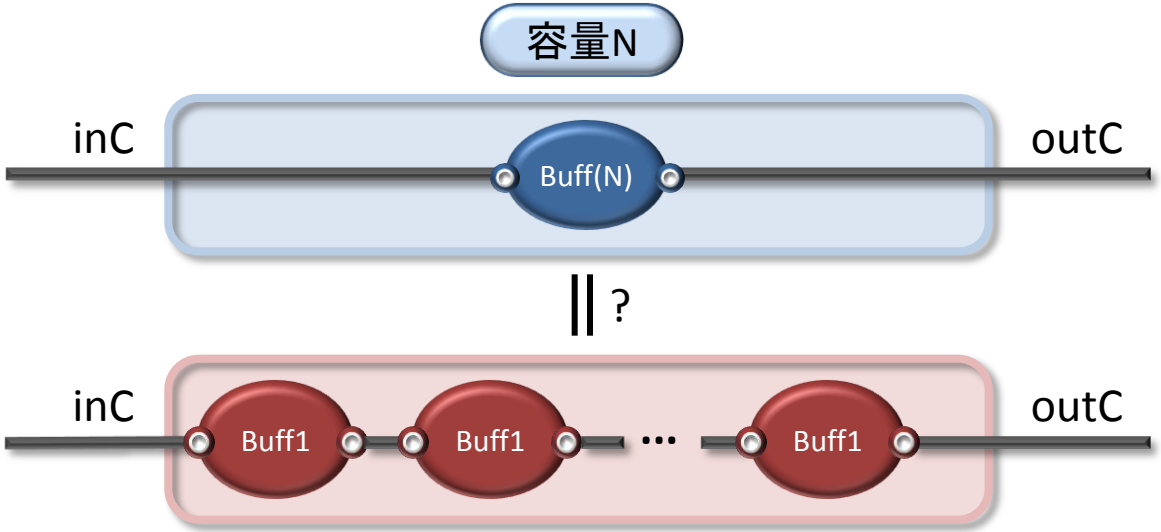
プロセス代数CSPの定理証明器 CSP-Proverの紹介

磯部祥尚 （産業技術総合研究所）

第2回CSP研究会 2009.06.30

講演内容

- CSPによる並行システムのモデル化の例
- CSPによる検証の例
  - モデル検査器（例: **FDR**）による検証とは？
  - 定理証明器（例: **CSP-Prover**）による検証とは？
- CSPによるスケーラブルな検証の例
  - モデル化
  - N個のプロセス
- まとめ

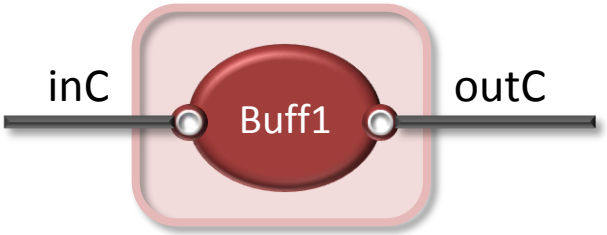


CSPによるモデル化

- 逐次動作のモデル化の例
- 並行動作のモデル化の例

CSPによるモデル化の例

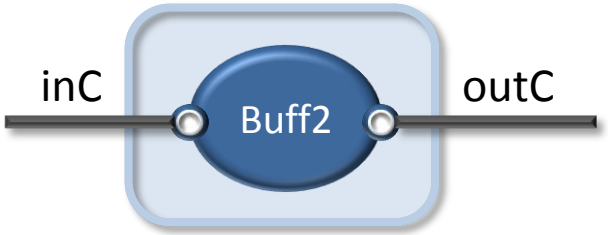
■ Buff1 （容量1のバッファ）



イベント実行

Buff1 = inC → Buff1’
Buff1’ = outC → Buff1

■ Buff2 （容量2のバッファ）



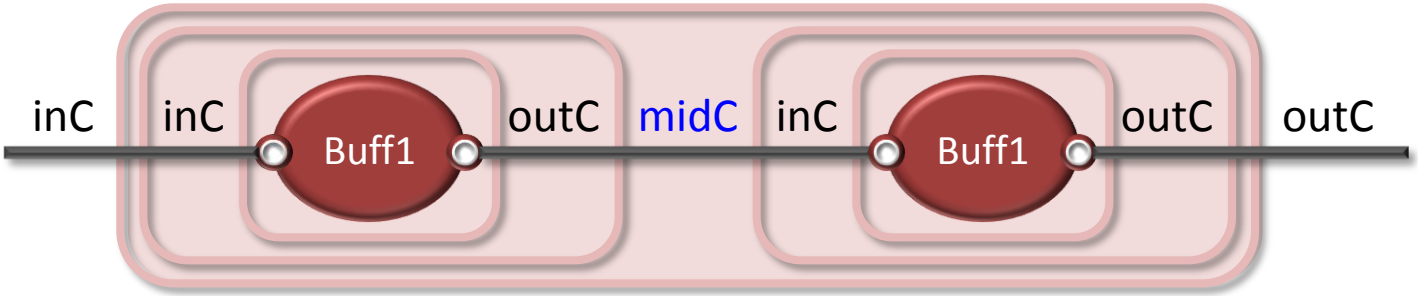
CSPモデル

Buff2 = inC → Buff2’
Buff2’ = inC → Buff2’’ □
outC → Buff2
Buff2’’ = outC → Buff2’

外部選択

CSPによるモデル化の例

- LinkBuff2 （2個の容量1のバッファを結合した並行システム）



並行合成　名前変更

```
LinkBuff2 = Link(Buff1, Buff1)
Link(P,Q) = (P [[outC←midC]] [|{|midC|}|] Q [[inC←midC]]) ＼{|midC|}
```

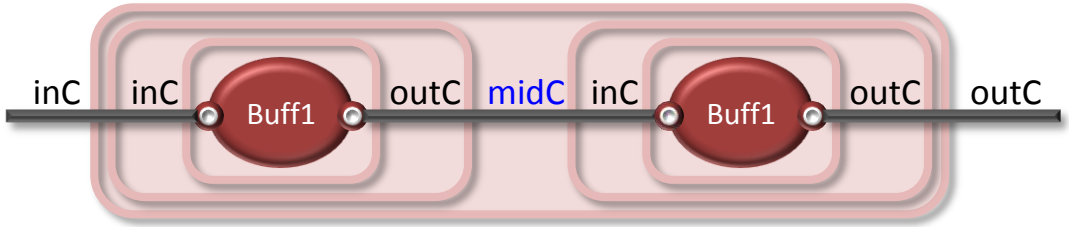
隠蔽

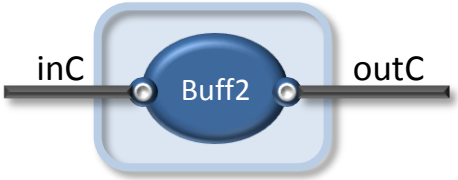
CSPによる検証

- 操作的意味論による検証の例
- 公理的意味論による検証の例

CSPによる検証の例 （操作的意味論に基づく検証）

■ LinkBuff2 と Buff2 の等価性





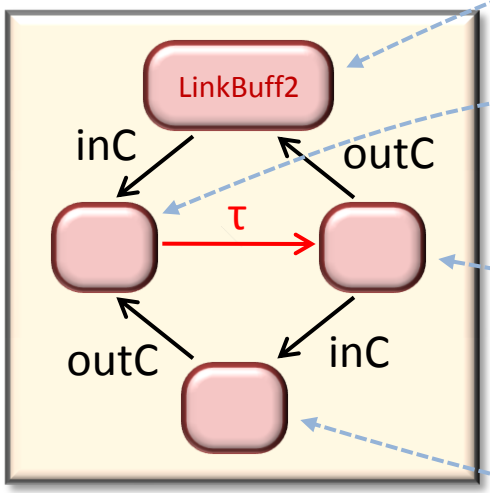
LinkBuff2 = Link(Buff1, Buff1)
Link(P,Q) = (P [[outC←midC]] [|{|midC|}|] Q [[inC←midC]]) ＼{|midC|}

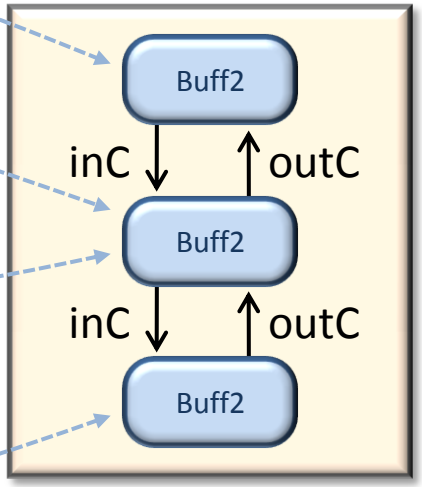
Bufff2 = inC → Buff2'
Buff2' = inC → Buff2'' □ outC → Buff1
Buff2'' = outC → Buff2'

CSPモデル

状態遷移図

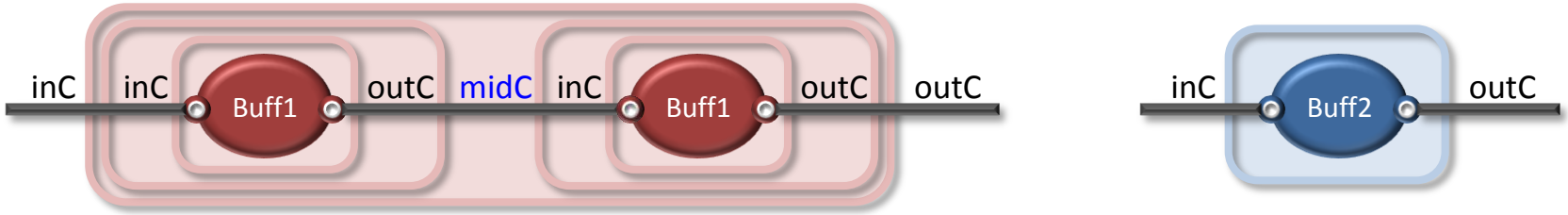
モデル検査器
FDR

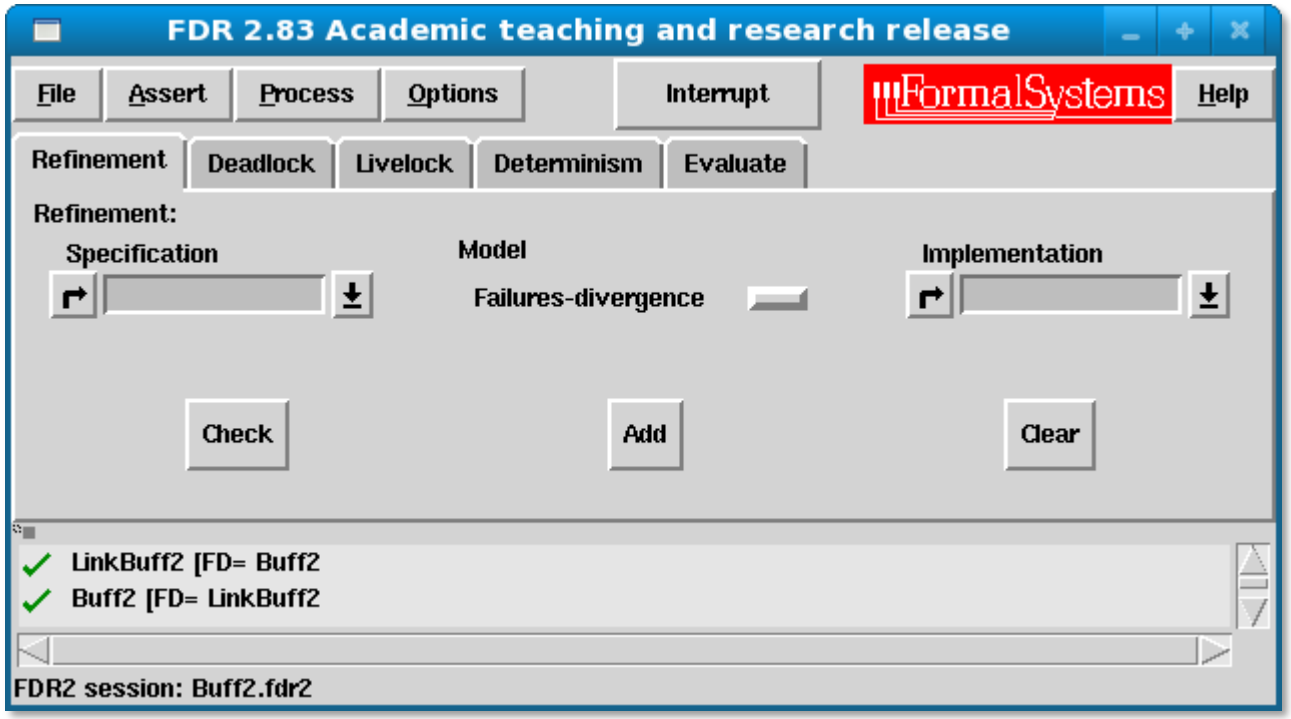




CSPによる検証の例 （操作的意味論に基づく検証）

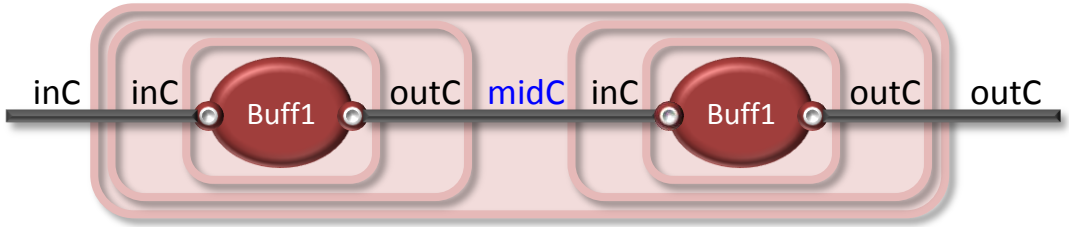
■ モデル検査器FDRによるLinkBuff2 = Buff2 の判定

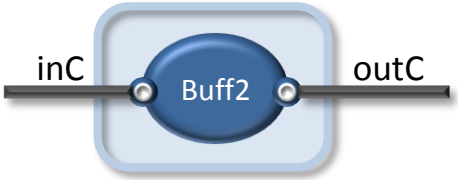




CSPによる検証の例 （公理的意味論に基づく検証）

■ LinkBuff2 と Buff2 の等価性





LinkBuff2 = Link(Buff1, Buff1)
Link(P,Q) = (P [[outC←midC]] [|{|midC|}|] Q [[inC←midC]]) ＼{|midC|}

CSPモデル

Bufff2 = inC → Buff2’
Buff2’ = inC → Buff2’’ □ outC → Buff1
Buff2’’ = outC → Buff2’

LinkBuff2 = (Buff1[[outC←midC]] [|{|midC|}|] Buff1[[inC←midC]]) ＼{|midC|}
= inC→ (Buff1’[[outC←midC]] [|{|midC|}|] Buff1[[inC←midC]]) ＼{|midC|}
= inC→ Buff2’
= Buff2

By CSP規則

参考 $(a + b)(c + a) = a(c + a) + b(c + a)$
$= ac + a^2 + bc + ba$
$= a^2 + ab + ac + bc$
$= a^2 + a(b + c) + bc$

CSPによる検証の例 （公理的意味論に基づく検証）

■ LinkBuff2 と Buff2 の等価性

定理証明器
CSP-Prover

$$
\begin{aligned}
&(Buff1'[[outC \leftarrow midC]] \ [|\{|midC|\}|] \ Buff1[[inC \leftarrow midC]]) \setminus \{|midC|\} \\
&= (Buff1[[outC \leftarrow midC]] \ [|\{|midC|\}|] \ Buff1'[[inC \leftarrow midC]]) \setminus \{|midC|\} \\
&= inC \rightarrow (Buff1'[[outC \leftarrow midC]] \ [|\{|midC|\}|] \ Buff1'[[inC \leftarrow midC]]) \setminus \{|midC|\} \ \Box \\
&\quad outC \rightarrow (Buff1[[outC \leftarrow midC]] \ [|\{|midC|\}|] \ Buff1[[inC \leftarrow midC]]) \setminus \{|midC|\} \\
&= inC \rightarrow Buff2'' \ \Box \ outC \rightarrow Buff2 \\
&= Buff2'
\end{aligned}
$$

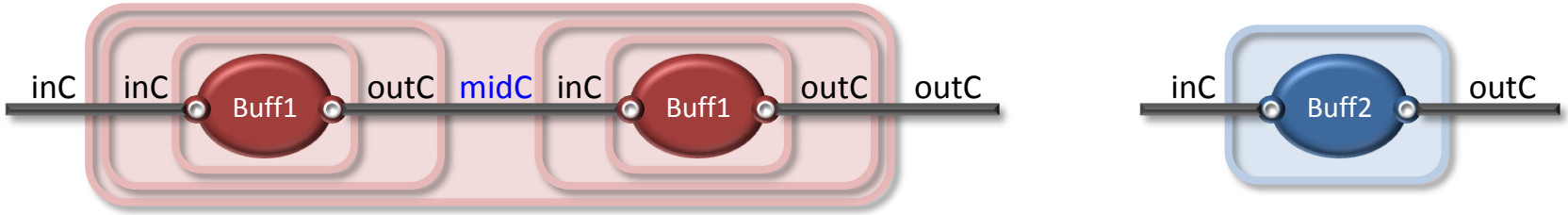
$$
\begin{aligned}
LinkBuff2 &= (Buff1[[outC \leftarrow midC]] \ [|\{|midC|\}|] \ Buff1[[inC \leftarrow midC]]) \setminus \{|midC|\} \\
&= inC \rightarrow (Buff1'[[outC \leftarrow midC]] \ [|\{|midC|\}|] \ Buff1[[inC \leftarrow midC]]) \setminus \{|midC|\} \\
&= inC \rightarrow Buff2' \\
&= Buff2
\end{aligned}
$$

By CSP規則

$$
\begin{aligned}
&(Buff1'[[outC \leftarrow midC]] \ [|\{|midC|\}|] \ Buff1'[[inC \leftarrow midC]]) \setminus \{|midC|\} \\
&= outC \rightarrow (Buff1'[[outC \leftarrow midC]] \ [|\{|midC|\}|] \ Buff1[[inC \leftarrow midC]]) \setminus \{|midC|\} \\
&= outC \rightarrow Buff2' \\
&= Buff2''
\end{aligned}
$$

CSPによる検証の例 （公理的意味論に基づく検証）

■ 定理証明器CSP-ProverによるLinkBuff2 = Buff2 の証明



Buff2.thy - emacs@localhost.localdomain

File Edit Options Buffers Tools Isabelle Proof-General Help

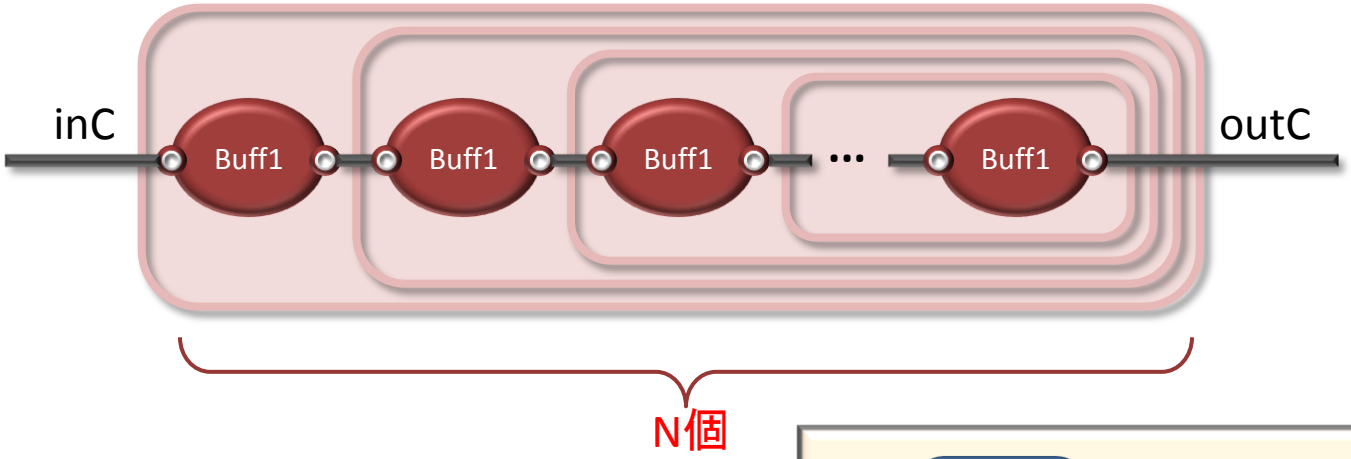
```
theorem Buff2_eq_LinkBuff2:
   "$Buff2 =F LinkBuff2"

(* by finxed point induction *)
apply (rule cspF_fp_induct_left[of _ "Buff2_to_LinkBuff2"])
apply (simp+, induct_tac p, auto, cspF_unwind+)

(* by sequentialsing *)
apply (cspF_hsf | auto)+
done
--:--  Buff2.thy       77% L99     (Isar script Scripting)---------------
proof (prove): step 3

goal:
No subgoals!
-u:--  *goals*         Bot L3      (proofstate)-------------------------
```

スケーラブルな並行システムの検証

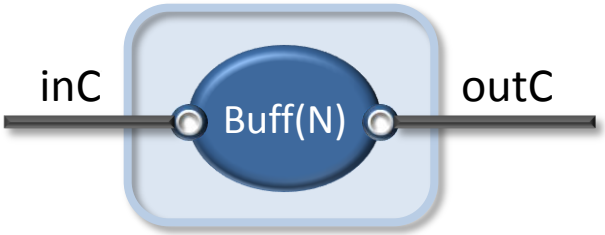
- モデル化の例
- 帰納法による検証の例

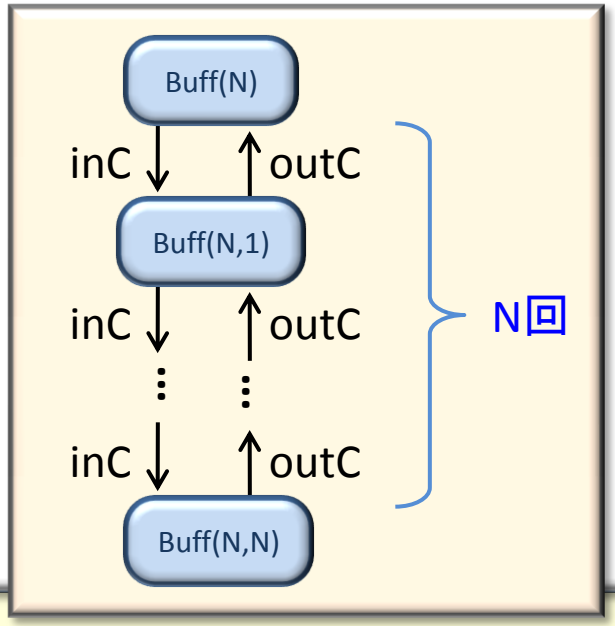
スケーラブルな並行システムの例

■ LinkBuff(N)　（N個の容量1のバッファを結合した並行システム）



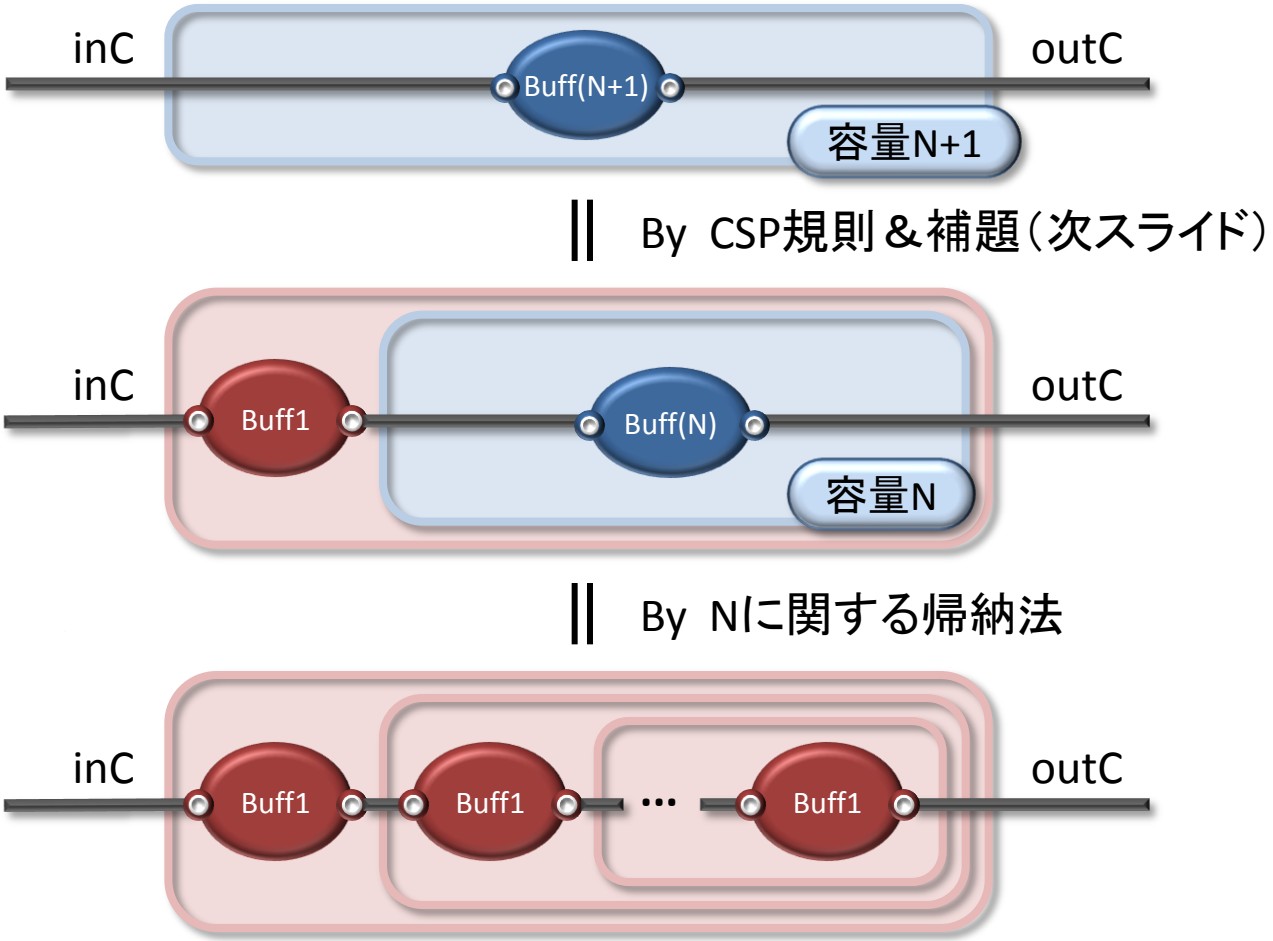
■ Buff(N)　（容量Nのバッファ）





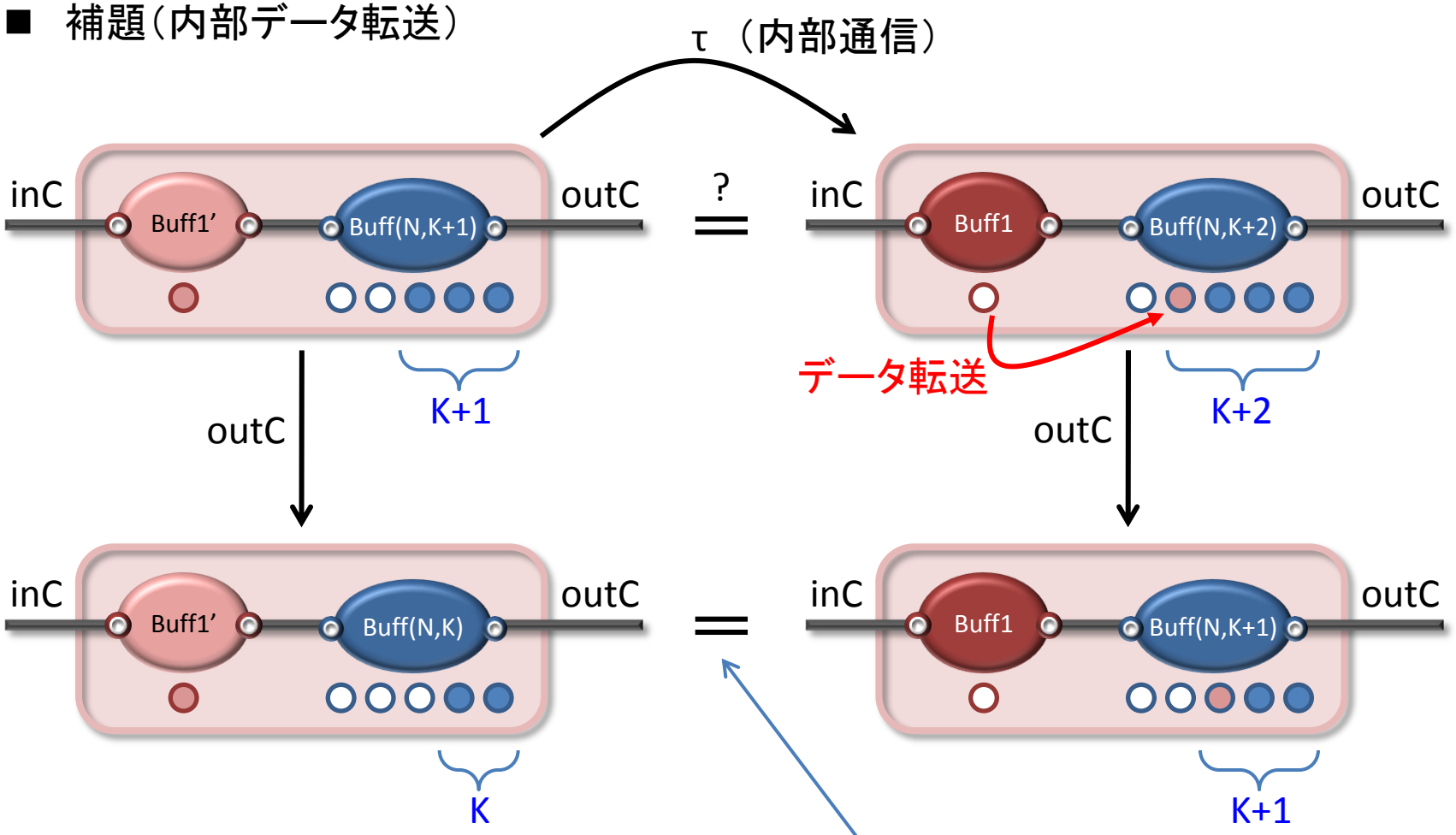
スケーラブルな並行システムの検証

■ Buff(N+1) = LinkBuff(N) の証明



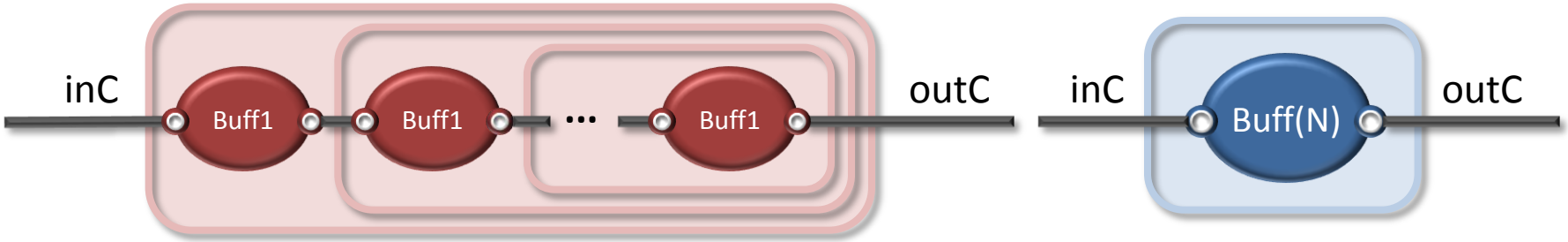
スケーラブルな並行システムの検証

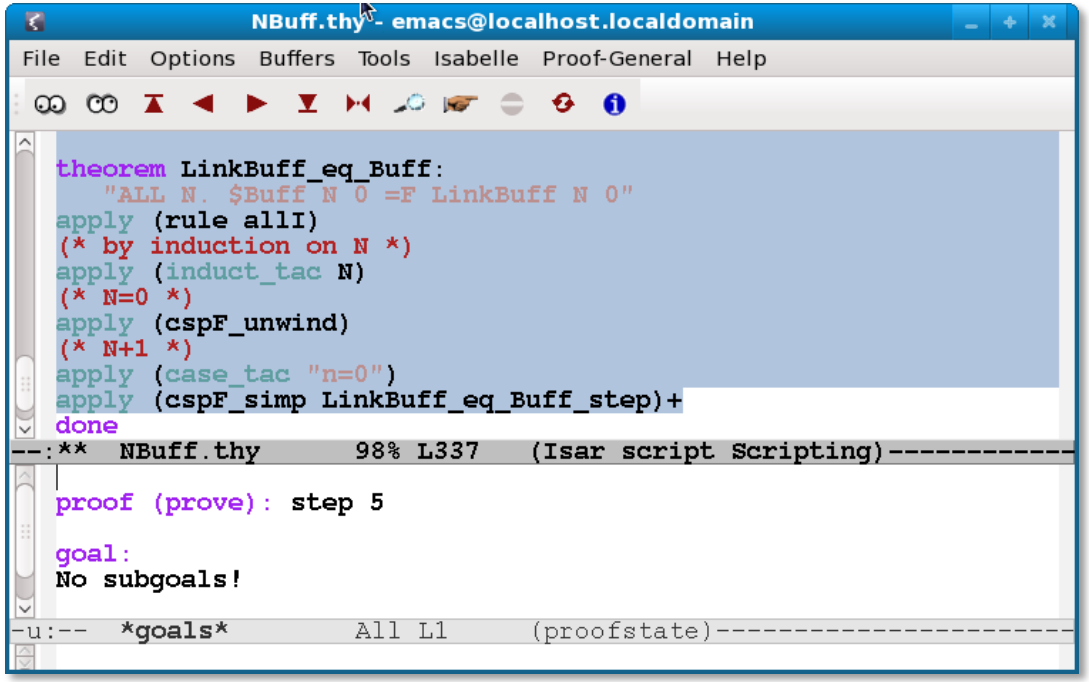
■ 補題（内部データ転送）



スケーラブルな並行システムの検証

■ 定理証明器CSP-Proverによる ∀N. LinkBuff(N) = Buff(N) の証明





まとめ

- モデル検査器の特徴
- 定理証明器の特徴

まとめ

■ **モデル検査器（例: FDR）の特徴**

- ○ 検証が完全に自動化されている。
- × パラメータを固定する必要がある（状態数は 10^7 ～10^8 程度まで）

■ **定理証明器（例: CSP-Prover）の特徴**

- ○ 帰納法等によってスケーラブルな並行システムを検証できる。
- ○ CSPの理論研究を支援できる（CSP理論に必要な様々な定理を含むため）。
- × 証明手続きをユーザが指示する必要がある。
- 検証例：Uniform Candy Distribution Puzzle
  （「コンピュータソフトウェア」, JSSST, Vol.25, No.4, pp.85-92, 2008）