Notes on Gazdar 1979
Updating Local Contexts Subject to Consistency

In what follows we assume the formalization of Gazdar 1979 given in the APPENDIX, for ease of comparison with Karttunen 1973 and Karttunen & Peters 1979. It is not clear how to deal with quantifiers in Gazdar's system. To avoid this issue, we adopt a non-quantificational analysis of definite descriptions (adapted from Heim 1982) and a closely analogous treatment of proper names (adapted from Kamp & Reyle 1993).

0. OVERVIEW OF KEY RESULTS

Correct predictions

- *No local inconsistency ⇒ no presupposition cancelled*

(1) John doesn't regret that he failed the exam.
Presupposes: John failed the exam.

(2) The king of Buganda is late.
Presupposes: There is a king of Buganda.

- *Cancellation due to inconsistency with initial context*

(3) John passed the exam. So he doesn't regret that he failed it.
Does **not** *presuppose*: John failed the exam.

- *Cancellation due to inconsistency with implicature*

(4) If John passed the exam, then he doesn't regret that he failed it.
Does **not** *presuppose*: John failed the exam.

- *Inconsistent presuppositions cancel each other*

(5) Either the king of Buganda is late or the president of Buganda is late.
Presupposes **neither** (i) There is a king of Buganda **nor** (ii) There is a president of Buganda.

- *Entailments cannot be cancelled*

(6) # Buganda is a monarchy and it is a republic.
Inconsistent entailments: (i) There is a king of Buganda *and* (ii) There is a president of Buganda.

Some problems

- *No cancellation if antecedent strictly entails presupposition of consequent*

(7) If John aced the exam, then he must be relieved that he passed it. cf. (4)
PREDICT: *Presupposes*: John passed the exam.
INTUITION: *Does* **not** *presuppose*: John passed the exam.

- *No cancellation with 'plug' verbs*

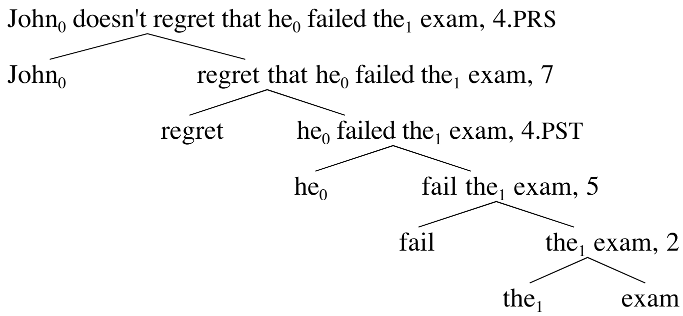
(8) {Lord Avon, Mad Charlie} claims that the king of Buganda is bald.
PREDICT: *Presupposes*: There is a king of Buganda.
INTUITION: *Does* **not** *presuppose*: There is a king of Buganda.

1. SAMPLE DERIVATIONS

- *No local inconsistency ⇒ no presupposition cancelled*

(1) John doesn't regret that he failed the exam.
Presupposes: John failed the exam.

(a) Step 1: *Computing the assertion* (**A**) *and potential presuppositions* (**P**)



Basic meanings

A(John$_0$) = $\lambda P_{e(st)}\, P\ x_0$
P(John$_0$) = $\lambda P_{e(st)}\, \lambda r_{st}(r = \lambda i(x_0 = john))$

A(he$_0$) = $\lambda P_{e(st)}\, P\ x_0$
P(he$_0$) = $\lambda P_{e(st)}\, \lambda r_{st}\ \bot$

A(the$_1$) = $\lambda P'_{e(st)}\, \lambda P_{e(st)}\, P x_1$
P(the$_1$) = $\lambda P'_{e(st)}\ \lambda P_{e(st)}\ \lambda r_{st}(r = P' x_1)$

A(exm) = $\lambda y_e\ \lambda i\ \ exm_i\ y$
P(exm) = $\lambda y_e\ \lambda r_{st}\ \bot$

A(fail) = $\lambda Q'_{(e(st))(st)}\ \lambda Q_{(e(st))(st)}\ (Q\ \lambda u_e(Q'\lambda z_e\ \lambda i\ \neg pass_i(y, z)))$
P(fail) = $\lambda Q'_{(e(st))(st)}\ \lambda Q_{(e(st))(st)}\ \lambda r\ \bot$

A(regret) = $\lambda p_{st}\ \lambda Q_{(e(st))(st)}\ (Q\ \lambda y_e \lambda i\ regret_i(y, p)))$
P(regret) = $\lambda p_{st}\ \lambda Q_{(e(st))(st)}\ \lambda r_{st}(r = p)$

Semantic composition

A(the$_1$ exam) = $\lambda P_{e(st)}\, P x_1$
P(the$_1$ exam) = $\lambda P_{e(st)}\ \lambda r_{st}(r = \lambda i\ exam_i\ \ x_1)$

A(fail the$_1$ exam) = $\lambda Q_{(e(st))(st)}\ (Q\ \lambda y_e \lambda i\ \neg pass_i(y,\ x_1))$
P(fail the$_1$ exam) = $\lambda Q_{(e(st))(st)}\ \lambda r_{st}(r = \lambda i\ exam_i\ \ x_1)$

A(he$_0$ failed the$_1$ exam) = $\lambda i\ \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0,\ x_1))$
P(he$_0$ failed the$_1$ exam) = $\lambda r_{st}(r = \lambda i\ \exists j(j \approx i \wedge j < i \wedge \text{exam}_j\ x_1))$ (note the binding problem)

In what follows, S$_0$:= he$_0$ failed the$_1$ exam.

A(regret that S$_0$) = $\lambda Q_{(e(st))(st)}\ (Q \lambda y_e \lambda i\ regret_i(y,\ \lambda i\ \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0,\ x_1))))$
P(regret that S$_0$) = $\lambda Q_{(e(st))(st)}\ \lambda r_{st}(r = \lambda i\ \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0,\ x_1))$
$\vee\ r = \lambda i\ \exists j(j \approx i \wedge j < i \wedge \text{exam}_j\ x_1))$

A(John$_0$ doesn't regret that S$_0$) = $\lambda i\ \neg regret_i(x_0,\ \lambda i\ \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0,\ x_1)))$
P(John$_0$ doesn't regret that S$_0$) = $\lambda r_{st}(r = \lambda i\ \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0,\ x_1))$
$\vee\ r = \lambda i\ \exists j(j \approx i \wedge j < i \wedge \text{exam}_j\ x_1))$
$\vee\ r = \lambda i(x_0 = john))$

(b) Step 2: *Computing potential clausal implicatures* (**I**)

Basic meanings ctd.

$\mathbf{I}(\text{John}_0) = \lambda P_{e(st)} \lambda r_{st} \bot$

$\mathbf{I}(\text{he}_0) = \lambda P_{e(st)} \lambda r_{st} \bot$

$\mathbf{I}(\text{the}_1) = \lambda P'_{e(st)} \lambda P_{e(st)} \lambda r_{st} \bot$

$\mathbf{I}(\text{exm}) = \lambda y_e \lambda r_{st} \bot$

$\mathbf{I}(\text{fail}) = \lambda Q'_{(e(st))(st)} \lambda Q_{(e(st))(st)} \lambda r \bot$

$\mathbf{I}(\text{regret}) = \lambda p_{st} \lambda Q_{(e(st))(st)} \lambda r_{st} \bot$

Semantic composition

$\mathbf{I}(\text{the}_1 \text{ exam}) = \lambda P_{e(st)} \lambda r_{st} \bot$

$\mathbf{I}(\text{fail the}_1 \text{ exam}) = \lambda Q_{(e(st))(st)} \lambda r_{st} \bot$

$\mathbf{I}(\text{he}_0 \text{ failed the}_1 \text{ exam}) = \mathbf{I}(S_0) = \lambda r_{st} \bot$

$\mathbf{I}(\text{regret that } S_0) = \lambda Q_{(e(st))(st)} \lambda r_{st} \bot$

$\mathbf{I}(\text{John}_0 \text{ doesn't regret that } S_0) = \lambda r_{st} \bot$

(c) Step 3: *Updating Successive Local Contexts Subject to Consistency*

General principles (from APPENDIX)

In Gazdar's system, a *context* is a consistent set of propositions (definition (X)) — that is, a set of propositions such that there is at least one world in which they are all true (definition (IX)). In our terms, if $\alpha_{(st)t}$ is (the characteristic function of) such a set of proposition, then consistency — in symbols, **con** α — is defined as follows:

$$\mathbf{con}\ \alpha_{(st)t} := \exists i\ (\cap\alpha)i$$

Gazdar defines contextual update in terms of set-theoretic *union*, $\cup$, and *satisfiable incrementation*, $\cup!$ (definition (XVI)). In our terms,

$$(\alpha_{(st)t} \cup \beta_{(st)t}) := \lambda r_{st}(\alpha r \vee \beta r)$$

$$(\alpha_{(st)t} \cup! \beta_{(st)t}) := \lambda r_{st}(\alpha r \vee (\beta r \wedge \forall W_{(st)t}(\forall q_{st}(Wq \rightarrow \alpha q \vee \beta q) \wedge \mathbf{con}\ W \rightarrow \mathbf{con}\ \lambda p_{st}(Wp \vee p = r))))$$

That is, (ignoring the difference between sets and characteristic functions), satisfiable incrementation of an initial context α with β yields a new context, $(\alpha \cup! \beta)$, which contains all of the propositions from α plus those propositions from β that cannot give rise to inconsistency. Formally, $(\alpha \cup! \beta)$ contains just those propositions r from β such that any consistent set W of propositions drawn from α and β remains consistent if augmented with r.

Finally, we are ready to transpose Gazdar's definitions (XV) and (XVII) of contextual update. When a sentence ϕ is uttered in a context α, then α is updated to a new context $(\alpha + \phi)$ as follows (ignoring scalar implicatures):

$$(\alpha + \phi) := ((\alpha \cup \lambda r_{st}(r = K\mathbf{A}(\phi)) \cup! \mathbf{I}(\phi)) \cup! \lambda r_{st} \exists q_{st}(\mathbf{P}(\phi)q \wedge r = Kq)$$

That is, we first add the proposition *I know that* ϕ. Next, we increment the resulting context with the potential clausal implicatures of ϕ, subject to consistency. The surviving implicatures are the *actual implicatures* of ϕ in α. And finally, the resulting context is incremented, again subject to consistency, with propositions of the form *I know that* ψ, where ψ is a potential presupposition of ϕ. Again, the surviving presuppositions ψ (entailed by *I know that* ψ, assuming Hintikka's semantics for *know*) are the *actual presuppositions* of ϕ in α. That is, they are those of the potential presuppositions of ϕ that are not cancelled on the grounds of inconsistency with either (1) the initial context α, or (2) the assertion of ϕ, or (3) the actual clausal implicatures of ϕ in α, or (4) each other. This is Gazdar's solution to the projection problem.

In particular, applied to the present example:

Let the initial context α be empty, i.e., $\alpha = \lambda r\, \bot$, and let ϕ = "John$_0$ doesn't regret that he$_0$ failed the$_1$ exam". We first add the proposition *I know that* ϕ, by set-theoretic union. The resulting local context is:

$$(\alpha \cup \lambda r_{st}(r = K\mathbf{A}(\phi))$$
$$= (\lambda r_{st}\bot \cup \lambda r_{st}(r = K\lambda i\, \neg regret_i(x_0, \lambda i\, \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1)))$$
$$= \lambda r_{st}(r = K\lambda i\, \neg regret_i(x_0, \lambda i\, \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1)))$$

Next, we increment this local context with the potential clausal implicatures of ϕ, subject to consistency:

$$(\alpha \cup \lambda r_{st}(r = K\mathbf{A}(\phi)) \cup!\ \mathbf{I}(\phi)$$
$$= \lambda r_{st}(r = K\lambda i\, \neg regret_i(x_0, \lambda i\, \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1))) \cup!\ \lambda r_{st}\bot$$
$$= \lambda r_{st}(r = K\lambda i\, \neg regret_i(x_0, \lambda i\, \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1)))$$

Finally, this new local context is incremented, again subject to consistency, with propositions of the form *I know that* ψ, where ψ is a potential presupposition of ϕ. Recall from above that

$$\mathbf{P}(\text{John}_0 \text{ doesn't regret that he}_0 \text{ failed the}_1 \text{ exam}) = \lambda r_{st}(r = \lambda i\, \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1)) \vee r = \lambda i\, \exists j(j \approx i \wedge j < i \wedge \text{exam}_j\, x_1) \vee r = \lambda i(x_0 = john))$$

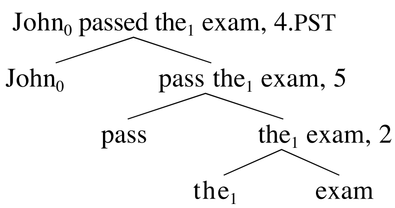
In our new local context none of these propositions can give rise to inconsistency. So the final update is:

$$\alpha + \text{"John}_0 \text{ doesn't regret that he}_0 \text{ failed the}_1 \text{ exam"}$$
$$= ((\alpha \cup \lambda r_{st}(r = K\mathbf{A}(\phi)) \cup!\ \mathbf{I}(\phi)) \cup!\ \lambda r_{st}\, \exists q_{st}(\mathbf{P}(\phi)q \wedge r = Kq)$$
$$= \lambda r_{st}(r = K\lambda i\, \neg regret_i(x_0, \lambda i\, \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1))) \cup!\ \lambda r_{st}\, \exists q_{st}((q = \lambda i\, \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1)) \vee q = \lambda i\, \exists j(j \approx i \wedge j < i \wedge \text{exam}_j\, x_1) \vee q = \lambda i(x_0 = john)) \wedge r = Kq)$$
$$= \lambda r_{st}(r = K\lambda i\, \neg regret_i(x_0, \lambda i\, \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1))) \cup!\ \lambda r_{st}\, (r = K\lambda i\, \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1)) \vee r = K\lambda i\, \exists j(j \approx i \wedge j < i \wedge \text{exam}_j\, x_1) \vee r = K\lambda i(x_0 = john))$$
$$= \lambda r_{st}(r = K\lambda i\, \neg regret_i(x_0, \lambda i\, \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1))) \vee r = K\lambda i\, \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1)) \vee r = K\lambda i\, \exists j(j \approx i \wedge j < i \wedge \text{exam}_j\, x_1) \vee r = K\lambda i(x_0 = john))$$

- *Cancellation due to inconsistency with initial context*

(3) [$_{\phi 1}$ John$_0$ passed the$_1$ exam]. [$_{\phi 2}$ So he$_0$ doesn't regret that he$_0$ failed it$_1$]
Does **not** *presuppose*: John$_0$ failed the$_1$ exam.

(a$_1$) Step 1 for ϕ_1: *Computing the assertion* (**A**) *and potential presuppositions* (**P**)



A(John$_0$ passed the$_1$ exam) = $\lambda i\ \exists j(j \approx i \wedge j < i \wedge pass_j(x_0, x_1))$
P(John$_0$ passed the$_1$ exam) = $\lambda r_{st}(r = \lambda i\ \exists j(j \approx i \wedge j < i \wedge \text{exam}_j\ x_1)$
$\vee\ r = \lambda i(x_0 = john)$)

(b$_1$) Step 2 for ϕ_1: *Computing potential clausal implicatures* (**I**)

I(John$_0$ passed the$_1$ exam) = $\lambda r_{st}\bot$

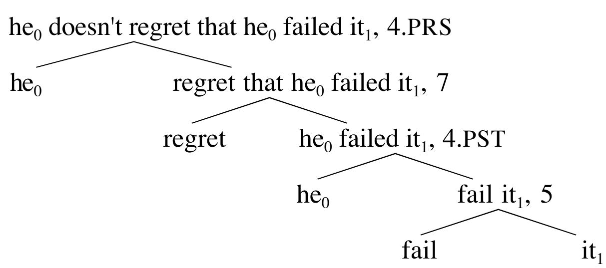
(c$_1$) Step 3 for ϕ_1: *Updating successive local contexts*, *beginning with* $\alpha = \lambda r_{st}\bot$, *subject to consistency*

$\lambda r_{st}\bot$ + "John$_0$ passed the$_1$ exam"

= $((\lambda r_{st}\bot \cup \lambda r_{st}(r = K\mathbf{A}(\phi)) \cup!\ \mathbf{I}(\phi)) \cup!\ \lambda r_{st}\ \exists q_{st}(\mathbf{P}(\phi)q \wedge r = Kq)$

= $\lambda r_{st}(r = K\lambda i\ \exists j(j \approx i \wedge j < i \wedge pass_j(x_0, x_1))$
$\vee\ r = K\lambda i\ \exists j(j \approx i \wedge j < i \wedge \text{exam}_j\ x_1)$
$\vee\ r = K\lambda i(x_0 = john))$

(a$_2$) Step 1 for ϕ_2: *Computing the assertion* (**A**) *and potential presuppositions* (**P**)



A(he$_0$ doesn't regret that he$_0$ failed it$_1$) = $\lambda i\ \neg regret_i(x_0, \lambda i\ \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1)))$
P(he$_0$ doesn't regret that he$_0$ failed it$_1$) = $\lambda r_{st}(r = \lambda i\ \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1))$

(b$_2$) Step 2 for ϕ_2: *Computing potential clausal implicatures* (**I**)

I(he$_0$ doesn't regret that he$_0$ failed it$_1$) = $\lambda r_{st}\bot$

(c$_2$) Step 3 for ϕ_2: *Updating successive local contexts*, *beginning with* $(\alpha + \phi_1)$, *subject to consistency*

($\lambda r_{st}\bot$ + "John$_0$ passed the$_1$ exam") + "he$_0$ doesn't regret that he$_0$ failed it$_1$"

= $\lambda r_{st}(r = K\lambda i\ \exists j(j \approx i \wedge j < i \wedge pass_j(x_0, x_1))$
$\vee\ r = K\lambda i\ \exists j(j \approx i \wedge j < i \wedge \text{exam}_j\ x_1)$
$\vee\ r = K\lambda i(x_0 = john)$
$\vee\ r = K\lambda i\ \neg regret_i(x_0, \lambda i\ \exists j(j \approx i \wedge j < i \wedge \neg pass_j(x_0, x_1)))$

APPENDIX A: GAZDAR 1979 FORMALIZED AS A REVISION + EXTENSION OF KARTTUNEN 1973

- Step 1: *Computing the assertion* (**A**) *and potential presuppositions* (**P**)

Assume the following *basic translations* and revise other basic translations as in the above sample derivations. Note the revised analysis of the definite article, adapted from Heim 1982:

$\mathbf{A}(\text{a}) = \lambda P'_{e(st)} \lambda P_{e(st)} \lambda i\, \exists x_e(P'xi \wedge Pxi)$

$\mathbf{P}(\text{a}) = \lambda P'_{e(st)} \lambda P_{e(st)} \lambda r_{st}\, \bot$

$\mathbf{A}(\text{the}_n) = \lambda P'_{e(st)} \lambda P_{e(st)} \lambda i\, Px_n i$

$\mathbf{P}(\text{the}_n) = \lambda P'_{e(st)} \lambda P_{e(st)} \lambda r_{st}(r = \lambda i\, P'x_n i)$

In each *compositional rule* below leave the syntax unchanged while revising the translation as follows. In terms of Karttunen 1973, Gazdar's functors are all HOLES (see definitions (23)–(25) and (VI)). Rules 7 and 11 have been adjusted accordingly. For the time being, we ignore the fact that Gazdar's presuppositions are epistemic statements of the form $K\phi$, where K abbreviates $\lambda p_{st}\lambda i\, know_i(me, p)$. The operator K will be introduced in the final step 3 (as suggested by Beaver 2001 on p. 66). In what follows we will also use the familiar *abbreviations*: $(\phi \models \psi)$ for $\forall i(\phi i \rightarrow \psi i)$, $(\phi \not\models \psi)$ for $\neg(\phi \models \psi)$, and $\cap\alpha_{(st)t}$ for $\lambda i\, \forall p_{st}(\alpha p \rightarrow pi)$. Finally, following Muskens 1995, we write "$i \approx j$" for "i has the same world coordinate as j", and "$j < i$" for "the time coordinate of j precedes that of i".

RULE 2 for $[\alpha_{T/CN}\, \beta_{CN}]_T$:

$\mathbf{A}(\alpha\, \beta) = \mathbf{A}(\alpha)\mathbf{A}(\beta)$

$\mathbf{P}(\alpha\, \beta) = \lambda P_{e(st)} \lambda r_{st}\, (\mathbf{P}(\alpha)\mathbf{A}(\beta)Pr \vee \exists x_e \mathbf{P}(\beta)xr)$

RULE 4.PRS for $[\beta_T\, prs(\alpha_{IV})]_S$, $[\beta_T\, negprs(\beta'_{IV})]_S$:

$\mathbf{A}(\beta\;\, prs(\alpha)) = \mathbf{A}(\alpha)\mathbf{A}(\beta)$

$\mathbf{P}(\beta\;\, prs(\alpha)) = \lambda r_{st}(\mathbf{P}(\alpha)\mathbf{A}(\beta)q \;\vee \exists P_{e(st)}\, \mathbf{P}(\beta)Pq)$

$\mathbf{A}(\beta'\;\, negprs(\beta)) = \lambda i\, \neg\mathbf{A}(\beta)\mathbf{A}(\beta')i$

$\mathbf{P}(\beta'\;\, negprs(\beta)) = \lambda r_{st}(\mathbf{P}(\alpha)\mathbf{A}(\beta)q \;\vee \exists P_{e(st)}\, \mathbf{P}(\beta)Pq))$

RULE 4.PST for $[\beta_T\, pst(\alpha_{IV})]_S$, $[\beta_T\, negpst(\beta'_{IV})]_S$:

$\mathbf{A}(\beta\;\, pst(\alpha)) = \lambda i\, \exists j(j \approx i \wedge j < i \wedge \mathbf{A}(\alpha)\mathbf{A}(\beta)j)$

$\mathbf{P}(\beta\;\, pst(\alpha)) = \lambda r_{st}\exists q_{st}((\mathbf{P}(\alpha)\mathbf{A}(\beta)q \;\vee \exists P_{e(st)}\, \mathbf{P}(\beta)Pq)$
$r = \lambda i\, \exists j(j \approx i \wedge j < i \wedge qj))$

$\mathbf{A}(\beta'\;\, negpst(\beta)) = \lambda i\, \neg\exists j(j \approx i \wedge j < i \wedge \mathbf{A}(\beta)\mathbf{A}(\beta')j)$

$\mathbf{P}(\beta'\;\, negpst(\beta)) = \lambda r_{st}\exists q_{st}((\mathbf{P}(\alpha)\mathbf{A}(\beta)q \;\vee \exists P_{e(st)}\, \mathbf{P}(\beta)Pq)$
$r = \lambda i\, \exists j(j \approx i \wedge j < i \wedge qj))$

RULE 5 for $[\alpha_{TV}\, acc(\beta_T)]_{IV}$:

$\mathbf{A}(\alpha\;\, acc(\beta)) = \mathbf{A}(\alpha)\mathbf{A}(\beta)$

$\mathbf{P}(\alpha\;\, acc(\beta)) = \lambda Q_{(e(st))st}\, \lambda r_{st}\, (\mathbf{P}(\alpha)\mathbf{A}(\beta)Qr \vee \exists P_{e(st)}\, \mathbf{P}(\beta)Pr)$

RULE 6 for $[gen(\beta_T)\; \alpha_{T/T}]_T$:

$\mathbf{A}(gen(\beta)\; \alpha) = \mathbf{A}(\alpha)\mathbf{A}(\beta)$

$\mathbf{P}(gen(\beta)\; \alpha) = \lambda P_{e(st)} \lambda r_{st}(\mathbf{P}(\alpha)\mathbf{A}(\beta)Pr \vee \exists P_{e(st)}\, \mathbf{P}(\beta)Pr)$

RULE 7 for $[\alpha_{IV/S} \text{ that } \beta_S]_{IV}$:

$\mathbf{A}(\alpha_{IV/S} \text{ that } \beta_S) \quad = \quad \mathbf{A}(\alpha)\mathbf{A}(\beta)$

$\mathbf{P}(\alpha_{IV/S} \text{ that } \beta_S) \quad = \quad \lambda Q_{(e(st))st}\, \lambda r_{st}\, (\mathbf{P}(\alpha)\mathbf{A}(\beta)Qr \vee \mathbf{P}(\beta)r)$

RULE 9 for $[\alpha_{S/S}\, \beta_S]_S$:

$\mathbf{A}(\alpha_{S/S}\, \beta_S) \quad = \quad \mathbf{A}(\alpha)\mathbf{A}(\beta)$

$\mathbf{P}(\alpha_{S/S}\, \beta_S) \quad = \quad \lambda r_{st}(\mathbf{P}(\alpha)\mathbf{A}(\beta)r \vee \mathbf{P}(\beta)r)$

RULE 11 for $[\text{if } \beta_S \text{ then } \beta'_S]$, $[\beta_S \text{ and } \beta'_S]_S$, $[\text{either } \beta_S \text{ or } \beta'_S]_S$:

$\mathbf{A}(\text{if } \beta \text{ then } \beta') \quad = \quad \lambda i\, (\mathbf{A}(\beta)i \rightarrow \mathbf{A}(\beta')i)$

$\mathbf{P}(\text{if } \beta \text{ then } \beta') \quad = \quad \lambda r_{st}(\mathbf{P}(\beta)q \vee \mathbf{P}(\beta)q)$

$\mathbf{A}(\beta \text{ and } \beta') \quad = \quad \lambda i\, (\mathbf{A}(\beta)i \wedge \mathbf{A}(\beta')i)$

$\mathbf{P}(\beta \text{ and } \beta') \quad = \quad \lambda r_{st}(\mathbf{P}(\beta)q \vee \mathbf{P}(\beta)q)$

$\mathbf{A}(\text{either } \beta \text{ or } \beta') \quad = \quad \lambda i\, (\mathbf{A}(\beta)i \vee \mathbf{A}(\beta')i)$

$\mathbf{P}(\beta \text{ and } \beta') \quad = \quad \lambda r_{st}(\mathbf{P}(\beta)q \vee \mathbf{P}(\beta)q)$

- <u>Step 2</u>: *Computing potential clausal implicatures* (**I**)

According to Gazdar's definition (V), a *potential clausal implicature* of ϕ is a proposition of the form *I don't know that* ψ or *I don't know that not* ψ, where ψ is a clausal constituent of ϕ such that neither ψ nor its negation is either entailed or presupposed by ψ. With K abbreviating $\lambda p_{st}\lambda i\ know_i(me, p)$, we can recast this definition as follows:

RULE 2 for $[\alpha_{T/CN}\, \beta_{CN}]_T$:

$\mathbf{I}(\alpha\ \beta) \quad = \quad \lambda P_{e(st)}\, \lambda r_{st}\, (\mathbf{I}(\alpha)\mathbf{A}(\beta)Pr \vee \exists x_e \mathbf{I}(\beta)xr)$

RULE 4.PRS for $[\beta_T\ prs(\alpha_{IV})]_S$, $[\beta_T\ negprs(\beta'_{IV})]_S$:

$\mathbf{I}(\beta\ \ prs(\alpha)) \quad = \quad \lambda r_{st}(\mathbf{I}(\alpha)\mathbf{A}(\beta)q\ \vee \exists P_{e(st)}\, \mathbf{I}(\beta)Pq)$

$\mathbf{I}(\beta'\ \ negprs(\beta)) \quad = \quad \lambda r_{st}(\mathbf{I}(\alpha)\mathbf{A}(\beta)q\ \vee \exists P_{e(st)}\, \mathbf{I}(\beta)Pq)$

RULE 4.PST for $[\beta_T\ pst(\alpha_{IV})]_S$, $[\beta_T\ negpst(\beta'_{IV})]_S$:

$\mathbf{I}(\beta\ \ pst(\alpha)) \quad = \quad \lambda r_{st}\exists q_{st}((\mathbf{I}(\alpha)\mathbf{A}(\beta)q\ \vee \exists P_{e(st)}\, \mathbf{I}(\beta)Pq)$
$\qquad \wedge\ r = \lambda i\ \exists j(j \approx i \wedge j < i \wedge qj))$

$\mathbf{I}(\beta\ \ negpst(\alpha)) \quad = \quad \lambda r_{st}\exists q_{st}((\mathbf{I}(\alpha)\mathbf{A}(\beta)q\ \vee \exists P_{e(st)}\, \mathbf{I}(\beta)Pq)$
$\qquad \wedge\ r = \lambda i\ \exists j(j \approx i \wedge j < i \wedge qj))$

RULE 5 for $[\alpha_{TV}\ acc(\beta_T)]_{IV}$:

$\mathbf{I}(\alpha\ \ acc(\beta)) \quad = \quad \lambda Q_{(e(st))st}\, \lambda r_{st}\, (\mathbf{I}(\alpha)\mathbf{A}(\beta)Qr \vee \exists P_{e(st)}\, \mathbf{I}(\beta)Pr)$

RULE 6 for $[gen(\beta_T)\ \alpha_{T/T}]_T$:

$\mathbf{I}(gen(\beta)\ \alpha) \quad = \quad \lambda P_{e(st)}\, \lambda r_{st}(\mathbf{I}(\alpha)\mathbf{A}(\beta)Pr \vee \exists P_{e(st)}\, \mathbf{I}(\beta)Pr)$

RULE 7 for $[\alpha_{IV/S} \text{ that } \beta_S]_{IV}$:

$\mathbf{I}(\alpha_{IV/S} \text{ that } \beta_S) \quad = \quad \lambda Q_{(e(st))st}\, \lambda r_{st}\, (\mathbf{I}(\alpha)\mathbf{A}(\beta)Qr \vee \mathbf{I}(\beta)r$
$\qquad \vee\ (r = \lambda i\ \neg K\mathbf{A}(\beta)i \vee r = \lambda i\ \neg(K\lambda j\neg\mathbf{A}(\beta)j)i)$
$\qquad \wedge\ ((\mathbf{A}(\alpha \text{ that } \beta)Q \not\models \mathbf{A}(\beta)) \wedge (\mathbf{A}(\alpha \text{ that } \beta)Q \not\models \lambda i\ \neg\mathbf{A}(\beta)i))$
$\qquad \wedge\ (\neg\exists Q_{(e(st))st}\mathbf{P}(\alpha \text{ that } \beta)Q\mathbf{A}(\beta) \wedge \neg\exists Q_{(e(st))st}\mathbf{P}(\alpha \text{ that } \beta)Q\lambda i\ \neg\mathbf{A}(\beta)i)))$

RULE 9 for $[\alpha_{S/S}\ \beta_S]_S$:

$$\mathbf{I}(\alpha_{S/S}\ \beta_S) = \lambda r_{st}(\mathbf{I}(\alpha)\mathbf{A}(\beta)r \vee \mathbf{I}(\beta)r \\ \vee (r = \lambda i\ \neg K\mathbf{A}(\beta)i \vee r = \lambda i\ \neg(K\lambda j \neg\mathbf{A}(\beta)j)i) \\ \wedge (\mathbf{A}(\alpha\ \beta) \not\models \mathbf{A}(\beta) \wedge \mathbf{A}(\alpha\ \beta) \not\models \lambda i\ \neg(\mathbf{A}(\beta)i) \\ \wedge (\neg\mathbf{P}(\alpha\ \beta)\mathbf{A}(\beta) \wedge \neg\mathbf{P}(\alpha\ \beta)\lambda i\ \neg\mathbf{A}(\beta)i)))$$

RULE 11 for [if β_S then β'_S], $[\beta_S$ and $\beta'_S]_S$, [either β_S or $\beta'_S]_S$:

$$\mathbf{I}(\text{if } \beta \text{ then } \beta') = \lambda r_{st}(\mathbf{I}(\beta)r \vee \mathbf{I}(\beta')r \\ \vee r = \lambda i\ \neg K\mathbf{A}(\beta)i \vee r = \lambda i\ \neg(K\lambda j \neg\mathbf{A}(\beta)j)i) \\ \vee r = \lambda i\ \neg K\mathbf{A}(\beta')i \vee r = \lambda i\ \neg(K\lambda j \neg\mathbf{A}(\beta')j)i)$$

$$\mathbf{I}(\beta \text{ and } \beta') = \lambda r_{st}(\mathbf{I}(\beta)r \vee \mathbf{I}(\beta')r)$$

$$\mathbf{I}(\text{either } \beta \text{ or } \beta') = \lambda r_{st}(\mathbf{I}(\beta)r \vee \mathbf{I}(\beta')r \\ r = \lambda i\ \neg K\mathbf{A}(\beta)i \vee r = \lambda i\ \neg(K\lambda j \neg\mathbf{A}(\beta)j)i) \\ \vee r = \lambda i\ \neg K\mathbf{A}(\beta')i \vee r = \lambda i\ \neg(K\lambda j \neg\mathbf{A}(\beta')j)i)$$

- Step 3: *Updating Successive Local Contexts Subject to Consistency*

In Gazdar's system, a *context* is a consistent set of propositions (definition (X)) — that is, a set of propositions such that there is at least one world in which they are all true (definition (IX)). In our terms, if $\alpha_{(st)t}$ is (the characteristic function of) such a set of proposition, then consistency — in symbols, **con** α — is defined as follows:

$$\mathbf{con}\ \alpha_{(st)t} := \exists i\ (\cap\alpha)i$$

Gazdar defines contextual update in terms of set-theoretic *union*, $\cup$, and *satisfiable incrementation*, $\cup!$ (definition (XVI)). In our terms,

$$(\alpha_{(st)t} \cup \beta_{(st)t}) := \lambda r_{st}(\alpha r \vee \beta r)$$

$$(\alpha_{(st)t} \cup! \beta_{(st)t}) := \lambda r_{st}(\alpha r \\ \vee (\beta r \wedge \forall W_{(st)t}(\forall q_{st}(Wq \rightarrow \alpha q \vee \beta q) \wedge \mathbf{con}\ W \rightarrow \mathbf{con}\ \lambda p_{st}(Wp \vee p = r))))$$

That is, (ignoring the difference between sets and characteristic functions), satisfiable incrementation of an initial context α with β yields a new context, $(\alpha \cup! \beta)$, which contains all of the propositions from α plus those propositions from β that cannot give rise to inconsistency. Formally, $(\alpha \cup! \beta)$ contains just those propositions r from β such that any consistent set W of propositions drawn from α and β remains consistent if augmented with r.

Finally, we are ready to transpose Gazdar's definitions (XV) and (XVII) of contextual update. When a sentence ϕ is uttered in a context α, then α is updated to a new context $(\alpha + \phi)$ as follows (ignoring scalar implicatures):

$$(\alpha + \phi) := ((\alpha \cup \lambda r_{st}(r = K\mathbf{A}(\phi)) \cup! \mathbf{I}(\phi)) \cup! \lambda r_{st}\ \exists q_{st}(\mathbf{P}(\phi)q \wedge r = Kq)$$

That is, we first add the proposition *I know that* ϕ. Next, we increment the resulting context with the potential clausal implicatures of ϕ, subject to consistency. The surviving implicatures are the *actual implicatures* of ϕ in α. And finally, the resulting context is incremented, again subject to consistency, with propositions of the form *I know that* ψ, where ψ is a potential presupposition of ϕ. Again, the surviving presuppositions ψ (entailed by *I know that* ψ, assuming Hintikka's semantics for *know*) are the *actual presuppositions* of ϕ in α. That is, they are those of the potential presuppositions of ϕ that are not cancelled on the grounds of inconsistency with either (1) the initial context α, or (2) the assertion of ϕ, or (3) the actual clausal implicatures of ϕ in α, or (4) each other. This is Gazdar's solution to the projection problem.