet de démontrer que le robot n’est plus à l’intérieur au temps 8: $\neg \boldsymbol{HoldsAt(Dedans,8)}$
(cf Rappel).

Calcul des événements de Shanahan (cadre 1)

Il reste toutefois un problème:

En fait, avec les axiomes donnés,
on ne pourra jamais conclure $\boldsymbol{HoldsAt(f,t)}$ ou $\neg \boldsymbol{HoldsAt(f,t)}$,
sauf en décrivant tout ce qui ne se produit pas
(ce qui est fait sur l’exemple en (R4)),
et tous les “non effets” des actions (ce qui est fait sur l’exemple: (R1), (R2)).

On est donc, en particulier, englué dans le “problème du cadre” (cf cours sur le calcul situationnel).

Calcul des événements de Shanahan (cadre 2)

On va d'abord supposer que les seuls effets des événements sont ceux que l'on peut déduire des données. Pour cela, on utilise un mécanisme de minimisation des informations positives. Shanahan a choisi l'un des plus connu, la circonscription (la version originale utilisait la négation par l'échec de Prolog, proche pour ce qui nous concerne ici de la complétion).

Soit donc (RA) l'ensemble des formules (dépendant du domaine considéré) décrivant les prédicats $Initiates$ et $Terminates$,
(RR) désignant les autres formules dépendant du domaine.

Au lieu de considérer $(EC1) + \cdots + (EC6) + (RA) + (RR)$,
on va considérer

$$(EC1) + \cdots + (EC6) + (RR) + Circ((RA) : (Initiates, Terminates)).$$

Événements de Shanahan: exemple (4)

Dans l'exemple, les données ne seraient plus traduites par les formes peu naturelles déjà complétées (R1) et (R2) , mais par les traductions plus immédiates, et "modulaires", (R1a–c) et (R2a,b):

$Initiates(PrendClé, PossèdeClé, t)$	(R1a)
$Initiates(MetClé, Fermé, t) \Leftarrow [\neg HoldsAt(Fermé, t) \wedge HoldsAt(PossèdeClé, t)]$	(R1b)
$Initiates(Franchit, Dedans, t) \Leftarrow [\neg HoldsAt(Fermé, t) \wedge \neg HoldsAt(Dedans, t)]$	(R1c)
$Terminates(Franchit, Dedans, t) \Leftarrow [\neg HoldsAt(Fermé, t) \wedge HoldsAt(Dedans, t)]$	(R2a)
$Terminates(MetClé, Fermé, t) \Leftarrow [HoldsAt(Fermé, t) \wedge HoldsAt(PossèdeClé, t)]$	(R2b)

Événements de Shanahan: exemple (5)

Dans l'exemple, les données seraient donc traduites par l'utilisateur à l'aide des formules (R1a–c) et (R2a,b) . En interne (notion de "couches de langages logiques"), ces données seraient formalisées ainsi:

$$(EC), Circ((R1a),(R1b),(R1c),(R2a),(R2b) : (Initiates, Terminates))$$

On a en effet ici:

$$Circ((R1a),(R1b),(R1c),(R2a),(R2b) : (Initiates, Terminates)) \equiv \{(R1),(R2)\}.$$

Il s'agit d'un cas où la circonscription des prédicats $Initiates$ et $Terminates$ équivaut à la complétion de ces prédicats.

Utiliser la circonscription est toutefois plus puissant et plus général qu'utiliser la complétion (cas de "récursivité",...).

Calcul des événements de Shanahan (cadre 3)

Il reste à traduire la supposition (naturelle) que ne se produisent que les événements dont on sait qu'ils se produisent. Pour cela, il faut encore utiliser un mécanisme de minimisation des informations positives.

Soit (RB) un ensemble de formules concernant
les prédicats $Happens$ et $<$ (ordre sur le temps).
Au lieu d'utiliser (RB) directement, on va effectuer la circonscription
$Circ((RB) : Happens)$.

Sur l'exemple, (R4) serait remplacé par $(R4a, b, c)$, plus naturel et plus modulaire ("elaboration tolerance"):

$Happens(PrendClé, 2)$ (R4a)
$Happens(MetClé, 4)$ (R4b)
$Happens(Franchit, 6)$ (R4c)

On a $Circ(\{(R4a),(R4b),(R4c)\} : Happens) \equiv (R4)$.
Sur cet exemple, cette circonscription équivaut elle aussi à une complétion.

Calcul des événements de Shanahan (cadre 4)

Ainsi, une description en calcul des événements se présentera ainsi:

- $(\boldsymbol{EC})$: les axiomes et définitions (EC1,2 , 3,4 , 5,6)
  (indépendants du domaine)
- $(\boldsymbol{RA})$: des formules en $\boldsymbol{Initiates}$ et $\boldsymbol{Terminates}$
  (sur l'exemple, (R1a)– (R2b)).
- $(\boldsymbol{RB})$: des formules en $\boldsymbol{Happens}$, et pour le temps, en $<, =$
  (sur l'exemple, (R4a–c)).
- Des formules $(\boldsymbol{RR})$, concernant $\boldsymbol{HoldsAt}$ (sur l'exemple, (R3)) ou indépendantes du temps (par exemple les formules d'unicité des noms).

Formellement, la théorie considérée "en interne" sera:

$$\{(\boldsymbol{EC}),\ (\boldsymbol{RR}),$$
$$\boldsymbol{Circ}((\boldsymbol{RA}):\boldsymbol{Initiates},\boldsymbol{Terminates}),\ \boldsymbol{Circ}((\boldsymbol{RB}):\boldsymbol{Happens})\}.$$

Événements de Shanahan: (cadre 5)

Ainsi, la méthode choisie ici, qui "résout le problème du cadre",

au lieu d'utiliser une seule circonscription compliquée (comme d'autres méthodes antérieures),

scinde les axiomes et données, et effectue des circonscriptions simples sur ces sous-théories.

C'est une caractéristique importante de la méthode, qui la rend plus compréhensible, et plus facile à calculer.

Malgré l'utilisation ici de la circonscription, il y en fait de fortes connexions entre cette méthode et celle du calcul situationnel de Toronto.

Calcul des événements de Shanahan: exemple (6)

Une illustration de la modularité et de la puissance théorique de la méthode:

Dans notre exemple, on apprend maintenant que le robot appuie sur un bouton de sonnette autour du moment où il met la clé. Il suffit d'ajouter à (R4a–c) la formule:

$\exists t_1\ [\boldsymbol{Happens}(\boldsymbol{Sonne}, t_1)\ \wedge\ \mathbf{2} < t_1 < \mathbf{6}]$	(R4d)

Alors, $\boldsymbol{Circ}(\{(\boldsymbol{R4a}), (\boldsymbol{R4b}), (\boldsymbol{R4c}), (\boldsymbol{R4d})\} : \boldsymbol{Happens})$ équivaut à:

$\exists t_1\ [\mathbf{2} < t_1 < \mathbf{6}\ \wedge\ [\boldsymbol{Happens}(a, t)\ \Leftrightarrow$ $((a = \boldsymbol{PrendClé}\ \wedge\ t = \mathbf{2})\ \vee$ $(a = \boldsymbol{MetClé}\ \wedge\ t = \mathbf{4})\ \vee$ $(a = \boldsymbol{Franchit}\ \wedge\ t = \mathbf{6})\ \vee$ $(a = \boldsymbol{Sonne}\ \wedge\ t = t_1))]]$	(R4')

Et on peut encore déduire $\neg \boldsymbol{HoldsAt}(\boldsymbol{Dedans}, \mathbf{8})$, comme avant.

(Dans ce cas, la complétion ne serait pas définie: formule non clausale.)

Calcul des événements de Shanahan (pile ou face)

Exemple de "non déterminisme" . Lancer une pièce produit pile, ou face. On lance une pièce à l'instant $\mathbf{2}$. On peut traduire cela ainsi (cas très particulier!):

$\boldsymbol{Initiates}(\boldsymbol{Lancer}, \boldsymbol{Face}, t)\ \wedge\ \boldsymbol{Terminates}(\boldsymbol{Lancer}, \boldsymbol{Face}, t)$	(C1)
$\boldsymbol{Happens}(\boldsymbol{Lancer}, \mathbf{2})$	(C2)

$\{(\boldsymbol{EC}), \boldsymbol{Circ}((\boldsymbol{C1}) : \boldsymbol{Initiates}, \boldsymbol{Terminates}),\ \boldsymbol{Circ}((\boldsymbol{C2}) : \boldsymbol{Happens})\}$
a quatre classes de modèles:

- une où $\boldsymbol{Face}$ est toujours vrai,
- une où $\boldsymbol{Face}$ n'est jamais vrai,
- une où $\boldsymbol{Face}$ est vrai jusqu'à l'instant 2, et faux ensuite,
- une où $\boldsymbol{Face}$ est faux jusqu'à l'instant 2, et vrai ensuite.

En effet, on a Clipped $(t_1, \boldsymbol{Face}, t_2)$, et Declipped $(t_1, \boldsymbol{Face}, t_2)$, ssi $t_1 \leq \mathbf{2} < t_2$. Donc (1) (EC3) et (EC4) sont ici trivialement satisfaits, et (2) jusqu'à $\mathbf{2}$ inclus, et après $\mathbf{2}$, $\boldsymbol{Face}$ garde sa valeur (principe d'"inertie").

Calcul des événements de Shanahan (pile ou face 2)

Voici une autre traduction possible du même exemple:

$Happens(Lancer,t) \Rightarrow [Happens(LancerF,t) \vee Happens(LancerP,t)]$	(C3)
$Initiates(LancerF, Face, t) \wedge Terminates(LancerP, Face, t)$	(C1')

On a introduit ici deux “événements indéterministes”
“lancer et obtenir pile”, et “lancer et obtenir face”.

La théorie considérée est maintenant $\{(EC), Circ((C1') : Initiates, Terminates), Circ(((C2), (C3)) : Happens)\}$.

Elle possède toujours les quatre classes de modèles déjà données, par rapport au fluent $Face$. (En effet, la circonscription de $Happens$ élimine les modèles où se produisent à la fois les deux événements $LancerF$ et $LancerP$, ce qui évite d’avoir à écrire explicitement cette condition).

(cf un corrigé d’examen antérieur où cette solution est détaillée.)

Calcul des événements de Shanahan (pile ou face 2bis)

Pour la planification par exemple, il resterait à introduire de nouveaux prédicats pour exprimer que seul l’événement $Lancer$ est réalisable (planifiable), alors que l’on ne peut pas exiger d’exécuter une “action” comme $LancerP$.

C’est très faisable, voici un début:

$Happens(a, t) \Leftrightarrow [Exécutable(a, t) \vee Incontrôlable(a, t)]$.

...

Calcul des événements de Shanahan
Actions concurrentes

Il est facile de traiter d'actions concurrentes, car différents $Happens$ peuvent se référer au même instant.

Il convient toutefois de préciser les éventuels effets combinés de plusieurs actions, et les annulations d'effets provoquées par d'autres actions simultanées.

Exemple: Un bol à deux anses rempli d'eau, soulever l'anse gauche (respectivement droite) va renverser de l'eau, mais soulever les deux anses ensemble va soulever le bol sans renverser d'eau. Voici une traduction possible:

Calcul des événements de Shanahan
Actions concurrentes (2)

On va exprimer des "préconditions" pour traduire combinaisons et annulations:

$Initiates(LeverG, Renversé, t)$	$\Leftarrow$	$\neg Happens(LeverD, t)$	
$Initiates(LeverD, Renversé, t)$	$\Leftarrow$	$\neg Happens(LeverG, t)$	(B1)
$Initiates(LeverD, Soulevé, t)$	$\Leftarrow$	$Happens(LeverG, t)$	

Supposons que l'on soit en présence de la situation et des événements suivants:

$\neg HoldsAt(Renversé, 0)$ (B2)

$Happens(LeverG, 2) \wedge Happens(LeverD, 2)$ (B3)

La théorie correspondante $\{(EC),\ (B2),\ Circ((B1) : Initiates, Terminates), Circ((B3) : Happens)\}$, permet par exemple de conclure:

$\neg HoldsAt(Renversé, 4)$ et $HoldsAt(Soulevé, 4)$.

Calcul des événements de Shanahan, variantes
1. Temps positif

Dans ce cas là (existence d'un instant initial), on introduit souvent deux prédicats unaires (portant sur les fluents, en fait il s'agit d' "abréviations" là encore) $InitiallyP$ et $InitiallyN$, satisfaisant:

$InitiallyP(\varphi) \equiv HoldsAt(\varphi, 0)$ et
$InitiallyN(\varphi) \equiv \neg HoldsAt(\varphi, 0)$ **(EC7a)**

On peut aussi simplifier l'axiomatisation en remplaçant (EC5) et (EC6) par:

$HoldsAt(f,t) \Leftarrow [InitiallyP(f) \wedge \neg Clipped(0,f,t)]$	(EC5a)
$\neg HoldsAt(f,t) \Leftarrow [InitiallyN(f) \wedge \neg Declipped(0,f,t)]$	(EC6a)
$InitiallyP(f) \vee InitiallyN(f)$	(EC8a)

Le gros avantage est que la traduction en programmation logique peut être plus directe, car on n'a pas les "boucles" de (EC5) et (EC6). De plus, (EC5a) et (EC6a) sont conséquences de (EC5) et (EC6) (avec (EC7a)). Par contre, on ne traduit pas ici la notion de persistance générale traduite par (EC5) et (EC6), même si on retrouve quand même une certaine notion de persistance.

Calcul des événements de Shanahan, variantes
2. Version déterministe

On a déjà évoqué cette variante. Dans le cas général, c'est assez compliqué, il faut garder $Clipped$ et $Declipped$ déjà définis, et ***ajouter***:

abréviation		**définition**	
$Clipped'(t_1, f, t_2)$	$=_{def}$	$\exists a, t\ [Happens(a,t) \wedge t_1 < t < t_2 \wedge Terminates(a,f,t)]$	(EC1a)
$Declipped'(t_1, f, t_2)$	$=_{def}$	$\exists a, t\ [Happens(a,t) \wedge t_1 < t < t_2 \wedge Initiates(a,f,t)]$	(EC2a)

Si on se place dans le cadre du temps positif décrit précédemment avec les axiomes (EC5a) et (EC6a) , c'est plus simple puisqu'il suffit de ***remplacer*** les définitions de $Clipped$ et $Declipped$ par les nouvelles, données dans cette page.

Calcul des événements de Shanahan, variantes
2bis. Version déterministe (suite)

Il faut supposer que ne se produit au temps $\mathbf{0}$ aucune action qui termine (respectivement initie) un fluent initialement positif (respectivement négatif).

Avec cette axiomatisation , $\boldsymbol{Initiates}(\alpha, \varphi, \tau)$ signifie:
φ est vrai immédiatement après une occurrence de α en τ.

Avec l' axiomatisation "classique" (la première version présentée, dite "non déterministe"), $\boldsymbol{Initiates}(\alpha, \varphi, \tau)$ signifie seulement:
une occurrence de α au temps τ a une influence qui tend à initier φ, cette influence peut cependant être contrecarrée par une influence (simultanée et "terminante") contraire (cf exemple).

Calcul des événements de Shanahan, variantes
3. Préconditions d'actions

On a déjà exprimé des pré-conditions pour des effets d'actions (R1b–R2b) (***pré-conditions de fluents***). Il est parfois utile d'exprimer des ***préconditions d'actions*** qui donnent des conditions sans lesquelles une action ne peut pas être exécutée du tout. Une façon de faire consiste à introduire un nouveau symbole de prédicat, Impossible$\subseteq \mathcal{A} \times \mathcal{T}$. Par exemple, on ajouterait à (R1a–R2b) :

$\boldsymbol{Impossible}(\boldsymbol{PrendClé}, t) \Leftarrow \neg \boldsymbol{HoldsAt}(\boldsymbol{MuniDePince}, t)$	(R5)
$\boldsymbol{Impossible}(\boldsymbol{Franchit}, t) \Leftarrow \boldsymbol{HoldsAt}(\boldsymbol{Fermé}, t)$	(R6)

Le problème de qualification consiste en partie à exprimer de façon succinte et modulaire l'idée que "les actions sont en général possibles". Une façon de procéder utilise $\boldsymbol{Circ}((\boldsymbol{R5}, \boldsymbol{R6}) : \boldsymbol{Impossible})$, ce qui donne

$$\boldsymbol{Impossible}(a, t) \Leftrightarrow [[a = \boldsymbol{PrendClé} \wedge \neg \boldsymbol{HoldsAt}(\boldsymbol{MuniDePince}, t)] \vee [a = \boldsymbol{Franchit} \wedge \boldsymbol{HoldsAt}(\boldsymbol{Fermé}, t)]].$$

Calcul des événements de Shanahan, variantes
3. Préconditions d'actions (2)

En ***planification***, qui invoque un raisonnement (hypothétique) sur des événements futurs, $Impossible(\alpha, \tau)$ signifie "il est impossible de prédire les effets de l'action α au temps τ", et sa négation est similaire au $Poss(\alpha, \sigma)$ du calcul situationnel.

On peut alors bloquer toute modification après une (tentative) d'action impossible, ce qui peut se faire en modifiant (EC1–EC2) ainsi:

$Clipped(t_1, f, t_2) =_{def}$ $\exists a, t\ [Happens(a, t) \wedge t_1 \leq t < t_2 \wedge Terminates(a, f, t)]$ $\vee\ \exists a, t\ [Happens(a, t) \wedge t < t_2 \wedge Impossible(a, t)]$	$(EC1b)$
$Declipped(t_1, f, t_2) =_{def}$ $\exists a, t\ [Happens(a, t) \wedge t_1 \leq t < t_2 \wedge Initiates(a, f, t)]$ $\vee\ \exists a, t\ [Happens(a, t) \wedge t < t_2 \wedge Impossible(a, t)]$	$(EC2b)$

Calcul des événements de Shanahan
(conclusion)

L'approche présentée ici est plus difficile à implémenter que les versions initiales, au moins en théorie. En pratique, des méthodes comme la programmation par "Answer sets" permettent de l'implémenter.

De plus les objectifs initiaux étaient trop ambitieux pour être réellement implémentables dans leurs totalité, et les version initatiales implémentées étaient trop intimement liées à certaines particularités de Prolog pour être facilement compréhensibles.

Le fait que la version du calcul situationnel de Toronto soit réellement implémentée et pas la version du calcul des événements de Shanahan tient surtout à la taille et aux objectifs des équipes respectives et ne semble pas significatif.

Par rapport au calcul situationnel, le temps est plus facile à traiter, et les actions simultanées aussi. En réalité, ces différences ne sont pas toujours aussi nettes mais, en dépit des points communs, il n'est pas certain que les diverses versions de l'un et de l'autre formalisme puissent converger.

Les problèmes du cadre et de la qualification demeurent des problèmes difficiles, quel que soit le formalisme choisi.