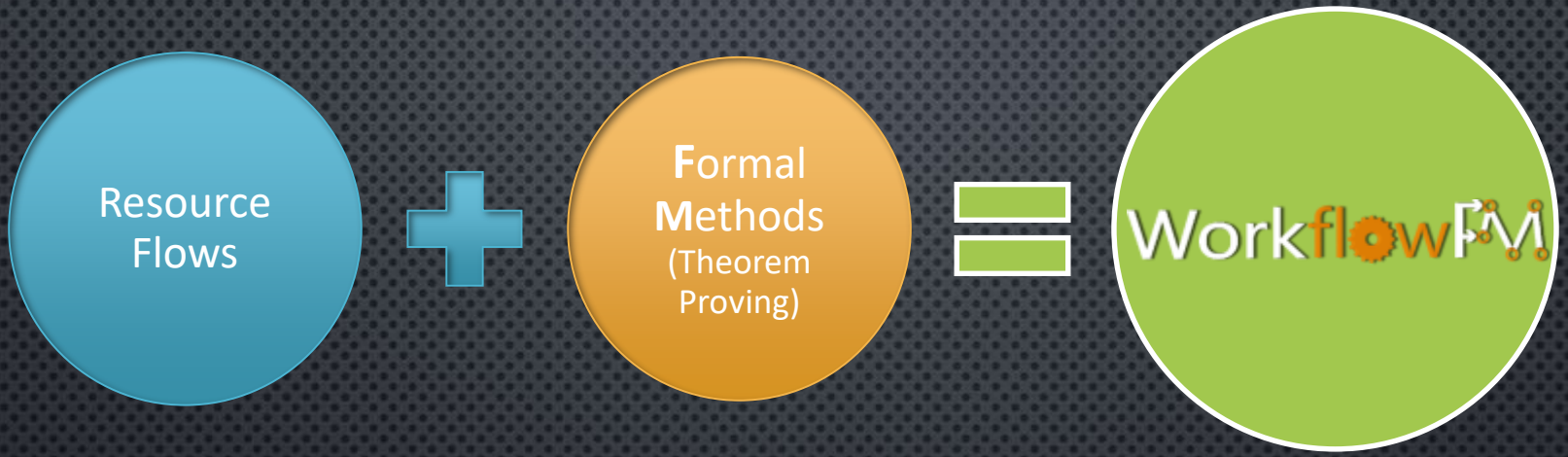
CORRECT-BY-CONSTRUCTION PROCESS COMPOSITION USING CLASSICAL LINEAR LOGIC INFERENCE

PETROS PAPAPANAGIOTOU

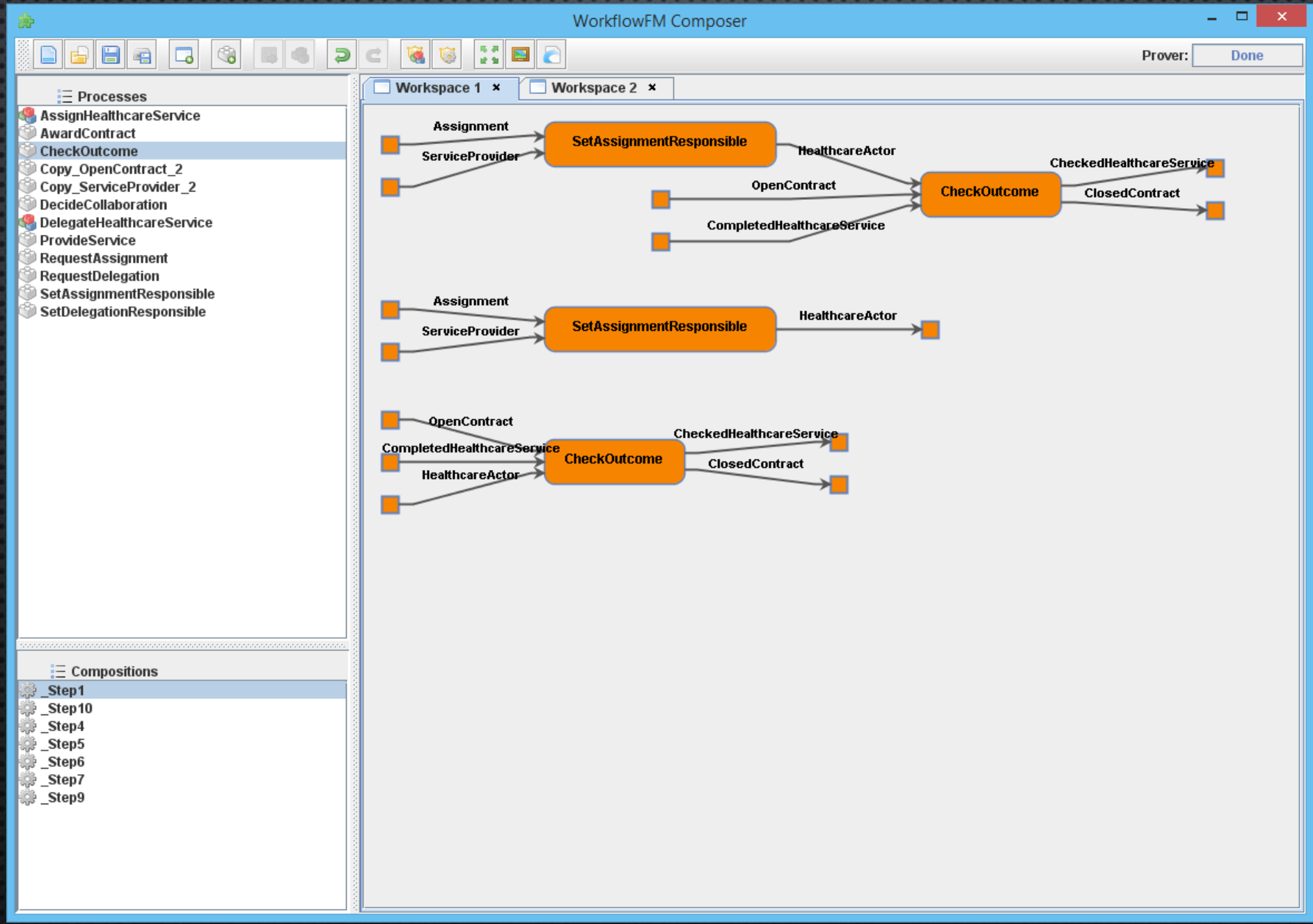
JACQUES FLEURIOT

LOPSTR 4-6 SEPTEMBER 2018

OUR SOLUTION



- ✓ VISUAL, RESOURCE-BASED WORKFLOW MODEL
- ✓ CORRECT-BY-CONSTRUCTION DESIGN
  - ✓ TYPE CHECKED
  - ✓ SYSTEMATICALLY TRACKED RESOURCES
  - ✓ DEADLOCK-FREE ASYNCHRONOUS EXECUTION

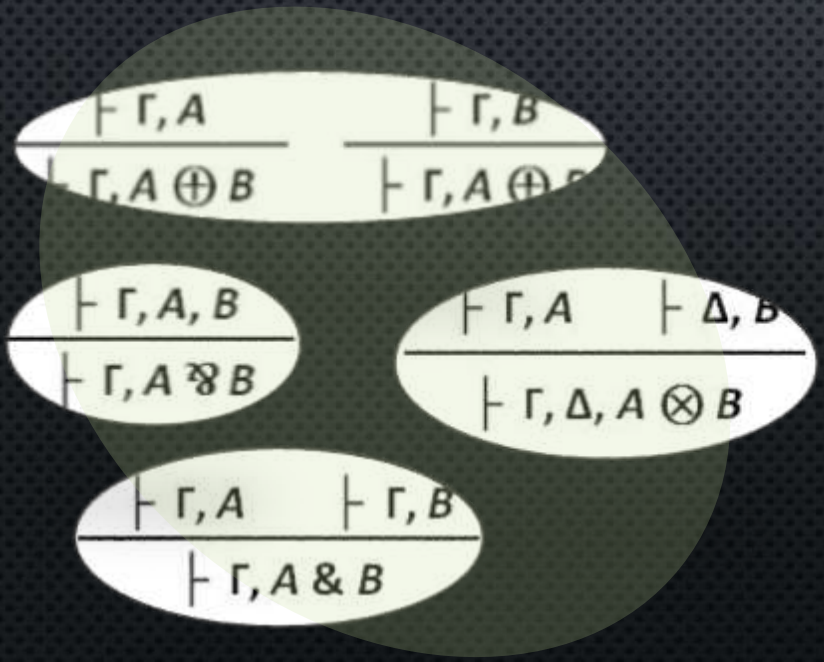

WorkflowFM Composer
Prover:
Done
Processes
AssignHealthcareService
AwardContract
CheckOutcome
Copy_OpenContract_2
Copy_ServiceProvider_2
DecideCollaboration
DelegateHealthcareService
ProvideService
RequestAssignment
RequestDelegation
SetAssignmentResponsible
SetDelegationResponsible
Compositions
_Step1
_Step10
_Step4
_Step5
_Step6
_Step7
_Step9
Workspace 1
Workspace 2
Assignment
ServiceProvider
SetAssignmentResponsible
HealthcareActor
OpenContract
CompletedHealthcareService
CheckOutcome
CheckedHealthcareService
ClosedContract
Assignment
ServiceProvider
SetAssignmentResponsible
HealthcareActor
OpenContract
CompletedHealthcareService
HealthcareActor
CheckOutcome
CheckedHealthcareService
ClosedContract

LINEAR LOGIC

PROOFS-AS-PROCESSES

LOGICAL PROOFS → CONCURRENT PROCESSES

CLASSICAL LINEAR LOGIC → Π-CALCULUS



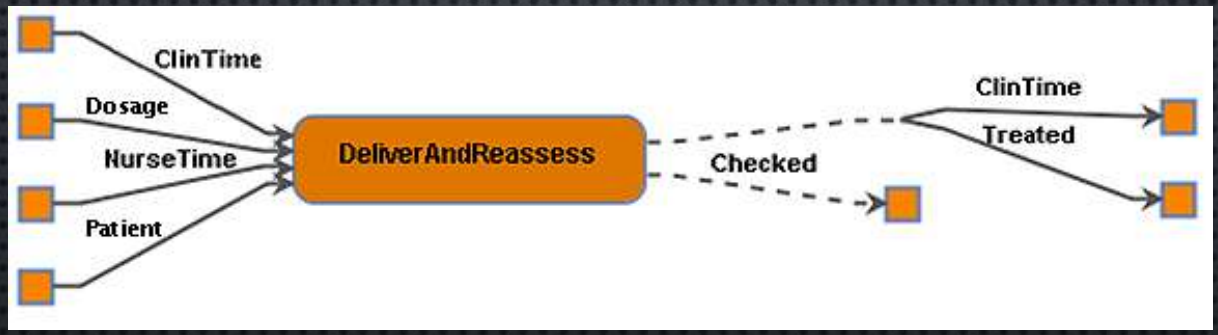
EXAMPLE


Patient
Dosage
NurseTime
DeliverDrug
Treated
Failed

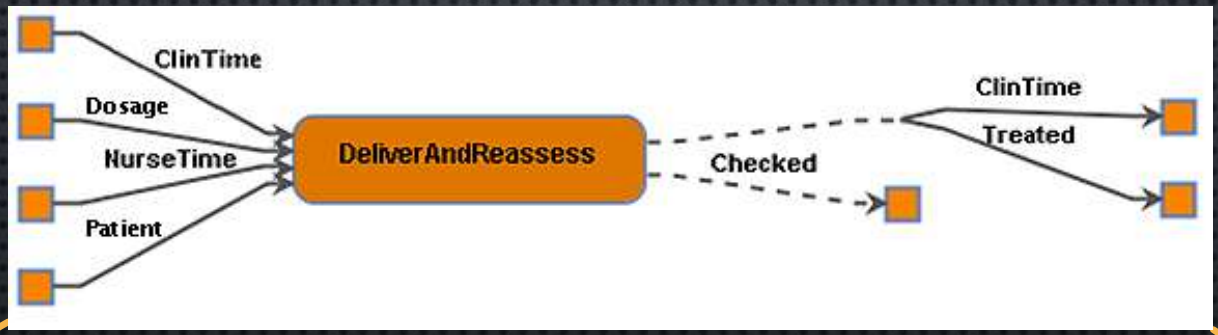

Failed
ClinTime
Reassess
Checked

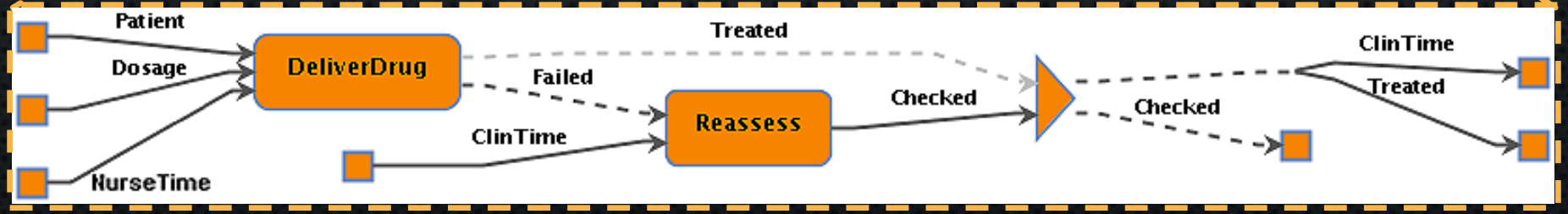

DeliverAndReassess

LESSON: TIME IS A VALUABLE RESOURCE!



LESSON: TIME IS A VALUABLE RESOURCE!





LINEAR LOGIC: PROCESS SPECIFICATION

$$\vdash A^{\perp}, (B \oplus C)^{\perp}, D \otimes E$$

$\vdash$ PATIENT$^{\perp}$, DOSAGE$^{\perp}$, NURSETIME$^{\perp}$, (TREATED $\oplus$ FAILED)

$\vdash$ FAILED$^{\perp}$, CLINTIME$^{\perp}$, CHECKED

LINEAR LOGIC ENGINE

$$\frac{}{\vdash A^{\perp}, A}\ Id \qquad \frac{\vdash \Gamma, C \quad \vdash \Delta, C^{\perp}}{\vdash \Gamma, \Delta}\ Cut$$

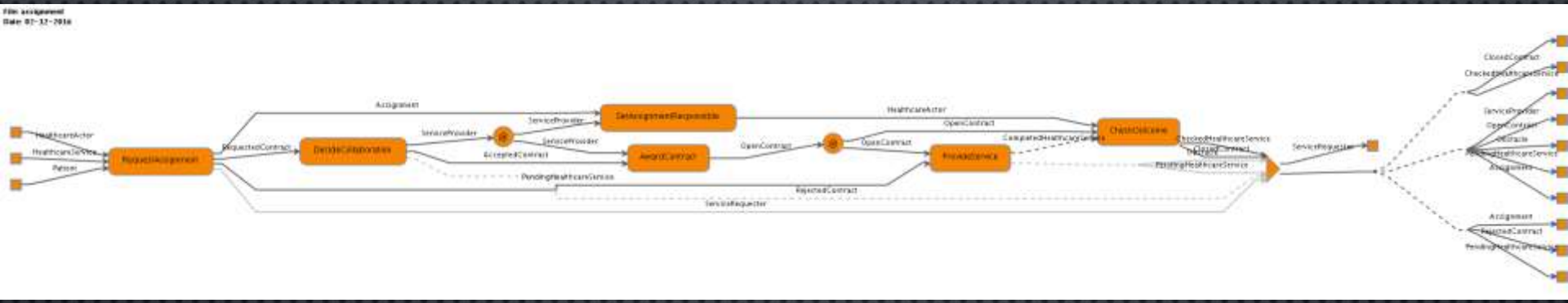
$$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\ \otimes \qquad \frac{\vdash \Gamma, A^{\perp}, B^{\perp}}{\vdash \Gamma, (A \otimes B)^{\perp}}\ \parr$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}\ \oplus_L \qquad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B}\ \oplus_R \qquad \frac{\vdash \Gamma, A^{\perp} \quad \vdash \Gamma, B^{\perp}}{\vdash \Gamma, (A \oplus B)^{\perp}}\ \&$$

A PRAGMATIC APPROACH

Case Study Theme	Processes	Resource Types	Binary Actions	Workflows
Patient Handovers	9	16	13	2
Tracheostomy care pathway	33	47	32	3
HIV care pathways	128	129	121	13
Pen manufacturing	42	45	60	20
Total	**212**	**237**	**226**	**38**

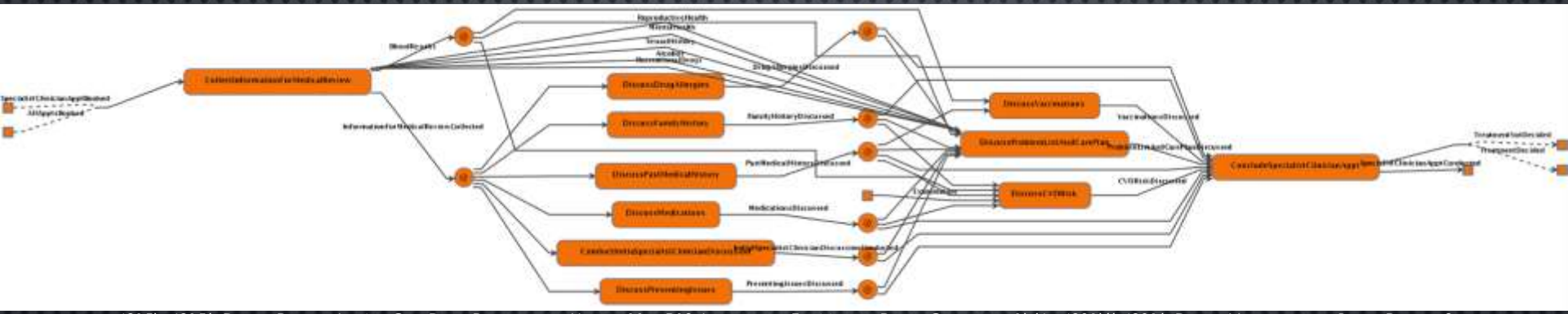
COMPLEX EXAMPLE



(^ z176)((z176(cSetDelegationResponsible_Delegation_1, z173),z173(cDecideCollaboration_RequestedContract_1, z171),z171(cSetDelegationResponsible_ServiceRequester_2, cProvideService_PendingHealthcareService_2).(^ z163)(((^ u155, v155)('z163<u155, v155>.(u155(z154).(^ y130)(y155(up155, vp155).'up155<y130>.z154(cAwardContract_AcceptedContract_1, cAwardContract_ServiceProvider_2).(^ z148)(((^ z146)((z146(cCheckOutcome_OpenContract_1, cProvideService_OpenContract_1).(^ z141)(((^ u130, v130)('z141<u130, v130>.(u130(cCheckOutcome_CompletedHealthcareService_2).(^ oCheckOutcome_lB_CheckedHealthcareService_x_ClosedContract_rB_)(y130(up130, vp130).'up130<oCheckOutcome_lB_CheckedHealthcareService_x_ClosedContract_rB_>.(^ z127)(((**CheckOutcome**(cCheckOutcome_OpenContract_1, cCheckOutcome_CompletedHealthcareService_2, z127, oCheckOutcome_lB_CheckedHealthcareService_x_ClosedContract_rB_)) | (**SetDelegationResponsible**(cSetDelegationResponsible_Delegation_1, cSetDelegationResponsible_ServiceRequester_2, z127))))) + v130(c130).(^ d130)(y130(uq130, vq130).'vq130<d130>.(^ x131, y131)('d130<x131, y131>.(cSetDelegationResponsible_Delegation_1(m132).'x131<m132>.0 | (^ x133, y133)('y131<x133, y133>.(cCheckOutcome_OpenContract_1(m134).'x133<m134>.0 | (^ x135, y135)('y133<x135, y135>.(cSetDelegationResponsible_ServiceRequester_2(m136).'x135<m136>.0 | c130(x137, y137).(^ x138, y138)('y135<x138, y138>.(x137(m139).'x138<m139>.0 | y137(m140).'y138<m140>.0)))))))))) | (**ProvideService**(cProvideService_OpenContract_1, cProvideService_PendingHealthcareService_2, z141)))) | (**Copy_OpenContract_2** (z148,z146)))) | (**AwardContract**(cAwardContract_AcceptedContract_1,cAwardContract_ServiceProvider_2,z148))))) + v155(c155).(^ d155)(y155(uq155, vq155).'vq155<d155>.(^ x156, y156)('d155<x156, y156>.(cSetDelegationResponsible_Delegation_1(m157).'x156<m157>.0 | (^ x158, y158)('y156<x158, y158>.(cProvideService_PendingHealthcareService_2(m159).'x158<m159>.0 | (^ x160, y160)('y158<x160, y160>.(cSetDelegationResponsible_ServiceRequester_2(m161).'x160<m161>.0 | c155(m162).'y160<m162>.0)))))))) | (**DecideCollaboration**(cDecideCollaboration_RequestedContract_1,z163)))) | (**RequestDelegation**(cRequestDelegation_Patient_1, cRequestDelegation_HealthcareActor_2, cRequestDelegation_HealthcareService_3, z176))))

Papapanagiotou, Fleuriot. **Modelling and Implementation of Correct by Construction Healthcare Workflows**, Lecture Notes in Business Information Processing, BPM 2014.

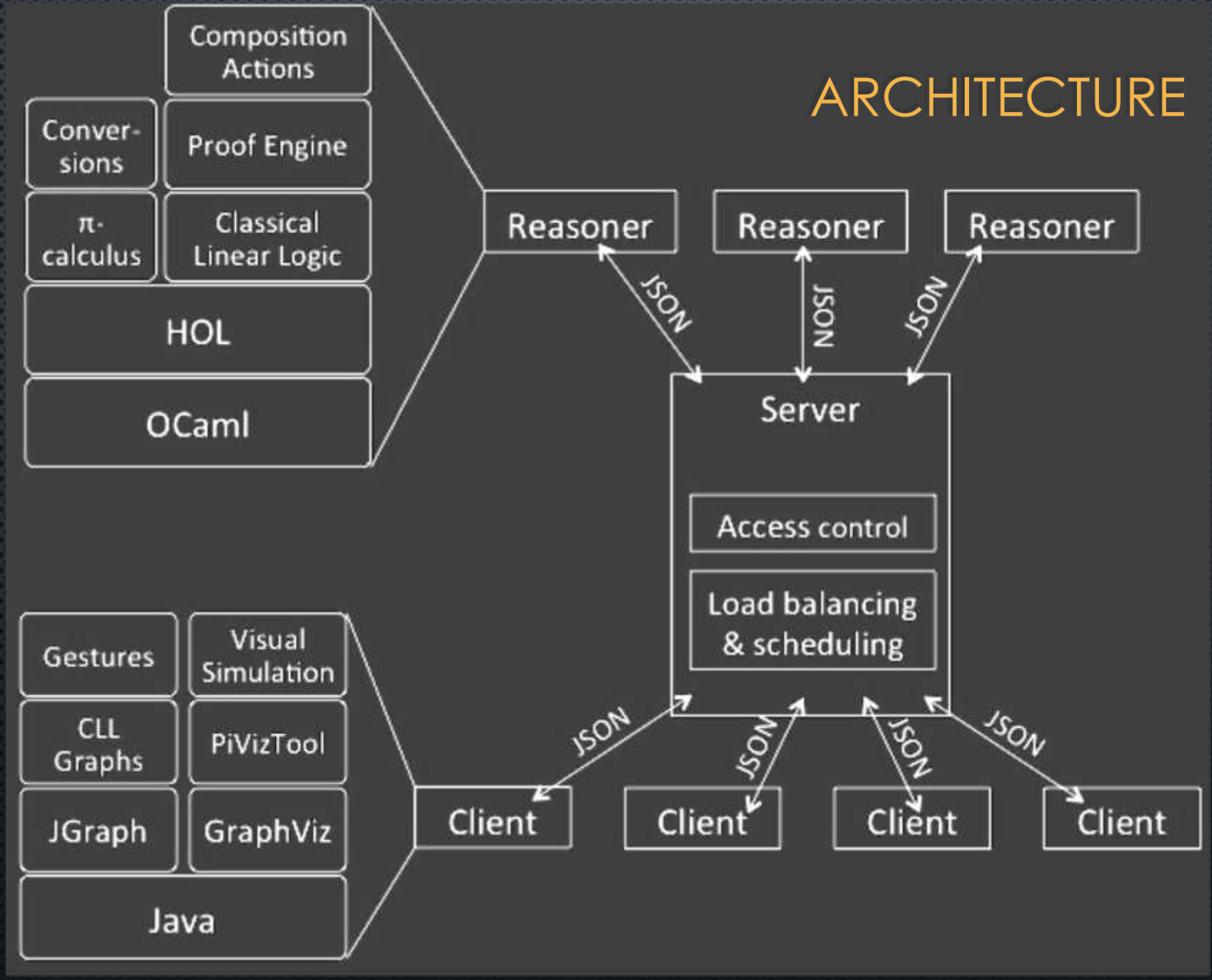
EVEN MORE COMPLEX REAL-WORLD EXAMPLE: HIV CARE IN THE NHS



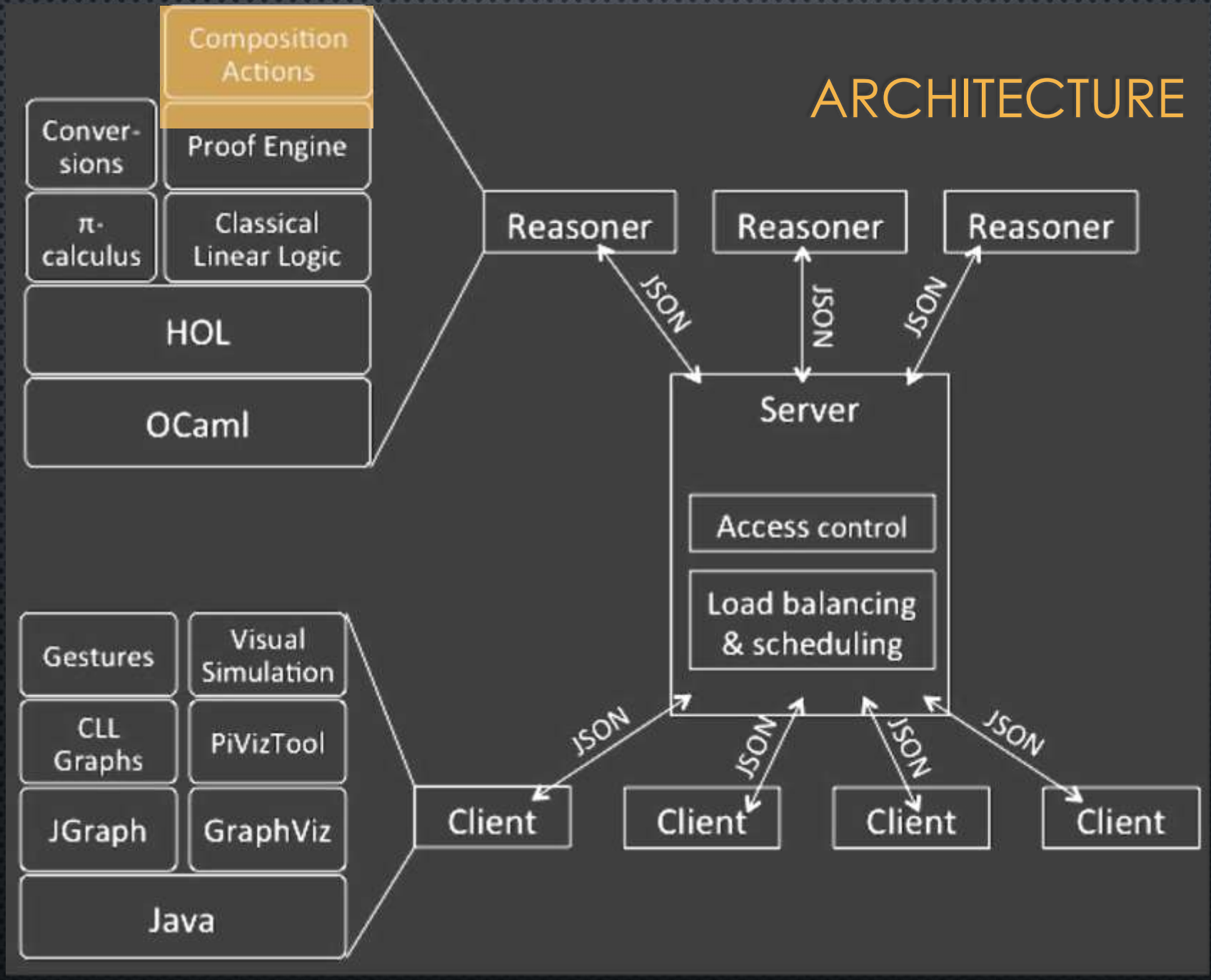
z4215).z4215(cDiscussProblemListAndCarePlan_ReproductiveHealth_11, cP13_InformationForMedicalReviewCollected_1).(^ z4201)((z4201(cDiscussVaccinations_BloodResults_2, z4199).z4199(cDiscussCVDRisk_BloodResults_5, cDiscussProblemListAndCarePlan_BloodResults_10).(^ z4192)((z4192(cDiscussPresentingIssues_InformationForMedicalReviewCollected_1, z4190).z4190(cDiscussPastMedicalHistory_InformationForMedicalReviewCollected_1, z4189).z4189(cDiscussMedications_InformationForMedicalReviewCollected_1, z4188).z4188(cDiscussFamilyHistory_InformationForMedicalReviewCollected_1, z4187).z4187(cConductInitiaSpecialistClinicianDiscussion_InformationForMedicalReviewCollected_1, cDiscussDrugAllergies_InformationForMedicalReviewCollected_1).(^ z4171)(((^ z4169)(((^ z4167)(((^ z4165)(((^ z4163)(((^ z4161)(((^ z4159)((z4159(cDiscussProblemListAndCarePlan_MedicationsDiscussed_6, z4156).z4156(cConcludeSpecialistClinicianAppt_MedicationsDiscussed_5, cDiscussCVDRisk_MedicationsDiscussed_1).(^ z4150)((z4150(cConcludeSpecialistClinicianAppt_DrugAllergiesDiscussed_6, cDiscussProblemListAndCarePlan_DrugAllergiesDiscussed_12).(^ z4145)((z4145(cConcludeSpecialistClinicianAppt_PresentingIssuesDiscussed_2, cDiscussProblemListAndCarePlan_PresentingIssuesDiscussed_9).(^ z4140)((z4140(cConcludeSpecialistClinicianAppt_InitialSpecialistClinicianDiscussionConducted_1, cDiscussProblemListAndCarePlan_InitialSpecialistClinicianDiscussionConducted_1).(^ z4135)((z4135(cDiscussProblemListAndCarePlan_FamilyHistoryDiscussed_8, z4132).z4132(cConcludeSpecialistClinicianAppt_FamilyHistoryDiscussed_4, cDiscussCVDRisk_FamilyHistoryDiscussed_3).(^ z4126)((z4126(cDiscussVaccinations_PastMedicalHistoryDiscussed_1, z4123).z4123(cDiscussProblemListAndCarePlan_PastMedicalHistoryDiscussed_7, z4121).z4121(cConcludeSpecialistClinicianAppt_PastMedicalHistoryDiscussed_3, cDiscussCVDRisk_PastMedicalHistoryDiscussed_2).(^ z4113)(((^ z4111)(((^ z4109)(((ConcludeSpecialistClinicianAppt(cConcludeSpecialistClinicianAppt_InitialSpecialistClinicianDiscussionConducted_1, cConcludeSpecialistClinicianAppt_PresentingIssuesDiscussed_2, cConcludeSpecialistClinicianAppt_PastMedicalHistoryDiscussed_3, cConcludeSpecialistClinicianAppt_FamilyHistoryDiscussed_4, cConcludeSpecialistClinicianAppt_MedicationsDiscussed_5, cConcludeSpecialistClinicianAppt_DrugAllergiesDiscussed_6, z4113, z4111, z4109, oConcludeSpecialistClinicianAppt_lB_SpecialistClinicianApptConducted_x_lB_TreatmentDecided_Plus_TreatmentNotDecided_rB_rB_)) | (DiscussVaccinations(cDiscussVaccinations_PastMedicalHistoryDiscussed_1, cDiscussVaccinations_BloodResults_2, z4109)))) | (DiscussCVDRisk(cDiscussCVDRisk_MedicationsDiscussed_1, cDiscussCVDRisk_PastMedicalHistoryDiscussed_2, cDiscussCVDRisk_FamilyHistoryDiscussed_3, cDiscussCVDRisk_Examination_4, cDiscussCVDRisk_BloodResults_5, z4111)))) | (DiscussProblemListAndCarePlan(cDiscussProblemListAndCarePlan_InitialSpecialistClinicianDiscussionConducted_1, cDiscussProblemListAndCarePlan_SexualHistory_2, cDiscussProblemListAndCarePlan_MentalHealth_3, cDiscussProblemListAndCarePlan_Alcohol_4, cDiscussProblemListAndCarePlan_RecreationalDrugs_5, cDiscussProblemListAndCarePlan_MedicationsDiscussed_6, cDiscussProblemListAndCarePlan_PastMedicalHistoryDiscussed_7, cDiscussProblemListAndCarePlan_FamilyHistoryDiscussed_8, cDiscussProblemListAndCarePlan_PresentingIssuesDiscussed_9, cDiscussProblemListAndCarePlan_BloodResults_10, cDiscussProblemListAndCarePlan_ReproductiveHealth_11, cDiscussProblemListAndCarePlan_DrugAllergiesDiscussed_12, z4113)))) | (P60 (z4167, z4126)))) | (P61 (z4169, z4135)))) | (P63 (z4161, z4140)))) | (P64 (z4163, z4145)))) | (P65 (z4171, z4150)))) | (P59 (z4165, z4159)))) | (ConductInitiaSpecialistClinicianDiscussion(cConductInitiaSpecialistClinicianDiscussion_InformationForMedicalReviewCollected_1, z4161)))) | (DiscussPresentingIssues(cDiscussPresentingIssues_InformationForMedicalReviewCollected_1, z4163)))) | (DiscussMedications(cDiscussMedications_InformationForMedicalReviewCollected_1, z4165)))) | (DiscussPastMedicalHistory(cDiscussPastMedicalHistory_InformationForMedicalReviewCollected_1, z4167)))) | (DiscussFamilyHistory(cDiscussFamilyHistory_InformationForMedicalReviewCollected_1, z4169)))) | (DiscussDrugAllergies(cDiscussDrugAllergies_InformationForMedicalReviewCollected_1, z4171)))) | (P13 (cP13_InformationForMedicalReviewCollected_1, z4192)))) | (P15 (cP15_BloodResults_1, z4201)))) | (CollectInformationForMedicalReview(cCollectInformationForMedicalReview_lB_AllApptsBooked_Plus_SpecialistClinicianApptBooked_rB_1, z4226))))

Alexandru, Clutterbuck, **Papapanagiotou**, Fleuriot, Manataki. **A step towards the standardisation of HIV care practices**, HEALTHINF 2017

ARCHITECTURE



ARCHITECTURE



```
val state : Actionstate.t =  {Actionstate.ctr = 0; metas = []; buffered = []; joined = []; iprov = []; prov = []}
# let deliverDrug = Proc.create "DeliverDrug" [`Patient`;`Dosage`;`NurseTime`] `Treated ++ Failed`;;
val deliverDrug : Proc.t = {Proc.name = "DeliverDrug";
  inputs = [(`Patient`, `cDeliverDrug_Patient_1`); (`Dosage`, `cDeliverDrug_Dosage_2`); (`NurseTime`, `cDeliverDrug
  output = (`Treated ++ Failed`, `oDeliverDrug_lB_Treated_Plus_Failed_rB_`);
  proc = (...); actions = []; copier = false; intermediate = false}
# let reassess = Proc.create "Reassess" [`Failed`;`ClinTime`] `Checked`;;
val reassess : Proc.t = {Proc.name = "Reassess";
  inputs = [(`Failed`, `cReassess_Failed_1`); (`ClinTime`, `cReassess_ClinTime_2`)];
  output = (`Checked`, `oReassess_Checked_`);
  proc =(...); actions = []; copier = false; intermediate = false}
# let cAction = Action.create "JOIN" "DeliverDrug" "r" "Reassess" "NEG Failed" "DeliverAndReassess";;
val cAction : Action.t = {Action.act = "JOIN"; larg = "DeliverDrug"; lsel = "r"; rarg = "Reassess"; rsel = "(NEG Failed)"; res
# let deliverAndReassess,state = compose "DeliverAndReassess" [deliverDrug;reassess] [cAction];;
Buffering: ClinTime ** Treated
PARBUF_TAC: |-- ' ((ClinTime ** Treated) <> b3)^' (NEG ClinTime <> cReassess_ClinTime_2)' (NEG Treated <> a3) (...
PARBUF_TAC: |-- ' (ClinTime <> x4)^' (NEG ClinTime <> cReassess_ClinTime_2) (...)
PARBUF_TAC: |-- ' (Treated <> y4)^' (NEG Treated <> a3) (...)
Joining: NEG Failed <> cReassess_Failed_1
*** Action complete: JOIN: DeliverDrug (r) Reassess (NEG Failed) -> DeliverAndReassess (0.0775)
*** Theorem reconstruction complete. (0.0144)
val deliverAndReassess : Proc.t = {Proc.name = "DeliverAndReassess";
  inputs = [(`ClinTime`, `cReassess_ClinTime_2`); (`Dosage`, `cDeliverDrug_Dosage_2`); (`NurseTime`, `cDeliverDrug
  output = (`(ClinTime ** Treated) ++ Checked`, `y3`);
  proc = (...);
  actions = [{Action.act = "JOIN"; larg = "DeliverDrug"; lsel = "r"; rarg = "Reassess"; rsel = "(NEG Failed)"; res = "DeliverA
  copier = false; intermediate = false}
val state : Actionstate.t =
 {Actionstate.ctr = 7; metas = [`z7`; `y7`; `Q7`; ... ]; buffered = [`ClinTime ** Treated`]; joined = [`NEG Failed <> cRe
```

EXAMPLE (LINEAR LOGIC)

$\vdash$ Patient$^{\perp}$, Dosage$^{\perp}$, NurseTime$^{\perp}$, (Treated $\oplus$ Failed)

$\vdash$ Failed$^{\perp}$, ClinTime$^{\perp}$, Checked

$$\frac{\vdash \Gamma, C \quad \vdash \Delta, C^{\perp}}{\vdash \Gamma, \Delta}\, Cut$$

EXAMPLE (LINEAR LOGIC)

$\vdash$ PATIENT$^{\perp}$, DOSAGE$^{\perp}$, NURSETIME$^{\perp}$, (TREATED $\oplus$ FAILED)

$\vdash$ FAILED$^{\perp}$, CLINTIME$^{\perp}$, CHECKED

$$\frac{\vdash \Gamma, C \quad \vdash \Delta, C^{\perp}}{\vdash \Gamma, \Delta}\ Cut$$

(Treated $\oplus$ Failed)$^{\perp}$

EXAMPLE (LINEAR LOGIC)

$\vdash$ PATIENT$^{\perp}$, DOSAGE$^{\perp}$, NURSETIME$^{\perp}$, (TREATED $\oplus$ FAILED)

$\vdash$ FAILED$^{\perp}$, CLINTIME$^{\perp}$, CHECKED

$$\frac{\vdash \Gamma, C \quad \vdash \Delta, C^{\perp}}{\vdash \Gamma, \Delta}\, Cut$$

Treated$^{\perp}$ & Failed$^{\perp}$

EXAMPLE (LINEAR LOGIC)

$\vdash$ PATIENT$^{\perp}$, DOSAGE$^{\perp}$, NURSETIME$^{\perp}$, (TREATED $\oplus$ FAILED)

$\vdash$ FAILED$^{\perp}$, CLINTIME$^{\perp}$, CHECKED

$$\frac{\vdash \Gamma, C \quad \vdash \Delta, C^{\perp}}{\vdash \Gamma, \Delta}\, Cut$$

Treated$^{\perp}$ & Failed$^{\perp}$

$$\frac{\vdash \Gamma, A^{\perp} \quad \vdash \Gamma, B^{\perp}}{\vdash \Gamma, (A \oplus B)^{\perp}}\, \&$$

EXAMPLE (LINEAR LOGIC)

$\vdash$ FAILED$^{\perp}$, CLINTIME$^{\perp}$, CHECKED

$$\frac{}{\vdash Treated^{\perp}, Treated}\ Id$$

$$\frac{\vdash \Gamma, A^{\perp} \quad \vdash \Gamma, B^{\perp}}{\vdash \Gamma, (A \oplus B)^{\perp}}\ \&$$

EXAMPLE (LINEAR LOGIC)

$\vdash$ FAILED$^{\perp}$, CLINTIME$^{\perp}$, CHECKED

(Treated $\oplus$ Failed) $^{\perp}$

$$\frac{\vdash \Gamma, A^{\perp} \quad \vdash \Gamma, B^{\perp}}{\vdash \Gamma, (A \oplus B)^{\perp}} \,\&$$

$$\frac{\dfrac{}{\vdash Treated^{\perp}, Treated}\, Id \qquad \dfrac{}{\vdash ClinTime^{\perp}, ClinTime}\, Id}{\vdash Treated^{\perp}, ClinTime^{\perp}, Treated \otimes ClinTime}\, \otimes$$

EXAMPLE (LINEAR LOGIC)

$\vdash$ FAILED$^{\perp}$, CLINTIME$^{\perp}$, CHECKED

(Treated $\oplus$ Failed)$^{\perp}$

$$\frac{\vdash \Gamma, A^{\perp} \quad \vdash \Gamma, B^{\perp}}{\vdash \Gamma, (A \oplus B)^{\perp}}\ \&$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash Treated^{\perp}, Treated}\,Id \quad \dfrac{}{\vdash ClinTime^{\perp}, ClinTime}\,Id}{\vdash Treated^{\perp}, ClinTime^{\perp}, Treated \otimes ClinTime}\,\otimes}{\vdash Treated^{\perp}, ClinTime^{\perp}, (Treated \otimes ClinTime) \oplus Reassessed}\,\oplus_L \qquad (1)}{(1) \quad \vdash Failed^{\perp}, ClinTime^{\perp}, (Treated \otimes ClinTime) \oplus Reassessed}\,\oplus_R \qquad \dfrac{}{\vdash Failed^{\perp}, ClinTime^{\perp}, Reassessed}\,Reassess$$

EXAMPLE (LINEAR LOGIC)

$\vdash$ FAILED$^{\perp}$, CLINTIME$^{\perp}$, CHECKED

$$\frac{\vdash \Gamma, A^{\perp} \quad \vdash \Gamma, B^{\perp}}{\vdash \Gamma, (A \oplus B)^{\perp}} \;\&$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash Treated^{\perp}, Treated}\,Id \quad \dfrac{}{\vdash ClinTime^{\perp}, ClinTime}\,Id}{\vdash Treated^{\perp}, ClinTime^{\perp}, Treated \otimes ClinTime}\,\otimes}{\vdash Treated^{\perp}, ClinTime^{\perp}, (Treated \otimes ClinTime) \oplus Reassessed}\,\oplus_L \qquad (1)}{\dfrac{(1) \quad \dfrac{\dfrac{}{\vdash Failed^{\perp}, ClinTime^{\perp}, Reassessed}\,Reassess}{\vdash Failed^{\perp}, ClinTime^{\perp}, (Treated \otimes ClinTime) \oplus Reassessed}\,\oplus_R}{\vdash (Treated \oplus Failed)^{\perp}, ClinTime^{\perp}, (Treated \otimes ClinTime) \oplus Reassessed}\,\& \qquad (2)}$$

EXAMPLE (LINEAR LOGIC)

$\vdash$ PATIENT$^{\perp}$, DOSAGE$^{\perp}$, NURSETIME$^{\perp}$, (TREATED $\oplus$ FAILED)

$\vdash$ FAILED$^{\perp}$, CLINTIME$^{\perp}$, CHECKED

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash Treated^{\perp}, Treated}\,Id \quad \dfrac{}{\vdash ClinTime^{\perp}, ClinTime}\,Id}{\vdash Treated^{\perp}, ClinTime^{\perp}, Treated \otimes ClinTime}\,\otimes}{\vdash Treated^{\perp}, ClinTime^{\perp}, (Treated \otimes ClinTime) \oplus Reassessed}\,\oplus_L \quad (1)}{\vdash (Treated \oplus Failed)^{\perp}, ClinTime^{\perp}, (Treated \otimes ClinTime) \oplus Reassessed}\,\& \quad (2)}{\vdash Patient^{\perp}, Dosage^{\perp}, NurseTime^{\perp}, ClinTime^{\perp}, (Treated \otimes ClinTime) \oplus Reassessed}\,Cut$$

$$\dfrac{\dfrac{}{\vdash Failed^{\perp}, ClinTime^{\perp}, Reassessed}\,Reassess}{\vdash Failed^{\perp}, ClinTime^{\perp}, (Treated \otimes ClinTime) \oplus Reassessed}\,\oplus_R \quad (1)$$

$$\dfrac{}{\vdash Patient^{\perp}, Dosage^{\perp}, NurseTime^{\perp}, Treated \oplus Failed}\,DeliverDrug \quad (2)$$

AUTOMATION: COMPOSITION ACTIONS

BUFFERS &FILTERS	• Buffers push things forward • Filters manipulate structure
TENSOR	• Parallel composition
WITH	• Conditional composition • *if $A^{\perp}$ then P ; if $B^{\perp}$ then Q*
JOIN	• Sequential composition • *Connect (part of) an output with (part of) an input*

BUFFERS

BUFFER_TAC

$$\dfrac{\dfrac{}{\vdash A^{\perp}, A}\,Id}{\vdash A^{\perp}, A \oplus (B \otimes C)}\,\oplus R \qquad \dfrac{\dfrac{\dfrac{}{\vdash B^{\perp}, B}\,Id \quad \dfrac{}{\vdash C^{\perp}, C}\,Id}{\vdash B^{\perp}, C^{\perp}, B \otimes C}\,\otimes}{\vdash (B \otimes C)^{\perp}, B \otimes C}\,\parr}{\vdash (A \oplus (B \otimes C))^{\perp}, A \oplus (B \otimes C)}\,\oplus L \;\&$$

PARBUF_TAC

$$\dfrac{\dfrac{}{\vdash A^{\perp}, A}\,Id \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash B^{\perp}, B}\,Id}{\vdash B^{\perp}, B \oplus C}\,\oplus R \quad \dfrac{\dfrac{}{\vdash C^{\perp}, C}\,Id}{\vdash C^{\perp}, B \oplus C}\,\oplus L}{\vdash (B \oplus C)^{\perp}, B \oplus C}\,\&}{\vdash (B \oplus C)^{\perp}, D^{\perp}, (B \oplus C) \otimes D}\,\otimes}{\vdash A^{\perp}, (B \oplus C)^{\perp}, D^{\perp}, A \otimes (B \oplus C) \otimes D}\,\otimes$$

FILTERS

$\vdash X^{\perp}, A \oplus B$

$\vdash (B \oplus A)^{\perp}, Y$

$$\frac{\vdash \Gamma, C \quad \vdash \Delta, C^{\perp}}{\vdash \Gamma, \Delta}\, Cut$$

FILTERS

$\vdash X^{\perp}, A \oplus B$

$\vdash (B \oplus A)^{\perp}, Y$

$$\dfrac{\dfrac{\dfrac{}{\vdash A^{\perp}, A}\,Id}{\vdash A^{\perp}, B \oplus A}\,\oplus_L \quad \dfrac{\dfrac{}{\vdash B^{\perp}, B}\,Id}{\vdash B^{\perp}, B \oplus A}\,\oplus_R}{\vdash (A \oplus B)^{\perp}, B \oplus A} \qquad \vdash (B \oplus A)^{\perp}, Y}{\vdash (A \oplus B)^{\perp}, Y}\,Cut$$

TENSOR

General Rule

$$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} \otimes$$

Example

$$\frac{\vdash A^{\perp}, C^{\perp}, D \quad \vdash B^{\perp}, E}{\vdash A^{\perp}, B^{\perp}, C^{\perp}, D \otimes E} \otimes$$

WITH

General Rule

$$\dfrac{\dfrac{\vdash \Gamma, A^\perp, X}{\vdash \Gamma, A^\perp, X \oplus Y}\oplus L \quad \dfrac{\vdash \Gamma, C^\perp, Y}{\vdash \Gamma, C^\perp, X \oplus Y}\oplus R}{\vdash \Gamma, (A \oplus C)^\perp, X \oplus Y}\,\&$$

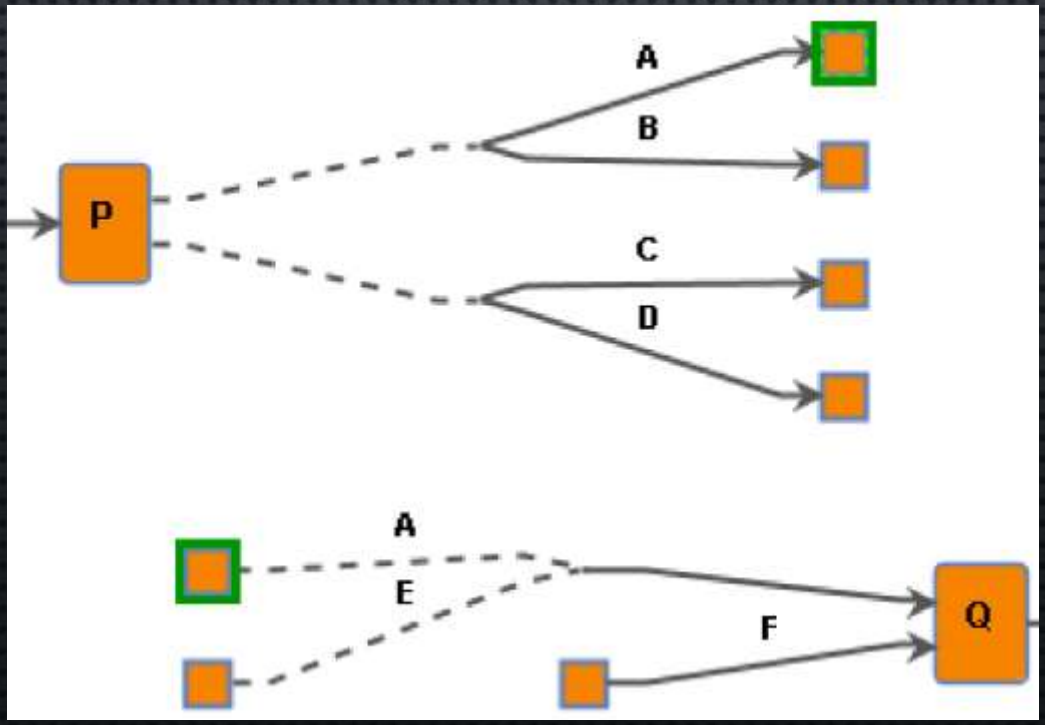
Example

$$\dfrac{\dfrac{\dfrac{}{\vdash A^\perp, B^\perp, X}\,\mathtt{P}}{\vdash A^\perp, B^\perp, X \oplus (Y \otimes B)}\oplus R \quad \dfrac{\dfrac{\dfrac{}{\vdash C^\perp, Y}\,\mathtt{Q} \quad \dfrac{}{\vdash B^\perp, B}\,Id}{\vdash C^\perp, B^\perp, Y \otimes B}\otimes}{\vdash C^\perp, B^\perp, X \oplus (Y \otimes B)}\oplus L}{\vdash (A \oplus C)^\perp, B^\perp, X \oplus (Y \otimes B)}\,\&$$

JOIN

General Rule

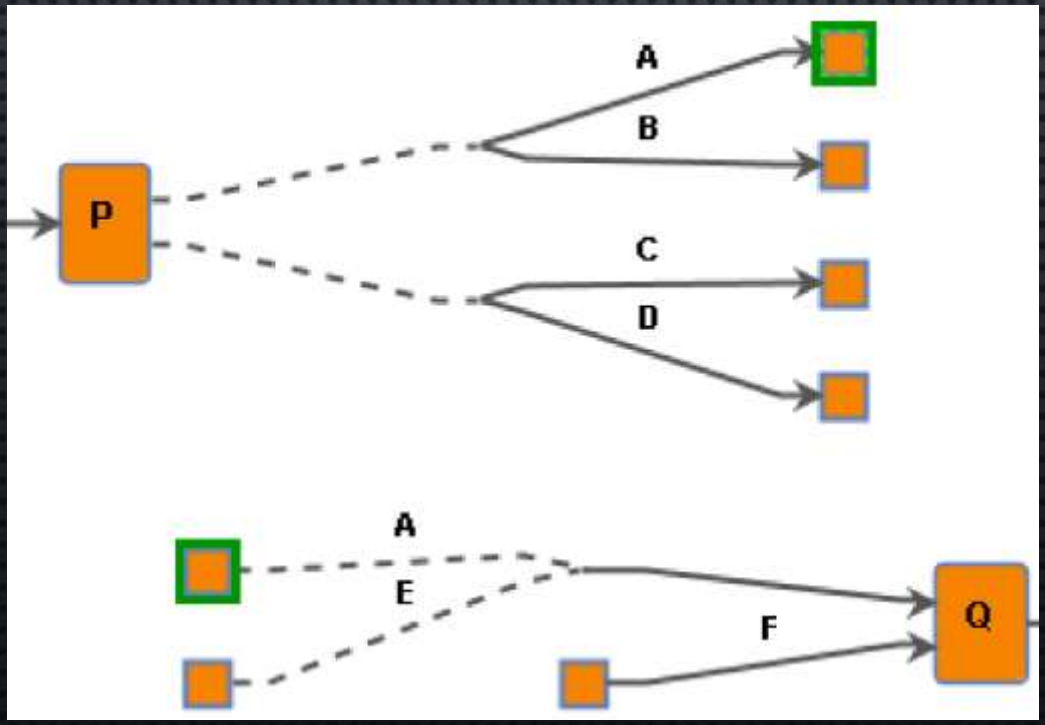
$$\frac{\vdash \Gamma, C \quad \vdash \Delta, C^{\perp}}{\vdash \Gamma, \Delta}\ Cut$$



JOIN

General Rule

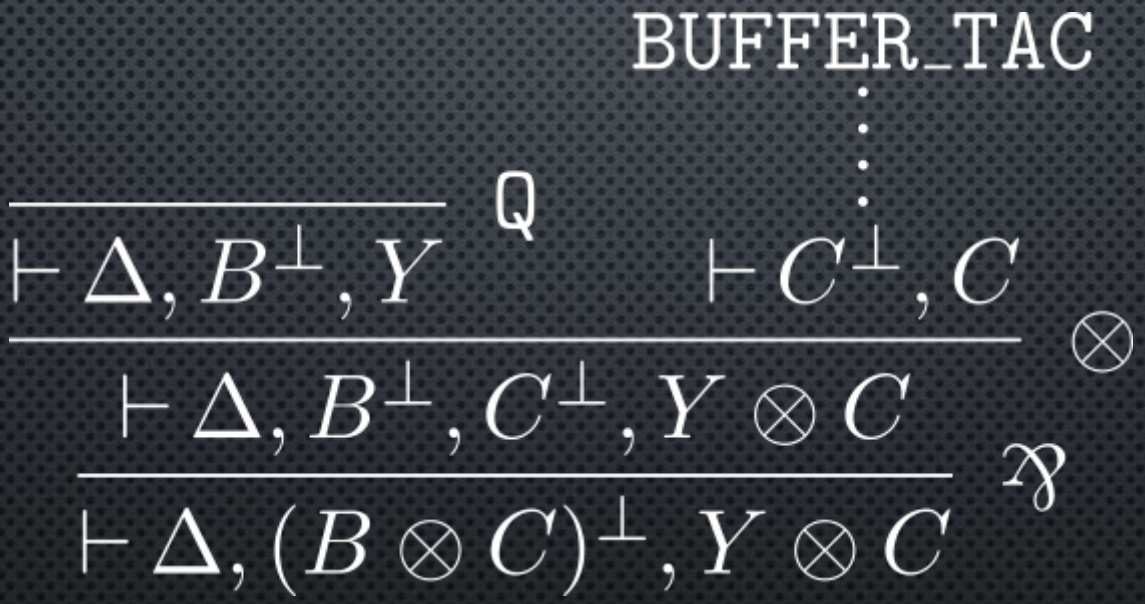
$$\dfrac{\vdash \Delta, \mathbf{X} \qquad \begin{array}{c} \vdash \Gamma, \mathbf{C}^{\perp}, Y \\ \vdots\ ? \\ \vdash ?, X^{\perp}, ? \end{array}}{?}\ Cut$$



JOIN: ATOMIC X

$$\dfrac{\dfrac{}{\vdash A^{\perp}, B^{\perp}, X}\,\mathtt{P} \qquad \dfrac{}{\vdash X^{\perp}, Z}\,\mathtt{Q}}{\vdash A^{\perp}, B^{\perp}, Z}\,Cut$$

JOIN: X = B⊗C


BUFFER_TAC
Q
$\vdash \Delta, B^{\perp}, Y$
$\vdash C^{\perp}, C$
$\otimes$
$\vdash \Delta, B^{\perp}, C^{\perp}, Y \otimes C$
$\parr$
$\vdash \Delta, (B \otimes C)^{\perp}, Y \otimes C$

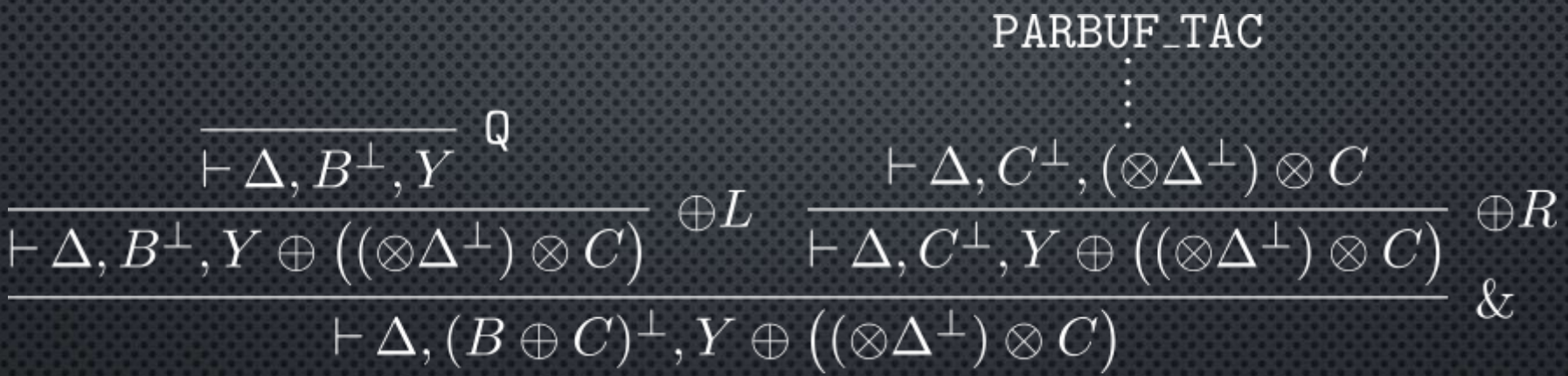
JOIN: X = B⊗C

$$\dfrac{\dfrac{}{\vdash \Delta, B^{\perp}, C^{\perp}, Y}\,\mathsf{Q}}{\vdash \Delta, (B \otimes C)^{\perp}, Y}\,\parr$$

JOIN: X = B⊕C

$$\dfrac{\dfrac{\dfrac{}{\vdash \Delta, B^{\perp}, Y}\,\mathtt{Q}}{\vdash \Delta, B^{\perp}, Y \oplus \left((\otimes \Delta^{\perp}) \otimes C\right)} \oplus L \qquad \dfrac{\begin{array}{c}\mathtt{PARBUF\_TAC}\\ \vdots \\ \vdash \Delta, C^{\perp}, (\otimes \Delta^{\perp}) \otimes C\end{array}}{\vdash \Delta, C^{\perp}, Y \oplus \left((\otimes \Delta^{\perp}) \otimes C\right)} \oplus R}{\vdash \Delta, (B \oplus C)^{\perp}, Y \oplus \left((\otimes \Delta^{\perp}) \otimes C\right)}\,\&$$

JOIN: X = B⊕C



Special case(s): Y = R⊕C , Δ={ }

JOIN: X = B⊕C

$$\dfrac{\dfrac{\dfrac{}{\vdash B^{\perp}, Y}\,Q}{\vdash B^{\perp}, Y \oplus C}\,\oplus L \qquad \dfrac{\dfrac{}{\vdash C^{\perp}, C}\,Id}{\vdash C^{\perp}, Y \oplus C}\,\oplus R}{\vdash (B \oplus C)^{\perp}, Y \oplus C}\,\&$$

Special case(s): Y = R⊕C , Δ={ }

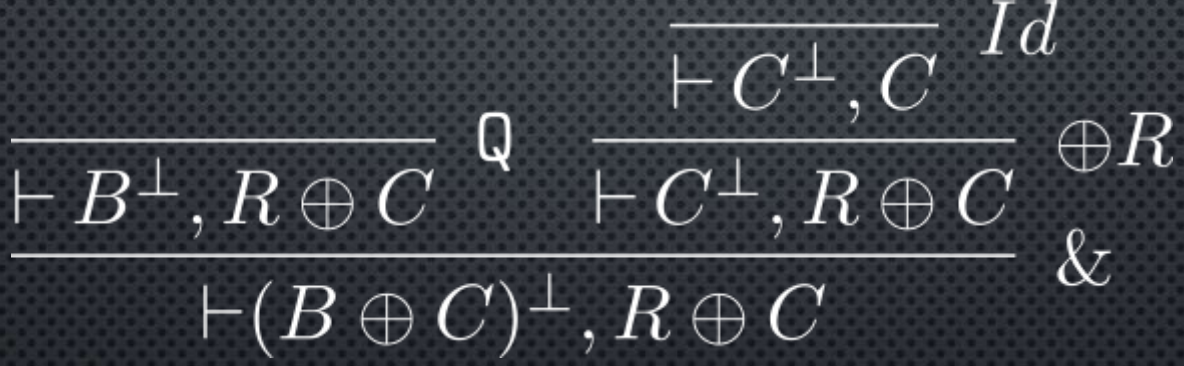
JOIN: X = B⊕C

$$\dfrac{\dfrac{}{\vdash B^{\perp}, R \oplus C}\,Q}{\vdash B^{\perp}, (R \oplus C) \oplus C}\,\oplus L \qquad \dfrac{\dfrac{}{\vdash C^{\perp}, C}\,Id}{\vdash C^{\perp}, (R \oplus C) \oplus C}\,\oplus R}{\vdash (B \oplus C)^{\perp}, (R \oplus C) \oplus C}\,\&$$

Special case(s): Y = R⊕C , Δ={ }

e.g. (Resource ⊕ Exception) ⊕ Exception

JOIN: X = B⊕C



Special case(s): Y = R⊕C , Δ={ }

e.g. (Resource ⊕ Exception) ⊕ Exception

```
1: function INPUT_TAC(sel, priority, orient, inputs, target, proc)
2:     Try to match target with sel (if provided) or one of the inputs
3:     if it matches then return proc
4:     else if target is atomic then
5:         if priority ≠ None then fail
6:         else Create a target buffer depending on orient
7:         end if
8:     else if target is L ⊗ R then
9:         if priority = Left then
10:            proc' = INPUT_TAC(sel, tail(priority), orient, inputs, L, proc)
11:            proc = INPUT_TAC(None, None, Right, inputs - {L}, R, proc')
12:        else
13:            proc' = INPUT_TAC(sel, tail(priority), orient, inputs, R, proc)
14:            proc = INPUT_TAC(None, None, Left, inputs - {R}, L, proc')
15:        end if
16:        Use the ⅋ rule to create the (L ⊗ R)^⊥ input
17:    else if target is L ⊕ R then
18:        if priority = Left then
19:            proc = INPUT_TAC(sel, tail(priority), orient, inputs, L, proc)
20:            Try special case derivations orElse Use derivation for optional X
21:        else if priority = Right then
22:            proc = INPUT_TAC(sel, tail(priority), orient, inputs, R, proc)
23:            Try special case derivations orElse Use derivation for optional X
24:        else
25:            Try as if priority = Left orElse Try as if priority = Right
26:            else Create a target buffer depending on orient
27:        end if
28:    end if
29:    return proc
30: end function
```

Conclusion

- Proofs-as-Processes give us the means to build Resource-based process models that:
  - Are Type-checked
  - Are Deadlock-Free
  - Have all resources explicitly accounted for (Leak-free)
  - Can be executed fully Asynchronously
- But at the cost of complex reasoning
- We automate this through 3 high-level composition actions:
  - Parallel (Tensor)
  - Conditional (With)
  - Sequential (Join)
- All implemented as a formal embedding in HOL Light
- Developed in a pragmatic evidence-based approach