Coalgebra, Automata, and Document Synchronization

Kazuhiro Inaba

Reading "Merging Hierarchically-Structured Documents in Workflow Systems" [E. Badouel and M. T. Tchendji, CMCS'08]

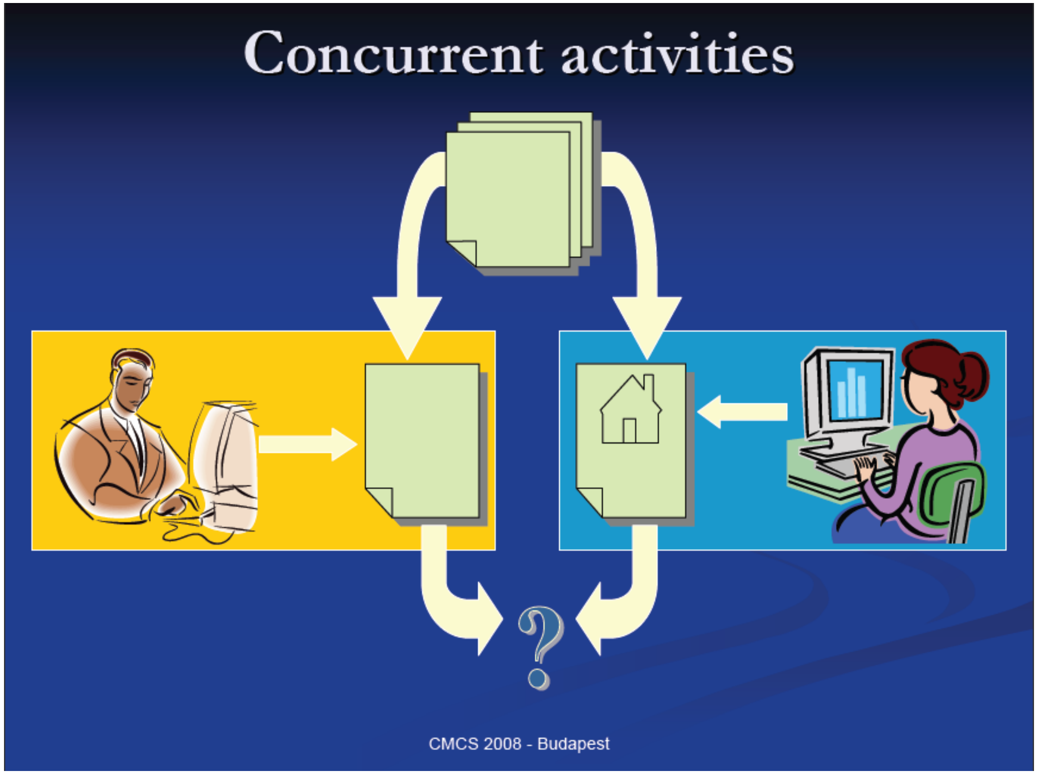
IPLAS 2009/06/09

- ((All pictures are cited from the authors presentation slide: http://old-www.cwi.nl/projects/cmcs08/slides/index.html thanks.))

≪Problem≫

- Partial views of a (big) document
- Concurrently updated

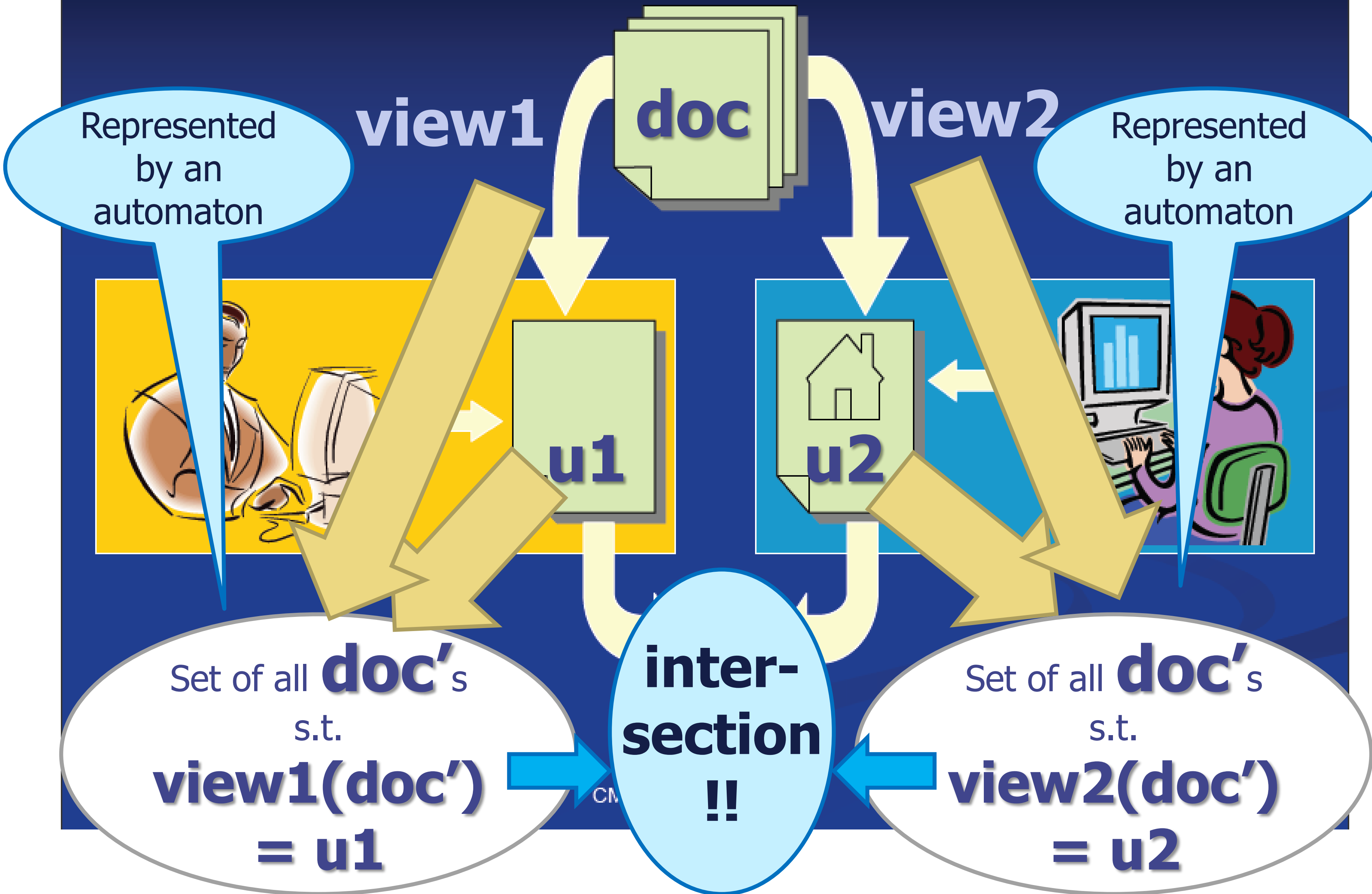
- **How to merge them?**



Approach of this paper

- **Use coalgebra (= tree automaton)!**

Concurrent activities

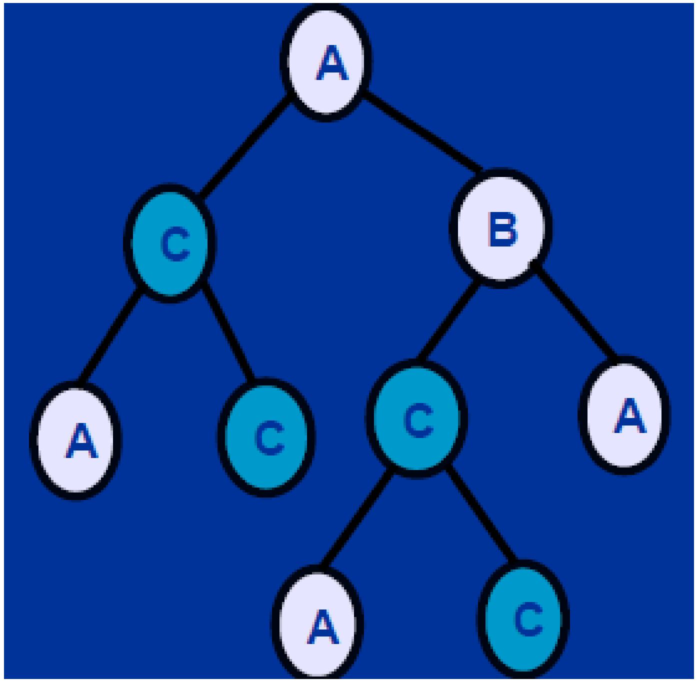


Basic Notions

- Document
- View
  - Projection
  - Expansion
- Grammar

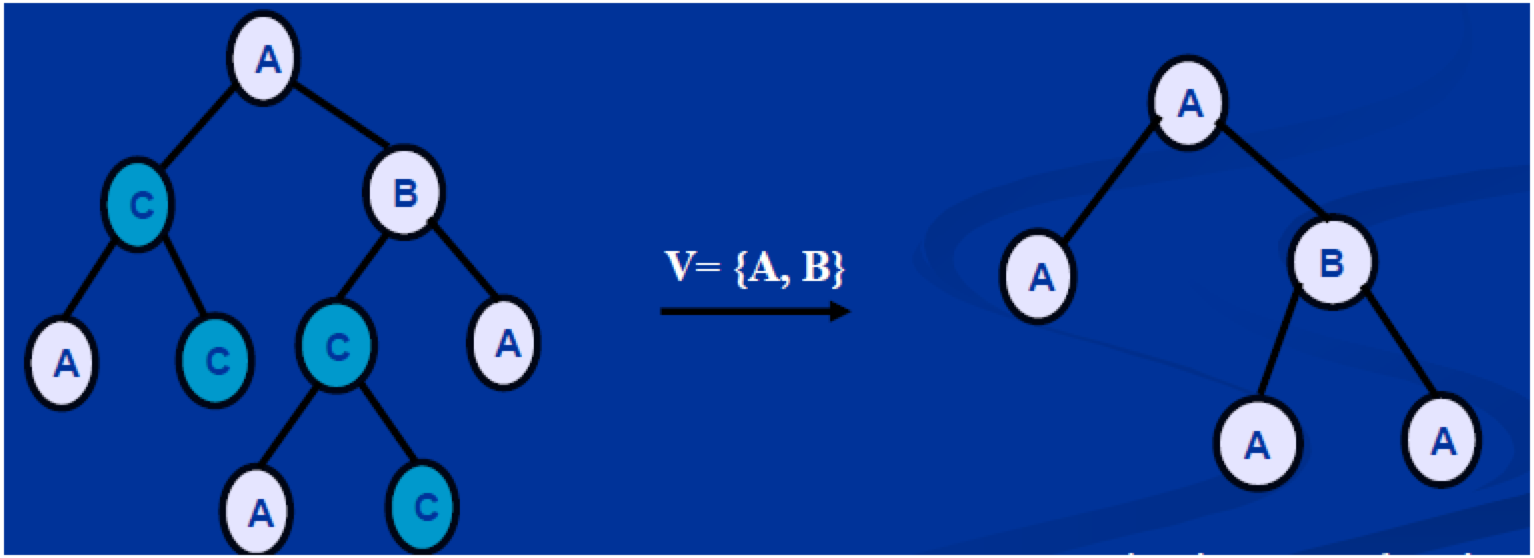
(Untyped) Document

- Let *S* be a finite alphabet
  - E.g. *S* = {A,B,C}
- A "document" (over *S*) is an unranked tree over *S*



(Untyped) View

- A "view" is a subset of *S*
- The "projection associated with a view" is the function (of type: tree → forest) that erases all symbols not in the view



(Untyped) Expansion

- The "expansion associated with a view" is the function (of type: forest → 2^{tree}) which is the inverse of the projection
  - Note: the output set is regular!
    ➔ they're represented as a tree automaton
  - How precisely? Wait a moment…

Grammar

- In the paper, the authors consider only "typed" documents.
  - Typed = Conformance to a grammar

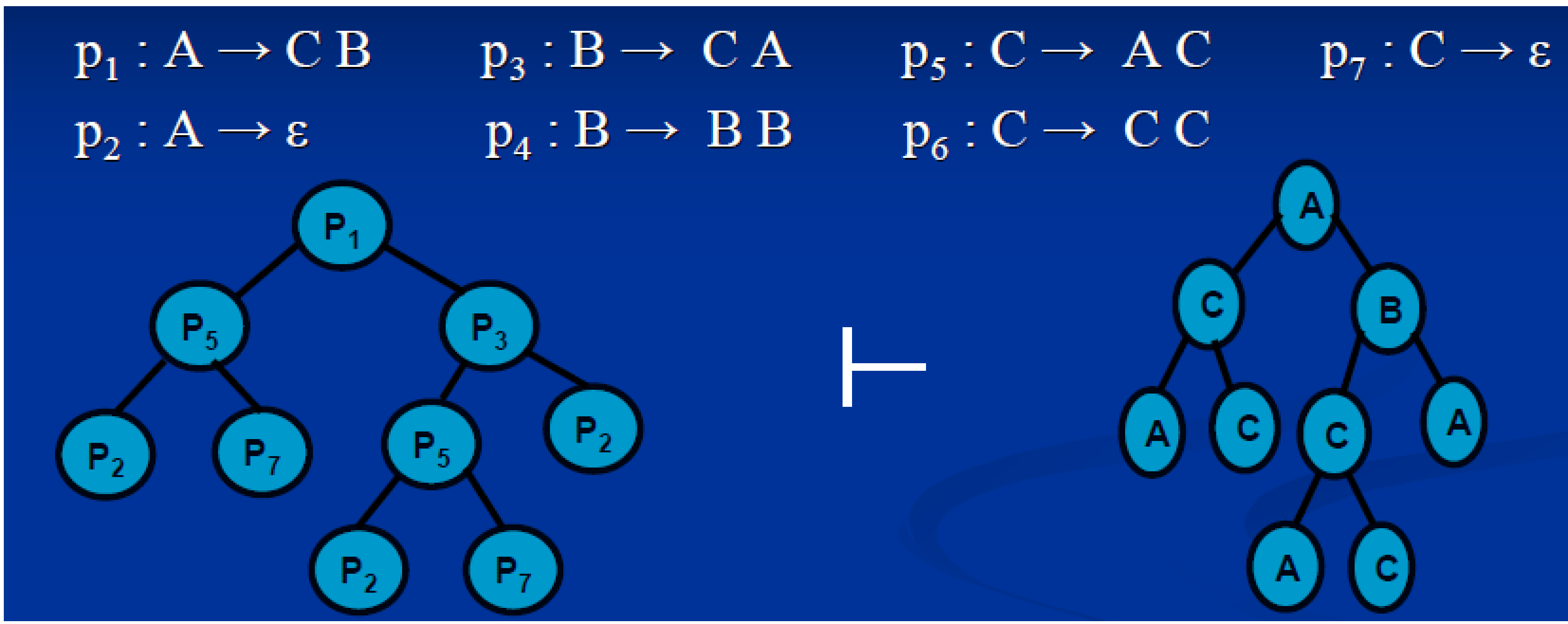
- A "Grammar" over $\mathcal{S}$ is a triple ($\mathcal{S}$, A, P):
  - $A \in \mathcal{S}$ axiom (initial symbol)
  - $P \subseteq \mathcal{S} \times \mathcal{S}^*$ set of productions

(Typed) Document

- A grammar $G = (\mathcal{S}, A, P)$ corresponds to a set of trees called "derivation trees", defined for each $X \in \mathcal{S}$ as follows:
  - $Der(G, X) = \{X(t_1,\ldots, t_n) \mid \exists X \rightarrow X_1 \ldots X_n \in P: \forall i: ti \in Der(G,X_i)\}$

- The members of Der(G, A) are the document confirming the grammar G
- From now on, we deal with such docs only

Example of a Grammar & a Doc.



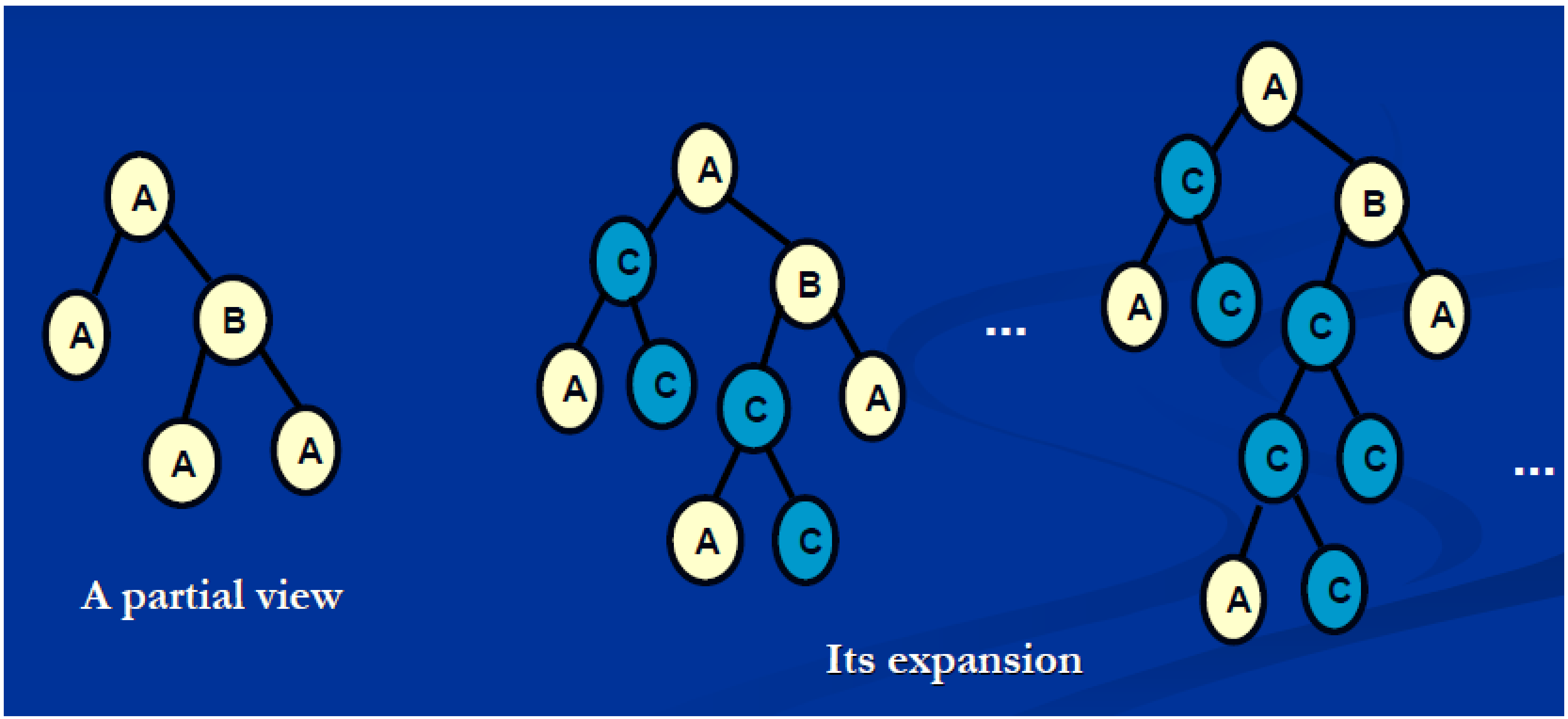
(Typed) View

- Same as before
  - A "view" $\subseteq \mathcal{S}$, "projection" is an erasure
- "expansion" takes two more parameters
  - G = ($\mathcal{S}$, A, P) : Grammar
  - X $\in \mathcal{S}$: Axiom
- expansion(V, ts, G, X) returns the set of trees in Der(G,X) whose projection with V are equal to ts

Example of an Expansion

$p_1 : A \rightarrow C\ B$ $\quad p_3 : B \rightarrow C\ A$ $\quad p_5 : C \rightarrow A\ C$ $\quad p_7 : C \rightarrow \varepsilon$

$p_2 : A \rightarrow \varepsilon$ $\quad p_4 : B \rightarrow B\ B$ $\quad p_6 : C \rightarrow C\ C$

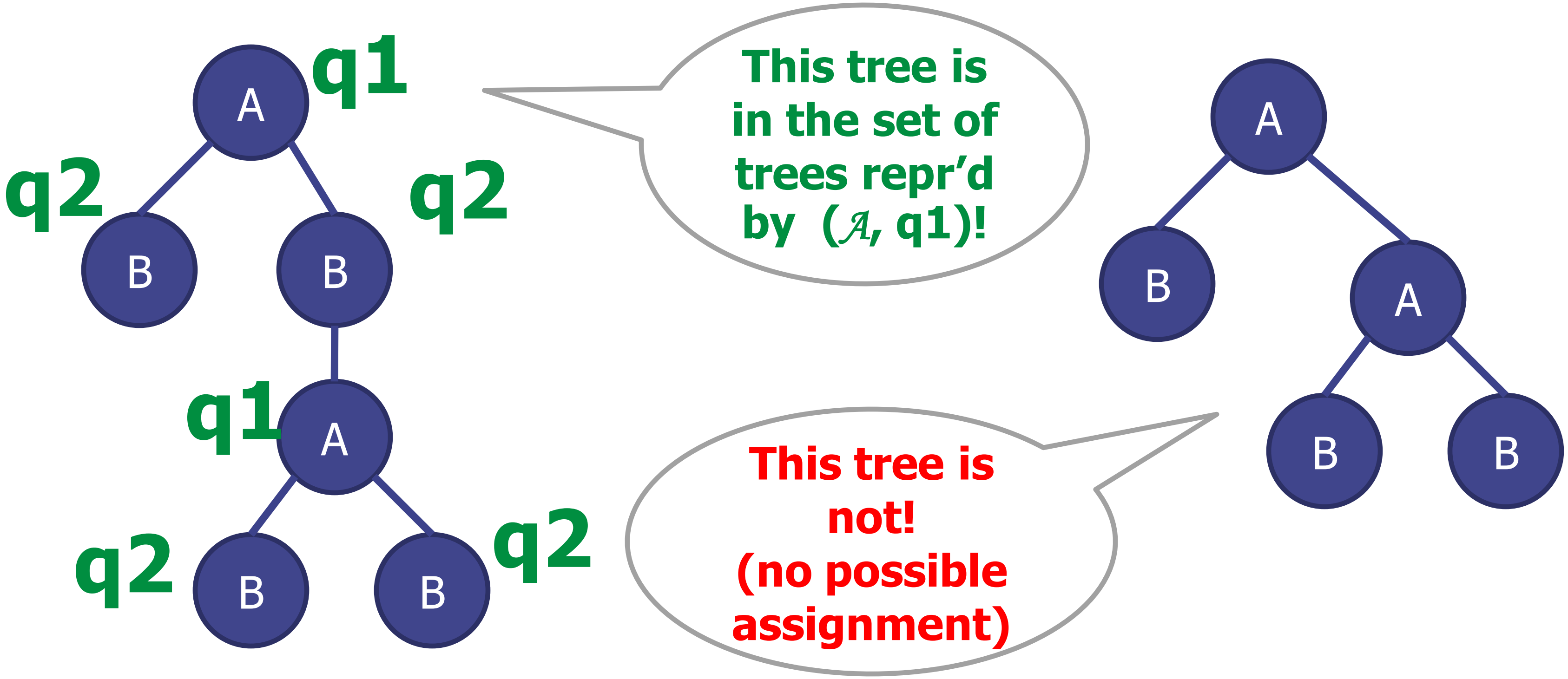


Top-Down Nodeterministic
Tree Automata

- Tree Automaton $\mathcal{A}$ is a tuple <$\mathcal{S}$, Q, δ>:
  - $\mathcal{S}$: node labels
  - Q : set of states
  - δ : $Q \rightarrow 2^{\mathcal{S} \times Q^*}$: transition relation
- Grammars are straightforwardly converted to tree automata: G = ($\mathcal{S}$, A, P) ~~>
  - $\mathcal{A}$ = ($\mathcal{S}$, Q, δ) where
    - Q = $\mathcal{S}$
    - δ(q) = {(q, q1 … qn) | q→q1…qn ∈ P}

Example

- $\mathcal{A}$ = ({A,B}, {q1,q2}, δ)
  - δ(q1) = { (A, q1q2), (A, q2q2) }
  - δ(q2) = { (B,), (B, q1) }



Representing Expansions by TA

- Recall:
  - expansion(**V**, **ts**, **G**, **X**) returns the set of trees in Der(G,X) whose projection by V are equal to ts.
  - **V** $\subseteq$ *S*
  - **G = (*S*, P)**

Input: V, G=($\mathcal{S}$, P), ts

Output: the corresponding automaton

- Corresponding automaton is $\mathcal{A} = (\mathcal{S}, Q, \delta)$:
  - $Q = \mathcal{S} \times \underline{T}$ $\underline{T}$ is the list of subtrees of **ts**
  - $\delta(\langle s, \underline{t} \rangle) =$
    - Φ if $s \in V$ and $\underline{t} \neq [s(\ldots)]$
    - $\{ (s, \langle s1, \underline{t1}\rangle\langle s2, \underline{t2}\rangle\ldots\langle sn, \underline{tn}\rangle) \mid$

      $s \rightarrow s1\ldots sn \in P$,

      $\underline{t1'} \ldots \underline{tn'} = \underline{t'}$

      $\}$

      if $s \in V$ and $\underline{t}=[s(\underline{t'})]$

      or $s \notin V$ and $\underline{t}=\underline{t'}$

Proposition (Correctness)

- Member of expansion(V, ts, G, X)

  iff

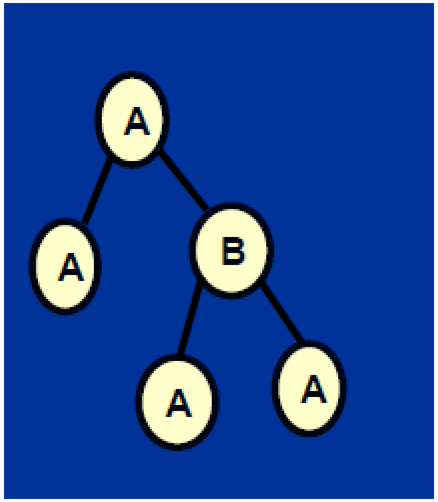
- Accepted by the automaton A from the state <X, ts>

Example

- $\mathcal{S}$ = {A,B,C}
- G = ($\mathcal{S}$, A, P) where P is:

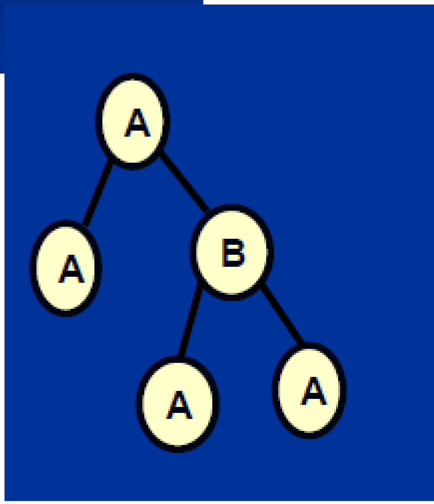
$p_1 : A \rightarrow C\ B$	$p_3 : B \rightarrow C\ A$	$p_5 : C \rightarrow A\ C$	$p_7 : C \rightarrow \varepsilon$
$p_2 : A \rightarrow \varepsilon$	$p_4 : B \rightarrow B\ B$	$p_6 : C \rightarrow C\ C$	

- V = {A, B}
- ts =



$p_1 : A \rightarrow C\ B$ $\quad p_3 : B \rightarrow C\ A$ $\quad p_5 : C \rightarrow A\ C$ $\quad p_7 : C \rightarrow \varepsilon$

$p_2 : A \rightarrow \varepsilon$ $\quad p_4 : B \rightarrow B\ B$ $\quad p_6 : C \rightarrow C\ C$

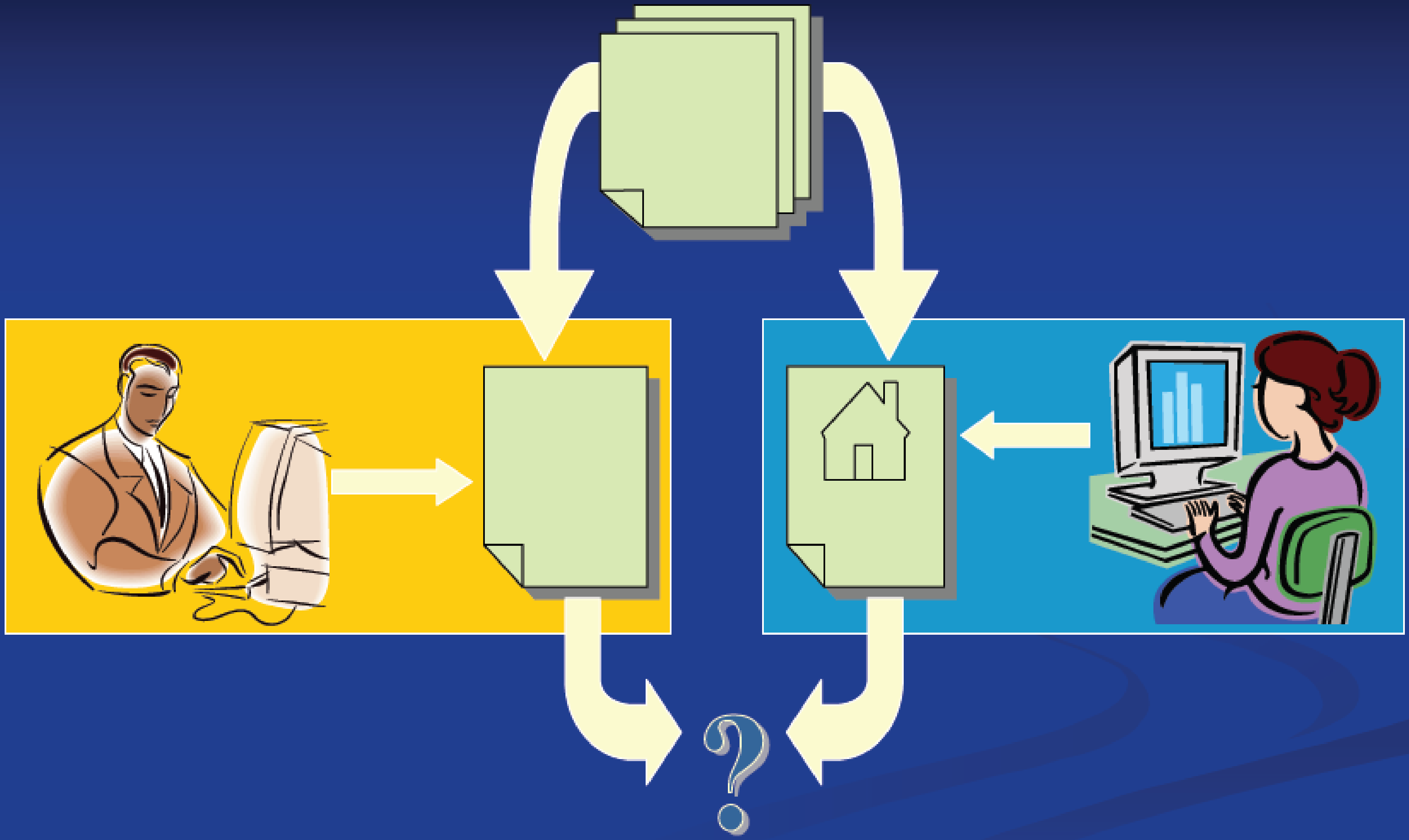


- δ(<A, **A(A,B(A,A))**>) = {
  (A, <C, **A,B(A,A))**><B,ε>),
  (A, <C, **A**><B, **B(A,A))**>),
  (A, <C, ε><B, **A,B(A,A))** >) }
- δ(<C, **A,B(A,A))**>) = {
  (C, <A, **A,B(A,A)**><C, ε>),
  (C, <A, **A**><C, **B(A,A)**>),
  (C, <A, ε><C, **A,B(A,A)**>),
  (C, <C, **A,B(A,A)**><C, ε>),
  (C, <C, **A**><C, **B(A,A)**>),
  (C, <C, ε><C, **A,B(A,A)**>) }
- ...

Now, On Tree Automata…

- We can compute
  - Emptiness of the represented set
    - Coinductively
  - Intersection between two sets
    - By Product Construction
      - $(Q, \delta) \cap (P, \gamma) = (Q \times P, \beta)$ where
        - $\beta((q, p)) =$
          $\{ (\sigma, (q1,p1)\ldots(qn,pn)) \mid$
          $(\sigma,q1\ldots qn) \in \delta(q)$ and $(\sigma,p1\ldots pn) \in \gamma(p) \}$

Concurrent activities



Algorithm: "Coherence" check

- Given
  - Grammar G = (*S*, A, P)
  - View V1 and Forest ts1
  - View V2 and Forest ts2
- Are these two views coherent? (i.e., can we "merge" them into a single document that generates the two views simultaneously?)

expansion(V1,ts1,G,A) ∩ expansion(V2,ts2,G,A) ≠ Φ?

Algorithm: “Synchronization”

- Given
  - Grammar G = (*S*, A, P)
  - View V1 and Forest ts1
  - View V2 and Forest ts2
- How can we get the merged document?

➔ If **expansion(V1,ts1,G,A) ∩ expansion(V2,ts2,G,A)**
is a singleton set, that’s it!
Otherwise, ambiguous → error

Singleton check
(Not in the paper…)

- Step 1 (Cleaning): $\mathcal{A}$ ~~> $\mathcal{A}_{cl}$
  - Eliminate all "failure" states and transitions
- Step 2 (Thinning): $\mathcal{A}_{cl}$ ~~> $\mathcal{A}_{cl,th}$
  - Determinize the automaton by dropping nondeterministic rules
- Step 3 (Equivalence Check) $\mathcal{A}_{cl}$ =? $\mathcal{A}_{cl,th}$
  - Check Bisimilarity

Example 1

- ADDRESSBOOK → @
- @ → ε | PERSON @
- PERSON → NAME ADDRESS TEL
- NAME → …, ADDRESS → …, TEL → …

- V1 = {NAME}
- V2 = {TEL}

- Example of a document: ADDRESSBOOK[@[
  - PERSON[
    - NAME[...]   ADDRESS[...]   TEL[...]
  -]@[
  - PERSON[
    - NAME[...]   ADDRESS[...]   TEL[...]
  -]@[]]]]
- expansion(V1, NAME[...]NAME[...])
  is consistent with
  expansion(V2, TEL[...]TEL[...])
  but not with
  expansion(V2, TEL[...]TEL[...]TEL[...])

Example 2

- ADDRESSBOOK → @
- @ → ε | PERSON @
- PERSON → NAME ADDRESS TEL #
- # → ε | TEL #
- NAME → …, ADDRESS → …, TEL → …

- V1 = {NAME}
- V2 = {TEL}

- Example of a document: ADDRESSBOOK[@[
  - PERSON[
    - NAME[...] ADDRESS[...] TEL[...] #[]
  -]@[
  - PERSON[
    - NAME[...] ADDRESS[...] TEL[...] #[TEL[...] #[]]
  -]@[]]]]
- expansion(V1, NAME[...]NAME[...])
  is consistent with
  expansion(V2, TEL[...]TEL[...])
  and also with
  expansion(V2, TEL[...]TEL[...]TEL[...])

Ambiguity

- Example of a document: ADDRESSBOOK[@[
  - PERSON[
    - NAME[...] ADDRESS[...] #[TEL[...] #[]]
  -]@[
  - PERSON[
    - NAME[...] ADDRESS[...] #[TEL[...] #[TEL[...] #[]]
  -]@[]]]]
- Ambiguity in the synchronization of
  expansion(V1, NAME[...]NAME[...])
  with
  expansion(V2, TEL[...]TEL[...]TEL[...])

How to resolve ambiguity

- Be more careful in choosing views
  - Set of views that "covers" the whole structure
- Such as
  - V1 = {NAME}
  - V2 = {PERSON, TEL}

  or

  - V1 = {PERSON, NAME}
  - V2 = {TEL, #}

  etc.

“Static” Singleton Checking?

- How can the merging system support users to choose appropriate views?
- For example, Given a set of views, can we check whether they cause ambiguity or not?
  ➔ Undecidable Problem
  (reduces to the ambiguity of CFG)
  - (No clear solution is given in the paper…)

Summary

- Synchronization of multiple (edited) views can be computed by using tree automaton
  - Compute inverse-image of the view
  - Compute intersection, emptiness, & singularity
- Further topic (in the paper, but not in this presentation)
  - "To-be-written" node
    - For each symbol X in the grammar, add $\underline{X}$ and $\underline{X} \to \varepsilon$
    - Synchronizer allows X in u2 to occur at $\underline{X}$ position in u1, etc…

APPENDIX:

Automata as Coalgebra

- J.J.M.M. Rutten, "Automata and Coinduction (An Exercise in Coalgebra)", CONCUR 1998
  - The classical theory of deterministic automata is presented in terms of the notions of ***homomorphism and bisimulation***, which are the cornerstones of the theory of (universal) coalgebra. This leads to a transparent and uniform presentation of automata theory and yields some new insights, amongst which coinduction proof methods for language equality and language inclusion. At the same time, the present treatment of automata theory may serve as an introduction to coalgebra.