NSCによるZeno問題およびデッドロックの推論について

中村勝則† 房岡璋‡
立命館大学 理工学部情報学科† ‡

1 はじめに

Situation Calculusは時間推論のための一階述語論理の枠組みであるが，これを超準解析を用いて非標準に拡張したNSC（Nonstandard Situation Calculus）を提案し，これを用いてZeno問題とデッドロックを取り扱う試みについて報告する．

2 NSC(Nonstandard Situation Calculus)

標準の Situation Calculus (SSC)の Situation と流量を非標準に拡張した NSC[1]を提唱する．NSC では超準解析の枠組みを用いて連続的変化をする流量も離散的に扱う．NSC を用いることで，物理現象を解析学的な扱いをせずに初等的な演算だけで扱うことができる．（図 1，2）

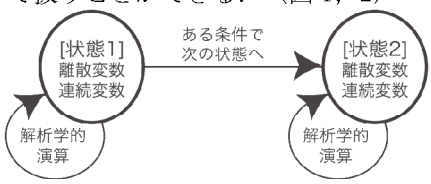


図 1. SSC

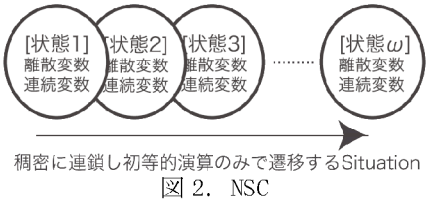


図 2. NSC

3 Zeno問題の扱い

Hybrid System において，有限時間内に無限回の状態遷移が発生してしまう，Zeno 問題[2]がある．Zeno 現象を起こすシステムの状態を SSC で推論すると推論は停止しないが，NSC を用いることでこの問題は解決する．Zeno 現象を起こす非常に単純なシステムの例として，バウンドするボールを考える．ある高さ h からボールを地面に落とすとボールはバウンドを繰り返し，有限時間内にボールは停止する．（反射係数は k とする．）しかし，ボールは停止するまでに無限回のバウンドをする．このシステムを NSC で記述すると次のようになる．

ある Situation S における時刻，速度，位置，バウンド回数をそれぞれ $t(S)$, $v(S)$, $x(S)$, $n(S)$（全て非標準の実数：超実数）とする．アクション A を Situation S に施して得られる Situation を $do(A,S)$ とする．Situation S におけるボールがバウンドせずに落下する微小アクションを ∂a として次のように定義する．$t(do(\partial a,S))=t(S)+\varepsilon$, $v(do(\partial a,S))=v(S)+g\varepsilon$, $x(do(\partial a,S))=x(S)+v(S)\varepsilon$, $n(do(\partial a,S))=n(S)$ （$\varepsilon\in{}^*R$ ：${}^*R$ は超実数で，$\forall x\in R[\varepsilon<x]$）．また Situation S におけるボールがバウンドするアクションを sw とし，次のように定義する．

$t(do(sw,S))=t(S)$， $v(do(sw,S))=kv(S)$，（$k\in R$，$-1<k<0$），$x(do(sw,S))=x(S)$，$n(do(sw,S))=n(S)+1$．このシステムの状態推移の制御規則を次のように定義する．

$st(x(S))=0\supset S'=do(sw,S)$，$st(x(S))\neq 0\supset S'=do(\partial a,S)$

（$st(X)$ は超実数 X の標準部分を与える関数）．

またボールが地面に到達するまでにアクション ∂a を非標準の回数繰り返すが，これをアクション a として表現すると，次のようになる．

$$do(a,S)=iterate\left(\partial a,\frac{-v(S)+\sqrt{v(S)^2-2gx(S)}}{g\varepsilon},S\right)$$

ここで，$iterate(A,N,S)$ は，アクション A を Situation S に N 回施して得られる状態を与えるものとする．

これらを知識として推論システムに与えて，初期状態 S_0 から n 回バウンドした直後のボールの状態 S_n をシステムに推論させると，次のようになる． $S_n=iterate([a;sw],n,S_0)$ （$[a;sw]$ は a，sw の連続アクション）SSC の場合は，バウンド回数はいくらでも大きくとることができ，推論は終了しないが，NSC では非標準の実数 ω を用いることで無限回のバウンドを繰り返した後の状態 S_ω を得ることができる． $S_\omega=iterate([a;sw],\omega,S_0)$，$\forall n\in R\exists\omega\in{}^*R[n<\omega]$

ボールが停止する時刻を実際に求めるには，$t(S_\omega)$ を形式的に求めた後，関数 st により標準の実数として時刻を計算する．関数 st の実装には次に示すような ${}^*R\Leftrightarrow R$ の変換規則が必要となる．

Reasoning about Zeno and Dead Lock on NSC
†Katsunori Nakamura,Graduate School of Ritsumeikan University
‡Prof.Akira Fusaoka, Dept. of Computer Science, Ritsumeikan University

変換規則の例

$$^o(x+y)={}^ox+{}^oy,\quad {}^o(xy)={}^ox{}^oy,\ {}^ox\neq 0\supset {}^o\left(\frac{1}{x}\right)=\frac{1}{{}^ox},$$

$$x\geq y\supset {}^ox\geq {}^oy,\ {}^ox>{}^oy\supset x>y,\quad \frac{1}{\varepsilon}=\omega,\quad {}^o\left(\frac{1}{\omega}\right)=0,$$

$$\omega\in Inf\supset\left(1+\frac{x}{\omega}\right)^{\omega}\approx e^x\wedge {}^o\left\{\left(1+\frac{x}{\omega}\right)^{\omega}\right\}=e^x$$

3．1　ZenoPoint

このシステムの記述において，アクションaの実行にかかる時間をτとすると，n回目のバウンドから次のバウンドまでに要する時間τ_nは次の様になる．

$\tau_0=\sqrt{\frac{2h}{g}},\tau_n=2\sqrt{\frac{2h}{g}}k^n$　これよりシステムの時刻は，$t_n=\sqrt{\frac{2h}{g}}\left(1+2\sum_{i=1}^{n}k^i\right)$となり，無限回のバウンドを繰り返しながらシステム時刻は有限の時刻t_ωに収束するのがわかる．（下記）

$$st(t_\omega)=\sqrt{\frac{2h}{g}}\left(1+\frac{2k}{1-k}\right)$$

時刻t_ωの状態をZenoPointと呼ぶ．

推論システムに，状態遷移の一般化を行う機能を組み込むことで，状態遷移がZenoとなるかどうかの判定ができるようになり，システム制御プランの自動生成や検証に応用できる．

このシステムではZenoPointに到達した時点で状態遷移は終了するが，古典的な「アキレスと亀」の問題の場合は，ZenoPoint以降も状態遷移が行われる．推論システムにおいて，ZenoPoint以降の状態遷移を推論するにはZenoPoint近傍の無限に小さい領域において非標準の値を用いた制御規則を適用し，ZenoPoint以降の状態を生成する必要がある．これについては次の章で説明する．

4　デッドロックの解消

水平に動くアームが2つあり，それらがぶつからないようにどちらかがパッケージをつかむシステムを考える．（図3）

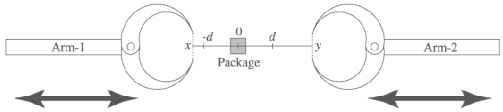


図3．デッドロックを起こすアーム

パッケージの位置を0，左のアームの位置をx，右のアームの位置をy，とする．2つのアームは同時にパッケージに近付きパッケージをつかもうとする．

r0: $|x(S)|>d\wedge|y(S)|>d\supset\ st=A\wedge$
$x(do(\partial a,S))=x(S)+\varepsilon\wedge y(do(\partial a,S))=y(S)-\varepsilon$

2つのアームが衝突を避けるために，Critical Regionとしてdを設け，次の2つのルールを設定する．

r1: $|x(S)|>d\wedge|y(S)|\leq d\supset st=B\wedge$,
$x(do(\partial b,S))=x(S)\wedge y(do(\partial b,S))=y(S)-\varepsilon$

r2: $|x(S)|\leq d\wedge|y(S)|>d\supset st=C\wedge$
$x(do(\partial c,S))=x(S)+\varepsilon\wedge y(do(\partial c,S))=y(S)$

2つのアームが同時にCritical Regionに入った場合のために次のルールを設定する．

r3: $|x(S)|\leq d\wedge|y(S)|\leq d\supset$

$$\begin{bmatrix}|x(S)|-|y(S)|>\varepsilon\supset st=D\wedge\\ \quad x(do(\partial d,S))=x(S)-\varepsilon\wedge y(do(\partial d,S))=y(S)-\varepsilon,\\ |y(S)|-|x(S)|>\varepsilon\supset st=E\wedge\\ \quad x(do(\partial e,S))=x(S)+\varepsilon\wedge y(do(\partial e,S))=y(S)+\varepsilon,\\ \left(|x(S)|-|y(S)|\right)\leq\varepsilon\supset st=F\wedge\\ \quad x(do(\partial f,S))=x(S)-\varepsilon\wedge y(do(\partial f,S))=y(S)+\varepsilon\end{bmatrix}$$

（εは無限小の超実数）

$\left(|x(S)|-|y(S)|\right)\leq\varepsilon$のときに，左のアームは$-d$の近傍で，右のアームは$d$の近傍でデッドロックを起こすが，ルールr3の最後の部分で，どちらか一方のアームのみを取り下げるような規則に換えることでデッドロックは防止できる．

上記ルールr3は無限小の超実数を用いた制御であり，NSCでは無限小の制御によりZeno問題やデッロドックの問題を解決することができる．

5　おわりに

NSCを用いた推論の有効性を検証するために，現在，市販の数式処理システムの上に，NSC推論のシミュレータを構築しつつある．NSCシミュレータを用いて，システムの自動検証や，システム制御プランの自動生成などについて研究してゆきたいと考えている．

参考文献

[1] 房岡,平塚 “Situation Calculusの非標準モデルについて”,人工知能学会論文誌, 17巻5号D,2002

[2] Zhang,et al “Dynamical Systems Revisited:Hybrid System with Zeno Executions”,HSCC 2000, LNGS 1790,pp451-464,2000